IN-71
73438
P-80

NASA Contractor Report 189582

AN INTRODUCTION TO HIGH SPEED AIRCRAFT NOISE PREDICTION

MARK R. WILSON

Lockheed Engineering and Sciences Company
Hampton, Virginia 23666

Contract NAS1-19000

February 1992

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665-5225

Summary

The purpose of this document is to introduce users to the Aircraft Noise Prediction Program's High Speed Research prediction system (ANOPP-HSR). Included within this mini-manual is an introduction which gives a brief overview of the ANOPP system and the components of the HSR prediction method. ANOPP information resources are given. Twelve of the most common ANOPP-HSR control statements are described. Each control statement's purpose and format are stated and relevant examples are provided. More detailed examples of the use of the control statements are presented in the manual along with ten ANOPP-HSR templates. The purpose of the templates is to provide the user with working ANOPP-HSR programs which can be modified to serve particular prediction requirements. Also included in this manual is a brief discussion of common errors and how to solve these problems. The appendices include the following useful information: a summary of all ANOPP-HSR functional research modules, a data unit directory, a discussion of one of the more complex control statements, and input data unit/input table examples.

INTRODUCTION TO THE USE OF ANOPP-HSR

1. Introduction

This document has been prepared to introduce users to the Aircraft Noise Prediction Program (ANOPP) executive system and the High Speed Research (HSR) prediction system. It has been based on an existing manual entitled "Introduction to the Use of ROTONET" (ref. 1). Much of the information contained in this document has been condensed from the Aircraft Noise Prediction Program User's Manual (ref. 2) to provide users with the basics for preparing modules for execution. It is written for the user who is interested in occasionally making high speed noise predictions on the Langley Research Center VAX computers without obtaining a detailed understanding of the executive system capabilities.

A brief overview of the ANOPP system is beneficial to understanding some of the program concepts to be discussed in this manual. The ANOPP System is divided into two parts, the Executive System and th Functional Module Library. The Executive System controls execution of the ANOPP run. It consists of several managing routines and a group of general utilities. The Executive Manager controls the execution of the ANOPP control statements. The Data Base Manager controls the activities of the data tables and data members. The Dynamic Storage Manager allows core sharing and dynamic dimensioning of the variable arrays. The General Utilities provide access to interpolation routines and other general functions. More information dealing with the executive system can be found in the ANOPP Programmer's Reference Manual for the Executive System (ref 3). The Functional Module Library contains all of the research functional modules. Each functional module is an independent group of subprograms which performs noise prediction functions. A hierarchical representation of the ANOPP components is shown in figure 1.

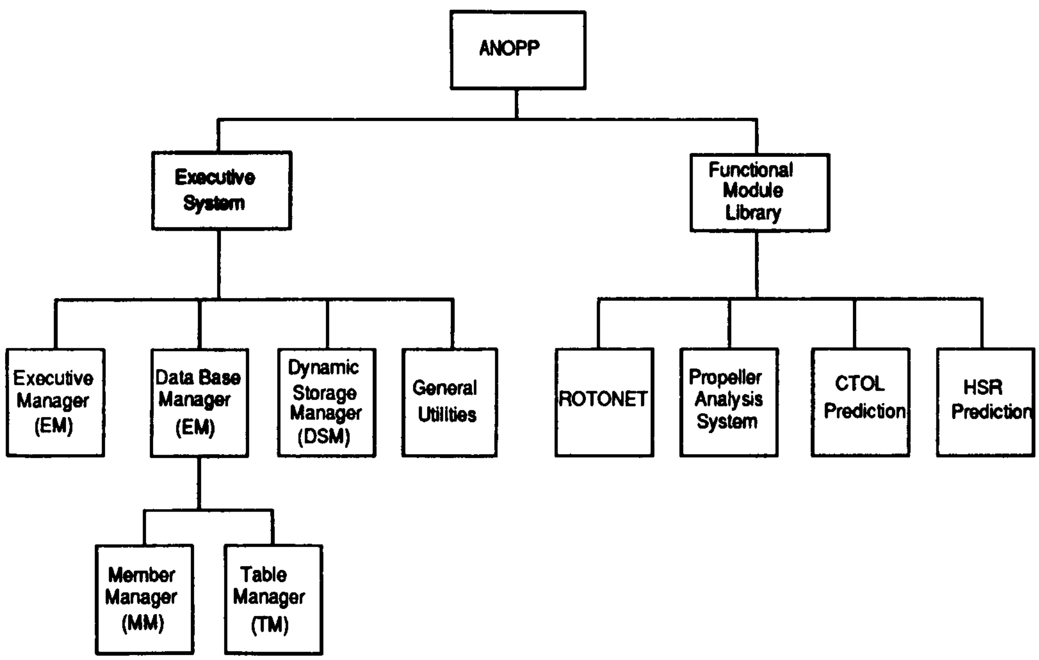

Figure 1. ANOPP Functional Hierarchy

A flow chart of the ANOPP-HSR system is shown in figure 2a. A procedure begins by defining an atmosphere using the Atmospheric Module (ATM). If atmospheric absorption effects are to be considered, the Atmospheric Absorption Module (ABS) is executed. The user has a choice when defining the flight path. If the test will consist of a steady flyover only, the Steady Flyover Module (SFO) is used. If the test includes takeoff, the Jet Takeoff Module (JTO) is used. And, if the test includes a landing, the Jet Landing Module (JLD) is used. The Geometry Module (GEO), which computes the range and directivity angles from the observer to the noise source, is executed next. At this point in the procedure, several noise source modules may be executed.

Modules specific to the HSR project are currently being developed. In the meantime, the Conventional Take-Off and Landing (CTOL) noise source modules are being used as a basis for HSR predictions.

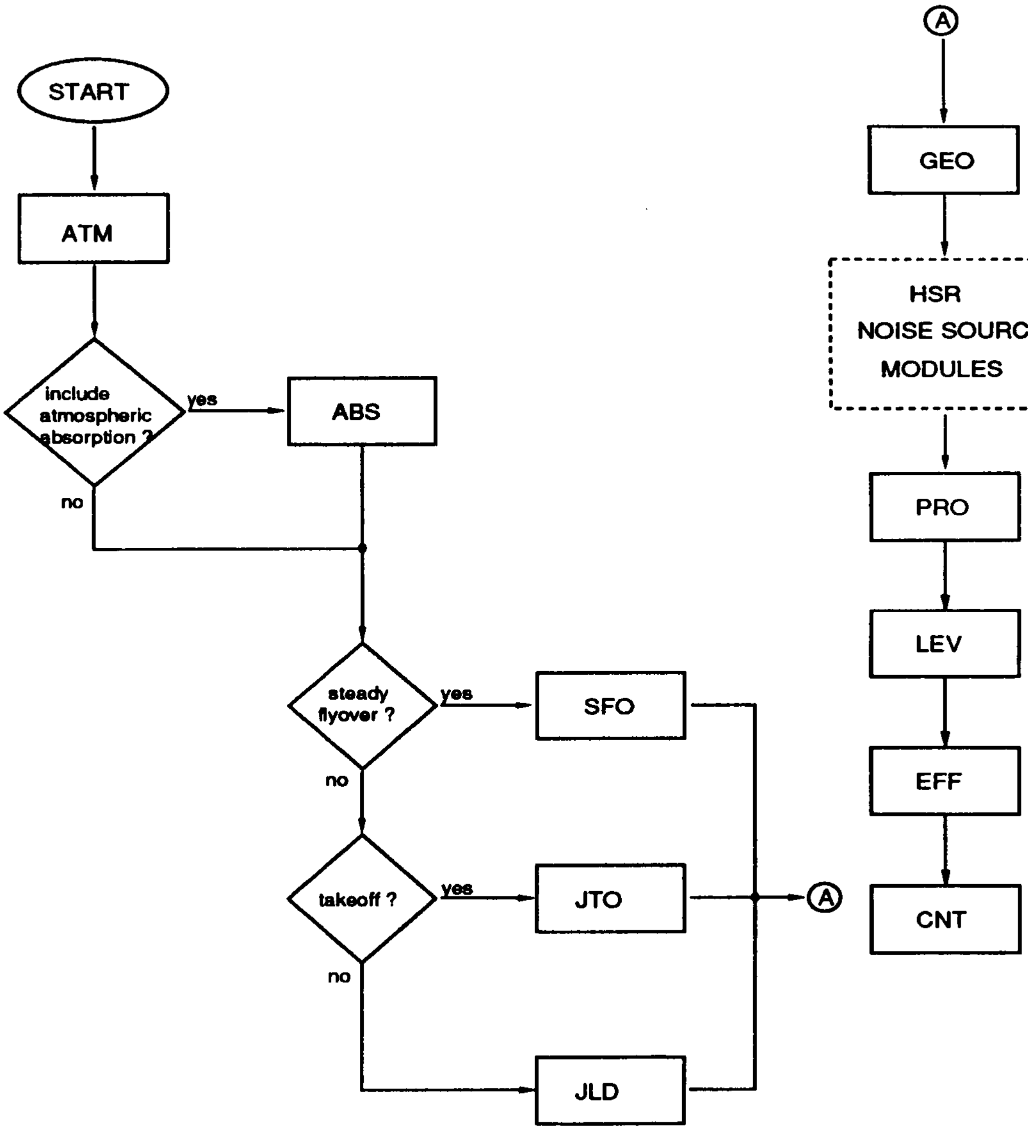

Figure 2a. Flow Chart of ANOPP-HSR program modules

The ANOPP-HSR source noise prediction modules are shown in figure 2b. For the prediction of jet noise, the user has the option of using any of the following modules: the Dual Stream Coannular Jet Noise Module (CNLJET), the Single Stream Circular Jet Noise Module (SGLJET), or the Stone Jet Noise Module (STNJET). Broadband shock noise prediction for circular jets is accomplished using the Circular Jet Shock Cell Noise Module (SAESHK). Fan noise predictions are generated using the Heidman Fan Noise Module (HDNFAN). Combustor noise is predicted using the Core Noise Module (GECOR). Turbine noise can be predicted using either the Turbine Noise Module (GETUR) or the Smith and Bushell Turbine Noise Module (SMBTUR). Airframe noise predictions are generated using the Airframe Noise Module (FNKAFM).

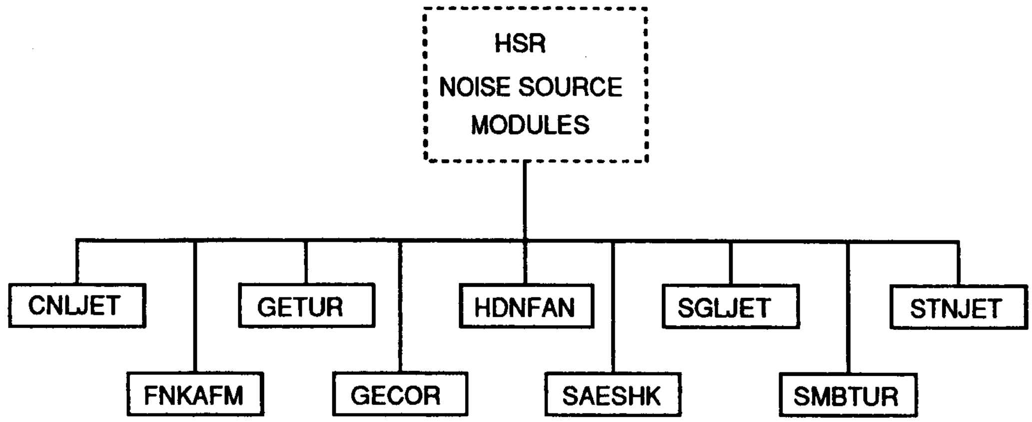

Figure 2b. ANOPP-HSR Source Noise Prediction Modules

Once data has been generated by the noise source modules, the Propagation Module (PRO) applies corrections to the noise data in the source frame of reference to transfer it to the observer frame of reference. It is in the Propagation Module that atmospheric absorption effects and ground effects are applied if requested. The Noise Levels Module (LEV) computes the Overall Sound Pressure Level (OASPL), the A-weighted Sound Pressure Level, the D-weighted Sound Pressure Level, the Perceived Noise Level (PNL), and the Tone-Corrected Perceived Noise Level (PNLT). The Effective Noise Module (EFF) computes the Effective Perceived Noise Level (EPNL). The Contour Module (CNT) provides a data file which can be used by an 'in-house' graphics package to plot either Overall Sound Pressure Level (OASPL), max. Perceived Noise Level, or max. A-weighted Sound Pressure Level values. A summary of the ANOPP functional modules which are used as a basis for HSR prediction is given in Appendix A.

Section 2 of this manual lists information resources available to users. Section 3 discusses the format of module documentation with an example containing informative comments. Section 4 describes the twelve most often used ANOPP control statements which will enable a user to execute any HSR module. Section 5 gives five example programs which demonstrate how to obtain user documentation and execute a module. Section 6 include ten ANOPP-HSR template programs. Section 7 indicates what to do if an error occurs.

The appendices provide supplemental information which will be useful as reference material. Appendix A is a summary of the functional modules provided in tabular format. Included in this table are the full title for each module, the associated ANOPP abbreviation, and a brief description of the function of that module. Appendix B is a data unit directory. Each data unit/member combination encountered in this document is listed along with the ANOPP abbreviations of the modules which use that data unit/member and the source from which that data unit/member is generated. Appendix C includes a more indepth discussion of the TABLE control statement than is presented in section 4.2.2 of this document. Appendix D presents examples of common input data units and data tables.

2. Informational Resources

Three manuals are available which aid users in preparing modules for execution. The first document is the Aircraft Noise Prediction Program User's Manual (ref. 2) which contains a detailed explanation of the ANOPP executive system. This document also contains a copy of the documentation preface, as described in Section 3, for each of the ANOPP-CTOL modules. Much of the information for this document has been condensed from reference 2. The second and third documents are the Aircraft Noise Prediction Program Theoretical Manual, Parts 1 and 2, (ref. 4) which contain a theoretical development of each module. HSR users will need the functional module documentation as discussed in Section 3.

3. Module Documentation

User documentation is maintained as a preface to the FORTRAN source code. This is done to ensure that the correct documentation is available for each version of the program in existence. This documentation is maintained on line and is accessible to the user. Example 5.1 in Section 5 explains how to obtain this documentation.

Figure 3 shows the format for the documentation of each module. The most important sections to the user are the INPUT, OUTPUT, and DATA BASE STRUCTURE sections. Under the INPUT and OUTPUT sections there are user parameters and unit members. A user parameter retains its value for each execution of a module. A unit member is closely related to a file, and contains a large block of data. Unit members will be discussed in more detail in section 4.2. The DATA BASE STRUCTURES section provides details concerning all unit members. The ERRORS section provides useful error diagnostics. The LDS (Local Dynamic Storage) and GDS (Global Dynamic Storage) REQUIREMENTS indicate computer core requirements.

```
***
*       PURPOSE - short description of the functional module (1-2 sentences)
*
*       AUTHOR - programmers initials and level number, such as L01/00/00
*
*       INPUT
*          USER PARAMETERS
*             Name1 - description - default value
*             .
*             .
*             .
*             Namen - description - default value
*          DATA BASE UNIT MEMBERS
*             DATA UNIT(DATA MEMBER) - short description of data requirement
*
*       OUTPUT
*          SYSTEM PARAMETERS
*             Name - description
*          USER PARAMETER - same as for INPUT
*          DATA BASE UNIT MEMBERS
*             DATA UNIT(DATA MEMBER) - short description of data requirement
*
*       DATA BASE STRUCTURES
*          DATA UNIT(DATA MEMBER) - complete description of data and required
*                                   format for all input and output data units
*
*       ERRORS
*          NON-FATAL - description of errors that are possible within the
*                      functional module.
*          FATAL - functional modules do not use fatal errors.
*
*       LDS REQUIREMENTS - describes the amount of local dynamic storage
*                          required by this module.
*
*       GDS REQUIREMENTS - describes the amount of global dynamic storage
*                          required for this module.
*
***
```

Figure 3. - ANOPP functional module prologue format.

The core sizes of LDS and GDS are preset to accommodate the execution of most modules. Should either of these core allocations be insufficient, Section 7 shows how to increase their sizes.

User documentation in example 3.1 is for the Atmospheric Module (ATM). Included in this documentation are various types of user parameters: integer (I), real single (RS), and alphanumeric (A). Two examples of table members are included. Example 3.1 will be referred to extensively in section 4 with further examples demonstrating how to use this documentation.

Example 3.1 Atmospheric Module Prologue

```
***
*       PURPOSE - BUILD TABLE OF ATMOSPHERIC MODEL DATA AS FUNCTIONS
*                 OF ALTITUDE
*
*       AUTHOR  - SWP(L03/00/00)
*
*       INPUT
*         USER PARAMETERS                                  TYPE DEFAULT
*           DELH       ALTITUDE INCREMENT FOR OUTPUT         RS    100.
*                      M (FT)
*           H1         GROUND LEVEL ALTITUDE                 RS    0.
*                      REFERENCED TO SEA LEVEL
*                      M (FT)
*           IUNITS     INPUT UNITS CODE                      A     2HSI
*                      =2HSI     , INPUTS ARE IN SI UNITS
*                      =7HENGLISH, INPUTS ARE IN ENGLISH
*                                  UNITS
*           NHO        NUMBER OF ALTITUDES FOR OUTPUT        I     1
*                      ATMOSPHERIC FUNCTIONS
*           P1         ATMOSPHERIC PRESSURE AT               RS    101325.
*                      GROUND LEVEL     N/M**2 (LBF/FT**2)
*           IPRINT     PRINT CODE FOR FORTRAN WRITE          I     3
*                        0   NO PRINT DESIRED
*                        1   INPUT PARAMETER PRINT ONLY
*                        2   OUTPUT PRINT ONLY
*                        3   BOTH INPUT PARAMETER AND OUTPUT PRINT
*         MEMBER
*           ATM( IN )
*
*       TEMPORARIES
*         MEMBER
*           SCRATCH( TAB1 )
*
*       OUTPUT
*         SYSTEM PARAMETER
*           NERR       EXECUTIVE SYSTEM PARAMETER FOR ERROR ENCOUNTERED
*                      DURING EXECUTION OF A FUNCTIONAL MODULE. NERR
*                      SET TO .TRUE. IF ERROR ENCOUNTERED.
*         MEMBER
*           ATM( TMOD )
*
*       DATA BASE STRUCTURES
*         ATM( IN )    CONTAINS DATA INPUT TO ATM IN FOLLOWING FORMAT
*                      ----------------------------------------------
*                      RECORD    FORMAT    DESCRIPTION
*                      ----------------------------------------------
*                        1        3RS      ALT, TEMP, RELATIVE HUMIDITY
```

```
*                                           (ALTITUDE, "ALT", IS
*                                           REFERENCED TO SEA LEVEL AND
*                                           SHOULD NOT BE LESS THAN USER
*                                           PARAMETER H1.)
*                       .         .
*                       .         .
*                     -------------
*                      ALTITUDE UNITS        M(FT)
*                      TEMPERATURE UNITS     KELVIN(RANKINE)
*                      RELATIVE HUMIDITY     PERCENT
*         SCRATCH( TAB1 )
*                     TEMPORARY TWO-DIMENSIONAL  TYPE 1 DATA TABLE
*                       INDEPENDENT VARIABLES
*                         1. ALTITUDE
*                         2. ORDERED POSITION
*                       DEPENDENT VARIABLES IN FOLLOWING ORDER
*                         TEMPERATURE
*                         HUMIDITY
*         ATM( TMOD )  OUTPUT TWO-DIMENSIONAL TYPE 1 DATA TABLE OF
*                     ATMOSPHERIC MODEL VALUES IN DIMENSIONLESS UNITS
*                       INDEPENDENT VARIABLES
*                         1. ALTITUDE (REFERENCED TO GROUND LEVEL)
*                         2. ORDERED POSITION
*                       DEPENDENT VARIABLES IN FOLLOWING ORDER
*                         PRESSURE
*                         DENSITY
*                         TEMPERATURE
*                         SPEED OF SOUND
*                         AVERAGE SPEED OF SOUND
*                         HUMIDITY
*                         COEFFICIENT OF VISCOSITY
*                         COEFFICIENT OF THERMAL CONDUCTIVITY
*                         CHARACTERISTIC IMPEDANCE(RHO*C)
*
*       ERRORS
*         NON-FATAL
*          1.  USER PARAMETER NHO IS OUT OF RANGE
*          2.  MEMBER CONTAINING INPUT DATA NOT AVAILABLE
*          3.  LOCAL DYNAMIC STORAGE INSUFFICIENT
*          4.  ERROR OCCURRED IN TABLE BUILD ROUTINE WHICH PREVENTED
*              THE BUILDING OF A TABLE.
*          5.  MEMBER CONTAINING INPUT DATA INVALID
*         FATAL - NONE
***
```

4. Control Statements

Described in this section are twelve of the more frequently used statements for preparing an HSR module for execution. A complete description of all the ANOPP control statements can be found in the Aircraft Noise Prediction Program User's Manual, NASA TM 84486 (ref. 2).

Each executive control statement has a specific format indicated in the following subsections. All control statement formats adhere to the following conventions:

- Each control statement directive is a free-form sequence, using columns 1 to 80
- A control statement may begin in any column and continue across as many as 5 lines to complete the directive.
- Each control statement is terminated by the $ character.
- Comments may appear in columns following the $ character terminator.
- Comments may continue across lines only if the first character on the line is the $ character terminator.

The general format of a control statement (CS) is as follows:

CSNAME OPERANDS $ COMMENTS

CSNAME — control statement name

Listed below are the twelve most frequently used ANOPP control statements:

ANOPP	LOAD
CALL	PARAM
DATA	STARTCS
ENDCS	TABLE
EVALUATE	UNLOAD
EXECUTE	UPDATE

OPERANDS — These are the operand fields that are required for each of the individual control statements.

COMMENTS — Any user desired comment can be included.

ANOPP control statements can be divided into two categories; Single Directive and Multiple Directive. As the name implies, single directive control statements require only one statement to execute a given function. These commands are described in section 4.1. Multiple directive control statements, described in section 4.2, require sub-commands to execute a given function.

4.1 Single Directive Control Statements

4.1.1 ANOPP

Purpose: The ANOPP control statement is the first CS in the user's input deck.

Format: ANOPP JECHO=.TRUE. JLOG=.TRUE. $

JECHO: print control during edit phase
JLOG: print control during execution phase

A complete list of system parameters has been tabulated on page 3-8 of the ANOPP User's Manual (ref. 2).

4.1.2 STARTCS

Purpose: The STARTCS control statement is the second CS in the user's input deck. STARTCS begins the execution.

Format: STARTCS $

4.1.3 LOAD

Purpose: The LOAD control statement loads unit members from an ANOPP library which has been previously stored on an external file via the UNLOAD control statement.

Format: LOAD/external file/unit1,...,unitn $

Example: Load the unit SAE from the external file LIBRARY. Unit SAE contains tables which are required by the SGLJET module.

```
LOAD/LIBRARY/ SAE $
```

4.1.4 UNLOAD

Purpose: The UNLOAD control statement establishes an ANOPP library for storage of one or more units on an external file.

Format: UNLOAD/external file/unit1,...,unitn $

Example: Create an ANOPP library with units UN1 and UN2 and store it on external file EXTFIL.

```
UNLOAD/EXTFIL/UN1,UN2 $
```

4.1.5 PARAM

Purpose: The PARAM control statement establishes values of one or more user parameters.

Format: PARAM $pname_1$=$value_1$,...,$pname_n$=$value_n$ $

pname: user parameter name
value: any required integer, real single precision, logical, or alphanumeric value

Example: Referring to example 3.1, assign values to the following user parameters:

AW	reference area	50 m^2
CTK	time constant	10.
DELDB	limiting noise level	15. dB
DIRECT	interpolation flag	true
IPRINT	print option	output only
IUNITS	units	metric

```
PARAM        AW=50.,CTK=10.,DELDB=15.,
             DIRECT=.TRUE.,IPRINT=2,
             IUNITS=2HSI $
```

4.1.6 EVALUATE

Purpose: The EVALUATE control statement establishes the value of a user parameter via an arithmetic expression.

Format: EVALUATE Pname=expression $

Pname: user parameter name
expression: a sequence of constants, user parameters and functions separated by operators and parentheses

The arithmetic operators are as follows:

+	addition
-	subtraction
*	multiplication
/	division
**	exponentiation

It is important to note that the arithmetic operators '+' and '-' must be preceded and followed by at least one blank space when used in the EVALUATE statement.

Additional functions are available as shown in table 1.

Example: Evaluate the nondimensional velocity V given a velocity of 102 meters per second and the default speed of sound, C, equals 340.294 meters per second.

```
EVALUATE V=102./C $
```

Name	Definition	Number of Arguments	Type of Arguments	Example
ABS	$\|x\|$	1	any type	Y = ABS(X)
ANTILOG	10^x	1	I,RS,RD	Y = ANTILOG(X)
COS	cos(x)	1	any type	Y = COS(X) X in degrees
INT	convert to integer	1	any type	Y = INT(X)
LOG	$\log_{10}(x)$, x>0	1	I,RS,RD	Y = LOG(X)
REAL	convert to real	1	any type	Y = REAL(X)
SIN	sin(x)	1	any type	Y = SIN(X) X in degrees
SQRT	$\sqrt{x}$, $x \geq 0$	1	any type	Y = SQRT(X)
TAN	sin(x)/cos(x)	1	any type	Y = TAN(X) X in degrees

Table 1. Generic Functions for the EVALUATE Control Statement

4.1.7 EXECUTE

Purpose: The EXECUTE control statement calls a specific functional module into execution.

Format: EXECUTE functional module name $

Example: Execute the Geometry module, GEO.

```
EXECUTE GEO $
```

4.1.8 CALL

<u>Purpose</u>: The CALL control statement allows the user to transfer run processing control to a secondary input stream.

<u>Format</u>: CALL unit1(pmname) $oldvalue_1$=$newvalue_1$,..., $oldvalue_n$=$newvalue_n$ $

unit1:	name of the run-life unit on which pmname resides
pmname:	name of the procedure member containing control statement lines to be used in the generation of the secondary input stream to be executed
oldvalue:	the exact field values which, if found to occur in a pmname resident CS line, are to be replaced with the corresponding newvalue
newvalue:	the CS image field to replace each occurrence of the oldvalue

<u>Example</u>: Call into execution the secondary input stream consisting of the lines residing on data unit DATA, procedure member SGLJET (see example 5.5 for actual implementation) replacing all occurrences of the value JETUNIT with JET2.

```
CALL DATA(SGLJET) JETUNIT=JET2 $
```

4.1.9 ENDCS

<u>Purpose</u>: The ENDCS control statement is the last line in the user's input deck and terminates the ANOPP run.

<u>Format</u>: ENDCS $

4.2 Multiple Directive Control Statements

The control statements discussed so far are single directive statements. The UPDATE, TABLE, and DATA statements are multiple directive statements. The purpose of these three statements is to provide a unit of information to a module.

As indicated in figure 4, a library is a collection of units and a unit is a collection of members. Two types of members are described: Data members and tables. Data members are input using the UPDATE control statement and provide a unit of information to a module that does not require interpolation. A unit requiring interpolation is input using the TABLE control statement. A table is a member with a specific structure. The DATA statement enables the user to set aside a block of information which can be accessed later in the procedure. This is a convenient method of executing several different cases within the same procedure as illustrated in example 5.5.

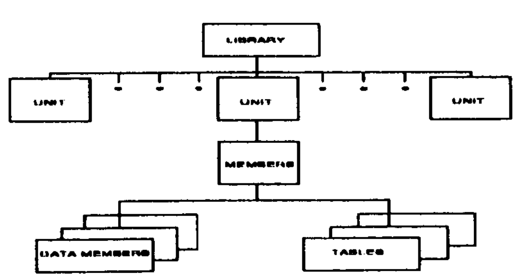

Figure 4. Library Hierarchy

4.2.1 UPDATE

Purpose: The UPDATE control statement allows the user to input a unit.

Format: UPDATE NEWU=unitname SOURCE=* $

unitname: name of data unit onto which new members are to be generated

-ADDR

Purpose: The -ADDR control statement allows the user to input a member on a specific unit with the aid of the UPDATE control statement.

Format: -ADDR OLDM=*,NEWM=mname,FORMAT=format $

mname: input member name

Valid format specifications are:

FORMAT=0	Unformatted
FORMAT=2HCI	Card Image
FORMAT=nHet,...,et$	Fixed Length Format
FORMAT=nH*et,...,et$	Variable Length Format

n: number of Hollerith characters in the format specification valid element types (et) are:

I	Integer
RS	Real Single
CS	Complex Single
L	Logical
A	Alphanumeric

The input deck follows the -ADDR statement, is separated by blanks or commas, and may take as many lines as necessary.

END*

Purpose: The END* control statement signals the termination of input to the unit. This statement is also used with the TABLE and DATA statements.

Example: A user is required to input unit member OBSERV(COORD) with each record having three real single precision values.

```
UPDATE NEWU=OBSERV SOURCE=* $
-ADDR OLDM=* NEWM=COORD FORMAT=4H3RS$ $
          10.   20.   30.   $
          20.   20.   20.   $
          30.   20.   10.   $
END* $
```

Example: A user is required to input unit SFIELD which consists of members FREQ, THETA, and PHI. This unit member represents the 1/3-octave band frequencies, polar directivity angles, and azimuthal directivity angles required by every source noise module for calculation purposes.

```
UPDATE NEWU=SFIELD SOURCE=* $
   -ADDR OLDM=* NEWM=FREQ FORMAT=4H*RS$ $
         50.     63.     80.    100.    125.
        160.    200.    250.    315.    400.
        500.    630.    800.   1000.   1250.
       1600.   2000.   2500.   3150.   4000.
       5000.   6300.   8000.  10000.       $
   -ADDR OLDM=* NEWM=THETA FORMAT=4H*RS$ $
    10.    30.    50.    70.    90.   110.
   130.   150.   170. $
   -ADDR OLDM=* NEWM=PHI FORMAT=4H*RS$ $
     0. $
END* $
```

Example: A user is required to input unit ATM which consists of the member IN. This unit member is required as input to the Atmosphere Module. It consists of a temperature and humidity profile as a function of altitude.

```
UPDATE NEWU=ATM SOURCE=* $
   -ADDR OLDM=* NEWM=IN FORMAT=4H3RS$ $
          0.    313.2    70.       $
       1000.    306.7    70.       $
       2000.    300.2    70.       $
       3000.    293.7    70.       $
       4000.    287.2    70.       $
       5000.    280.7    70.       $
   END* $
```

4.2.2 TABLE**

Purpose: The TABLE control statement builds a table member in accordance with a set of user supplied instructions for interpolation.

Format: Type 1 Tables (only type currently available).

```
TABLE UNIT(MEMBER) 1 SOURCE=* $
   INT=0,1,2
   IND1=RS,n1,2,2, independent variable values
                   separated by commas or blanks
   IND2=RS,n2,2,2, independent variable values
                   separated by commas or blanks
   IND3=RS,n3,2,2, independent variable values
                   separated by commas or blanks
   IND4=RS,n4,2,2, independent variable values
                   separated by commas or blanks
   DEP=RS, dependent variable values separated by
           commas or blanks
END* $
```

The integer values n1,...,n4 are the number of values of the corresponding independent variables. If the table has less than four dimensions, then fewer independent variables are needed. If the independent variable is ordered position, then the "RS" is replaced by a "0" (zero) and no independent variable values are needed.

** See Appendix C of this manual for a discussion of the TABLE control statement

Independent and dependent variable values may take as many lines as needed.

Example: The following two functions, pressure and temperature, are input as table ATM(SAMPLE) using ordered position. The tabulated pressure values are entered first followed by the temperature values. IND2 is used to indicate ordered position by replacing "RS" with "0" and setting n2 equal to 2 indicating the two functions: pressure and temperature.

altitude	pressure	temperature
0.	2116.	510.
2000.	1965.	506.
4000.	1824.	502.
6000.	1692.	498.

```
TABLE ATM(SAMPLE) 1 SOURCE=* $
INT=0  1  2
IND1=RS 4  2  2  0. 2000. 4000. 6000.
IND2=0  2  2  2
DEP=RS 2116.  1965.  1824.  1692.
        510.   506.   502.   498.
END* $
```

4.2.3 DATA

Purpose: The DATA control statement creates or recreates a formatted member on system work unit DATA. The input set is terminated by the first END* line encountered following the DATA CS.

Format:

```
DATA DM=mname $
     <control statement> $
          .
          .
          .
     <control statement> $
END* $
```

mname: name of the member to be created on system data unit DATA

Example: Generate a formatted member called FLIGHT as in example 5.4.

```
DATA DM=FLIGHT $
        PARAM IOUT=1        $
        PARAM J=1           $
        EVALUATE VA=MACH*CA $
END* $
```

5. Example Programs

In this section examples will be given showing how to obtain user documentation for the ATM module and prepare input for execution. The examples include the control statements necessary to prepare any module for execution.

Example 5.1

In order to obtain the user documentation for the ATM module, the following ANOPP input deck can be executed. Appendix A lists the names of modules currently included in the ANOPP system. To obtain the user documentation for any one of these modules, replace ATM in the following example with the name of the desired module.

```
ANOPP  JECHO=.TRUE. $
STARTCS $
LOAD/LIBRARY/MANUAL $
MEMLIST MANUAL(ATM) FORMAT=2HCI $
ENDCS $
```

Example 5.2

A demonstration of the use of the Atmospheric Module (ATM) is presented in this example. The purpose of this module is to generate tables of atmospheric data that can be used by other modules for subsequent calculations. One table is generated in this example. This table provides conditions for a standard sea level atmosphere based on a 70% relative humidity (i.e. 0.2 percent mole fraction). Refer to the Atmospheric Module prologue, presented as Example 3.1, for more detailed information concerning the input and output of this module.

```
ANOPP  JECHO=.TRUE.  $
STARTCS $
$
$ CREATE THE REQUIRED INPUT DATA BASE MEMBERS
$
UPDATE NEWU=ATM SOURCE=* $
      -ADDR OLDM=* NEWM=IN FORMAT=4H3RS$ $
                   0.      288.15    70.     $
                   200.    286.85    70.     $
                   400.    285.55    70.     $
                   600.    284.25    70.     $
                   800.    282.95    70.     $
                   1000.   281.65    70.     $
END* $
$
$ GENERATE ATMOSPHERIC PROPERTIES
$
PARAM  DELH=100. H1=0. NH0=11 P1=101325. IPRINT=3 $
$
EXECUTE  ATM  $
$
ENDCS $
```

Example 5.3

The geometry module (GEO) is executed in this example. In order for any module to function properly, it must be supplied with certain tables or units of information. Normally the data will be generated by one module and then used in subsequent modules as explained in example 5.2. In some cases, it may be more convenient for the user to provide the input data required by a module. This is accomplished using the UPDATE control statement. For example, when examining pages 4-5 and 4-6 of the ANOPP User's Manual (ref. 2), it can be seen that the Geometry Module, GEO, requires the following data base structures: ATM(TMOD), FLI(PATH), and OBSERV(COORD) as input. The table ATM(TMOD) will be generated using the Atmospheric Module, ATM. The unit member FLI(PATH) can be generated by any of the flight dynamics modules (SFO, JTO, or JLD) or it can be generated by the user. A detailed description of the unit member FLI(PATH) is given on page 4-7 of reference 2. Based on this information, the unit member can be defined by the user. In this example, three flight trajectories will be examined. Each unit member, FLI(TAKOFF), FLI(LEVFLI), and FLI(LAND), is defined using the UPDATE control statement. In order to use the different flight trajectories when executing the Geometry Module, a name override is used for the member name PATH, as can be seen in the EXECUTE statements below.

```
ANOPP JECHO=.TRUE. JLOG=.TRUE. $
STARTCS $
$
$ DEMONSTRATION PROBLEMS FOR GEOMETRY MODULE
$
$ CREATE REQUIRED INPUT DATA BASE MEMBERS
$
UPDATE NEWU=ATM SOURCE=* $
      -ADDR OLDM=* NEWM=IN FORMAT=4H3RS$ $
                    0.      536.670         50.         $
                    200.    535.957         50.         $
                    400.    535.244         50.         $
                    600.    534.530         50.         $
                    800.    533.817         50.         $
                    1000.   533.104         50.         $
                    1500.   532.604         50.         $
                    2000.   532.236         50.         $
                    2500.   532.082         50.         $
END* $
$
PARAM        DELH=100.     H1=0.           IUNITS=7HENGLISH
             NH0=26        P1=2116.22      IPRINT=3                 $
$
EXECUTE ATM    $
$
UPDATE NEWU=FLI SOURCE=* $
      -ADDR OLDM=* NEWM=TAKOFF FORMAT=5H10RS$ $
           0.0      0.     0.        0.     0.   0.   0.   0.   0.   0.   $
           10.0    500.    0.     -100.     0.   0.   0.   0.   0.   0.   $
           20.0   1000.    0.     -200.     0.   0.   0.   0.   0.   0.   $
           30.0   1500.    0.     -300.     0.   0.   0.   0.   0.   0.   $
           40.0   2000.    0.     -400.     0.   0.   0.   0.   0.   0.   $
           50.0   2500.    0.     -450.     0.   0.   0.   0.   0.   0.   $
           60.0   3000.    0.     -500.     0.   0.   0.   0.   0.   0.   $
           70.0   3500.    0.     -550.     0.   0.   0.   0.   0.   0.   $
      -ADDR OLDM=* NEWM=LEVFLI FORMAT=5H10RS$ $
           0.0      0.    50.    -1000.     0.   0.   0.   0.   0.   0.   $
           20.0    700.   50.    -1000.     0.   0.   0.   0.   0.   0.   $
           40.0   1400.   50.    -1000.     0.   0.   0.   0.   0.   0.   $
           60.0   2100.   50.    -1000.     0.   0.   0.   0.   0.   0.   $
           80.0   2800.   50.    -1000.     0.   0.   0.   0.   0.   0.   $
      -ADDR OLDM=* NEWM=LAND FORMAT=5H10RS$ $
           0.0      0.     0.    -2500.     5.   5.   5.   5.   5.   5.   $
           10.0    500.    0.    -2000.     5.   5.   5.   5.   5.   5.   $
           20.0   1000.    0.    -1500.     5.   5.   5.   5.   5.   5.   $
           30.0   1500.    0.    -1000.     5.   5.   5.   5.   5.   5.   $
```

```
            40.0    2000.     0.     -500.     5.   5.   5.   5.   5.   5.   $
            50.0    2500.     0.        0.     5.   5.   5.   5.   5.   5.   $
END* $
UPDATE NEWU=OBSERV SOURCE=* $
      -ADDR OLDM=* NEWM=COORD FORMAT=4H3RS$ $
                  100.      50.       5.        $
                  100.       0.      10.        $
                 1000.     -50.       5.        $
                 1000.       0.      10.        $
                 2000.     100.       5.        $
                 2000.    -100.      10.        $
END* $
$
$=====================================================================
$
$ TEST NUMBER 1.    TAKEOFF WITH SOURCE COORDINATE
$                   DESCRIPTIONS GIVEN AND CTK=10,
$                   DELB=10,  START=0.  AND  STOP=70.
$
PARAM        CTK=10.  DELB=10.  START=0.  STOP=70.  IPRINT=3
             IUNIT=7HENGLISH $
$
EXECUTE GEO PATH=TAKOFF   $
$
$=====================================================================
$
$ TEST NUMBER 2.    SAME AS NUMBER 1 EXCEPT FOR LEVEL
$                   FLIGHT PATH AND START=10 AND STOP=50
$
PARAM CTK=1.0  START=10.  STOP=50. $
$
EXECUTE GEO PATH=LEVFLI   $
$
$=====================================================================
$
$ TEST NUMBER 3.    SAME AS NUMBER 1 EXCEPT FOR LANDING
$                          FLIGHT PATH
$
EXECUTE GEO PATH=LAND   $
$
ENDCS$
```

Example 5.4

The DATA and CALL control statements are demonstrated in this example. The input job is designed to provide noise predictions for a turbofan-powered aircraft in two different flight regimes. The first prediction case is a low speed level flyover at an altitude of 5000 feet. The second prediction is calculated at a speed of Mach 0.8 and an altitude of 35000 feet. After the data unit SAE is loaded from the LIBRARY and the sound field grid is established, an atmosphere is defined so that ambient conditions can be obtained by the Atmospheric Parameters module (APM). The APM module outputs ambient conditions for any altitude specified. These conditions are interpolated from the ATM(TMOD) data table. Once the atmosphere is defined, the noise prediction modules are executed. In order to avoid unnecessary duplication of this section of the input job, the DATA statement is used to define a command procedure that will be executed for each case. The procedure is entitled SGLJET and its purpose is to normalize the input data and execute the Single Stream Circular Jet Noise module and the SAE Shock Noise Module (if there are shocks present in the flow). The command procedure is activated by the CALL control statement. Data units for each case are defined by using name overrides when invoking the command procedure. Input parameters may also be redefined prior to the use of the CALL control statement.

```
ANOPP JECHO=.TRUE. JLOG=.TRUE.  $
STARTCS $
$
$
UPDATE NEWU=ATM SOURCE=* $
      -ADDR OLDM=* NEWM=IN FORMAT=4H3RS$ $
               0.    510.     70.    $
            7500.    495.     40.    $
           18000.    450.     23.    $
           34000.    420.     20.    $
END* $
$
$
$     THE SOURCE NOISE OF THE ENGINE IS ESTIMATED USING
$     SINGLE STREAM CIRCULAR JET NOISE PREDICTION MODEL AND
$     THE CIRCULAR JET SHOCK CELL NOISE PREDICTION MODEL AS
$     IMPLEMENTED IN MODULES SGLJET AND SAESHK
$
$
$     DATA TABLES ARE LOADED FROM THE LIBRARY AND THE SOUND
$     FIELD GRID IS DEFINED
$
LOAD /LIBRARY/ SAE  $
$
UPDATE NEWU=SFIELD SOURCE=* $
      -ADDR OLDM=* NEWM=FREQ FORMAT=4H*RS$ $
          50.     63.     80.    100.    125.    160.    200.
         250.    315.    400.    500.    630.    800.   1000.
        1250.   1600.   2000.   2500.   3150.   4000.   5000.
        6300.   8000.  10000. $
      -ADDR OLDM=* NEWM=THETA FORMAT=4H*RS$ $
          10.   30.   50.   70.   90.  110.  130.  150.  170.   $
      -ADDR OLDM=* NEWM=PHI FORMAT=4H*RS$ $
           0.    $
END* $
$
$
$     THE USER PARAMETERS FOR THE ATMOSPHERIC MODULE IS NOW DEFINED
$
$
PARAM   IUNITS  =  7HENGLISH    $   ENGLISH UNITS
PARAM   DELH    =   1000.       $   ALTITUDE INCREMENT FOR
                                $   OUTPUT TABLE IN FEET
PARAM   H1      =      0.       $   GROUND LEVEL ALTITUDE(FT)
PARAM   NH0     =     36        $   NUMBER OF ALTITUDE
                                $   VALUES IN OUTPUT TABLE
PARAM   P1      =   2116.22     $   GROUND LEVEL PRESSURE
                                $   IN POUNDS/SQ.FT.
PARAM   IPRINT  =      3        $   REQUEST ALL PRINT
$
$
$     THE ATMOSPHERIC MODULES IS NOW EXECUTED FOR THE INPUT PROFILE
$
$
EXECUTE  ATM   $
```

```
$
$
$     THE FOLLOWING USER PARAMETERS ARE REQUIRED FOR ALL
$     NOISE PREDICTION RUNS
$
$
PARAM   ALT     =   35000.0     $   ALTITUDE IN FEET
PARAM   AE      =       1.      $   SET ENGINE REFERENCE AREA
                                $   TO 1 SQUARE FOOT
PARAM   RS      =      10.      $   SET EDGE OF FARFIELD TO TEN FEET
PARAM   NENG    =       2       $   TWO ENGINES
PARAM   IOUT    =       1       $   PRINT dB VALUES ONLY
PARAM   AJ      =       3.45    $   PRIMARY JET EXIT AREA FT**2
$
$
$     A COMMAND PROCEDURE IS USED TO NORMALIZE THE INPUT
$     AND EXECUTE THE SGLJET MODULE AS FOLLOWS
$
$
DATA DM=SGLJET $
      EXECUTE  APM Z=ALT            $   FETCH ATMOSPHERIC PARAMETERS
      EVALUATE RHOTOT =  PJ / TJ / 1715.
                                    $   COMPUTE TOTAL DENSITY
                                    $   FROM IDEAL GAS LAW
      EVALUATE RHOJ = (PRES / PJ) ** (1./1.4)*RHOTOT/RHOA
                                    $   COMPUTE STATIC DENSITY
                                    $   FROM ISENTROPIC RELATION
      EVALUATE TJ = TJ / TA         $   NORMALIZE TEMPERATURE
      EVALUATE VJ = VJ / CA         $   NORMALIZE VELOCITY
      EXECUTE  SGLJET SGLJET=JETUNIT
                                    $   EXECUTE SGLJET MODULE
                                    $   WITH NAME OVERRIDE
      EVALUATE MJ = SQRT(((PJ/PRES)**(0.4/1.4) - 1.)*2./0.4)
                                    $   COMPUTE JET MACH NUMBER
      IF ( MJ .LT. 1. ) GOTO LOOP   $     CHECK FOR SHOCK NOISE MODULE
      EXECUTE  SAESHK SAESHK=SHKUNIT
                                    $   EXECUTE SHOCK NOISE MODULE
      LOOP CONTINUE                 $   END LOOP
END* $
$
$     THE REMAINING PARAMETERS DEFAULT EXCEPT FOR JET
$     THERMODYNAMIC VARIABLES WHICH CHANGE FOR EACH CASE
$
$======================================================================
$
$     CASE 1 IS 5000 FT ALTITUDE AND 0.385 MACH NUMBER
$
PARAM   ALT     =   5000.       $   ALTITUDE IN FEET
PARAM   MA      =      0.385    $   AIRCRAFT MACH NUMBER
PARAM   PJ      =   3345.       $   JET TOTAL PRESSURE,
                                $   POUND/SQ.FT.
PARAM   TJ      =    876.0      $   JET TOTAL TEMPERATURE,
                                $   DEG RANKINE
PARAM   VJ      =   1328.       $   JET VELOCITY, FT/SEC
$
$      PREDICT THE NOISE FOR CASE 1
$
CALL DATA(SGLJET) JETUNIT=JET1    $
```

```
$
$=====================================================================
$
$     CASE 2 IS 35000 FT ALTITUDE AND 0.8 MACH NUMBER
$
PARAM   ALT     = 35000.        $   ALTITUDE IN FEET
PARAM   MA      =      0.8      $   AIRCRAFT MACH NUMBER
PARAM   PJ      =   4425.       $   JET TOTAL PRESSURE,
                                $   POUND/SQ.FT.
PARAM   TJ      =    818.0      $   JET TOTAL TEMPERATURE,
                                $   DEG RANKINE
PARAM   VJ      =   1588.       $   JET VELOCITY, FT/SEC
$
$       PREDICT THE NOISE FOR CASE 2
$
CALL DATA(SGLJET) JETUNIT=JET2 SHKUNIT=SHOCK2   $
$
$
ENDCS $
```

6. HSR Program Templates

This section contains ten ANOPP-HSR program templates which have been developed to provide a better understanding of what kind of problems ANOPP-HSR can handle and how to develop code to solve these problems. The Aircraft Noise Prediction Program User's Manual (Reference 2) should be kept close at hand for reference purposes. Each template begins by establishing a problem. A solution is then posed. Throughout the solutions, input and output data units will be mentioned. All of this information can be found in greater detail within the pages of Reference 2. The Aircraft Prediction Program Theoretical Manual, Parts 1 & 2, (Reference 4) should also be examined. Within these two volumes, the theoretical aspects of each module are discussed.

Template 6.1 Free Field Lossless Jet Mixing Noise Prediction

Problem: Predict the free field lossless jet mixing noise for a stationary circular nozzle, as shown in figure 5, with the following exit conditions:

jet velocity	=	340.294	m/sec
jet density	=	1.225	kg/m^3
jet total temperature	=	288.000	Kelvin
fully expanded jet area	=	1.000	m^2

and the following ambient conditions:

ambient density	=	1.225	kg/m^3
ambient temperature	=	288.0	Kelvin
ambient speed of sound	=	340.294	m/sec

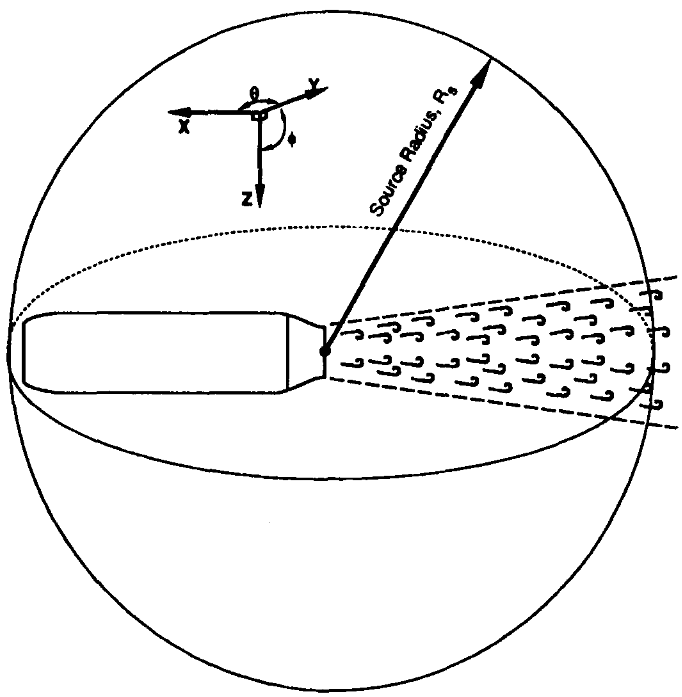

Figure 5. Stationary Circular Jet Noise Prediction

Solution: An ANOPP input file using the Single Stream Circular Jet Noise Module (SGLJET) is generated. Upon examining the input section of the Aircraft Noise Prediction User's Guide, it can be seen that the data units SAE and SFIELD are required as input. The SAE tables are obtained from the ANOPP permanent data base (unit LIBRARY) using the LOAD command. The unit SFIELD consists of three members whose values are specified by -ADDR control statements as explained in an example in Section 4.2.1. FREQ represents the third-octave band center frequencies, THETA represents the polar directivity angles, and PHI represents the azimuthal directivity angles. In this template, THETA varies from 10 to 170 degrees in 20 degree increments. Input parameters have been split into three sections. First, ambient conditions are defined. Next, the remainder of the input parameters for the SGLJET module are defined in dimensional form. Certain input parameters are required by ANOPP to be nondimensional. The output from this module will be two tables. The first is mean square pressure and the second is sound pressure level (in dB) both as a function of frequency and polar directivity angle (it is assumed that predictions for this noise source do not vary with azimuthal directivity angle).

ANOPP Listing:

```
ANOPP $
STARTCS $
$
$     Load SAE table from the ANOPP permanent data base LIBRARY
$
LOAD /LIBRARY/ SAE $
$
$     Specify the frequency and directivity angles
$
UPDATE NEWU=SFIELD SOURCE=* $
   -ADDR OLDM=* NEWM=FREQ FORMAT=4H*RS$ $
         50.     63.     80.    100.    125.    160.
        200.    250.    315.    400.    500.    630.
        800.   1000.   1250.   1600.   2000.   2500.
       3150.   4000.   5000.   6300.   8000.  10000.    $
   -ADDR OLDM=* NEWM=THETA FORMAT=4H*RS$ $
         10.   30.   50.   70.   90.  110.   130.   150.   170. $
   -ADDR OLDM=* NEWM=PHI FORMAT=4H*RS$ $
          0. $
   END* $
$
$     Define ambient conditions
$
PARAM RHOA   =     1.225  $  ambient density, kg/m3
PARAM TA     =   288.0    $  ambient temperature, Kelvin
PARAM CA     =   340.294  $  ambient speed of sound, m/sec
$
$     Define input parameters
$
PARAM AE       =    1.0    $  engine reference area, m2
PARAM AJ       =    1.0    $  area of fully expanded primary jet, m2
PARAM RHOJ     =    1.225  $  density of primary jet, kg/m3
PARAM TJ       =  288.0    $  jet total temperature, Kelvin
PARAM VJ       =  340.294  $  jet velocity relative to nozzle exit,
                           $  m/sec
PARAM RS       =   10.0    $  radial distance from nozzle exit to
                           $  observer, m
PARAM STIME    =    0.0    $  source noise calculation time, sec
PARAM MA       =    0.0    $  aircraft Mach number
PARAM NENG     =    1      $  number of engines
PARAM DELTA    =    0.0    $  engine inclination angle, deg
PARAM SCRXXX   =    3HXXX  $  table unit member identifier
```

```
PARAM SCRNNN  =    1      $  table unit member identifier
PARAM IUNITS  =    2HSI   $  define input units to be SI
PARAM IOUT    =    3      $  output code for table and printed output
PARAM IPRINT  =    3      $  printed output option code
$
$     Nondimensionalize certain input parameters
$
EVALUATE AJ    =   AJ/AE        $
EVALUATE RHOJ  =   RHOJ/RHOA    $
EVALUATE TJ    =   TJ/TA        $
EVALUATE VJ    =   VJ/CA        $
EVALUATE RS    =   RS/SQRT(AE)  $
$
$     Execute the noise module
$
EXECUTE SGLJET $
$
ENDCS $
```

Template 6.2 Free Field Lossless Jet Mixing Noise Prediction Including Suppression

Problem: Predict the free field lossless jet mixing noise for the circular nozzle, shown in figure 5 and described in template 6.1, suppressing the noise by 5 dB for all frequencies and directivity angles.

Solution: A suppression factor may be applied to any noise source using the General Suppression Module (GENSUP). As described in the ANOPP listing below, all of the input parameters necessary for the execution of the GENSUP module are defined initially for use in the SGLJET module. Also required as input are a table of mean-square pressures and table of suppression factors. The mean-square pressure table is provided by executing the SGLJET module. The table of suppression factors must be generated by the user. The suppression factor is defined as

$$<p^2> S = <p^2>_S$$

where $<p^2>_S$ is the suppressed mean-square acoustic pressure and $<p^2>$ is the unsuppressed mean-square acoustic pressure. In this template the following relation is used to establish the 5 dB suppression factor:

$$S = 10^{(-5/10)} = 0.31623$$

The suppression factor is applied as a function of frequency, polar directivity angle, and azimuthal directivity angle by means of the TABLE control statement. In this template, 24 frequencies, 9 polar directivity angles and 1 azimuthal directivity angle are defined. Since the 5 dB suppression is to be applied over all frequencies and directivity angles, there will be 216 [24 x 9 x 1] entries in the TABLE control statement.

A more detailed description of this module is available starting on pages 4-42 of the ANOPP User's Manual (ref. 2) and 5.2-1 of the ANOPP Theoretical Manual, Part 1 (ref. 4). Output is similar to that obtained from template 6.1. Results of the predictions from the SGLJET module and predictions for the suppressed noise source are provided. Two tables are generated for each prediction. The first table contains the mean-square pressure. The second table contains sound pressure level in decibels. It is important to note that the GENSUP module requires the name of the noise prediction table to which it will apply the suppression. In this example, the name of that table is SGLJET. GENSUP works with its own table called NOISE. In order to inform GENSUP to apply the suppression factor to the SGLJET noise table, the table name NOISE is equivalenced to SAEJET. This is appended to the EXECUTE control statement as indicated by

```
EXECUTE GENSUP NOISE=SGLJET  $
```

The suppressed mean-square acoustic pressure values are stored in an output table which has the same name as the input source noise table with the exception of a letter "s" appended to the name. For example, output from the GENSUP module using the SGLJET source noise table as input would be a table entitled SGLJETS(XXXNNN).

ANOPP Listing:

```
ANOPP $
STARTCS $
$
$     Load SAE table from the ANOPP permanent data base LIBRARY
```

```
$
LOAD /LIBRARY/ SAE $
$
$     Specify the frequency and directivity angles
$
UPDATE NEWU=SFIELD SOURCE=* $
   -ADDR OLDM=* NEWM=FREQ FORMAT=4H*RS$ $
          50.     63.     80.     100.    125.    160.
         200.    250.    315.     400.    500.    630.
         800.   1000.   1250.    1600.   2000.   2500.
        3150.   4000.   5000.    6300.   8000.  10000.    $
   -ADDR OLDM=* NEWM=THETA FORMAT=4H*RS$ $
          10.   30.   50.   70.   90.  110.  130.  150.  170. $
   -ADDR OLDM=* NEWM=PHI FORMAT=4H*RS$ $
           0. $
   END* $
$
$     Define ambient conditions
$
PARAM RHOA  =     1.225  $  ambient density, kg/m3
PARAM TA    =   288.0    $  ambient temperature, Kelvin
PARAM CA    =   340.294  $  ambient speed of sound, m/sec
$
$     Define input parameters
$
PARAM AE       =     1.0    $  engine reference area, m2
PARAM AJ       =     1.0    $  area of fully expanded primary jet, m2
PARAM RHOJ     =     1.225  $  density of primary jet, kg/m3
PARAM TJ       =   288.0    $  jet total temperature, Kelvin
PARAM VJ       =   340.294  $  jet velocity relative to nozzle exit,
                            $  m/sec
PARAM RS       =    10.0    $  radial distance from nozzle exit to
                            $  observer, m
PARAM STIME    =     0.0    $  source noise calculation time, sec
PARAM MA       =     0.0    $  aircraft Mach number
PARAM NENG     =     1      $  number of engines
PARAM DELTA    =     0.0    $  engine inclination angle, deg
PARAM SCRXXX   =     3HXXX  $  table unit member identifier
PARAM SCRNNN   =     1      $  table unit member identifier
PARAM IUNITS   =     2HSI   $  define input units to be SI
PARAM IOUT     =     3      $  output code for table and printed output
PARAM IPRINT   =     3      $  printed output option code
$
$     Nondimensionalize certain input parameters
$
EVALUATE AJ    =  AJ/AE        $
EVALUATE RHOJ  =  RHOJ/RHOA    $
EVALUATE TJ    =  TJ/TA        $
EVALUATE VJ    =  VJ/CA        $
EVALUATE RS    =  RS/SQRT(AE)  $
$
$     Execute the noise module
$
EXECUTE SGLJET $
$
$     The following input parameters are required for execution of the
$     GENSUP module.  Since each of these parameters has already been
$     defined for use in the SGLJET module, they do not need to be
```

```
$     re-defined, unless a value has changed:
$
$           SCRXXX       RHOA         IUNITS
$           SCRNNN       CA           IOUT
$           IPRINT
$
$     Define the suppression factor for each frequency and directivity angle
$
TABLE SUPPRESS(FACTOR) 1 SOURCE=* $
   INT= 1 2
   IND1= RS 24 2 2
          50.     63.     80.    100.    125.    160.
         200.    250.    315.    400.    500.    630.
         800.   1000.   1250.   1600.   2000.   2500.
        3150.   4000.   5000.   6300.   8000.  10000.
   IND2= RS 9 2 2
          10.    30.    50.    70.    90.   110.   130.   150.   170.
   IND3= RS 1 2 2
           0.
   DEP = RS
            .31623 .31623 .31623 .31623 .31623 .31623 .31623 .31623
            .31623 .31623 .31623 .31623 .31623 .31623 .31623 .31623
            .31623 .31623 .31623 .31623 .31623 .31623 .31623 .31623
            .31623 .31623 .31623 .31623 .31623 .31623 .31623 .31623
            .31623 .31623 .31623 .31623 .31623 .31623 .31623 .31623
            .31623 .31623 .31623 .31623 .31623 .31623 .31623 .31623
            .31623 .31623 .31623 .31623 .31623 .31623 .31623 .31623
            .31623 .31623 .31623 .31623 .31623 .31623 .31623 .31623
            .31623 .31623 .31623 .31623 .31623 .31623 .31623 .31623
            .31623 .31623 .31623 .31623 .31623 .31623 .31623 .31623
            .31623 .31623 .31623 .31623 .31623 .31623 .31623 .31623
            .31623 .31623 .31623 .31623 .31623 .31623 .31623 .31623
            .31623 .31623 .31623 .31623 .31623 .31623 .31623 .31623
            .31623 .31623 .31623 .31623 .31623 .31623 .31623 .31623
            .31623 .31623 .31623 .31623 .31623 .31623 .31623 .31623
            .31623 .31623 .31623 .31623 .31623 .31623 .31623 .31623
            .31623 .31623 .31623 .31623 .31623 .31623 .31623 .31623
            .31623 .31623 .31623 .31623 .31623 .31623 .31623 .31623
            .31623 .31623 .31623 .31623 .31623 .31623 .31623 .31623
            .31623 .31623 .31623 .31623 .31623 .31623 .31623 .31623
            .31623 .31623 .31623 .31623 .31623 .31623 .31623 .31623
            .31623 .31623 .31623 .31623 .31623 .31623 .31623 .31623
            .31623 .31623 .31623 .31623 .31623 .31623 .31623 .31623
            .31623 .31623 .31623 .31623 .31623 .31623 .31623 .31623
            .31623 .31623 .31623 .31623 .31623 .31623 .31623 .31623
            .31623 .31623 .31623 .31623 .31623 .31623 .31623 .31623
            .31623 .31623 .31623 .31623 .31623 .31623 .31623 .31623
   END* $
$
$     Execute the general suppression module
$
EXECUTE GENSUP NOISE=SGLJET $
$
ENDCS $
```

<u>Template</u> 6.3 Free Field Lossless Jet Mixing and Broadband Shock Noise Prediction

Problem: Predict the free field lossless jet mixing noise using Stone's Method for the circular nozzle shown in figure 6 with the following exit conditions:

jet velocity	=	425.368	m/sec
jet Mach number	=	1.25	
jet density	=	1.225	kg/m^3
jet total temperature	=	288.000	Kelvin
fully expanded jet area	=	1.000	m^2

and the following ambient conditions:

ambient density	=	1.225	kg/m^3
ambient temperature	=	288.0	Kelvin
ambient speed of sound	=	340.294	m/sec

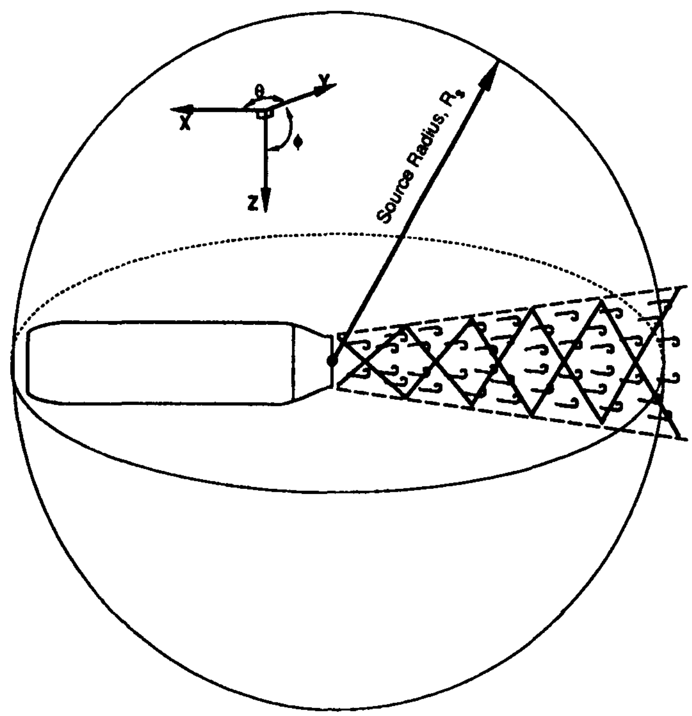

Figure 6. Stationary Circular Jet Noise Prediction Including Broadband Shock Noise

Solution: An ANOPP input file using the Stone Jet Noise Module (STNJET) is generated. The data units STNTBL and SFIELD are required as input. The data tables STNTBL are located in the ANOPP permanent data base and need to accessed using the LOAD control statement as described in template 6.1. To predict jet mixing and broadband shock noise, the STNJET module will be used. This module is also capable of predicting dual stream or coannular jets but for this template, a single stream circular nozzle will be defined. Output is provided as mean-square pressure and dB formats. More detailed information regarding the Stone Jet Noise Module can be found in the ANOPP Theoretical Manual, Part 2 beginning on page 8.6-1.

ANOPP Listing:

```
ANOPP $
STARTCS $
$
$     Load STNTBL tables from the ANOPP permanent data base LIBRARY
$
LOAD /LIBRARY/ STNTBL $
$
$     Specify the frequency and directivity angles
$
UPDATE NEWU=SFIELD SOURCE=* $
   -ADDR OLDM=* NEWM=FREQ FORMAT=4H*RS$ $
          50.     63.     80.    100.    125.    160.
         200.    250.    315.    400.    500.    630.
         800.   1000.   1250.   1600.   2000.   2500.
        3150.   4000.   5000.   6300.   8000.  10000.    $
   -ADDR OLDM=* NEWM=THETA FORMAT=4H*RS$ $
          10.    30.    50.    70.    90.   110.   130.   150.   170. $
   -ADDR OLDM=* NEWM=PHI FORMAT=4H*RS$ $
           0. $
   END* $
$
$     Define ambient conditions
$
PARAM RHOA  =     1.225   $  ambient density, kg/m3
PARAM TA    =   288.0     $  ambient temperature, Kelvin
PARAM CA    =   340.294   $  ambient speed of sound, m/sec
$
$     Define input parameters
$
PARAM AE       =    1.0     $  engine reference area, m2
PARAM A1       =    1.0     $  area of fully expanded primary jet, m2
PARAM RHO1     =    1.225   $  density of primary jet, kg/m3
PARAM T1       =  288.0     $  primary jet total temperature, Kelvin
PARAM V1       =  340.294   $  primary jet velocity rel. to nozzle exit,
                            $  m/sec
PARAM M1       =    1.25    $  primary jet Mach number
PARAM DE1      =    0.5     $  actual primary stream equivalent
                            $  diameter, m
PARAM DH1      =    0.5     $  actual primary stream hydraulic diameter, m
PARAM RS       =   10.0     $  radial distance from nozzle exit to
                            $  observer, m
PARAM STIME    =    0.0     $  source noise calculation time, sec
PARAM MA       =    0.0     $  aircraft Mach number
PARAM NENG     =    1       $  number of engines
PARAM DELTA    =    0.0     $  engine inclination angle, deg
PARAM SCRXXX   =    3HXXX   $  table unit member identifier
PARAM SCRNNN   =    1       $  table unit member identifier
PARAM IUNITS   =    2HSI    $  define input units to be SI
PARAM PLUG     =  .FALSE.   $  implies nozzle without plug
PARAM CIRCLE   =  .TRUE.    $  implies a circular nozzle
PARAM SUPER    =  .TRUE.    $  implies supersonic circular nozzle
PARAM IOUT     =    3       $  output code for table and printed output
PARAM IPRINT   =    3       $  printed output option code
$
$     Nondimensionalize input parameters
```

```
$
EVALUATE A1    =  A1/AE          $
EVALUATE RHO1  =  RHO1/RHOA      $
EVALUATE T1    =  T1/TA          $
EVALUATE V1    =  V1/CA          $
EVALUATE DE1   =  DE1/SQRT(AE)   $
EVALUATE DH1   =  DH1/SQRT(AE)   $
EVALUATE RS    =  RS/SQRT(AE)    $
$
$     Execute the noise module
$
EXECUTE STNJET $
$
ENDCS $
```

Template 6.4 Free Field Lossless Jet Mixing and Broadband Shock Noise Prediction For A Coannular Jet

Problem: Predict the free field lossless jet mixing noise for a coannular (i.e. coaxial) nozzle as shown in figure 7 with the following jet exit conditions:

		Primary Jet	Secondary Jet	
velocity	=	425.368	323.279	m/sec
Mach number	=	1.25	0.95	
density	=	1.35	1.225	kg/m^3
total temperature	=	325.000	288.000	Kelvin
fully expanded area	=	1.000	1.250	m^2

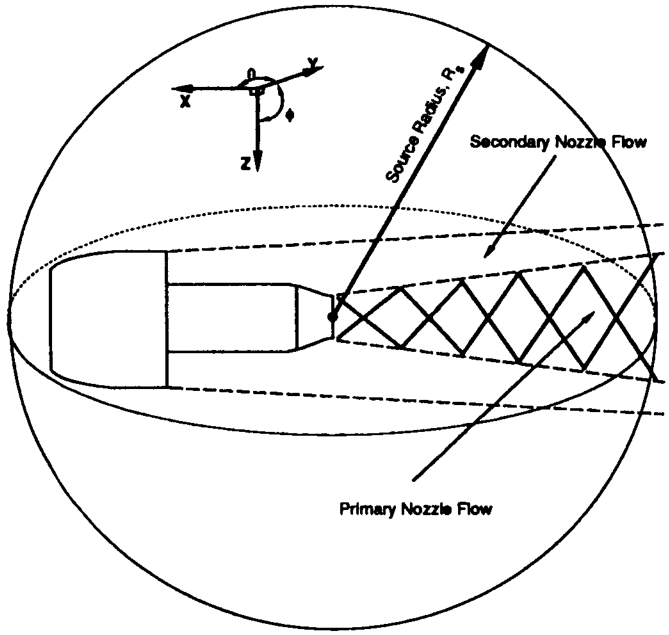

Figure 7. Stationary Coannular Jet Noise Prediction

Solution: An ANOPP input file using the Stone Jet Noise Module (STNJET) is generated. The required input data units are STNTBL and SFIELD as described in template 6.3. To predict jet mixing and broadband shock noise for a coannular nozzle, the dual stream or coannular option offered in the STNJET module will be used. The input for this template is essentially the same that is used in template 6.3 with the addition of secondary jet flow parameter definitions. Output is provided in mean-square pressure and dB formats.

ANOPP Listing:

```
ANOPP $
STARTCS $
$
$     Load STNTBL tables from the ANOPP permanent data base LIBRARY
$
LOAD /LIBRARY/ STNTBL $
```

```
$
$     Specify the frequency and directivity angles
$
UPDATE NEWU=SFIELD SOURCE=* $
   -ADDR OLDM=* NEWM=FREQ FORMAT=4H*RS$ $
         50.     63.     80.    100.    125.    160.
        200.    250.    315.    400.    500.    630.
        800.   1000.   1250.   1600.   2000.   2500.
       3150.   4000.   5000.   6300.   8000.  10000.    $
   -ADDR OLDM=* NEWM=THETA FORMAT=4H*RS$ $
         10.   30.   50.   70.   90.  110.  130.  150.  170. $
   -ADDR OLDM=* NEWM=PHI FORMAT=4H*RS$ $
          0. $
   END* $
$
$     Define ambient conditions
$
PARAM RHOA  =    1.225  $  ambient density, kg/m3
PARAM TA    =  288.0    $  ambient temperature, Kelvin
PARAM CA    =  340.294  $  ambient speed of sound, m/sec
$
$     Define input parameters
$
PARAM AE      =    1.0    $  engine reference area, m2
PARAM A1      =    1.0    $  area of fully expanded primary jet, m2
PARAM A2      =    1.25   $  area of fully expanded secondary jet, m2
PARAM RHO1    =    1.35   $  density of primary jet, kg/m3
PARAM RHO2    =    1.225  $  density of secondary jet, kg/m3
PARAM T1      =  325.0    $  primary jet total temperature, Kelvin
PARAM T2      =  288.0    $  secondary jet total temperature, Kelvin
PARAM V1      =  425.368  $  primary jet velocity rel. to nozzle exit,
                          $  m/sec
PARAM V2      =  323.279  $  secondary jet vel. rel. to nozzle exit,
                          $  m/sec
PARAM M1      =    1.25   $  primary jet Mach number
PARAM M2      =     .95   $  secondary jet Mach number
PARAM DE1     =    0.5    $  actual primary stream equivalent
                          $  diameter, m
PARAM DH1     =    0.5    $  actual primary stream hydraulic diameter, m
PARAM RS      =   10.0    $  radial distance from nozzle exit to
                          $  observer, m
PARAM STIME   =    0.0    $  source noise calculation time, sec
PARAM MA      =    0.0    $  aircraft Mach number
PARAM NENG    =    1      $  number of engines
PARAM DELTA   =    0.0    $  engine inclination angle, deg
PARAM SCRXXX  =    3HXXX  $  table unit member identifier
PARAM SCRNNN  =    1      $  table unit member identifier
PARAM IUNITS  =    2HSI   $  define input units to be SI
PARAM PLUG    =  .FALSE.  $  implies nozzle without plug
PARAM CIRCLE  =  .FALSE.  $  implies a coannular nozzle
PARAM SUPER   =  .TRUE.   $  implies supersonic circular nozzle
PARAM IOUT    =    3      $  output code for table and printed output
PARAM IPRINT  =    3      $  printed output option code
$
$     Nondimensionalize input parameters
$
EVALUATE A1   =   A1/AE        $
EVALUATE A2   =   A2/AE        $
```

```
EVALUATE RHO1  =   RHO1/RHOA      $
EVALUATE RHO2  =   RHO2/RHOA      $
EVALUATE T1    =   T1/TA          $
EVALUATE T2    =   T2/TA          $
EVALUATE V1    =   V1/CA          $
EVALUATE V2    =   V2/CA          $
EVALUATE DE1   =   DE1/SQRT(AE)   $
EVALUATE DH1   =   DH1/SQRT(AE)   $
EVALUATE RS    =   RS/SQRT(AE)    $
$
$     Execute the noise module
$
EXECUTE STNJET $
$
ENDCS $
```

Template 6.5 Standard Atmosphere and Atmospheric Absorption

Problem: Generate the standard atmosphere and standard atmospheric absorption tables.

Solution: The standard atmosphere tables have been created and stored in the ANOPP permanent data base, LIBRARY. The procedure ATMSTD, which is also stored in the the permanent data base, is used to load the tables from the data base. In order to use the procedure it must be extracted from the permanent data base using the LOAD control statement. The next step is to call the procedure ATMSTD which will load the two tables from the data base. In order to verify that the correct tables were loaded, they are included in the output listings by using the TABLIST control statement.

ANOPP Listing:

```
ANOPP $
STARTCS $
$
$      Load the procedure library from the ANOPP permanent data base
LIBRARY
$
LOAD /LIBRARY/ PROCLIB $
$
$      Call the procedure ATMSTD which loads the standard atmosphere
$      and standard atmospheric absorption tables from the permanent
$      data base.
$
CALL PROCLIB( ATMSTD ) $
$
$      List the tables ATM(TMOD) and ATM(AAC) to verify that the correct
$      tables were loaded from the permanent database
$
TABLIST LIST=FULL ATM(TMOD), ATM(AAC) $
$
ENDCS $
```

Template 6.6 Atmosphere and Atmospheric Absorption For A Non-standard Day

Problem: Generate atmospheric and atmospheric absorption tables for a 25 degree Celsius day layered atmosphere as described in figure 8.

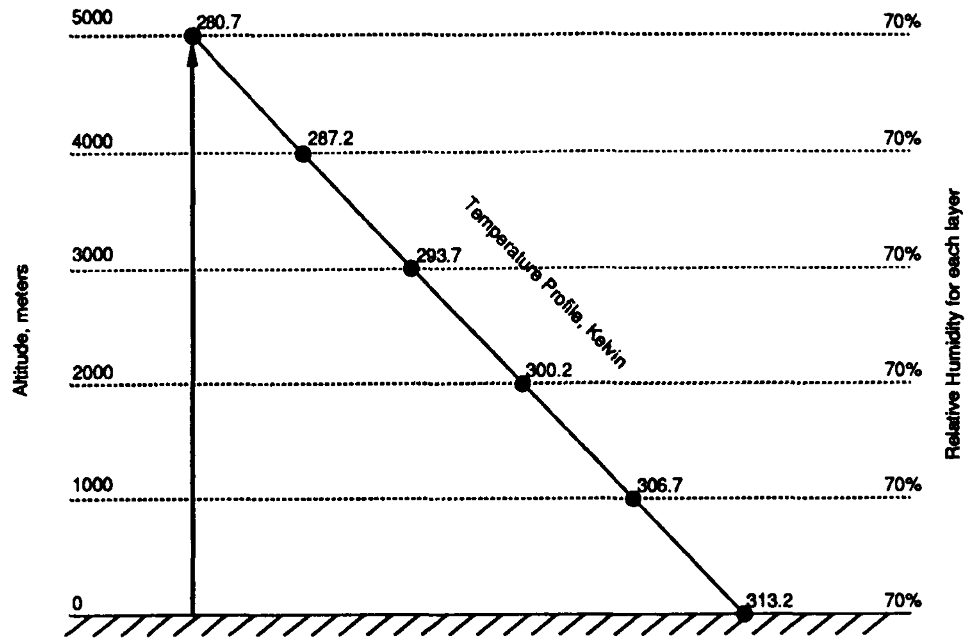

Figure 8. Layered Atmosphere for a 25 degree Celsius Day

Solution: In order to define atmospheric and atmospheric absorption tables for a non-standard, the Atmospheric Module (ATM) and the Atmospheric Absorption Module (ABS) are executed. Input to the Atmospheric Module consists of a table of altitudes, temperatures, and relative humidities and several input parameters as listed below. Input to the Atmospheric Absorption Module includes the frequency portion of the SFIELD unit member, the table generated by the Atmospheric Module (ATM(TMOD) - see below) and several input parameters which are listed below. Both modules use two of the same input parameters, IPRINT and IUNITS. Input parameters common to all modules only need to be defined once. If any input parameter needs to be changed for a particular module, that parameter must be redefined before execution of the module. Output from the Atmospheric Module includes a table called ATM(TMOD) which consists of pressure, density, temperature, speed of sound, average speed of sound, humidity, coefficient of viscosity, coefficient of thermal conductivity, and characteristic impedance as a function of altitude. Output from the the Atmospheric Absorption Module includes the table ATM(AAC) which consists of atmospheric absorption coefficients as a function of altitude and frequency.

ANOPP Listing:

```
ANOPP $
STARTCS $
$
$     Define the input unit member ATM(IN), each record defines the
$     temperature and relative humidity at a specific altitude
$
```

```
UPDATE NEWU=      ATM SOURCE=* $
   -ADDR OLDM=* NEWM=IN FORMAT=4H3RS$ $
           0.    313.2    70.  $
        1000.    306.7    70.  $
        2000.    300.2    70.  $
        3000.    293.7    70.  $
        4000.    287.2    70.  $
        5000.    280.7    70.  $
   END* $
$
$     Define input user parameters for the Atmospheric Module
$
PARAM DELH     =   1000.    $  altitude increment for output, m
PARAM H1       =      0.    $  ground level altitude referenced to
                            $  sea level, m
PARAM IUNITS   =    2HSI    $  define input units to be SI
PARAM NHO      =      6     $  number of altitudes for output
PARAM P1       =  101325.   $  atmospheric pressure at ground level, N/m2
PARAM IPRINT   =      3     $  printed output option code
$
$     Execute the Atmospheric Module
$
EXECUTE ATM $
$
$     Specify the frequency array for input to the Atmospheric
$     Absorption Module
$
UPDATE NEWU=SFIELD SOURCE=* $
   -ADDR OLDM=* NEWM=FREQ FORMAT=4H*RS$ $
          50.     63.     80.    100.    125.    160.
         200.    250.    315.    400.    500.    630.
         800.   1000.   1250.   1600.   2000.   2500.
        3150.   4000.   5000.   6300.   8000.  10000.   $
   END* $
$
$
$     Define input user parameters for the Atmospheric Absorption
$     Module - two input parameters have already been defined for
$     use by the previous module
$
PARAM ABSINT   =     5      $  number of integration steps
PARAM SAE      =  .TRUE.    $  method option - use SAE method
$
$     Execute the Atmospheric Module
$
EXECUTE ABS $
$
ENDCS $
```

Template 6.7 Steady Flyover Using a Single Noise Source

Problem: Predict the jet mixing noise associated with an aircraft flying straight and level at an altitude of 2500 meters and a velocity of 300 meters/second. Propagate the noise to three observers 1000 meters from the center line, 1000 meters apart and 4 meters above the ground. The flight path and observer locations are shown in figure 9.

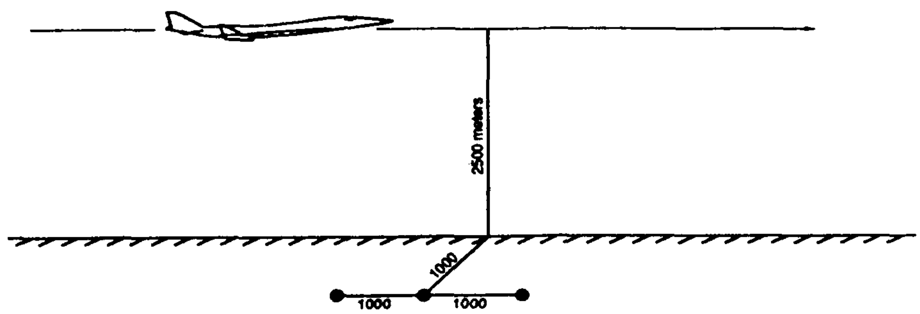

Figure 9. Observer Locations and Flight Path for Steady Flyover

Solution: This ANOPP input file combines several of the templates that have been already been described and adds additional modules which complete the job. In order to successfully execute an ANOPP input job, certain modules need to be executed in a particular order. For this template, an atmosphere must first be defined using the Atmospheric Module. Next the flight trajectory must be generated by executing the Steady Flyover Module. The observer coordinates are defined next and the Geometry Module is executed to establish a relationship between the flight trajectory and the observer locations. The noise source is defined by executing the Single Stream Circular Jet Noise Module. ANOPP does not "see" the aircraft as is shown in figure 9. ANOPP considers the aircraft to be a point source travelling along the flight trajectory which is divided into time intervals. This concept is illustrated in figure 10. The Propagation Module transfers the noise source from a source frame of reference to an observer frame of reference. Finally, the Noise Levels Module is executed to calculate various noise metrics such as overall sound pressure level, A-weighted sound pressure level, and perceived noise level as requested by the user. This template is divided into sections each starting with a brief description of function, input, and output. More detailed information can be obtained from the ANOPP User's Guide and the ANOPP Theoretical Manuals.

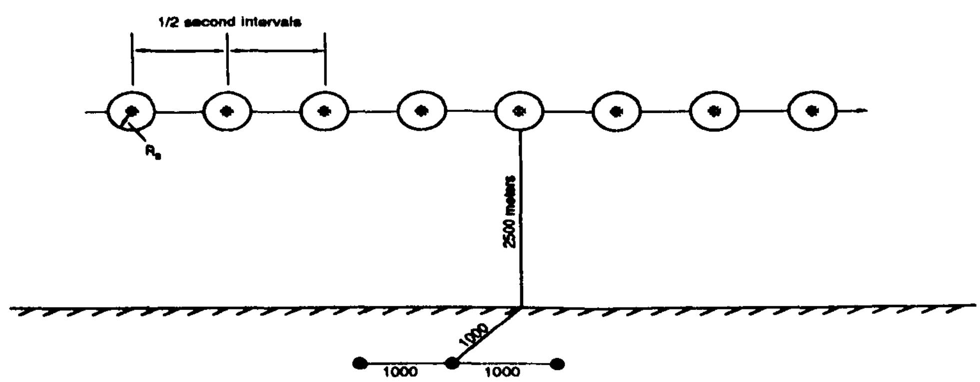

Figure 10. ANOPP interprets the aircraft's flight path as a moving point source shown here as a gray circle

ANOPP Listing:

```
ANOPP $
STARTCS $
$
$     Load SAE table from the ANOPP permanent data base LIBRARY
$
LOAD /LIBRARY/ SAE $
$
$     Specify the frequency and directivity angles
$
UPDATE NEWU=SFIELD SOURCE=* $
   -ADDR OLDM=* NEWM=FREQ FORMAT=4H*RS$ $
          50.     63.      80.    100.    125.    160.
         200.    250.     315.    400.    500.    630.
         800.   1000.    1250.   1600.   2000.   2500.
        3150.   4000.    5000.   6300.   8000.  10000.     $
   -ADDR OLDM=* NEWM=THETA FORMAT=4H*RS$ $
          10.    30.    50.    70.    90.   110.   130.   150.   170. $
   -ADDR OLDM=* NEWM=PHI FORMAT=4H*RS$ $
           0. $
   END* $
$
$     These two input parameters will be used by every module
$     executed in this template.  Since they will not be modified,
$     they are defined once before and module is executed.
$
PARAM IUNITS    =     2HSI    $  define input units to be SI
PARAM IPRINT    =     3       $  printed output option code
$
$=======================================================================
$  Atmospheric Module - ATM
$
$    The purpose of this module is to build an atmospheric table.
$    Input includes the user parameters listed below and the unit member
$    ATM(IN).  Output consists of the table ATM(TMOD) which is a table
$    of atmospheric values in dimensionless form.  The values include
$    pressure, density, temperature, speed of sound, average speed of
$    sound, humidity, coefficient of viscosity, coefficient of thermal
$    conductivity, and characteristic impedance all as a function of
$    altitude.  The ATM(TMOD) table will be used as input to the SFO,
$    GEO, and PRO modules executed later in this template.
$
$-----------------------------------------------------------------------
$
$     Define the input unit member ATM(IN), each record defines the
$     temperature and relative humidity at a specific altitude
$
UPDATE NEWU=ATM SOURCE=* $
   -ADDR OLDM=* NEWM=IN FORMAT=4H3RS$ $
           0.     288.2     70.  $
        1000.     281.7     70.  $
        2000.     275.2     70.  $
        3000.     268.7     70.  $
        4000.     262.2     70.  $
        5000.     255.7     70.  $
   END* $
$
```

```
$     Define input user parameters for the Atmospheric Module
$
PARAM DELH   =    1000.   $  altitude increment for output, m
PARAM H1     =       0.   $  ground level altitude, m
PARAM NHO    =       6    $  number of altitudes for output
PARAM P1     =  101325.   $  atmospheric pressure at ground level, N/m^2
$
$     Execute the Atmospheric Module
$
EXECUTE ATM $
$
$======================================================================
$  Steady Flyover Module - SFO
$
$   The purpose of this module is to provide flight dynamics data in
$   the case of a steady state flyover.  One record of trajectory data
$   is written to a unit member at each time step.  This module
$   requires the user parameters listed below and the unit member
$   generated by the Atmospheric Module, ATM(TMOD), as input.  SFO
$   generates two unit members as output.  FLI(PATH) contains the
$   following flight trajectory data: time, aircraft position
$   (x,y,z), Euler angles from vehicle-carried to body axis and Euler
$   angles from body to wind axis.  FLI(FLIXXX) contains flight data in
$   the following order: time, Mach number, power setting, speed of
$   sound, density, viscosity, landing gear indicator, flap setting,
$   and humidity.
$
$----------------------------------------------------------------------
$
$     Define input user parameters for the Steady Flyover Module
$
PARAM NJO     =      0      $  number of time steps already completed
PARAM IOUT    =      1      $  output in original units
PARAM J       =      1      $  initial time step
PARAM TSTEP   =      0.5    $  time interval between step, sec
PARAM Z1      =      0.0    $  altitude at brake release, m
PARAM ENGNAM  =    3HXXX    $  engine identifier name
PARAM DELTA   =      0.0    $  engine inclination angle, deg
PARAM TT      =      0.0    $  initial time, sec
PARAM VA      =    300.0    $  aircraft velocity, m/sec
PARAM XA      =  -5000.0    $  initial distance from origin, m
PARAM YA      =      0.0    $  initial lateral distance from origin, m
PARAM ZA      =   2500.0    $  initial altitude, m
PARAM THW     =      0.0    $  inclination of flight vector with respect
                            $  to horizontal, deg
PARAM PLG     =     4HUP    $  initial landing gear position
PARAM TLG     =      0.0    $  time at which landing gear position
                            $  was reset, sec
PARAM JF      =    500.0    $  final step number limit
PARAM TF      =    100.0    $  final time limit, sec
PARAM XF      =   5000.0    $  final distance limit, m
PARAM ZF      =   2500.0    $  final altitude limit, m
PARAM ALPHA   =      2.0    $  angle of attack, deg
PARAM THROT   =      1.0    $  power setting
$
$     Execute the Steady Flyover Module
$
EXECUTE SFO $
```

```
$
$=====================================================================
$  Geometry Module - GEO
$
$    The purpose of the Geometry Module is to calculate the source to
$    observer geometry.  Input parameters are given below.  Input data
$    units include ATM(TMOD), FLI(PATH), and OBSERV(COORD).  ATM(TMOD)
$    is generated by the Atmospheric Module.  The unit FLI(PATH) is
$    generated as output from the Steady Flyover Module.  OBSERV(COORD)
$    is generated using the UPDATE control statement as shown below.
$    This unit member contains the observer locations where the noise
$    source will be propagated.  The data required are the x, y, and z
$    coordinates of the observer (i.e. microphone) locations.  The value
$    of the user parameter ICOORD determines the output generated by
$    this module.  In this example, ICOORD has a value of 1 which
$    indicates that geometry associated with the body axis will be
$    output in a table called GEO(BODY).  Body axis calculations are
$    used for all of the engine noise sources while wind axis
$    calculations are used for airframe source noise.
$
$---------------------------------------------------------------------
$
$      Define the observer coordinates
$
UPDATE NEWU=OBSERV SOURCE=* $
   -ADDR OLDM=* NEWM=COORD FORMAT=4H3RS$ $
       -1000.   1000.      4.  $
           0.   1000.      4.  $
        1000.   1000.      4.  $
   END* $
$
$      Define input user parameters for the Geometry Module
$
PARAM AW      =     1.0   $  reference area, m^2
PARAM CTK     =     0.1   $  characteristic time constant
PARAM DELDB   =    20.0   $  limiting noise level down from peak, dB
PARAM MASSAC  =   416.8   $  reference mass of aircraft, kg
PARAM START   =     0.0   $  initial flight time to be considered, sec
PARAM STOP    =  1000.0   $  final flight time to be considered, sec
PARAM DELT    =     0.5   $  reception time increment, sec
PARAM DELTH   =    10.0   $  maximum polar directivity angle limit, deg
PARAM ICOORD  =     1     $  generate body axis output
PARAM DIRECT  =  .FALSE.  $  interpolate from FLI(PATH) observer
                          $  reception times based on user parameters
                          $  start, stop, delth, and delt
$
$      Execute the Geometry Module
$
EXECUTE GEO $
$
$=====================================================================
$  Single Stream Circular Jet Noise Module - SGLJET
$
$    This module predicts 1/3-octave band circular jet mixing noise
$    incorporating forward flight effects with methods developed by the
$    SAE-A21 jet noise subcommittee.  Input user parameters required by
$    this module are listed below.  The input data units required are
$    SFIELD(FREQ), SFIELD(THETA), and SFIELD(PHI).  These unit members
```

```
$    were created using the UPDATE control statement at the beginning of
$    this input deck.  The SAE data tables are also required input to
$    this module.  These tables were obtained from the permanent data
$    base using the LOAD control statement at the beginning of this
$    input deck.  The output generated by this module is a table of
$    mean-square acoustic pressure values as a  function of frequency,
$    polar directivity angle and azimuthal directivity angle.  The table
$    is entitled SGLJET(XXXNNN) where the XXX is replaced by the value
$    of the user parameter SCRXXX and the NNN is replaced by the value
$    of the user parameter SCRNNN.
$
$---------------------------------------------------------------------
$
$     Define ambient conditions
$
PARAM RHOA  =     1.225  $  ambient density, kg/m3
PARAM TA    =   288.0    $  ambient temperature, Kelvin
PARAM CA    =   340.294  $  ambient speed of sound, m/sec
$
$     Define input parameters
$
PARAM AE      =   1.0    $  engine reference area, m2
PARAM AJ      =   1.0    $  area of fully expanded primary jet, m2
PARAM RHOJ    =   1.225  $  density of primary jet, kg/m3
PARAM TJ      = 288.0    $  jet total temperature, Kelvin
PARAM VJ      = 340.294  $  jet velocity relative to nozzle exit, m/sec
PARAM RS      =  10.0    $  radial distance from nozzle exit to
                         $  observer, m
PARAM STIME   =   0.0    $  source noise calculation time, sec
PARAM MA      =   0.0    $  aircraft Mach number
PARAM NENG    =   1      $  number of engines
PARAM DELTA   =   0.0    $  engine inclination angle, deg
PARAM SCRXXX  =   3HXXX  $  table unit member identifier
PARAM SCRNNN  =   1      $  table unit member identifier
PARAM IOUT    =   3      $  output code for table and printed output
$
$     Nondimensionalize input parameters
$
EVALUATE AJ    =  AJ/AE         $
EVALUATE RHOJ  =  RHOJ/RHOA     $
EVALUATE TJ    =  TJ/TA         $
EVALUATE VJ    =  VJ/CA         $
EVALUATE RS    =  RS/SQRT(AE)   $
$
$     Execute the noise module
$
EXECUTE SGLJET $
$
$=====================================================================
$  Propagation Module - PRO
$
$    The Propagation Module takes noise data which has been generated by
$    the noise source module(s) in the source frame of reference and
$    applies all of the appropriate computations to transfer it to the
$    observer frame of of reference.  Input user parameters required by
$    this module are listed below.  Input data base units include the
$    following:
```

```
$        ATM(TMOD)         -  generated as output from the Atmospheric
$                             Module
$        ATM(AAC)          -  generated as output from the Atmospheric
$                             Absorption Module and used only if
$                             absorption effects are requested
$        GEO(GEOM)         -  generated as output from the Geometry Module
$        FLI(FLIXXX)       -  generated as output from the flight dynamics
$                             module - SFO in this template
$        YYYYYY(XXXNNN)    -  output generated by the noise source
$                             module(s) where YYYYYY is the unit name
$                             associated with the noise module(s) used to
$                             calculate the source noise - SGLJET in this
$                             example
$    Output generated by this module includes the data unit PRO(PRES)
$    which contains dimensionless mean-square pressure at the observer
$    as a function of frequency and time.
$
$--------------------------------------------------------------------------
$
$     Define input parameters for the Propagation Module
$
PARAM IOUT     =    3        $  print output in both SPL (dB) and
                             $  mean-square acoustic pressure
PARAM SIGMA    =  2.5E05     $  specific flow resistance of the
                             $  ground kg/(sec m^3)
PARAM NBAND    =    5        $  number of subbands per 1/3-octave band
PARAM SURFACE  =  4HSOFT     $  type of surface to be used in
                             $  calculating ground effects
PARAM COH      =   0.01      $  incoherence coefficient
PARAM ABSORP   =   .FALSE.   $  do not include atmospheric absorption
                             $  effects
PARAM GROUND   =  .FALSE.    $  do not include ground effects
PARAM PROTIME  =  3HXXX      $  3 letter id from unit member FLI(FLIXXX)
PARAM PROSUM   =  6HSGLJET   $  name(s) of source unit(s) to be summed
$
$     Execute the Propagation Module - a name override is used to inform
$     the Propagation Module that the Geometry Module generated the unit
$     member GEO(BODY) while the Propagation Module is expecting
$     GEO(GEOM)
$
EXECUTE PRO GEOM=BODY $
$
$==========================================================================
$  Noise Levels Module - LEV
$
$    The Noise Levels Module computes overall sound pressure level,
$    A-weighted sound pressure level, D-weighted sound pressure level
$    perceived noise level, and tone-corrected perceived noise level as
$    a function of time and observer as requested by the user.  The
$    input user parameters required by this module are listed below.
$    The Noise Levels Module uses the data unit PRO(PRES), which was
$    generated by the Propagation Module, as input.  Also required as
$    input are the data units SFIELD(FREQ) and OBSERV(COORD) which
$    both were generated using the UPDATE control statement earlier in
$    this input deck.  If tone-corrected perceived noise levels
$    calculations are requested then the data unit LEV(PNLT) is
$    generated as output.
$
```

```
$-----------------------------------------------------------------------
$
$     Define input parameters for the Noise Levels Module
$
PARAM IAWT    =    .TRUE.      $  A-weighted sound pressure level option
PARAM IDWT    =    .FALSE.     $  D-weighted sound pressure level option
PARAM IOSPL   =    .TRUE.      $  overall sound pressure level option
PARAM IPNL    =    .TRUE.      $  perceived noise level (PNL) option
PARAM IPNLT   =    .FALSE.     $  tone-corrected PNL option
PARAM MEMSUM  =  4HPRO 4HPRES  $  unit name and member name of the noise
                               $  member to be summed
$
$     Execute the Noise Levels Module
$
EXECUTE LEV $
$
ENDCS $
```

Template 6.8 Steady Flyover Using a Single Noise Source Applying Atmospheric Absorption and Ground Effects

Problem: Predict the Jet mixing noise associated with and aircraft flying straight and level at an altitude of 2500 meters. Propagate the noise source to three observers 1000 meters off the center line, 1000 meters apart and 4 meters off the ground applying atmospheric absorption and ground effects.

Solution: This template is based on template 6.7. Atmospheric absorption is applied by executing the Atmospheric Absorption Module to generate a table of absorption coefficients and setting a flag in the Propagation Module. Ground effects are applied by setting a flag in the Propagation Module. Section descriptions will only be provided for modules which have not been encountered yet.

ANOPP Listing:

```
ANOPP $
STARTCS $
$
$     Load SAE table from the ANOPP permanent data base LIBRARY
$
LOAD /LIBRARY/ SAE $
$
$     Specify the frequency and directivity angles
$
UPDATE NEWU=SFIELD SOURCE=* $
   -ADDR OLDM=* NEWM=FREQ FORMAT=4H*RS$ $
          50.     63.     80.    100.    125.    160.
         200.    250.    315.    400.    500.    630.
         800.   1000.   1250.   1600.   2000.   2500.
        3150.   4000.   5000.   6300.   8000.  10000.    $
   -ADDR OLDM=* NEWM=THETA FORMAT=4H*RS$ $
          10.    30.    50.    70.    90.   110.   130.   150.   170.     $
   -ADDR OLDM=* NEWM=PHI FORMAT=4H*RS$ $
           0. $
   END* $
$
$     These two input parameters will be used by every module executed
$     in this template.  Since they will not be modified, they are
$     defined once before and module is executed.
$
PARAM IUNITS      =      2HSI      $     define input units to be SI
PARAM IPRINT      =      3         $     printed output option code
$
$=========================================================================
$  Atmospheric Module - ATM
$
$-------------------------------------------------------------------------
$
$     Define the input unit member ATM(IN), each record defines the
$     temperature and relative humidity at a specific altitude
$
UPDATE NEWU=ATM SOURCE=* $
   -ADDR OLDM=* NEWM=IN FORMAT=4H3RS$ $
           0.    288.2    70.  $
        1000.    281.7    70.  $
        2000.    275.2    70.  $
        3000.    268.7    70.  $
```

```
        4000.    262.2    70.  $
        5000.    255.7    70.  $
   END* $
$
$     Define input user parameters for the Atmospheric Module
$
PARAM DELH  =     1000.   $  altitude increment for output, m
PARAM H1    =        0.   $  ground level altitude, m
PARAM NHO   =        6    $  number of altitudes for output
PARAM P1    =   101325.   $  atmospheric pressure at ground level, N/m2
$
$     Execute the Atmospheric Module
$
EXECUTE ATM $
$
$=======================================================================
$  Atmospheric Absorption Module - ABS
$
$   This module computes average absorption/wavelength as a function of
$   altitude and frequency.  Input parameters for this module are
$   Input data units required include SFIELD(FREQ), which has been
$   generated using the UPDATE control statement, and ATM(TMOD), which
$   has been generated by the Atmospheric Module.  The Atmospheric
$   Absorption Module generates the data table ATM(AAC) as output.
$   This table contains atmospheric absorption coefficients as a
$   function of altitude and frequency.  It will be used by the
$   Propagation Module to apply atmospheric absorption effects to the
$   noise sources.
$
$-----------------------------------------------------------------------
$
$     Define input user parameters for the Atmospheric Absorption Module
$
PARAM ABSINT   =     5      $  number of integration steps
PARAM SAE      =  .TRUE.    $  method option - use SAE ARP 866 method
$
$     Execute the Atmospheric Absorption Module
$
EXECUTE ABS $
$
$=======================================================================
$  Steady Flyover Module - SFO
$
$-----------------------------------------------------------------------
$
$     Define input user parameters for the Steady Flyover Module
$
PARAM NJO      =      0     $  number of time steps already completed
PARAM IOUT     =      1     $  output in original units
PARAM J        =      1     $  initial time step
PARAM TSTEP    =      0.5   $  time interval between step, sec
PARAM Z1       =      0.0   $  altitude at brake release, m
PARAM ENGNAM   =   3HXXX    $  engine identifier name
PARAM DELTA    =      0.0   $  engine inclination angle, deg
PARAM TT       =      0.0   $  initial time, sec
PARAM VA       =    300.0   $  aircraft velocity, m/sec
PARAM XA       =  -5000.0   $  initial distance from origin, m
PARAM YA       =      0.0   $  initial lateral distance from origin, m
```

```
PARAM ZA      =   2500.0  $  initial altitude, m
PARAM THW     =      0.0  $  inclination of flight vector with respect
                          $  to horizontal, deg
PARAM PLG     =     4HUP  $  initial landing gear position
PARAM TLG     =      0.0  $  time at which landing gear position
                          $  was reset, sec
PARAM JF      =    500.0  $  final step number limit
PARAM TF      =    100.0  $  final time limit, sec
PARAM XF      =   5000.0  $  final distance limit, m
PARAM ZF      =   2500.0  $  final altitude limit, m
PARAM ALPHA   =      2.0  $  angle of attack, deg
PARAM THROT   =      1.0  $  power setting
$
$     Execute the Steady Flyover Module
$
EXECUTE SFO $
$
$======================================================================
$  Geometry Module - GEO
$
$----------------------------------------------------------------------
$
$     Define the observer coordinates
$
UPDATE NEWU=OBSERV SOURCE=* $
   -ADDR OLDM=* NEWM=COORD FORMAT=4H3RS$ $
       -1000.   1000.     4.  $
           0.   1000.     4.  $
        1000.   1000.     4.  $
   END* $
$
$     Define input user parameters for the Geometry Module
$
PARAM AW      =      1.0  $  reference area, m^2
PARAM CTK     =      0.1  $  characteristic time constant
PARAM DELDB   =     20.0  $  limiting noise level down from peak, dB
PARAM MASSAC  =    416.8  $  reference mass of aircraft, kg
PARAM START   =      0.0  $  initial flight time to be considered, sec
PARAM STOP    =   1000.0  $  final flight time to be considered, sec
PARAM DELT    =      0.5  $  reception time increment, sec
PARAM DELTH   =     10.0  $  maximum polar directivity angle limit, deg
PARAM ICOORD  =    1      $  generate body axis output
PARAM DIRECT  =  .FALSE.  $  interpolate from FLI(PATH) observer
                          $  reception times based on user parameters
                          $  start, stop, delth, and delt
$
$     Execute the Geometry Module
$
EXECUTE GEO $
$
$======================================================================
$  Single Stream Circular Jet Noise Module - SGLJET
$
$----------------------------------------------------------------------
$
$     Define ambient conditions
$
PARAM RHOA  =      1.225  $  ambient density, kg/m^3
```

```
PARAM TA     =   288.0    $  ambient temperature, Kelvin
PARAM CA     =   340.294  $  ambient speed of sound, m/sec
$
$     Define input parameters
$
PARAM AE       =    1.0    $  engine reference area, m2
PARAM AJ       =    1.0    $  area of fully expanded primary jet, m2
PARAM RHOJ     =    1.225  $  density of primary jet, kg/m3
PARAM TJ       =  288.0    $  jet total temperature, Kelvin
PARAM VJ       =  340.294  $  jet velocity relative to nozzle exit, m/sec
PARAM RS       =   10.0    $  radial distance from nozzle exit to
                           $  observer, m
PARAM STIME    =    0.0    $  source noise calculation time, sec
PARAM MA       =    0.0    $  aircraft Mach number
PARAM NENG     =    1      $  number of engines
PARAM DELTA    =    0.0    $  engine inclination angle, deg
PARAM SCRXXX   =    3HXXX  $  table unit member identifier
PARAM SCRNNN   =    1      $  table unit member identifier
PARAM IOUT     =    3      $  output code for table and printed output
$
$     Nondimensionalize input parameters
$
EVALUATE AJ     =   AJ/AE          $
EVALUATE RHOJ   =   RHOJ/RHOA      $
EVALUATE TJ     =   TJ/TA          $
EVALUATE VJ     =   VJ/CA          $
EVALUATE RS     =   RS/SQRT(AE)    $
$
$     Execute the noise module
$
EXECUTE SGLJET $
$
$======================================================================
$  Propagation Module - PRO
$
$----------------------------------------------------------------------
$
$     Define input parameters for the Propagation Module
$
PARAM IOUT      =     3        $  print output in both SPL (dB) and
                               $  mean-square acoustic pressure
PARAM SIGMA     =  2.5E05      $  specific flow resistance of the
                               $  ground kg/(sec m3)
PARAM NBAND     =     5        $  number of subbands per 1/3-octave band
PARAM SURFACE   =   4HSOFT     $  type of surface to be used in calculating
                               $  ground effects
PARAM COH       =     0.01     $  incoherence coefficient
PARAM PROTIME   =    3HXXX     $  3 letter id from unit member FLI(FLIXXX)
PARAM PROSUM    =  6HSGLJET    $  name(s) of source unit(s) to be summed
$
$     In order to include atmospheric absorption and ground effects,
$     these two input parameters are given a value of TRUE
$
PARAM ABSORP =   .TRUE.  $  include atmospheric absorption effects
PARAM GROUND =   .TRUE.  $  include ground effects
$
$     Execute the Propagation Module - a name override is used to inform
$     the Propagation Module that the Geometry Module generated the unit
```

```
$      member GEO(BODY) while the Propagation Module is expecting
$      GEO(GEOM)
$
EXECUTE PRO GEOM=BODY $
$
$=======================================================================
$  Noise Levels Module - LEV
$
$-----------------------------------------------------------------------
$
$      Define input parameters for the Noise Levels Module
$
PARAM IAWT    =    .TRUE.       $  A-weighted sound pressure level option
PARAM IDWT    =    .FALSE.      $  D-weighted sound pressure level option
PARAM IOSPL   =    .TRUE.       $  overall sound pressure level option
PARAM IPNL    =    .TRUE.       $  perceived noise level (PNL) option
PARAM IPNLT   =    .FALSE.      $  tone-corrected PNL option
PARAM MEMSUM  =  4HPRO 4HPRES   $  unit name and member name of the noise
                                $  member to be summed
$
$      Execute the Noise Levels Module
$
EXECUTE LEV $
$
ENDCS $
```

Template 6.9 Takeoff Maneuver Using Two Noise Sources

Problem: Predict the jet mixing noise and the broadband shock noise for an aircraft during takeoff. The following lift and drag coefficients are to be used:

	angle of attack, α			
	-3.0	0.0	3.0	6.0
flap setting, δ	lift coefficient			
0	.03	.15	.29	.39
10	.07	.19	.31	.43
20	.12	.24	.36	.48
30	.15	.27	.39	.51
	drag coefficient			
0	.016	.016	.026	.045
10	.018	.019	.030	.052
20	.025	.026	.038	.062
30	.034	.037	.051	.079

The following table of thrust values, as a function of power setting and aircraft Mach number, is also to be used:

	Engine Power Setting, percent			
Mach Number	1.00	0.75	0.50	0.25
0.00	213504.	160128.	106752.	53376.
0.20	210123.	157606.	105093.	52544.
0.30	200365.	143982.	100182.	50084.
0.40	191758.	143808.	95894.	47945.

The following engine parameters are included in this prediction as the engine deck ENG(PRIM):

	Engine Power Setting							
	0.5	1.0	0.5	1.0	0.5	1.0	0.5	1.0
Aircraft Mach	Jet Area		Mass Flow Rate		Total Temp. Ratio		Pressure Ratio	
0.00	0.06	0.06	.1155	.0754	1.7371	2.9855	1.3341	1.5894
.030	0.06	0.06	.1171	.0878	1.8417	3.1560	1.3221	1.5478
.035	0.06	0.06	.1179	.0932	1.8539	3.1710	1.3245	1.5434
.050	0.06	0.06	.1017	.0506	2.1969	4.0123	1.3192	1.6638

The fuel-to-air ratio for each condition is set equal to zero (0). In order to retain the continuity of the engine deck format, all variables neccessary for each engine component (i.e. turbine, fan, jet, etc.) are included in the engine deck. In this case, rotational speed is not a relevant parameter so it is assigned a value of zero(0) as it will not be used by the jet noise routines.

The noise sources are to be summed and propagated to two observers, applying atmospheric absorption and ground effects. One observer is the FAA centerline noise measurement point and the other observer is the FAA sideline noise measurement point. EPNL values will be calculated at these locations. Figure 11 illustrates a typical flight path and the observer locations.

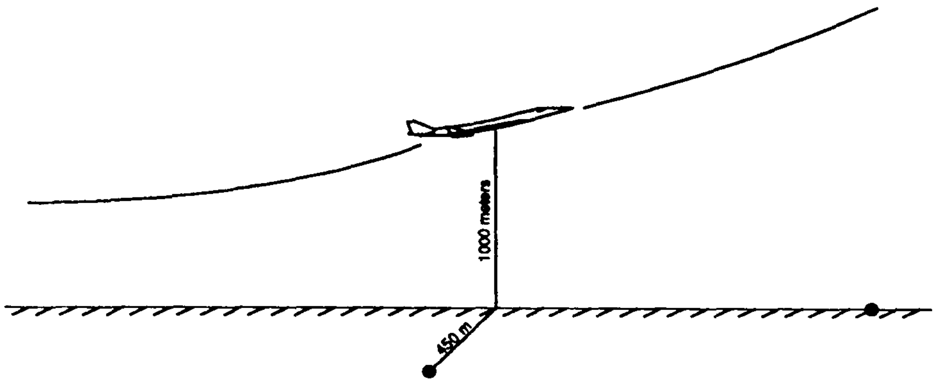

Figure 11. Observer Locations and Flight Path for Takeoff

Solution: This template is based on template 6.8. In order to obtain a takeoff flight trajectory, the Jet Takeoff Module (JTO) will be executed in the place of the Steady Flyover Module (SFO). Two ANOPP procedures are used to reduce the number of control statements required for the prediction of this time dependant problem. The ANOPP procedure library (PROCLIB) is stored in the permanent data base, LIBRARY.

ANOPP Listing:

```
ANOPP $
STARTCS $
$
$      Load SAE table and the Procedure Library from the ANOPP permanent
$      data base LIBRARY
$
LOAD /LIBRARY/ SAE PROCLIB $
$
$      Specify the frequency and directivity angles
$
UPDATE NEWU=SFIELD SOURCE=* $
   -ADDR OLDM=* NEWM=FREQ FORMAT=4H*RS$ $
          50.     63.     80.    100.    125.    160.
         200.    250.    315.    400.    500.    630.
         800.   1000.   1250.   1600.   2000.   2500.
        3150.   4000.   5000.   6300.   8000.  10000.    $
   -ADDR OLDM=* NEWM=THETA FORMAT=4H*RS$ $
          10.    30.    50.    70.    90.   110.   130.   150.   170. $
   -ADDR OLDM=* NEWM=PHI FORMAT=4H*RS$ $
           0. $
   END* $
$
$      These two input parameters will be used by every module executed
$      in this template.  Since they will not be modified, they are
$      defined once before and module is executed.
$
PARAM IUNITS      =      2HSI       $     define input units to be SI
PARAM IPRINT      =      3          $     printed output option code
$
$
```

```
$=====================================================================
$  Atmospheric Module - ATM
$
$---------------------------------------------------------------------
$
$     Define the input unit member ATM(IN), each record defines the
$     temperature and relative humidity at a specific altitude
$
UPDATE NEWU=ATM SOURCE=* $
   -ADDR OLDM=* NEWM=IN FORMAT=4H3RS$ $
           0.    288.2    70.  $
        1000.    281.7    70.  $
        2000.    275.2    70.  $
        3000.    268.7    70.  $
        4000.    262.2    70.  $
        5000.    255.7    70.  $
   END* $
$
$     Define input user parameters for the Atmospheric Module
$
PARAM DELH  =     1000.   $  altitude increment for output, m
PARAM H1    =        0.   $  ground level altitude, m
PARAM NHO   =        6    $  number of altitudes for output
PARAM P1    =   101325.   $  atmospheric pressure at ground level, N/m2
$
$     Execute the Atmospheric Module
$
EXECUTE ATM $
$
$=====================================================================
$  Atmospheric Absorption Module - ABS
$
$---------------------------------------------------------------------
$
$     Define input user parameters for the Atmospheric Absorption Module
$
PARAM ABSINT    =    5      $  number of integration steps
PARAM SAE       =  .TRUE.   $  method option - use SAE ARP 866 method
$
$     Execute the Atmospheric Absorption Module
$
EXECUTE ABS $
$
$=====================================================================
$  Jet Takeoff Module - JTO
$
$   The purpose of this module is to simulate the flight trajectory of
$   an aircraft during takeoff.  One record of trajectory data is
$   written to a unit member at each time step.  This module requires
$   the user parameters listed below.  Several input data units are
$   required.  They include the unit member generated by the
$   Atmospheric Module, ATM(TMOD), and the user provided unit members
$   ENG(DYNS), AERO(COEF), and AERO(CDLG) which are described below.
$   JTO generates two unit members as output.  FLI(PATH) contains the
$   following flight trajectory data: time, aircraft position (x,y,z),
$   Euler angles from vehicle-carried to body axis and Euler angles
$   from body to wind axis.  FLI(FLIXXX) contains flight data in the
$   following order: time, Mach number, power setting, speed of sound,
```

```
$    density, viscosity, landing gear indicator, flap setting, and
$    humidity.
$
$------------------------------------------------------------------------
$
$     Define the engine dynamics table which consists of thrust values
$     as a function of power setting (IND1) and Mach number (IND2)
$
TABLE ENG(DYNS) 1 SOURCE=* $
   INT= 0 1 2
   IND1= RS 4 1 2  1.00    0.75    0.50    0.25
   IND2= RS 4 1 2  0.00    0.20    0.30    0.40
   DEP  = RS
            213504.  160128.  106752.   53376.
            210123.  157606.  105093.   52544.
            200365.  143982.  100182.   50084.
            191758.  143808.   95894.   47945.
   END* $
$
$     Define the lift and drag coefficients as a function of angle
$     of attack and flap setting
$
TABLE AERO(COEF) 1 SOURCE=* $
   INT= 0 1 2
   IND1= RS 17 1 2 -6.0  -4.5  -3.0  -1.5   0.0   1.5   3.0   4.5   6.0
                    7.5   9.0  10.5  12.0  13.5  15.0  16.5  18.0
   IND2= RS 4 1 2   0.0  10.0  20.0  30.0
   IND3= 0 2 0 0
   DEP  = RS      -.093 -.032  .030  .090  .154  .209  .297  .333  .398
                   .457  .519  .580  .640  .699  .759  .818  .878
                  -.039  .016  .078  .143  .195  .256  .314  .375  .433
                   .494  .557  .618  .677  .735  .793  .851  .909
                   .000  .062  .120  .180  .240  .300  .360  .420  .480
                   .540  .599  .658  .716  .774  .832  .889  .945
                   .035  .096  .157  .215  .277  .337  .395  .455  .514
                   .572  .631  .690  .748  .804  .859  .915  .970
                   .018  .017  .016  .015  .016  .018  .026  .033  .045
                   .060  .080  .103  .130  .162  .201  .247  .297
                   .022  .020  .018  .018  .019  .023  .030  .039  .052
                   .069  .090  .114  .143  .177  .218  .264  .315
                   .030  .026  .025  .025  .026  .031  .038  .048  .062
                   .079  .100  .124  .153  .190  .233  .282  .334
                   .039  .036  .034  .034  .037  .042  .051  .063  .079
                   .098  .120  .147  .179  .217  .260  .311  .367
   END* $
$
$     Define the landing gear drag coefficients as a function of lift
$     coefficient
$
TABLE AERO(CDLG) 1 SOURCE=* $
   INT= 0 1 2
   IND1= RS 18 1 2   .00   .10   .15   .20   .25   .30   .35   .40   .45
                     .50   .55   .60   .65   .70   .75   .80   .85   .90
   DEP  = RS
      .0010 .0015 .0020 .0025 .0030 .0035 .0040 .0045 .0050
      .0055 .0060 .0065 .0070 .0075 .0080 .0085 .0090 .0095
   END* $
$
```

```
$     Define input user parameters for the Jet Takeoff Module
$
PARAM TAU      =     0.01    $  coefficient of rolling friction
PARAM DELT     =     0.5     $  flight time increment, sec
PARAM ALPHDOT  =     3.0     $  rate of increase in angle of
                             $  attack, deg/sec
PARAM VROT     =   103.0     $  rotation velocity, m/sec
PARAM WEIGHT   =  2.23E05    $  weight of aircraft, N
PARAM AW       =   800.0     $  aircraft wing area, m2
PARAM E        =     0.0     $  engine inclination angle, deg
PARAM DELTAF   =    20.0     $  flap setting, deg
PARAM T0       =     0.0     $  initial time, sec
PARAM X0       =     0.0     $  initial distance from origin, m
PARAM Z0       =     0.0     $  initial altitude, m
PARAM ALPH0    =     0.0     $  initial angle of attack, deg
PARAM TMAX     =   120.0     $  maximum time, sec
PARAM XMAX     =  7500.0     $  maximum distance from origin, m
PARAM ZMAX     =  3050.0     $  maximum altitude, m
PARAM ALFMAX   =     6.0     $  maximum angle of attack, deg
PARAM POW      =     1.0     $  power setting
PARAM THETAC   =    10.0     $  desired flight path angle
                             $  during climb, deg
PARAM VCLIMB   =    90.0     $  climb velocity, m/sec
PARAM TOL      =     0.001   $  error tolerance
PARAM HSTEP    =     0.1     $  integration step
PARAM HMAX     =     0.5     $  maximum integration step
PARAM HMIN     =  0.125E-10  $  minimum integration step
PARAM NENG     =     1       $  number of engines
PARAM ENGNAM   =    3HXXX    $  engine name parameter
$
$     Execute the Jet Takeoff Module
$
EXECUTE JTO $
$
$========================================================================
$  Geometry Module - GEO
$
$------------------------------------------------------------------------
$
$     Define the observer coordinates
$
UPDATE NEWU=OBSERV SOURCE=* $
   -ADDR OLDM=* NEWM=COORD FORMAT=4H3RS$ $
        6500.     0.      1.  $
        1845.   450.      1.  $
   END* $
$
$     Define input user parameters for the Geometry Module
$
PARAM AW       =    1.0    $  reference area, m2
PARAM CTK      =    0.1    $  characteristic time constant
PARAM DELDB    =   20.0    $  limiting noise level down from peak, dB
PARAM MASSAC   = 22739.7   $  reference mass of aircraft, kg
PARAM START    =    0.0    $  initial flight time to be considered, sec
PARAM STOP     = 1000.0    $  final flight time to be considered, sec
PARAM DELT     =    0.5    $  reception time increment, sec
PARAM DELTH    =   10.0    $  maximum polar directivity angle limit, deg
PARAM ICOORD   =    1      $  generate body axis output
```

```
PARAM DIRECT   =    .FALSE. $  interpolate from FLI(PATH) observer
                            $  reception times based on user parameters
                            $  start, stop, delth, and delt
$
$     Execute the Geometry Module
$
EXECUTE GEO $
$
$=========================================================================
$  Procedure TSGLJET
$
$    This procedure computes the jet exhaust noise using the Single
$    Stream Circular Jet Noise Module (SGLJET) for a time dependant
$    problem.  Input user parameters required by this procedure are
$    listed below.  The input data units required are SFIELD(FREQ),
$    SFIELD(THETA), and SFIELD(PHI).  These unit members were created
$    using the UPDATE control statement at the beginning of this
$    input deck.  Input tables required for the execution of this
$    procedure are the SAE data tables which were obtained from the
$    permanent data base using the LOAD control statement at the
$    beginning of this input deck.  This procedure requires an engine
$    deck as input.  Defined below is the jet noise engine deck
$    ENG(PRIM).  It consists of the flow state variables area,
$    fuel-to-air ratio, mass flow rate, total pressure, and total
$    temperature all as a function of engine power setting and aircraft
$    Mach number.  The last row of zeros designates a null parameter
$    (rotational speed) which is included only to keep the table format
$    consistent - rotational speed is not required input for the jet
$    table.  The procedure generates tables of mean-square acoustic
$    pressure values as a function of frequency, polar directivity angle
$    and azimuthal directivity angle.  The tables are entitled
$    SGLJET(XXXNNN) where the XXX is replaced by the value of the user
$    parameter SCRXXX.  The values of NNN range for 1 to n, where n is
$    the number of source times.
$
$-------------------------------------------------------------------------
$
$     Define the engine deck which consists of flow state variables
$
TABLE ENG(PRIM) 1 SOURCE=* $
   INT= 0 1 2
   IND1= RS 2 2 2  0.50    1.00
   IND2= RS 4 2 2  0.00    0.30    0.35    0.50
   IND3= 0 6 0 0
   DEP = RS
        0.06    0.06    0.06    0.06    0.06    0.06    0.06    0.06
        0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
        0.1155  0.0754  0.1171  0.0878  0.1179  0.0932  0.1017  0.0506
        1.7371  2.9855  1.8417  3.1560  1.8539  3.1710  2.1969  4.0123
        1.3341  1.5894  1.3221  1.5478  1.3245  1.5434  1.3192  1.6638
        0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
   END* $
$
$     Define ambient conditions
$
PARAM RHOA   =     1.225      $  ambient density, kg/m3
PARAM TA     =   288.0        $  ambient temperature, Kelvin
PARAM CA     =   340.294      $  ambient speed of sound, m/sec
```

```
$
$     Define input parameters
$
PARAM AE       =  0.06        $  engine reference area, m^2
PARAM AP       =  0.0         $  nozzle plug area, m^2
PARAM RS       = 10.0         $  radial distance from nozzle exit to
                              $  observer, m
PARAM NENG     =  1           $  number of engines
PARAM IRATIO   =  8HCONSTANT  $  constant specific heat ratio method
PARAM ITYPE    =  2           $  use cubic spline interpolation on
                              $  engine deck
PARAM DELTA    =  0.0         $  engine inclination angle, deg
PARAM SCRXXX   =  3HXXX       $  table unit member identifier
PARAM PREPRT   =  3           $  output print option for Prejet Module
PARAM IOUT     =  3           $  output code for table and printed
                              $  output
$
$     Nondimensionalize input parameters
$
EVALUATE AP  =  AP/AE       $
EVALUATE RS  =  RS/SQRT(AE) $
$
$     Execute the jet noise procedure
$
CALL PROCLIB (TSGLJET) $
$
$=====================================================================
$  Procedure TSAESHK
$
$   This procedure computes the shock cell noise using the Circular Jet
$   Shock Cell Noise Module (SAESHK) for a time dependant
$   problem.  Input user parameters required by this procedure are
$   listed below.  The input data units required by this procedure are
$   the same as those required by the previous procedure, TSGLJET.  A
$   description of each table is given under the title Procedure
$   TSGLJET above.  This procedure also generates tables of mean-square
$   acoustic pressure as a function of frequency, polar directivity
$   angle, and azimuthal directivity angle.  The tables are entitled
$   SAESHK(XXXNNN) where the XXX is replaced by the value of the user
$   parameter SCRXXX.  The values of NNN range for 1 to n, where n is
$   the number of source times.
$
$---------------------------------------------------------------------
$
$     Define the input parameters for this noise source procedure - all
$     input parameters that are required by this procedure, except for
$     the following, have already been defined for use in the Single
$     Stream Circular Jet Noise Module procedure above - they do not
$     need to be redefined since their values have not changed
$
PARAM NSHK  =     8      $  number of shocks
$
$     Execute the shock noise procedure
$
CALL PROCLIB (TSAESHK) $
$
$=====================================================================
$  Propagation Module - PRO
```

```
$
$-------------------------------------------------------------------------
$
$     Define input parameters for the Propagation Module
$
PARAM IOUT     =  3        $  print output in both SPL (dB) and
                           $  mean-square acoustic pressure
PARAM SIGMA    =  2.5E05   $  specific flow resistance of the
                           $  ground kg/(sec m³)
PARAM NBAND    =  5        $  number of subbands per 1/3-octave band
PARAM SURFACE  =  4HSOFT   $  type of surface to be used in calculating
                           $  ground effects
PARAM COH      =  0.01     $  incoherence coefficient
PARAM PROTIME  = 3HXXX     $  3 letter id from unit member FLI(FLIXXX)
$
$     This input parameter, PROSUM, is an array which contains the names
$     of all source noise tables to be summed for propagation - each
$     variable must be in a Hollerith format of the same length, i.e.,
$     if there was to be an additional source noise table called 'core'
$     to be included, it would have to be added to this list in the form
$     6HCORE    with two blank spaces left to fill the 6H format
$
PARAM PROSUM =  6HSGLJET   6HSAESHK      $
$
$     To include atmospheric absorption and ground effects, these
$     two input parameters are given a value of TRUE
$
PARAM ABSORP =  .TRUE.  $  include atmospheric absorption effects
PARAM GROUND =  .TRUE.  $  include ground effects
$
$     Execute the Propagation Module
$
EXECUTE PRO GEOM=BODY $
$
$=========================================================================
$  Noise Levels Module - LEV
$
$-------------------------------------------------------------------------
$
$     Define input parameters for the Noise Levels Module
$
PARAM IAWT    =    .TRUE.      $  A-weighted sound pressure level option
PARAM IDWT    =    .FALSE.     $  D-weighted sound pressure level option
PARAM IOSPL   =    .TRUE.      $  overall sound pressure level option
PARAM IPNL    =    .TRUE.      $  perceived noise level (PNL) option
PARAM IPNLT   =    .TRUE.      $  tone-corrected PNL option
PARAM MEMSUM  =  4HPRO 4HPRES  $  unit name and member name of the noise
                               $  member to be summed
$
$     Execute the Noise Levels Module
$
EXECUTE LEV $
$
$=========================================================================
$  Effective Noise Module - EFF
$
$    The Effective Noise Levels Module computes the effective perceived
$    noise levels (EPNL) as a function of observer location.  The input
```

```
$   user parameter required by this module is listed below.  Required
$   input data units include OBSERV(COORD), which has been previously
$   defined using the UPDATE control statement, and LEV(PNLT), which
$   has been generated by the Noise Levels Module (LEV) by setting the
$   value of the user parameter IPNLT to TRUE.  The output member
$   EFF(EPNL) is generated by this module.  EPNL values are printed in
$   the output listing if the user parameter IPRINT has a value of
$   either 2 or 3.
$
$---------------------------------------------------------------------
$
$     Define input parameter for the Effective Noise Module
$
PARAM DTIME  =   0.5          $  reception time increment, sec
$
$     Execute the Effective Noise Module
$
EXECUTE EFF $
$
ENDCS $
```

Template 6.10 Landing Maneuver Using Two Noise Sources

Problem: Predict the jet mixing noise and the shock noise associated with an aircraft executing a landing maneuver. The noise sources are to be summed and propagated to one observer, applying atmospheric absorption and ground effects. The one observer is the approach FAA centerline noise measurement point . An EPNL value will be calculated at this location. A flight trajectory and the location of the observer point is shown in figure 12.

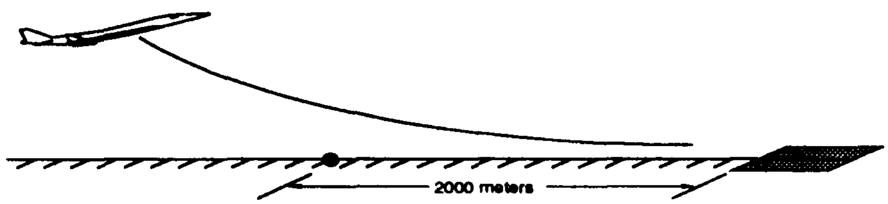

Figure 12. Observer Location and Flight Path for Landing

Solution: The structure of this template is also based on template 6.8. In order to obtain a landing flight trajectory, the Jet Landing Module (JLD) will be executed in place of the Steady Flyover Module (SFO). Two ANOPP procedures are used to reduce the number of control statements required for the prediction of this time dependant problem. The ANOPP procedure library (PROCLIB) is stored in the permanent data base, LIBRARY.

ANOPP Listing:

```
ANOPP $
STARTCS $
$
$     Load SAE table and the Procedure Library from the ANOPP
$     permanent data base LIBRARY
$
LOAD /LIBRARY/ SAE PROCLIB $
$
$     Specify the frequency and directivity angles
$
UPDATE NEWU=SFIELD SOURCE=* $
   -ADDR OLDM=* NEWM=FREQ FORMAT=4H*RS$ $
          50.     63.     80.    100.    125.    160.
         200.    250.    315.    400.    500.    630.
         800.   1000.   1250.   1600.   2000.   2500.
        3150.   4000.   5000.   6300.   8000.  10000.    $
   -ADDR OLDM=* NEWM=THETA FORMAT=4H*RS$ $
          10.    30.    50.    70.    90.   110.   130.   150.   170. $
   -ADDR OLDM=* NEWM=PHI FORMAT=4H*RS$ $
           0. $
   END* $
$
$     These two input parameters will be used by every module executed
$     in this template.  Since they will not be modified, they are
$     defined once before any module is executed.
$
PARAM IUNITS     =     2HSI      $     define input units to be SI
PARAM IPRINT     =     3         $     printed output option code
```

```
$
$===========================================================================
$  Atmospheric Module - ATM
$
$---------------------------------------------------------------------------
$
$     Define the input unit member ATM(IN), each record defines the
$     temperature and relative humidity at a specific altitude
$
UPDATE NEWU=ATM SOURCE=* $
   -ADDR OLDM=* NEWM=IN FORMAT=4H3RS$ $
           0.    288.2    70.  $
        1000.    281.7    70.  $
        2000.    275.2    70.  $
        3000.    268.7    70.  $
        4000.    262.2    70.  $
        5000.    255.7    70.  $
   END* $
$
$     Define input user parameters for the Atmospheric Module
$
PARAM DELH   =     1000.   $  altitude increment for output, m
PARAM H1     =        0.   $  ground level altitude, m
PARAM NHO    =        6    $  number of altitudes for output
PARAM P1     =   101325.   $  atmospheric pressure at ground level, N/m^2
$
$     Execute the Atmospheric Module
$
EXECUTE ATM $
$
$===========================================================================
$  Atmospheric Absorption Module - ABS
$
$---------------------------------------------------------------------------
$
$     Define input user parameters for the Atmospheric Absorption Module
$
PARAM ABSINT   =   5       $  number of integration steps
PARAM SAE      =  .TRUE.   $  method option - use SAE ARP 866 method
$
$     Execute the Atmospheric Absorption Module
$
EXECUTE ABS $
$
$===========================================================================
$  Jet Landing Module - JLD
$
$   The purpose of this module is to simulate the flight trajectory of
$   an aircraft during landing.  One record of trajectory data is
$   written to a unit member at each time step.  This module requires
$   the user parameters listed below.  Several input data units are
$   required.  They include the unit member generated by the
$   Atmospheric Module, ATM(TMOD), and the user provided unit members
$   ENG(DYNS), AERO(COEFF), and AERO(CDLG) which are described below.
$   JTO generates two unit members as output.  FLI(PATH) contains the
$   following flight trajectory data: time, aircraft position (x,y,z),
$   Euler angles from vehicle-carried to body axis and Euler angles
$   from body to wind axis.  FLI(FLIXXX) contains flight data in the
```

```
$    following order:  time, Mach number, power setting, speed of sound,
$    density, viscosity, landing gear indicator, flap setting, and
$    humidity.
$
$-------------------------------------------------------------------------
$
$      Define the engine dynamics table which consists of thrust values
$      as a function of power setting (IND1) and Mach number (IND2)
$
TABLE ENG(DYNS) 1 SOURCE=* $
   INT= 0 1 2
   IND1= RS 4 1 2  1.00     0.75     0.50     0.25
   IND2= RS 4 1 2  0.00     0.20     0.30     0.40
   DEP = RS
            213504.  160128.  106752.   53376.
            210123.  157606.  105093.   52544.
            200365.  143982.  100182.   50084.
            191758.  143808.   95894.   47945.
   END* $
$
$      Define the lift and drag coefficients as a function of angle
$      of attack and flap setting
$
TABLE AERO(COEFF) 1 SOURCE=* $
   INT= 0 1 2
   IND1= RS 17 1 2 -6.0   -4.5   -3.0   -1.5    0.0    1.5    3.0    4.5    6.0
                    7.5    9.0   10.5   12.0   13.5   15.0   16.5   18.0
   IND2= RS 4 1 2   0.0   10.0   20.0   30.0
   IND3= 0 2 0 0
   DEP = RS        -.093 -.032  .030   .090   .154   .209   .297   .333   .398
                    .457  .519  .580   .640   .699   .759   .818   .878
                   -.039  .016  .078   .143   .195   .256   .314   .375   .433
                    .494  .557  .618   .677   .735   .793   .851   .909
                    .000  .062  .120   .180   .240   .300   .360   .420   .480
                    .540  .599  .658   .716   .774   .832   .889   .945
                    .035  .096  .157   .215   .277   .337   .395   .455   .514
                    .572  .631  .690   .748   .804   .859   .915   .970
                    .018  .017  .016   .015   .016   .018   .026   .033   .045
                    .060  .080  .103   .130   .162   .201   .247   .297
                    .022  .020  .018   .018   .019   .023   .030   .039   .052
                    .069  .090  .114   .143   .177   .218   .264   .315
                    .030  .026  .025   .025   .026   .031   .038   .048   .062
                    .079  .100  .124   .153   .190   .233   .282   .334
                    .039  .036  .034   .034   .037   .042   .051   .063   .079
                    .098  .120  .147   .179   .217   .260   .311   .367
   END* $
$
$      Define the landing gear drag coefficients as a function of lift
$      coefficient
$
TABLE AERO(CDLG) 1 SOURCE=* $
   INT= 0 1 2
   IND1= RS 18 1 2    .00    .10    .15    .20    .25    .30    .35    .40    .45
                      .50    .55    .60    .65    .70    .75    .80    .85    .90
   DEP = RS
      .0010 .0015 .0020 .0025 .0030 .0035 .0040 .0045 .0050
      .0055 .0060 .0065 .0070 .0075 .0080 .0085 .0090 .0095
   END* $
```

```
$
$     Define input user parameters for the Jet Landing Module
$
PARAM DFLAP   =   30.0      $  flap setting, deg
PARAM ZD      = -125.0      $  altitude for flap and landing gear
                            $  extension, m
PARAM ZR      =  -15.24     $  altitude for end of runway crossing, m
PARAM TRV     =   -0.25     $  thrust reverser parameter, percent
PARAM VF      =    5.0      $  final aircraft velocity, m/sec
PARAM W       =    2.23E05  $  weight of aircraft, N
PARAM AW      =  800.0      $  aircraft wing area, m^2
PARAM NENG    =    1        $  number of engines
PARAM TMAX    =  213504.    $  maximum thrust, N
PARAM E       =    0.0      $  engine inclination angle, deg
PARAM TAU     =    0.01     $  coefficient of rolling friction
PARAM DELT    =    0.5      $  flight time increment, sec
PARAM ENGNAM  =    3HXXX    $  engine name parameter
PARAM LDPOW   =    1.0      $  power setting
$
$     The Jet Landing Module breaks the landing maneuver into segments.
$     The user has the option to specify the number of segments, the
$     starting altitude for each segment,an approach angle for each
$     segment, and a velocity.  The number of segments is defined in a
$     user parameter.  The altitude, flight path angle, and velocity
$     arrays are defined in the unit member LAND(PROF) as shown below:
$
PARAM NSEG    =     3                    $  number of segments
$
UPDATE NEWU=LAND SOURCE=* $
   -ADDR OLDM=* NEWM=PROF FORMAT=4H3RS$ $
       -1000.     -8.     60.  $
        -700.     -5.     50.  $
        -400.     -3.     45.  $
   END* $
$
$     Execute the Jet Landing Module
$
EXECUTE JLD $
$
$======================================================================
$  Geometry Module - GEO
$
$----------------------------------------------------------------------
$
$     Define the observer coordinates
$
UPDATE NEWU=OBSERV SOURCE=* $
   -ADDR OLDM=* NEWM=COORD FORMAT=4H3RS$ $
       -2000.      0.      1.  $
   END* $
$
$     Define input user parameters for the Geometry Module
$
PARAM AW      =      1.0   $  reference area, m^2
PARAM CTK     =      0.1   $  characteristic time constant
PARAM DELDB   =     20.0   $  limiting noise level down from peak, dB
PARAM MASSAC  =  22739.7   $  reference mass of aircraft, kg
PARAM START   =      0.0   $  initial flight time to be considered, sec
```

```
PARAM STOP     =   1000.0     $  final flight time to be considered, sec
PARAM DELT     =      0.5     $  reception time increment, sec
PARAM DELTH    =     10.0     $  maximum polar directivity angle limit, deg
PARAM ICOORD   =      1       $  generate body axis output
PARAM DIRECT   =   .FALSE.    $  interpolate from FLI(PATH) observer
                              $  reception times based on user parameters
                              $  start, stop, delth, and delt
$
$     Execute the Geometry Module
$
EXECUTE GEO $
$
$=====================================================================
$  Procedure TSGLJET
$
$---------------------------------------------------------------------
$
$     Define the engine deck which consists of flow state variables
$
TABLE ENG(PRIM) 1 SOURCE=* $
   INT= 0 1 2
   IND1= RS 2 2 2  0.50    1.00
   IND2= RS 4 2 2  0.00    0.30    0.35    0.50
   IND3= 0 6 0 0
   DEP = RS
        0.06    0.06    0.06    0.06    0.06    0.06    0.06    0.06
        0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
        0.1155  0.0754  0.1171  0.0878  0.1179  0.0932  0.1017  0.0506
        1.7371  2.9855  1.8417  3.1560  1.8539  3.1710  2.1969  4.0123
        1.3341  1.5894  1.3221  1.5478  1.3245  1.5434  1.3192  1.6638
        0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
   END* $
$
$     Define ambient conditions
$
PARAM RHOA   =      1.225  $  ambient density, kg/m3
PARAM TA     =    288.0    $  ambient temperature, Kelvin
PARAM CA     =    340.294  $  ambient speed of sound, m/sec
$
$     Define input parameters
$
PARAM AE       =      0.06       $  engine reference area, m2
PARAM AP       =      0.0        $  nozzle plug area, m2
PARAM RS       =     10.0        $  radial distance from nozzle exit
                                 $  to observer, m
PARAM NENG     =      1          $  number of engines
PARAM IRATIO   =  8HCONSTANT     $  constant specific heat ratio method
PARAM ITYPE    =      2          $  use cubic spline interpolation on
                                 $  engine deck
PARAM DELTA    =      0.0        $  engine inclination angle, deg
PARAM SCRXXX   =    3HXXX        $  table unit member identifier
PARAM PREPRT   =      3          $  output print option for Prejet Module
PARAM IOUT     =      3          $  output code for table and printed
                                 $  output
$
$     Nondimensionalize input parameters
$
EVALUATE AP   =  AP/AE           $
```

```
EVALUATE RS   = RS/SQRT(AE)   $
$
$     Execute the jet noise procedure
$
CALL PROCLIB (TSGLJET) $
$
$=======================================================================
$  Procedure TSAESHK
$
$-----------------------------------------------------------------------
$
$     Define the input parameters for this noise source procedure - all
$     input parameters that are required by this procedure, except for
$     the following, have already been defined for use in the Single
$     Stream Circular Jet Noise Module procedure above - they do not
$     need to be redefined since their values have not changed
$
PARAM NSHK  =      8      $  number of shocks
$
$     Execute the shock noise procedure
$
CALL PROCLIB (TSAESHK) $
$
$=======================================================================
$  Propagation Module - PRO
$
$-----------------------------------------------------------------------
$
$     Define input parameters for the Propagation Module
$
PARAM IOUT     =    3     $  print output in both SPL (dB) and
                          $  mean-square acoustic pressure
PARAM SIGMA    =  2.5E05  $  specific flow resistance of the
                          $  ground kg/(sec m^3)
PARAM NBAND    =    5     $  number of subbands per 1/3-octave band
PARAM SURFACE  =  4HSOFT  $  type of surface to be used in calculating
                          $  ground effects
PARAM COH      =   0.01   $  incoherence coefficient
PARAM PROTIME  =  3HXXX   $  3 letter id from unit member FLI(FLIXXX)
$
$     This input parameter, PROSUM, is an array which contains the names
$     of all source noise tables to be summed for propagation - each
$     variable must be in a Hollerith format of the same length, i.e.,
$     if there was to be an additional source noise table called 'core'
$     to be included, it would have to be added to this list in the form
$     6HCORE    with two blank spaces left to fill the 6H format
$
PARAM PROSUM =  6HSGLJET   6HSAESHK     $
$
$     In order to include atmospheric absorption and ground effects,
$     these two input parameters are given a value of TRUE
$
PARAM ABSORP =  .TRUE.  $  include atmospheric absorption effects
PARAM GROUND =  .TRUE.  $  include ground effects
$
$     Execute the Propagation Module - a name override is used to inform
$     the Propagation Module that the Geometry Module generated the unit
$     member GEO(BODY) while the Propagation Module is expecting
```

```
$       GEO(GEOM)
$
EXECUTE PRO GEOM=BODY $
$
$========================================================================
$  Noise Levels Module - LEV
$
$------------------------------------------------------------------------
$
$       Define input parameters for the Noise Levels Module
$
PARAM IAWT    =     .TRUE.        $  A-weighted sound pressure level option
PARAM IDWT    =     .FALSE.       $  D-weighted sound pressure level option
PARAM IOSPL   =     .TRUE.        $  overall sound pressure level option
PARAM IPNL    =     .TRUE.        $  perceived noise level option
PARAM IPNLT   =     .TRUE.        $  tone-corrected perceived noise level
                                  $  option
PARAM MEMSUM  =  4HPRO 4HPRES     $  unit name and member name of the noise
                                  $  member to be summed
$
$       Execute the Noise Levels Module
$
EXECUTE LEV $
$
$========================================================================
$  Effective Noise Module - EFF
$
$------------------------------------------------------------------------
$
$       Define input parameter for the Effective Noise Module
$
PARAM DTIME   =    0.5           $  reception time increment, sec
$
$       Execute the Effective Noise Module
$
EXECUTE EFF $
$
ENDCS $
```

7. Troubleshooting

The types of errors a user will encounter can be divided into two categories: programming errors and execution errors. Programming errors occur generally because a control statement format is invalid. The two most common errors of this type are:

- the omission of the $ terminator required by all ANOPP control statements - every line must included the $ terminator. If a blank line is to be used, a $ sign must be the first character in that line.
- a control statement is misspelled

Primary edit phase errors may occur due to the use of the tab key. All spaces between any words, symbols, or numbers must be entered using the space bar. Use of the tab key will result in an error. Another common error is the use of non-capital letters. Every example in this manual uses capital letters because non-capital letters will not be recognized by the ANOPP system as valid input.

Execution errors can occur for any number of reasons. Using a DEC VAX system eliminates the problems of insufficient Local Dynamic Storage (LDS) or Global Dynamic Storage (GDS) due to the nature of virtual memory.

Each module is tested and validated before it becomes a part of the functional module library. Modules are designed to test for error conditions and inform the user with an error message. With all the precautions taken to ensure that every possible error condition is tested, occasionally one will slip by.

8. References

1. Rawls, John W., Jr.: Introduction to the Use of ROTONET. Feb. 1985.

2. Gillian, Ronnie E.: Aircraft Noise Prediction Program User's Manual. NASA TM-84486, 1983.

3. Gillian, Ronnie E.; Brown, Christine G.; Bartlett, Robert W.; and Baucom, Patricia H.: ANOPP Programmer's Reference Manual for the Executive System. NASA TM X-74029, 1977.

4. Zorumski, William E.: Aircraft Noise Prediction Program Theoretical Manual, Parts 1 & 2. NASA TM-83199, 1981.

Appendix A. Summary of Functional Modules

Module Name	Module Title	Brief Description
ABS	Atmospheric Absorption Module	Computes absorption coefficient as a function of altitude & frequency using ANSI or SAE method
APM	Atmospheric Parameters Module	Retrieves ambient atmospheric data values for any altitude given as input
ATM	Atmospheric Module	Computes atmospheric properties as a function of altitude using hydrostatic method
CNLJET	Dual Stream Coannular Jet Noise Module	Predicts characteristics of a coannular jet exhaust nozzle with an inverted velocity profile
CNT	Contour Module	Generate an output data file for plotting purposes
EFF	Effective Noise Module	Computes the Effective Perceived Noise Levels
FNKAFM	Airframe Noise Module	Predicts broadband noise for the dominant components of the airframe by Fink's method
GECOR	Combustion Noise Module	Predicts noise from conventional combustors installed in gas turbine engines (SAE ARP 876)
GENSUP	General Suppression Module	Applies a noise suppression factor to a noise table produced by any ANOPP noise source module
GEO	Geometry Module	Calculates source-to-observer geometry
GETUR	Turbine Noise Module	Predicts the broadband noise and pure tones for an axial flow turbine
HDNFAN	Fan Noise Module	Predicts broadband noise and pure tones for an axial flow compressor or fan by Heidman method
JLD	Jet Landing Module	Computes aircraft takeoff parameters
JTO	Jet Takeoff Module	Computes aircraft landing parameters
LEV	Noise Levels Module	Computes OASPL, A-weighted SPL, D-weighted SPL, PNL, and/or PNLT
PREAFM	Airframe Noise Parameters Module	Generate parameters required to execute the Airframe Noise Module
PRECOR	Core Noise Parameters Module	Generate the physical parameters required for a core noise prediction module
PREFAN	Fan Noise Parameters Module	Generate the physical parameters required by Heidman's method for fan noise prediction
PREJET	Jet Noise Parameters Module	Generate the physical parameters required for the exhaust jet noise prediction modules
PRETUR	Turbine Noise Parameters Module	Generate the physical parameters required for a turbine noise prediction module
PRO	Propagation Module	Transfers noise data from the source frame of reference to the observer frame of reference
SAESHK	Circular Jet Shock Cell Noise Module	Predicts shock noise using the SAE method
SFO	Steady Flyover Module	Provide flight dynamics data for a steady flyover
SGLJET	Single Stream Circular Jet Noise Module	Predicts circular jet noise using the SAE method
SBMTUR	Smith & Bushell Turbine Noise Module	Predicts broadband noise for the 'vortex' component of an axial flow turbine
STNJET	Stone Jet Noise Module	Predicts far-field mean-square acoustic pressure for single stream and coaxial jets

Appendix B. Data Unit Directory

Data Unit	Required By	Source
AERO(CDLG)	JLD,JTO	User Supplied
AERO(COEF)	JLD,JTO	User Supplied
ATM(AAC)	PRO	ABS
ATM(IN)	ABS	User Supplied
ATM(TMOD)	ABS,GEO,JLD,JTO,PRO,SFO	ATM
CNLJET(XXXNNN)	GENSUP,PRO	CNLJET
CNLJETS(XXXNNN)	PRO	GENSUP
EFF(EPNL)	CNT	EFF
ENG(CORE1)	PRECOR	User Supplied
ENG(CORE2)	PRECOR	User Supplied
ENG(DYNS)	JLD,JTO	User Supplied
ENG(FAN1)	PREFAN	User Supplied
ENG(FAN2)	PREFAN	User Supplied
ENG(PRIM)	PREJET	User Supplied
ENG(SEC)	PREJET	User Supplied
ENG(TURBINE1)	PRETUR	User Supplied
ENG(TURBINE2)	PRETUR	User Supplied
FLI(FLIXXX)	All PRE- Modules,PRO	JLD,JTO,SFO
FLI(PATH)	GEO	JLD,JTO,SFO
FNKAFM(XXXNNN)	GENSUP,PRO	FNKAFM
FNKAFMS(XXXNNN)	PRO	GENSUP
GECOR(XXXNNN)	GENSUP,PRO	GECOR
GECORS(XXXNNN)	PRO	GENSUP
GEO(GEOM)	PRO	GEO
GETUR(XXXNNN)	GENSUP,PRO	GETUR
GETURS(XXXNNN)	PRO	GENSUP

Data Unit	Required By	Source
HDNFAN(XXXNNN)	GENSUP,PRO	HDNFAN
HDNFANS(XXXNNN)	PRO	GENSUP
JWRCOAN(...)[1]	CNLJET	Permanent Data Base - LIBRARY
LAND(PROF)	JLD	User Supplied
LEV(PNLT)	EFF,CNT	LEV
OBSERV(COORD)	CNT,EFF,GEO,LEV	User Supplied
PRO(PRES)	LEV	PRO
SAE(...)[2]	SGLJET	Permanent Data Base - LIBRARY
SAESHK(XXXNNN)	GENSUP,PRO	SAESHK
SAESHKS(XXXNNN)	PRO	GENSUP
SFIELD(FREQ)	ABS,LEV, and All Noise Source Modules	User Supplied
SFIELD(PHI)	All Noise Source Modules	User Supplied
SFIELD(THETA)	All Noise Source Modules	User Supplied
SGLJET(XXXNNN)	GENSUP,PRO	SGLJET
SGLJETS(XXXNNN)	PRO	GENSUP
SMBTUR(XXXNNN)	GENSUP,PRO	SMBTUR
SMBTURS(XXXNNN)	PRO	GENSUP
STNJET(XXXNNN)	GENSUP,PRO	STNJET
STNJETS(XXXNNN)	PRO	GENSUP
STNTBL(...)[3]	STNJET	Permanent Data Base - LIBRARY
SUPPRESS(FACTOR)	GENSUP	User Supplied

1 Includes the tables OM, PDF, S1, S2, NSF, CBF, APLHA, DIR, and MTH
2 Includes the tables MTH, NDF, OM,PDF, SLF, and SJC
3 Includes the tables SDF, JDF, and FSP

Appendix C. - TABLE Control Statement Discussion

The TABLE control statement builds an ANOPP data table which can be used as input to any of the functional modules. What follows is a brief description of the elements of a table control control statement and how these elements fit together to form a usable ANOPP table. For more detailed information refer to section 3.7.3 of the Aircraft Noise Prediction Program User's Manual (ref. 2).

User supplied tables are required as input for several ANOPP-HSR modules including the Jet Takeoff and Landing Modules (JTO and JLD, respectively) and all of the Source Noise Parameters Modules (PREJET, PRETUR, PRECOR, etc.). The table shown below defines the data unit AERO, table member (COEF), as the lift coefficients, as a function of angle of attack and flap setting, of a particular aircraft.

```
TABLE       AERO(COEF)        1     SOURCE=*     $
    INT     0    1    2
    IND1=   RS   4    1    2    -3.00     0.00     3.00     6.00
    IND2=   RS   4    1    2     0.00    10.00    20.00    30.00
    DEP=    RS                   0.03     0.15     0.29     0.39
                                 0.07     0.19     0.31     0.43
                                 0.12     0.24     0.36     0.48
                                 0.15     0.27     0.39     0.51
    END*    $
```

The table control statement always begins with the word TABLE followed by a data unit name and a table member name. In this example, AERO is the data unit and COEF is the table member. The table member name must be enclosed in parentheses. The number 1 following the data unit/table member definition indicates that this will be a type-1 data table. Type-1 data tables are the only type of data tables supported by ANOPP at this time. 'SOURCE=' specifies where the data from which the table will be built is located. The '*' indicates that the data will immediately follow the TABLE control statement. As with any ANOPP control statement, this line must end with a dollar sign symbol ($).

The next eight lines define the data unit AERO(COEF). The line beginning with INT determines which interpolation procedures will be permitted in this table. A zero (0) indicates no interpolation, a one (1) indicates linear interpolation, and a two (2) indicates cubic-spline interpolation. In this example, all three interpolation procedures will be permitted to be used on this data table. This table consists of one dependent variable as a function of two independent variables.

The next two lines define the independent variables. The IND1= line defines the angle of attack values. The character following the 'IND1=' indicates the data-type code for the angle of attack values. A value of 'RS' means the values will be real single precision. The next value in this line determines the number of independent variables in this line. There will be four (4) values of the angle of attack. The number which follows is an integer code which defines the extrapolation procedure to be used (during interpolation) if a specified value for the angle of attack falls beyond the last table value for the independent variable. A value of '1' indicates that the independent variable table value closest to the specified value will be used. The purpose of the next number is similar to that of the previous number in that it is an integer code for the extrapolation procedure to be used (during interpolation) if a specified value for the angle of attack falls before the first table value for the independent variable. A value of '2', in this case, indicates that the extrapolation is to be linear using the first two table values for the independent variable. Following these two integer codes are the values of the independent variable, angle of attack.

This same description applies to the next line, IND2=, which defines the flap angle setting as the second independent variable. After all of the independent variables have been defined, the dependent variable is defined following the symbol 'DEP='. As with the independent variable definitions, the character following the 'DEP=' symbol is a format data-type code. Once again, 'RS' indicates the values of the lift coefficient are to be read in as real single precision number. Following this character are the values of the dependent variable.

It is important to place the dependent variables in the correct order when working with more than one independent variable. In this example, ANOPP will read the flap angle settings (IND2) first and then the angle of attack values (IND1), i.e. the first flap setting will be held constant while lift coefficients for each angle of attack are defined then the second flap setting will be held constant, etc. The following table breaks the dependent variable data block down to show the location of independent variable combinations.

	angle of attack, α			
flap setting, δ	-3.0	0.0	3.0	6.0
0	.03	.15	.29	.39
10	.07	.19	.31	.43
20	.12	.24	.36	.48
30	.15	.27	.39	.51

The 'END*' symbol is the input terminator line which signifies the end of a table input section. This is also a control statement which requires a dollar sign ($) at the end of the line. The statements between the line beginning with 'TABLE' and the line beginning with 'END*' are table description lines, not control statements; therefore, they do not require dollar sign symbols ($) at the end of each line.

Multiple dependent variables can be assigned in one ANOPP table structure. To implement a multiple dependent variable table, the 'ordered position' format code is used on an additional independent variable symbol 'IND'. The additional dependent variable(s) is(are) added to the dependent variable list. In the following example, a drag coefficient table will be added to the lift coefficient table described above.

```
TABLE     AERO(COEF)     1    SOURCE=*     $
   INT    0   1   2
   IND1=  RS  4   1   2    -3.00    0.00    3.00    6.00
   IND2=  RS  4   1   2     0.00   10.00   20.00   30.00
   IND3=  0   2   0   0
   DEP=   RS                0.03    0.15    0.29    0.39
                            0.07    0.19    0.31    0.43
                            0.12    0.24    0.36    0.48
                            0.15    0.27    0.39    0.51
                            0.016   0.016   0.026   0.045
                            0.018   0.019   0.030   0.052
                            0.025   0.026   0.038   0.062
                            0.034   0.037   0.051   0.079
   END*   $
```

Ordered position has been indicated on the 'IND3=' line by using a zero (0) for the format data-type code. The next value, a two (2), indicates there are two dependent variables in this table. The extrapolation procedure values are irrelevant in this line so they have been given values of zero (0). As can be seen when examining the dependent data, the lift coefficients are listed first followed by the drag coefficients. The order of the drag coefficient values is as important as was described earlier in reference to the lift coefficient dependent variables. To visualize the order which must be observed another table is defined showing the lift and drag coefficients as a function of angle of attack and flap setting as they should appear following the 'DEP' symbol. The values within the double-line box have been taken directly from the table shown above. The flap setting and angle of attack values have been added in order to give a better description of how the table should be built. This table is the same as the table described above with the drag coefficient values appended in the correct order.

	angle of attack, α			
	-3.0	0.0	3.0	6.0
flap setting, δ	lift coefficient			
0	.03	.15	.29	.39
10	.07	.19	.31	.43
20	.12	.24	.36	.48
30	.15	.27	.39	.51
	drag coefficient			
0	.016	.016	.026	.045
10	.018	.019	.030	.052
20	.025	.026	.038	.062
30	.034	.037	.051	.079

Ordered position tables are common in complete ANOPP input decks. Engine state variable tables (i.e. engine decks) can be found in templates 6 and 7. They have the same format as the lift and drag coefficient table described here. The two independent variables are engine power setting and aircraft Mach number. The dependent variables include area, fuel-to-air ratio, mass flow rate, total pressure, and total temperature. Refer to the "Data Base Structures" heading in any of the Source Noise Parameters Module descriptions, found on pages 4-22 through 4-34 of the Aircraft Noise Prediction Program User's Manual (ref. 2), for more details concerning each individual engine deck.

Appendix D. - Input Data Unit and Table Examples

The following are common examples of the multiple directive control statements UPDATE and TABLE. Preceding each example is a brief description of the data unit.

The unit ATM(IN) defines an atmosphere. Three values for each layer of the atmosphere are required as a part of this unit. For each altitude, a temperature and a relative humidity must be defined. In this example, an atmosphere from 0 to 4000 meters will be defined in 1000 meter layers. The temperatures are given in Kelvin and the relative humidity values are given in percent.

```
UPDATE NEWU=ATM SOURCE=* $
  -ADDR OLDM=* NEWM=IN FORMAT=4H3RS$ $
        0.    313.2   70.  $
     1000.    306.7   70.  $
     2000.    300.2   70.  $
     3000.    293.7   70.  $
     4000.    287.2   70.  $
  END* $
```

The unit OBSERV(COORD) defines the observer locations. This unit is required as input if the noise source is to be propagated. It consists of the x, y, and z coordinates of the observer (i.e. microphone) location. This UPDATE statement defines two observers with coordinates (0.,0.,4.) and (1000.,1000.,4.), respectively.

```
UPDATE NEWU=OBSERV SOURCE=* $
  -ADDR OLDM=* NEWM=COORD FORMAT=4H3RS$ $
        0.      0.       4.  $
     1000.   1000.       4.  $
  END* $
```

As input to the Jet Landing Module (JLD), definition of the flight path is required. The unit LAND(PROF) defines the initial values for each of the landing profile flight segments. Due to the definition of the coordinate system in the JLD module, the x distances and flight path angle values are entered as negative numbers. Three segments are defined below:

```
UPDATE NEWU=LAND SOURCE=* $
  -ADDR OLDM=* NEWM=PROF FORMAT=4H3RS$ $
     -1000.     -8.      60.  $
      -700.     -5.      50.  $
      -400.     -3.      45.  $
  END* $
```

The unit SFIELD defines the third-octave band frequencies (FREQ), polar directivity angles (THETA), and azimuthal directivity angles (PHI). This unit defines the sound field grid. Noise predictions will be made over the range of frequencies and polar directivity angles specified. Azimuthal directivity angle is included for completeness as it is not used in any noise prediction method other than the Airframe Prediction Module (FNKAFM).

```
UPDATE NEWU=SFIELD SOURCE=* $
  -ADDR OLDM=* NEWM=FREQ FORMAT=4H*RS$ $
    50.     63.     80.    100.    125.    160.    200.    250.
   315.    400.    500.    630.    800.   1000.   1250.   1600.
  2000.   2500.   3150.   4000.   5000.   6300.   8000.  10000.   $
  -ADDR OLDM=* NEWM=THETA FORMAT=4H*RS$ $
     0.     45.     90.    135.    180. $
  -ADDR OLDM=* NEWM=PHI FORMAT=4H*RS$ $
         0. $
  END* $
```

Landing gear drag coefficients are required by the Jet Takeoff Module (JTO) and the Jet Landing Module (JLD). The table AERO(CDLG) defines the landing gear drag as a function of lift coefficient. Six values for the independent variable, lift coefficient, are specified in the IND1= line. The six corresponding dependent variable, landing gear drag coefficient, are given in the DEP line.

```
TABLE AERO(CDLG) 1 SOURCE=* $
   INT= 0 1 2
   IND1= RS 6 1 2      .00    .20    .40    .60    .80    .90
   DEP = RS
    .0010     .0025     .0045     .0065     .0085     .0095
   END* $
```

The takeoff and landing modules, JTO and JLD, also require lift and drag coefficients as a function of angle of attack and flap angle as input. The table AERO(COEF) defines the lift and drag coefficients as a function of angle of attack and flap setting. The first INT line specifies that any of the three valid interpolation options are allowed on the data presented in this table. The four values of the angle of attack are listed in the IND1= line as the first independent variable. The second independent variable, listed in the IND2= line, are the two values of the flap setting. The IND3= line uses a "0" as its first argument to indicate that there will be multiple dependent variables for each unique set of independent variables listed in this table. The number 2 which follows indicates there will be two dependent variables defined in this table, lift coefficient and drag coefficient, respectively. The dependent variables are defined after the DEP= indicator. The first line defines the lift coefficient as a function of angle of attack for the first flap setting. The second line defines the lift coefficient as a function of angle of attack for the second flap setting. Similarly, the third and fourth lines define the drag coefficient as a function of angle of attack for the first and second flap settings, respectively.

```
TABLE     AERO(COEF)       1    SOURCE=*      $
   INT        0    1    2
   IND1=      RS   4    1    2    -3.00    0.00    3.00    6.00
   IND2=      RS   2    1    2     0.00   10.00
   IND3=      0    2    0    0
   DEP=       RS                   0.03    0.15    0.29    0.39
                                   0.07    0.19    0.31    0.43
                                   0.016   0.016   0.026   0.045
                                   0.018   0.019   0.030   0.052
   END*       $
```

An engine deck must be specified if any of the noise source parameters modules, such as PREJET or PRETUR, are executed. For the execution of the Jet Noise Parameters Module (PREJET), the table ENG(PRIM) must be defined. The table ENG(SEC) must also be supplied if the jet has a secondary flow. In this example, only the table ENG(PRIM) is required. The table ENG(PRIM) defines the engine state variables for the primary jet exhaust flow as a function of engine power setting and aircraft Mach number. As indicated above, the INT= 0 1 2 allows any of the three interpolation options to be executed on this table. The two values of engine power setting are defined in the line IND1=. The next line, IND2=, defines the second independent variable - aircraft Mach number. The IND3= 0 6 indicates that six dependent variables will be assigned in this table statement. The first four lines after the DEP= define the jet area as a function of engine power setting for each aircraft Mach number. Each of the following sets of four lines define the fuel-to-air ratio, the mass flow rate, the total pressure, the total temperature, and the rotational speed, respectively. Each set is also defined as a function of engine power setting for each aircraft Mach number. The rotational speed values are not relevant to the jet noise prediction procedure; therefore, they are given values of zero. There are no restrictions on the structure of the dependent variable list. The list is divided into groups of four lines only for the explanation of this example.

```
TABLE ENG(PRIM) 1 SOURCE=* $
  INT= 0 1 2
  IND1= RS 2 2 2    0.50    1.00
  IND2= RS 4 2 2    0.00    0.30    0.35    0.50
  IND3= 0 6 0 0
  DEP = RS
     0.06         0.06
     0.06         0.06
     0.06         0.06
     0.06         0.06
     0.0          0.0
     0.0          0.0
     0.0          0.0
     0.0          0.0
     0.1155       0.0754
     0.1171       0.0878
     0.1179       0.0932
     0.1017       0.0506
     1.7371       2.9855
     1.8417       3.1560
     1.8539       3.1710
     2.1969       4.0123
     1.3341       1.5894
     1.3221       1.5478
     1.3245       1.5434
     1.3192       1.6638
     0.0          0.0
     0.0          0.0
     0.0          0.0
     0.0          0.0
  END* $
```

Another table required as input to the takeoff and landing modules (JTO and JLD) is the engine thrust table. The table ENG(DYNS) defines engine thrust as a function of engine power setting and aircraft Mach number. The thrust values given in this table are the sum of the thrusts of each individual engine. This table is built using the same format that has been previously described in this section. The first independent variable is the engine power setting. The second independent variable is the aircraft Mach number. The thrust values, given as a function of power setting for each Mach number, are given as the dependent variables.

```
TABLE ENG(DYNS) 1 SOURCE=* $
  INT= 0 1 2
  IND1= RS 4 1 2    1.00    0.75    0.50    0.25
  IND2= RS 4 1 2    0.00    0.20    0.30    0.40
  DEP = RS
      213504.  160128.  106752.   53376.
      210123.  157606.  105093.   52544.
      200365.  143982.  100182.   50084.
      191758.  143808.   95894.   47945.
  END* $
```

In order to define a suppression factor for use by the General Suppression Module (GENSUP), the table SUPPRESS(FACTOR) must be generated by the user. This table defines a suppression factor as a function of frequency, polar directivity angle, and azimuthal directivity angle. The INT= 1 2 line indicates that only linear interpolation or cubic-spline interpolation will be permitted on this table. The frequencies are defined as the the first independent variable. The polar directivity angles are defined as the second independent variable and the azimuthal directivity angles are defined as the third independent variable. An explanation of how to calculate the suppression factor is given in the description of Template 6.2 of this manual. The suppression factor is defined in the table as a function of frequency for each polar directivity angle for each azimuthal angle. In this example, the same suppression factor is applied over the entire range of frequencies and directivity angles. The values are arranged in five (the number of polar directivity angles) groups of twelve (the number of frequencies) numbers for each azimuthal angle as shown below.

```
      TABLE SUPPRESS(FACTOR) 1 SOURCE=* $
        INT= 1 2
        IND1= RS 12 2 2
                 50.      63.      80.      100.
                500.     630.     800.     1000.
               5000.    6300.    8000.    10000.
        IND2= RS 5 2 2
                  0.      45.      90.      135.      180.
        IND3= RS 1 2 2
                  0.
        DEP = RS
            .31623   .31623   .31623   .31623   .31623   .31623
            .31623   .31623   .31623   .31623   .31623   .31623
            .31623   .31623   .31623   .31623   .31623   .31623
            .31623   .31623   .31623   .31623   .31623   .31623
            .31623   .31623   .31623   .31623   .31623   .31623
            .31623   .31623   .31623   .31623   .31623   .31623
            .31623   .31623   .31623   .31623   .31623   .31623
            .31623   .31623   .31623   .31623   .31623   .31623
            .31623   .31623   .31623   .31623   .31623   .31623
            .31623   .31623   .31623   .31623   .31623   .31623
  END* $
```

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	February 1992	Contractor Report

4. TITLE AND SUBTITLE
An Introduction to High Speed Aircraft Noise Prediction

5. FUNDING NUMBERS
NAS1-19000
537-03-20

6. AUTHOR(S)
Mark R. Wilson

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Lockheed Engineering and Sciences Company
144 Research Drive
Hampton, VA 23666

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23665-5225

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA CR-189582

11. SUPPLEMENTARY NOTES
Langley Technical Monitor: Robert A. Golub
Interim Report

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified-Unlimited

Subject Category: 71

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The purpose of this document is to introduce users to the Aircraft Noise Prediction Program's High Speed Research prediction system (ANOPP-HSR). Included within this mini-manual is an introduction which gives a brief overview of the ANOPP system and the components of the HSR prediction method. ANOPP information resources are given. Twelve of the most common ANOPP-HSR control statements are described. Each control statement's purpose and format are stated and relevant examples are provided. More detailed examples of the use of the control statements are presented in the manual along with ten ANOPP-HSR templates. The purpose of the templates is to provide the user with working ANOPP-HSR programs which can be modified to serve particular prediction requirements. Also included in this manual is a brief discussion of common errors and how to solve these problems. The appendices include the following useful information: a summary of all ANOPP-HSR functional research modules, a data unit directory, a discussion of one of the more complex control statements, and input data unit and table examples.

14. SUBJECT TERMS
Aircraft Noise, Noise Prediction, High Speed Aircraft, Jet Noise, Broadband Shock Noise, Mixing Noise

15. NUMBER OF PAGES
80

16. PRICE CODE
A05

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified		