NASA Contractor Report 181689

AIRCRAFT NOISE PREDICTION PROGRAM PROPELLER ANALYSIS SYSTEM IBM-PC VERSION USER'S MANUAL VERSION 2.0

SANDRA K. NOLAN

PRC System Services, a Division of
Planning Research Corporation
Hampton, VA 23666

Contract NAS1-18000

August 1988

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

Contents

List of Figures

List of Tables

INTRODUCTION

Overview

The IBM-PC version of the Aircraft Noise Prediction Program (ANOPP) Propeller Analysis System (PAS) is a set of computational programs for predicting the aerodynamics, performance and noise of propellers. The ANOPP-PAS is a subset of a larger version of ANOPP which can be executed on CDC or VAX computers.

This manual provides a description of the ANOPP-PAS system and its capabilities, and instructions on how to install and use the system on an IBM-XT or IBM-AT Personal Computer. Section I provides an overview of the ANOPP-PAS prediction capabilities. Section II contains the IBM-PC system requirements, a list of programs on the IBM-PC version of ANOPP-PAS diskettes, and instructions for installation of the system. Section III describes the system design and includes overviews of the programs, data components, and terminology that make up ANOPP-PAS. Section IV covers ANOPP-PAS usage. Section V provides documentation and examples of how to use the 9 data entry programs, called preprocessors.

Section VI documents the installed ANOPP functional modules available on the system. These are the coded modules that perform specific acoustic-related functions and that are available for user execution during ANOPP runs. A brief functional description, input and output documentation, and examples are given for each functional module. For a detailed description of the methodology for each module, the user should refer to the ANOPP Theoretical Manual (ref. 1).

Appendixes A to C support the main sections: Appendix A contains a glossary of ANOPP terms, acronyms, and commands; Appendix B contains a summary of the ANOPP-PAS functional modules; and Appendix C contains techniques for error diagnostics and recovery.

All ANOPP-PAS programs have been compiled using Ryan McFarland Corporation's IBM Personal Computer Professional FORTRAN. The system requires the use of IBM's Disk Operating System (DOS), version 2.1 or higher. The user should have a copy of the IBM Personal Computer DOS Reference Manual (ref. 2).

ANOPP is also documented in the "Aircraft Noise Prediction Program Theoretical Manual" (ref. 1), and the "Aircraft Noise Prediction Program Users' Manual" (ref. 3).

The ANOPP Theoretical Manual provides a rigorous description of the noise prediction methods implemented in ANOPP. Equations, diagrams, and technical references allow the user to select the methods applicable to the problem. Data requirements are described carefully and limiting values are given for each data item. Every ANOPP user should have a copy of the Theoretical Manual in addition to this manual.

The ANOPP User's Manual provides the instructions needed to solve a wide variety of aircraft noise prediction problems. Input preparation, method selection, data storage, and programming shortcuts are described. The introduction of this manual contains brief overviews of the noise prediction problem, the capabilities and advantages of the ANOPP software, and the organization of the manual.

*NOTE: The user should refer to "Differences between the IBM-PC version of ANOPP and other versions", in Section III of this manual, before using reference 3.

Description of Predictions

The key feature of the ANOPP-PAS is its capability to predict wind tunnel and flyover noise with a limited amount of input data. The predictions require knowledge of the propeller geometry, propeller operating state, source to observer geometry, and atmospheric data. Table 1 provides a summary of the input data requirements.

A total of 18 program modules make up ANOPP-PAS. A system flowchart is shown in figure 1. The user has a choice of using the original Blade Geometry modules or the Improved Blade Geometry modules. The original Blade Shape (RBS) module and the Improved Blade Shape (IBS) module read the propeller geometry and store it in a convenient form for use by other modules. The original Blade Section Aerodynamics (RBA) Module and the Improved Blade Section Aerodynamics (IBA) Module compute the blade pressure and blade section lift distributions. The original Blade Section Boundary Layer (BLM) Module and the Improved Blade Section Boundary (IBL) Module compute the blade skin friction and section drag distributions.

The computed power coefficient is matched to the measured power coefficient using the Propeller Performance (PRP) Module. An initial guess of the blade 3/4 radius pitch angle is made and the PRP Module is executed. The computed power coefficient is compared to the measured value. Iterations will continue until the computed and measured power coefficients converge. Thus, the absorbed power for the predictions match the measured data, but the blade 3/4 radius pitch angles differ to account for limitations in the method. Then the final blade pressure and skin friction distributions are determined using the Propeller Loads (PLD) Module.

The propeller noise signature is predicted by the Subsonic Propeller Noise (SPN) Module using the blade geometry and performance data. The method is a full-surface implementation of the solution of the Ffowcs Williams-Hawkings equation as presented by Farassat (ref. 1). The module produces acoustic time histories and narrowband spectra of the loading, thickness, and total noise. The broadband noise due to the interaction of the blade turbulent boundary layer with the trailing edge is predicted by the Propeller Trailing Edge (PTE) Module.

The variation of atmospheric state variables and atmospheric absorption coefficients are computed with the Atmospheric (ATM) and Atmospheric Absorption (ABS) Modules, respectively. The aircraft flight path is defined by the Steady Flyover (SFO) Module and the range and directivity angles from the observer to the source at sound emission are computed by the Geometry (GEO) Module. The Tone Propagation (PRT) Module applies Doppler, spherical spreading, characteristic impedance, atmospheric absorption, and ground effects corrections in propagating the narrowband spectra from the source to the observer. The Propagation (PRO) Module applies spherical spreading, characteristic impedance, atmospheric absorption and ground effects corrections in propagating the 1/3 octave frequency band spectra from source to observer. Finally, the Noise Levels (LEV) Module computes levels such as Overall Sound Pressure Level (OASPL) and A-Weighted Sound Pressure Level

(L_A), and the Effective Noise (EFF) Module computes the Effective Perceived Noise Level (EPNL).

TABLE 1 - INPUT DATA REQUIREMENTS

Propeller Geometry

- Airfoil Section Coordinates
- Chord Distribution
- Twist Distribution
- Leading Edge Displacement Distribution
- Blade Length
- Number of Blade

Propeller Operating State

- Propeller RPM
- Forward Speed
- Absorbed Power

Source to Observer Geometry

- Flight Path Angle
- Propeller Angle of Attack
- Microphone Positions

Atmospheric Data

- Ambient Temperature Profile
- Ground Level Pressure

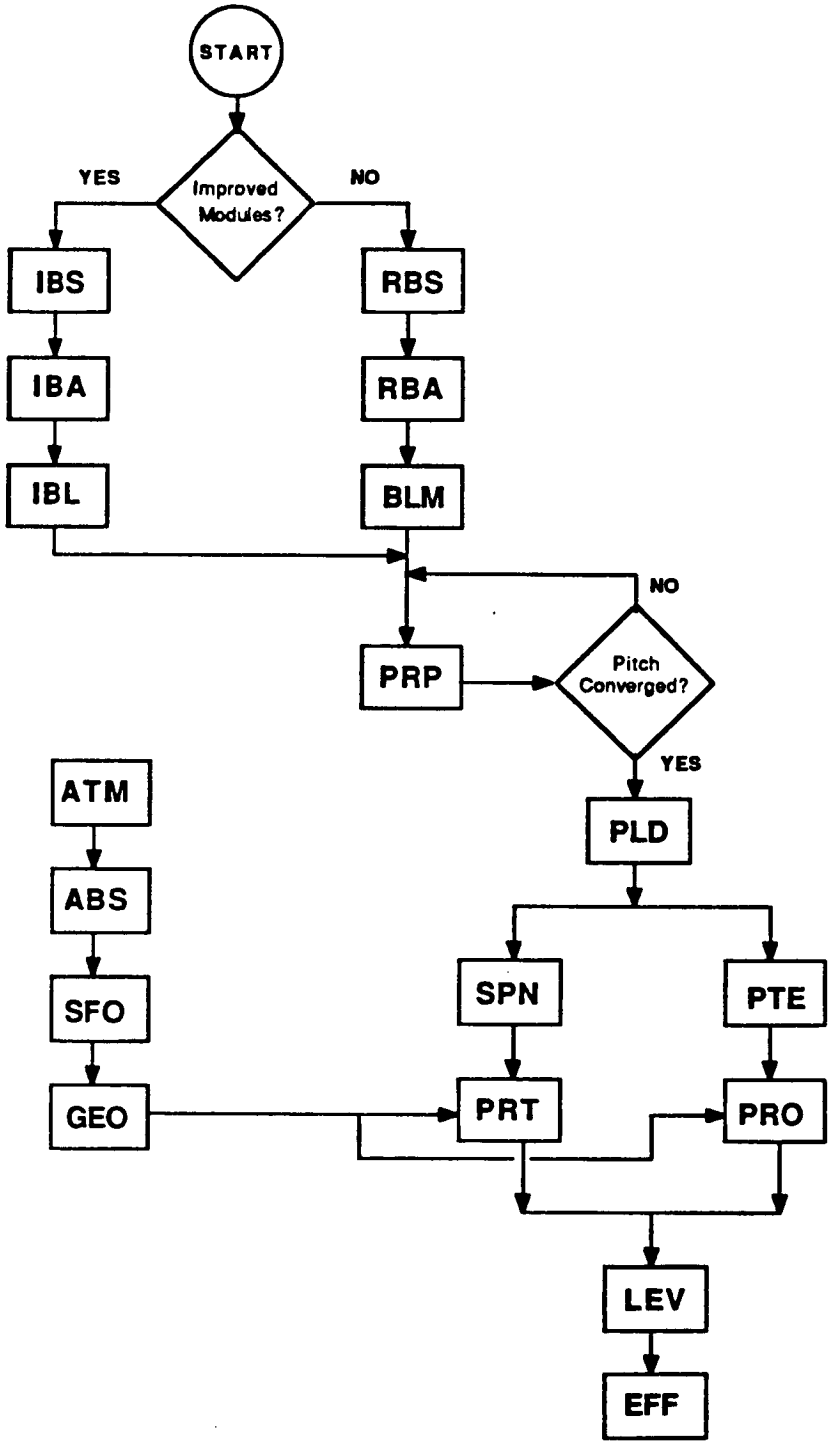

Figure 1.- Flowchart of ANOPP-PAS program modules used for predictions.

INSTALLATION

IBM-PC System Requirements

To successfully run the IBM-PC version of ANOPP-PAS, you need:

- IBM-XT or IBM-AT Personal Computer
- A fixed disk drive (10 MB or larger) and at least one 1.2 MB or 360 KB diskette drive
- 640 K bytes of memory
- A math coprocessor compatible with your system
- IBM Personal Computer Disk Operating System (DOS), version 2.1 or higher for IBM-PC XT or version 3.00 or higher for IBM-PC AT
- A printer (132 column capability recommended)
- IBM-PC version of ANOPP-PAS diskettes (total of 38 diskettes)

*NOTE: This manual refers to the fixed drive as Drive C and the diskette drive as Drive A. If your system uses different letters to reference those drives, then your systems drive letters must be substituted for A and/or C in the command instructions given in this manual.

Contents of the IBM-PC Version of ANOPP PAS Diskettes

Program Diskettes

Diskette Name	Program File Name	Size of File (bytes)	Command to Run Program
A-01	IBGPREP.EXE	235776	MAKE
B-01	UPFEDT.EXE	75072	PARAM
	ANTEDT.EXE	43744	ALT
C-01	MMEDT.EXE	301824	UNIT
D-01	TMEDT.EXE	336720	TABLE
E-01	RBS.EXE	342640	RUN RBS
F-01	RBA.EXE	339584	RUN RBA
G-01,G-02	BLM.EXE	394544	RUN BLM
H-01	PRP.EXE	350032	RUN PRP
I-01,I-02	PLD.EXE	388864	RUN PLD
J-01,J-02	SPN.EXE	580112	RUN SPN
K-01	ATM.EXE	300528	RUN ATM
L-01	ABS.EXE	300320	RUN ABS
M-01	SFO.EXE	300272	RUN SFO
N-01	GEO.EXE	322640	RUN GEO
O-01	PRT.EXE	348784	RUN PRT
P-01	LEV.EXE	280016	RUN LEV
Q-01	EFF.EXE	242368	RUN EFF
R-01	COMPACT.EXE	233232	REDUCE
S-01	PRFPREP.EXE	231200	INPUT
T-01	SNSPREP.EXE	237840	INPUT
U-01	FLPPREP.EXE	243312	INPUT
V-01	PLEPREP.EXE	220128	INPUT
W-01	UNMA.EXE	223344	LEVELS
X-01	IBS.EXE	345408	RUN IBS
Y-01,Y-02	IBA.EXE	364464	RUN IBA
Z-01,Z-02	IBL.EXE	377936	RUN IBL
AA-01	PTE.EXE	348448	RUN PTE
BB-01, BB-02	PRO.EXE	370384	RUN PRO

System Diskette I, II

Two diskettes which contain BATCH files and systems files.

Documentation Diskette I, II

Contain on-line documentation of Section V and Section VI of this manual.

Installing the ANOPP-PAS System

Step 1 System Check

- Refer to System Requirements at the front of this section to be sure that you have all of the hardware and software that the IBM-PC Version of ANOPP-PAS requires.

Step 2 System Configuration

- File CONFIG.SYS must contain the following statements:

  FILES = 10 (or larger number)

  BUFFERS = 10 (or larger number)

  (The configuration file (CONFIG.SYS) is explained in your IBM Personal Computer DOS Reference manual.) If your system does not have a CONFIG.SYS file installed, you must create one on your fixed disk. If you have a CONFIG.SYS file, use an editor and make sure that it contains the FILES = 10 and BUFFERS = 10 commands.

- File AUTOEXEC.BAT must contain a PATH command to C:\ANOPP\EXE and to the directory on which DOS is installed.

  EXAMPLE 1: If DOS is installed on the main directory of drive C, then PATH C:\; C:\ANOPP\EXE would be part of the path command.

  EXAMPLE 2: If DOS is installed on directory C:\DOS, then PATH C:\DOS; C:\ANOPP\EXE would be part of the path command.

  (The AUTOEXEC.BAT file is explained in your IBM Personal Computer DOS Reference manual. If your system does not have an AUTOEXEC.BAT file installed, you must create one and include the required PATH command. The PATH command tells DOS which directories to search for .EXE, .COM, and .BAT files, if not on the current directory. Additional directory references selected by the user may be included in the PATH command.)

*NOTE: If you create or change the CONFIG.SYS or AUTOEXE.BAT, then you must restart your system before running ANOPP-PAS programs.

- Create directories C:\ANOPP, and C:\ANOPP\EXE (if they do not exist on your system) by entering the following commands:

TYPE	C:
Press ENTER	Prompt C> appears on screen
Type	MD C:\ANOPP
Press ENTER	Directory C:\ANOPP is created
Type	MD C:\ANOPP\EXE
Press ENTER	Directory C:\ANOPP\EXE is created

Step 3 Installing the System Files

You will need approximately 10 MB of free space on your fixed disk to install and run the entire ANOPP-PAS System. (If you do not have this much space available, then the system can be installed and run in separate stages.)

- Install the Systems Diskettes (Version 2.0) by typing the following commands:

Type	CD C:\ANOPP\EXE
Press ENTER	C:\ANOPP\EXE is now current directory
Place System Diskette #1 in Drive A	
Type	Copy A:*.*
Press ENTER	Files on System Diskette #1 are installed
Place System Diskette #2 in Drive A	
Type	Copy A:*.*
Press ENTER	Files on System Diskette #2 are installed

- Installing the Executable Files

If you wish to install the entire ANOPP-PAS system, repeat the following instructions for diskettes A through BB. When there is a 2 diskette set, part 01 and part 02 with the same letter (EXAMPLE: G-01 and G-02), place diskette 01 in disk drive A and enter the RESTORE command following the instructions below. The program will prompt you to place diskette 02 in drive A.

If you do not wish to install the entire system, then choose the diskettes which contain the desired programs and repeat the following instructions for those diskettes.

*NOTE: When installing executable files (.EXE files) directory C:\ANOPP\EXE must be the current directory. To check the current directory:

TYPE	CD
Press ENTER	Display shows current directory

If C:\ANOPP\EXE is not shown as the current directory, then type the following:

TYPE	CD C:\ANOPP\EXE
Press ENTER	C:\ANOPP\EXE becomes the current directory

- Repeat the following instructions for each ANOPP-PAS program that is to be installed:

  Refer to Table 2 in this section to determine which diskette(s) contain the required program. In the following instructions substitute the actual diskette letter for x.

TYPE	RESTORE A: C:\ANOPP\EXE
Press ENTER	Display prompts user to place diskette 01 in drive A
Place diskette x-01 in disk drive A	
Press ENTER	Display will prompt user when the .EXE file has been copied to directory C:\ANOPP\EXE

*NOTE: The format for the RESTORE command may vary for different versions of DOS. Some versions require the following format:

RESTORE A: C:

- If the file is continued on a second diskette, then the display prompts the user to place diskette 02 in drive A.

  If prompted:

Remove diskette x-01 Place diskette x-02 in disk drive A	
Press ENTER	Display will prompt user when transfer is completed
Remove Diskette	

The above process can be repeated for as many programs as will fit on the available free space on the hard disk of your system and still allow 1/2-2 MB of free space for input and output data files. (The amount of reserved space depends on the number of programs installed on your system.)

If a limited amount of free space on the fixed disk is available, it is possible to install and run the system one program at a time with a minimum of 1 MB of free space for both program and data files. (See Section IV.)

SYSTEM DESIGN

The Noise Prediction System consists of 5 prediction procedures, 18 noise computational programs called functional modules and 9 data entry programs called preprocessors. Prediction procedures check the input data required by specific functional modules and initiate the execution of those modules. Functional modules perform specific noise prediction functions. Preprocessors create, edit and list data items used by the functional modules.

Prediction Procedures

Each of the prediction procedures controls the execution of a specific area of the noise prediction system. The prediction procedure sets up or verifies the existence of all required input files and controls the execution of one or more functional modules.

Functional Modules

Each functional module (FM) is made up of a program driver and 2 types of subprograms: executive system subprograms and noise-prediction-related subprograms.

The executive system subprograms are the same for all functional module programs. They provide all interfacing with DOS (IBM's Disk Operating System) and provide data base management of external files on which data items identified by the user reside. They also perform data storage management within the central memory established for the functional module. They manage global dynamic storage (GDS), where executive system and user tables are maintained. They also manage local dynamic storage (LDS), the portion of the program's central memory left available for the internal data storage required during the functional module execution.

Each noise-prediction-related subprogram performs a specific prediction function or a utility (data preparation or modification) function required by a single functional module. The functional module driver initializes the executive system, calls the main noise prediction subprogram, opens and closes the external files containing user parameter tables, alternate names tables and the functional module results.

Section VI of this manual documents the functional modules currently installed on the IBM-PC version of ANOPP-PAS. The section defines the noise prediction or utility capabilities of each FM and describes the data items (user parameters and data base unit members and tables) required as input or created as output by the FM.

The FM internally satisfies its data requirements by referencing alphanumeric ANOPP names of data base items and user parameters. These names are used in Section VI to identify the items required by the FM. ANOPP named data items (data base items and user parameters) are described in the following subsections. The preprocessors used to create, modify or list these data items are documented in Section V.

The user has the option of substituting alternate names, other than those recommended in Section VI, for the FM data. The alternate item may have been

created by a preprocessor, or by a functional module. The required name and the alternate name must be entered in an alternate names table file. During the FM execution, the identified alternate items are used. This alternate name concept also applies to output data items. The user can establish output items under different names than those used within and documented as output from the functional module. Alternate names are entered in the alternate names table by using the Alternate Names Table Editor Preprocessor.

Data Base

Data Base Components

All data items used within an ANOPP run reside within the ANOPP data base with the exception of user parameters. The ANOPP data base concept provides a method of storing and retrieving data on random-access files. In addition, through the ANOPP convention of named primary data base components and the capabilities of the ANOPP data base manager, the user is relieved of the responsibility for interfacing with the host operating system when opening, closing, reading, and writing external files.

The ANOPP data base is a hierarchical structure consisting, from top to bottom, of:

1. Data unit - An ANOPP named collection of members, which resides alone on a random-access mass-storage file.

2. Member - An ANOPP named collection of records.

3. Record - A collection of elements.

4. Element - A collection of words.

Figure 2 illustrates the basic structure of an ANOPP data unit and the relationship between its resident components.

The ANOPP data base convention of assigning a name to a data unit and to each of its members allows easy identification of the specific primary data base components required by functional modules and referenced by the user throughout the ANOPP run.

The executive data base manager records the record number of each named member on the data unit file. This technique results in efficient direct accessing of members in the internal system.

Data Units

The ANOPP data unit (DU) is a named collection of named members. Each data unit is stored in a direct access file that has the same name as the data unit with a '.UNT' extension. [EXAMPLE: data unit GEO will be stored in external DOS file GEO.UNT.]

The data unit is the primary component within the ANOPP data base. Section VI documents the data base item requirements of a functional module by

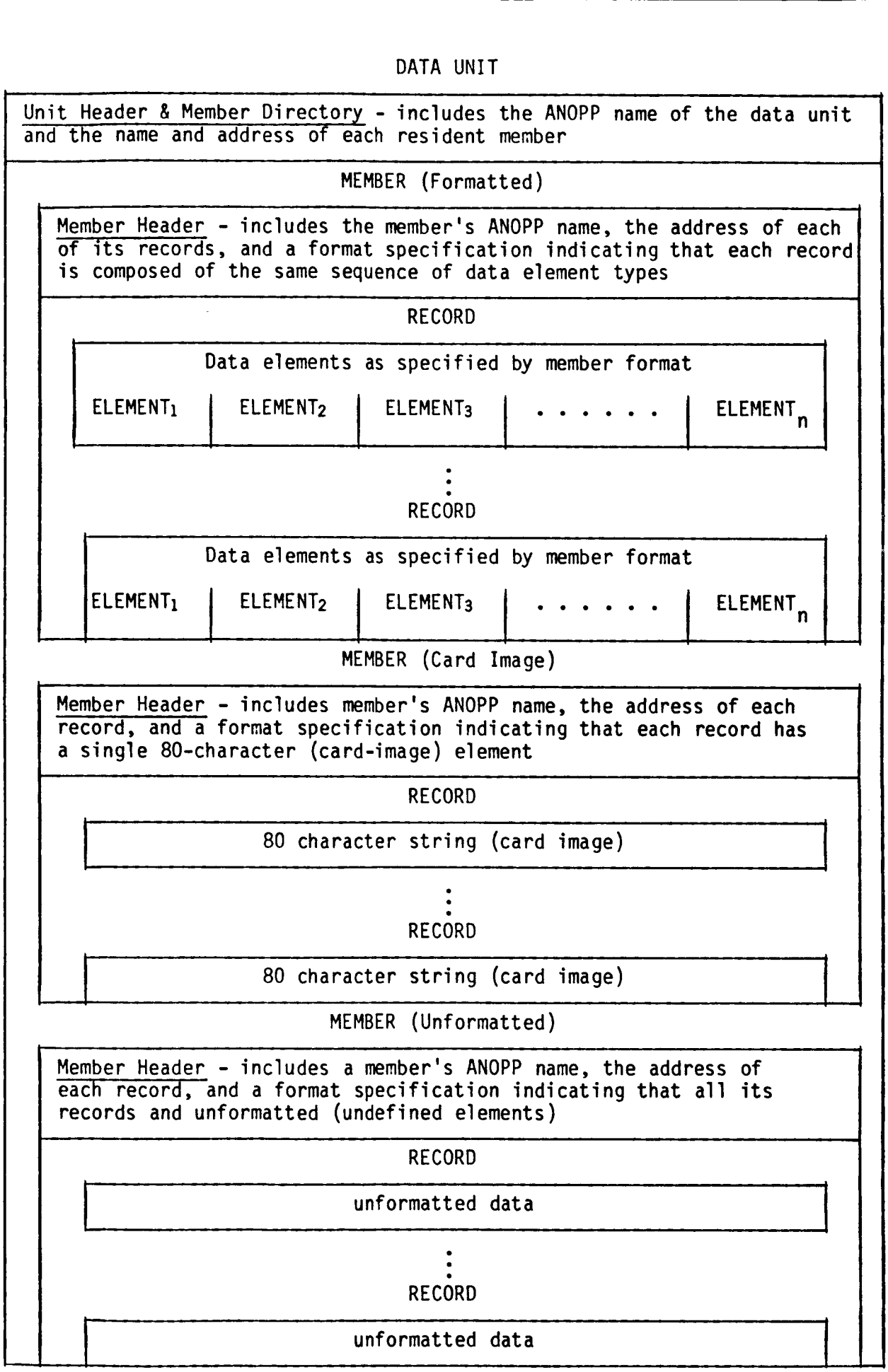

Figure 2.- ANOPP data unit structure

identifying each item by a combination of its data unit name and member name. As previously discussed in this section, the internal references to a specific data item within the FM are also made with this unit (member) combination. Normally the combination documented in Section VI is used. However, the user can substitute other data unit (member) combinations via the alternate name capability of the system.

Members

An ANOPP member is a named collection of logically related and organized records. Each member resides on a named data unit and must possess a name unique to all other members residing on the same unit. The member is the second and last of the named components in the ANOPP data base structure, the data unit being the first. Since no two data units may have the same name within an ANOPP run at the same time, and because members must be uniquely named within their data unit, any combination of data unit name and member name is also unique within an ANOPP run.

The unit (member) combination is used to address a specific data base item:

a) When referenced within a preprocessor, MMEDT

b) When referenced within a functional module

c) In Section VI, when identified as a functional module's data item requirement

When a user has established a data unit in a file, each named member that resides on that unit is automatically known in the run by its unit (member) combination. A preprocessor or internal functional module reference to the combination results in access to the specified member.

Preprocessor MMEDT can be used to create new members or new data units. The new members may contain combinations of records read directly from the user's input or records read from any other member currently established in a unit file.

A member consists of one or more records, each conforming in structure to the member's format specification. The ANOPP record is not a named data component. The user who builds or modifies a multirecord member must be aware of the sequential order of the records. The ANOPP record is not formatted in the sense of a format conversion as are FORTRAN computer language records. The format of an ANOPP record is specified for a member and must correspond to one of three general format types:

1. Unformatted records on a member are variable-length streams of data with no uniform structure (data element content) defined for the member.

2. Card-image records on a member are fixed-length (80-character) alphanumeric records, each with only one element corresponding to a Hollerith card image.

3. Formatted records on a member are fixed- or variable-length records with their structure defined by a member format specification of the sequence of data element types for each record in the member.

TABLE 2 - MEMBER-FORMAT SPECIFICATION RULES

Format type	Format-type specification rules	Examples
Unformatted	An unformatted member requires the integer 0 (zero).	FORMAT=0
Card image	A card-image formatted member requires the character string CI.	FORMAT=CI
Fixed-length format	A member containing formatted fixed-length records. The basic specification form is et,et,et$ where et is an ANOPP alpha data-type specification of I, R, C, L, or An that indicates a data element type occurring within each of the member's records. (See Table 3.) When two or more elements of the same type or two or more series of elements of the same types occur within the format of a record, they they may be shown by using a group specification of net. Integer n indicates the number of times the element type or series of element types repeat.	FORMAT=1I,1C,2R$ FORMAT=3A9,2C$ FORMAT=A9,L,I,A9,L,I$
Variable-length format	Variable-length records always contain a trailing portion of one or more elements which may occur from zero to an undefined number of times. If specified, each variable-length record will contain a leading portion of one or more elements. The * is used as a special character within variable-length format specifications to indicate the beginning of the variable portion of its record. The basic specification form is [et...,et]*([et...,et])$ The element-type specification rules, presented above for fixed-length records, also apply within the fixed and variable portions of variable-length format specifications. NOTE: For each occurrence of the variable portion in any variable length record, each element specified as part of that portion must exist.	FORMAT=*(R)$ FORMAT=*(R,I)$ FORMAT=*(2A8,10R)$ FORMAT=*(L,10R)$ FORMAT=A8,*(25R)$ FORMAT=L,I,*(10C)$ FORMAT=A7,*(I,C,5R)$

TABLE 3.- DATA TYPES

Data type	ANOPP type code	Output listing forms	Value range	Examples
Integer	I	±nnn...n (+ optional)	Largest, (2**31)-1; smallest, 0 (absolute values)	576460752 0
Real	R	±n.nnDn n.nnD+n n.nnD-n nD+n nDn nD-n (+ optional)	Largest, 10**+308; smallest, 10**-293 (absolute value)	17.72D27 36245.1029D+25 -522.365D-02 -1D+300 2000000D15 1D-293
Complex[a]	C	(R,R) (R,R) (Two real values within parentheses)	See R	(7.,1D279) (1D280 273.28D+2) (-1.7D02,0.52)
Logical	L	.TRUE. .FALSE.	.TRUE. .FALSE.	.TRUE. .FALSE.
Character string	A	xxx....x	1 to 132 characters (alphanumeric)	123456 ABCDEFGHIJK
Name	N	xxxxxxxx	1 to 8 characters (first character alpha)	JETUNIT FANTABLE

[a]The complex (C) value is specified as two R values enclosed within a required set of parentheses and separated by one or more delimiter characters (space or comma).

In the format specification of a member, the data types of the elements within records are designated by

I	integer
R	real
C	complex
L	logical
An	alphanumeric string of n characters

Internally, the executive system equates each element type specification with the number of computer words required to store a corresponding data value. Member format specification rules are documented in Table 2 and in MMEDT documentation in Section V.

In addition, there are special purpose data base members, called table members. Any member that does not fall within this category is simply called a data member.

Table Members

A table member is an unformatted member containing a single record referred to as a data table. A data table has an internal format that corresponds to a valid ANOPP data table type structure. Currently only type 1 data tables, defined below, are valid. A table manager editor, TMEDT, is available to create a table member and to print summary reports describing a table's structure and contents.

A type 1 data table may be output from a functional module or created using TMEDT. A type 1 data table may have from one to four independent variables. An independent variable in the table may be integer or real. A dependent variable may be integer, real, or complex. The independent variables need not be the same type, nor must the type of the dependent variables be the same as that of the independent variables.

While creating the table, the user specifies the acceptable interpolation procedures to be used on the table and, for each independent variable, the extrapolation method to be used if an interpolation request is outside the range of the independent variable. Interpolation and extrapolation procedures, number of independent variables, variable types, and data values are defined via Table Manager Editor entries.

User Parameters

The user parameter is a named data array of one or more elements of the same data type. User parameter elements may be integer, real, complex, logical or alphanumeric strings of characters.

User parameters are used as input to functional modules (FM). Section VI of this manual identifies the name, purpose, data type and number of elements

of all user parameters required by each FM. Alternate names can be used for user parameters. Functional modules may internally create new user parameters, or change the value of existing parameters.

Input user parameters are established before functional module execution by using the User Parameter File Editor, UPFEDT, to create a new user parameter file or edit existing user parameter files (Section V of this manual explains UPFEDT).

Alternate Names

The user has the option of substituting alternate data item names for those recommended in the functional module (FM) descriptions (see Section VI). These alternate names must exist in an Alternate Names Table in a DOS file, which has the same name as the FM but with a .ANT extension. [EXAMPLE: functional module PRO would access DOS file PRO.ANT to obtain any alternate names requested by the user.] An Alternate Names Table file must exist before each functional module is executed, even if the Alternate Names Table is empty. Alternate Names Table Files can be created using the ANTEDT preprocessor or by executing the DOS COPY or RENAME command. If no alternate names are to be used during a functional module execution, then an empty Alternate Names Table DOS file, KEEPME.ANT, is copied to the required [FM].ANT file. Alternate names can be used for user parameters and data base items which are required as input to a functional module or which are created as output from a functional module.

Preprocessors

There are four executive system preprocessors and five procedure preprocessors. The executive system preprocessors include the User Parameter File Editor (UPFEDT), the Alternate Names Table Editor (ANTEDT), the Member Manager Editor (MMEDT), and the Table Manager Editor (TMEDT). Each of these preprocessors allows the user to create or modify a different type of named data. The procedure preprocessors include the Blade Geometry Preprocessor (IBGPREP), the Performance Preprocessor (PRFPREP), the Noise Preprocessor (SNSPREP), the Flight Path Preprocessor (FLPPREP), and the Propagation Preprocessor (PLEPREP). Each of the prediction procedure preprocessors interactively reads the input data required by the corresponding prediction procedure.

UPFEDT provides the capability to create, change, or list named data arrays called user parameters.

ANTEDT allows the user to create, change or list alternate names that will be used in place of named data items referenced during functional module execution.

MMEDT provides the capability to create, modify, or list name data base items that do not require interpolation and are called unit members.

TMEDT allows the user to create, or list data tables, which are a special type of unit members which allow interpolation of the data.

IBGPREP interactively reads all data items required by the Blade Geometry Procedure. It executes functional modules RBS, RBA, and BLM, or IBS, IBA, and IBL.

PRFPREP interactively reads all data items required by the Performance Procedure (functional modules PRP and PLD).

SNSPREP interactively reads all data items required by the Noise Procedure (functional modules SPN and PTE).

FLPPREP interactively reads all data items required by the Flight Path Procedure (functional modules ATM, ABS, SFO, and GEO).

PLEPREP interactively reads all data items required by the Propagation Procedure (functional modules PRT, PRO, LEV, and EFF).

Documentation on each of the preprocessors is contained in Section V.

Differences Between IBM-PC Version of ANOPP and Other Versions of ANOPP

The IBM-PC version of ANOPP does not contain the control statements used by previous versions of ANOPP. Each functional module is contained in a separate program which can be executed alone or as part of a system run (a sequence of functional module program executions controlled by a DOS BATCH file). The User Parameter and Alternate Names Tables exist on external direct access files in the IBM-PC version and must be established by the user before the functional module program execution.

In the IBM-PC version, an ANOPP-PAS system run which contains multiple functional module executions can be established by creating a DOS BATCH file which contains DOS COPY commands, to set up user parameter and alternate names table files, and functional module execution commands.

Control Statements in Previous Versions of ANOPP	Preprocessors Used by IBM-PC Version of ANOPP
PARAM UPLIST EVALUATE	UPEDT interactively creates, changes or lists a user parameter file. (No equivalence on IBM-PC Version)
TABLE TABLIST CATALOG	TMEDT interactively creates changes or lists a data table, or creates a catalog listing of all members on a data unit.
CREATE UPDATE - COPY - ADDR - CHANGE - INSERT - DELETE - QUIT MEMLIST CATALOG	MMEDT interactively creates, changes, or lists a unit member, or creates a catalog listing of all members on a data unit.
EXECUTE	ANTEDT interactively creates, changes or lists an alternate names table file.

ANOPP-PAS USAGE

Overview

Two types of program make up the IBM-PC Version of ANOPP-PAS: executive system programs and functional module programs. The executive system programs include preprocessors which create, edit, and list data items used by functional modules, a file compacting program which enables data unit files to be stored in the least number of bytes, and a graphics postprocessor which plots noise prediction results. Functional module programs perform noise-prediction-related functions. Prediction Procedures control the execution of multiple functional modules.

All predictions are executed on subdirectories of C:\ANOPP. Each different propeller will have its own subdirectory and all predictions using that propeller will be run on that subdirectory. This section contains the instructions needed to create these subdirectories and to formulate and execute noise prediction applications.

Using ANOPP-PAS

Step 1 System Check

If the ANOPP-PAS system has not been installed on your computer system, refer to Section II for installation instructions. Your system must contain all of the requirements listed on page 2-1.

Step 2 File Preparation

If a new propeller is to be installed in the system, then a <propeller id> name of 1-8 characters must be chosen by the user and a file containing the blade geometry for that propeller must be created. (See Section V pages 5-6 to 5-8 for a description of this file.) The file must be named <propeller id>.DEF or <propeller id>.IDF and placed on directory C:\ANOPP.

EXAMPLE: If <propeller id>, ABCBLADE, is chosen by the user, then file ABCBLADE.DEF or ABCBLADE.IDF must be created.

Step 3 System Start

- COMMAND: ANOPP

To start the system:

Type	ANOPP
Press ENTER	The system displays:

- THE CURRENT DIRECTORY IS C:\ANOPP. A SUBDIRECTORY OF C:\ANOPP MUST BECOME THE CURRENT ANOPP DIRECTORY. [THERE MUST BE A SEPARATE SUBDIRECTORY FOR EACH PROPELLER AND THE SUBDIRECTORY WILL HAVE THE NAME OF THE PROPELLER ID (1-8 CHARACTERS).] DO YOU WISH TO SEE DIRECTIONS FOR MAKING OR CHANGING AN ANOPP SUBDIRECTORY (MAKE AND CHANGE COMMANDS)?
PLEASE ENTER Y OR N (DO NOT PRESS "ENTER"):

If the user types Y, then an explanation of the MAKE and CHANGE commands will be displayed on the screen.

The system then displays:

AFTER THE C> PROMPT TYPE: MAKE <propeller id>

or

CHANGE <propeller id>

Step 4 Subdirectory Commands

- Command: MAKE

If a new propeller is to be installed in the system:

Type Press ENTER	MAKE <propeller id> Subdirectory C:\ANOPP\<propeller id> will be created.

The following prompt will be displayed:

- DO YOU WISH TO USE THE IMPROVED BLADE GEOMETRY MODULES? PLEASE PRESS Y OR N (DO NOT PRESS "ENTER"):

The Improved Blade Geometry Modules, IBS, IBA, and IBL, allow the user to choose compressibility correction methods for lift and pressure coefficients, and include a zero pressure gradient flat plate model option. The user presses Y to choose the Improved Blade Geometry modules or N to use the original Blade Geometry modules, RBS, RBA, and BLM.

The Blade Geometry Preprocessor will prompt the user to enter the input data for the chosen modules.

The following prompt will be displayed:

- DO YOU WISH TO RUN BLADE GEOMETRY PROCEDURE AT THIS TIME? PLEASE PRESS Y OR N (DO NOT PRESS "ENTER"):

If the user presses Y, then functional modules RBS, RBA, and BLM, or IBS, IBA, and IBL, will be run.

If the user presses N, then the user must execute the Blade Geometry Procedure by typing the GEOM command, before running any other Prediction Procedures.

- Command: CHANGE

If the propeller has already been installed on subdirectory C:\ANOPP\ <propeller id>:

Type Press ENTER	CHANGE <propeller id> C:\ANOPP\<propeller id> will become the current directory

If the Blade Geometry input and output files were backed up to diskette, then the system displays:

- BLADE GEOMETRY FILES WERE BACKED UP. DO YOU WISH TO LOAD PROPELLER DATA FOR <PROPELLER ID> FROM A DISKETTE?
  PLEASE PRESS Y OR N (DO NOT PRESS "ENTER"):

  If the user presses Y, then the system instructs the user to place the backup diskette in Drive A and press any key. All backed up files will be restored to the current directory.

  If prediction input and output files were backed up to diskette, then the system displays:

- PREDICTION FILES HAVE BEEN BACKED UP. DO YOU WISH TO LOAD PREDICTION DATA FOR <PROPELLER ID> FROM A DISKETTE?
  PLEASE PRESS Y OR N (DO NOT PRESS "ENTER"):

  If the user presses Y, then the system instructs the user to place the backup diskette in Drive A and press any key. All backed up files will be restored to the current directory.

- After the CHANGE or MAKE command has been executed, the following is displayed:

  LIST OF COMMANDS THAT CAN BE RUN FROM AN ANOPP SUBDIRECTORY:

INPUT	- Create input files for any or all of the performance, noise, flight path or propagation modules.
GEOM	- Execute blade geometry modules.
PERF	- Execute performance modules.
NOISE	- Execute noise modules.
FPATH	- Execute flight path modules.
LEVELS	- Execute propagation modules.
BKUP	- Back up all input output files to floppy diskette.
PLOT	- Execute plot postprocessor.

  (The user must type one of these commands to continue. The user can display these commands at any time by typing LIST.)

Step 5 Input Data

Command: INPUT

The user must input data items, required by the Performance, Noise, Flight Path and Propagation Procedures, before executing these procedures. To run the data input preprocessors:

Type Press ENTER	INPUT The system displays the following prompts:

** DO NOT press the ENTER key after pressing Y or N in response to any of the following prompts. If N is pressed, then the system does not run the preprocessor.

- DO YOU WISH TO RUN PERFORMANCE PREPROCESSOR?
  PLEASE PRESS Y OR N (DO NOT PRESS "ENTER"):

  If Y is pressed, then the system asks the user to input or verify all data items required by the Performance Procedure, which executes functional modules PRP and PLD. (See PRFPREP in Section V for instructions for Performance Preprocessor.)

- DO YOU WISH TO RUN NOISE PREPROCESSOR?
  PLEASE PRESS Y OR N (DO NOT PRESS "ENTER"):

  If Y is pressed, then the system asks the user to input or verify all data items required by the Noise Procedure which executes functional modules SPN and/or PTE. (See SNSPREP in Section V for instructions for Noise Preprocessor.)

- DO YOU WISH TO RUN FLIGHT PATH PREPROCESSOR?
  PLEASE PRESS Y OR N (DO NOT PRESS "ENTER"):

  If Y is pressed, then the following prompt is displayed:

- THERE ARE 3 OPTIONS FOR ATMOSPHERIC TABLES:

  1. STANDARD ATMOSPHERE (PRESSURE:2116.22 LB/FT**2) EXISTING TABLES WILL BE USED.

  2. STANDARD ATMOSPHERE + 10 DEGREES CENTIGRADE (PRESSURE:2116.22 LB/FT**2) EXISTING TABLES WILL BE USED.

  3. TABLES WILL BE CREATED FROM ATMOSPHERIC PROFILE INPUT BY USER.

  DO YOU WISH TO USE STANDARD ATMOSPHERE (OPTION 1)?
  PLEASE PRESS Y OR N (DO NOT PRESS "ENTER"):

  If Y is pressed, then the atmospheric tables will be created from the following atmospheric profile:

  Standard Atmosphere
  Pressure: 2116.22 lb/ft^2

Altitude (ft)	Temperature (°R)	Rel. Humidity (%)
0.0	518.67	70.
1000.	515.10	70.

  If N is pressed, then the following is displayed:

- DO YOU WISH TO USE STANDARD ATMOSPHERE + 10 DEGREES CENTIGRADE (OPTION 2)?
  PLEASE PRESS Y OR N (DO NOT PRESS "ENTER"):

  If Y is pressed, then the atmospheric tables will be created from the following atmospheric profile:

Standard Atmosphere
Pressure: 2116.22 lb/ft^2

Altitude (ft)	Temperature (°R)	Rel. Humidity (%)
0.0	536.67	70.
1000.	533.10	70.

If N is pressed, then the following is displayed:

- OPTION 3 (USER INPUT) WILL BE USED.

The user will be asked to verify or enter the data required to create the atmospheric tables.

If the user requests that the Flight Path Preprocessor be run, then the system prompts the user to enter or verify all data required by the Flight Path Procedure, which executes functional modules ATM, ABS, SFO and GEO. (See FLPPREP in Section V for instructions for Flight Path Preprocessor.)

- DO YOU WISH TO RUN PROPAGATION PREPROCESSOR?
  PLEASE PRESS Y OR N (DO NOT PRESS "ENTER"):

If Y is pressed, then the system asks the user to enter or verify all data required by the Propagation Procedure which executes functional modules PRT, PRO, LEV and EFF. (See PLEPREP in Section V for instructions for Propagation Preprocessor.)

The system then displays the following message:

- ...ENDING PREPROCESSORS

Step 6 Input Data Modification (Optional)

To examine or modify input data created by the prediction preprocessors, executive system preprocessors must be used. The following table lists the required preprocessor commands, and the files and data types that they reference:

Command	Preprocessor File	Data Type
ALT	ANTEDT.EXE	Alternate Names
PARAM	UPFEDT.EXE	User Parameters
TABLE	TMEDT.EXE	Data Tables
UNIT	MMEDT.EXE	Unit Members

- The Preprocessor files (.EXE files) must be installed on directory C:\ANOPP\EXE. Typing the command and pressing ENTER will cause the appropriate menu to be displayed. Section V documents the instructions for using each preprocessor.

- Each functional module references a user parameter file and an alternate name file. The files have the same name as the functional module with a .PAR or .ANT extension, and reside on the current directory.
  (EXAMPLE: Functional module SPN references user parameter file SPN.PAR and alternate names file SPN.ANT.)

- A user parameter file (with the same name as the functional module and a .PAR extension) and files containing all required input unit member and data tables (documented in section VI) must exist before executing a functional module.

- If alternate names are to be used, they must be entered in an alternate names table file that has the same name as the functional module and a '.ANT' extension. EXAMPLE: RBS.ANT will contain the alternate names (if any), that will be used by RBS. An alternate names table file is required by each module. If they do not exist, then each Prediction Procedure will create an empty .ANT file for the functional modules that is executes.

Step 7 Prediction Procedure Execution

To execute a prediction procedure the user must type the procedure command and press the ENTER key. The procedures should be run in the following order:

Command	Procedure Name	Functional Modules	Approximate Run Time
GEOM	Blade Geometry	RBS, RBA, BLM or IBS, IBA, IBL	5-8 1/2 hrs.
PERF	Performance	PRP, PLD	30-60 minutes
NOISE	Noise	SPN, PTE	18-26 hrs.
FPATH	Flight Path	ATM, ABS, SFO, GEO	5-12 minutes
LEVELS	Propagation	PRT, PRO, LEV, EFF	15-25 minutes

The printed output from each procedure will be stored in a results file (one for each functional module [FM] that is executed). The files will be named RESULTS.[FM]
EXAMPLE: Printed output from the execution of the Blade Geometry Procedure will be found in files RESULTS.RBS, RESULTS.RBA and RESULTS.BLM or RESULTS.IBS, RESULTS.IBA, and RESULTS.IBL.

- Command: GEOM

This command is used only if the Blade Geometry Procedure is not run under the MAKE command, when a new propeller subdirectory is created. Functional modules RBS, RBA, and BLM or IBS, IBA, and IBL will be executed (See Section VI for a description of these functional modules.)

- Command: PERF

  This command will execute functional modules PRP and PLD. (See Section VI for a description of these functional modules.) The procedure will display the following prompt:

- DO YOU WISH CONVERGENCE OF POWER COEFFICIENT TO MEASURED POWER?
  PLEASE PRESS Y OR N (DO NOT PRESS THE ENTER KEY):

  If an exact blade pitch setting at the root was entered in the Performance Preprocessor, then the user presses N. (The ENTER key is not pressed after the N.) Functional module PRP will be executed only once and then PLD will be executed. If an initial guess for the blade pitch was made in the Performance Preprocessor, then the user presses Y. (The ENTER key is not pressed after the Y.) The system executes iterations of PRP, changing the blade pitch value until the calculated power coefficient matches the measured power. The system will calculate up to 10 iterations of PRP. If convergence does not occur, then an error message will be printed and PLD will not be run. If convergence does occur, then the system places the new value for the blade pitch in the Noise parameter file and runs PLD.

- Command: NOISE

  This command will execute functional modules SPN and/or PTE. (See Section VI for a description of these functional modules.) The procedure will display the following prompts:

```
*   NARROWBAND AND OR BROADBAND NOISE MAY BE PREDICTED    *
************************************************************
DO YOU WISH TO PREDICT NARROWBAND NOISE ?
PLEASE PRESS Y OR N (DO NOT PRESS "ENTER"):.
```

  If Y is pressed, then functional module SPN will be executed.

```
************************************************************
DO YOU WISH TO PREDICT BROADBAND NOISE ?
PLEASE PRESS Y OR N (DO NOT PRESS "ENTER"):.
```

  If Y is pressed, then functional module PTE will be executed.

- Command: FPATH

  If the standard or current atmospheric tables are not to be used, then functional modules ATM and ABS will be run. Functional modules SFO and GEO will be executed. (See Section VI for a description of these functional modules.)

- Command: LEVELS

  This command will execute functional modules PRT, PRO, LEV and EFF. (See Section VI for a description of these functional modules.) This command also executes UNMA.EXE, which prepares plot files, PRES.PLT, BBND.PLT, PNLT.PLT, OSPL.PLT, AWGT.PLT, and COORD.PLT, which are used by the plot postprocessor.

Step 8 Plotting Noise Prediction Data

- Command: PLOT

- In order to successfully execute the Graphics Plot Postprocessor, one of the following graphics boards must be installed in your computer system:

  Hercules Monochrome Graphics Board
  IBM Color Graphics Board
  IBM Extended Graphics Board

- To run the Graphics Plot Postprocessor:

Type Press ENTER	PLOT If the graphics board has not been identified on the system, then the following menu is displayed:

- GRAPHIC PLOTS ARE AVAILABLE USING ONE OF THE FOLLOWING GRAPHICS BOARDS (code H or I):

ID CODE	GRAPHICS BOARD
H	HERCULES MONOCHROME GRAPHICS BOARD.
I	IBM COLOR GRAPHICS BOARD/OR IBM EXTENDED GRAPHICS BOARD.
N	NO GRAPHICS BOARD/OR GRAPHICS BOARD NOT LISTED.

ENTER GRAPHIC ID CODE

If the user enters H, then file HERCULES.TYP will be created on subdirectory C:\ANOPP\EXE. If the user enters I, then file IBMCORE.TYP will be created on the same subdirectory. If the user enters N, then the following message is displayed:

** THE GRAPHICS PLOT PROGRAM CAN ONLY BE RUN IF ONE OF THE GRAPHICS BOARDS LISTED ABOVE IS INSTALLED IN THE COMPUTER SYSTEM.

- The system will prompt the user to enter the correct graphics ID only if file C:\ANOPP\EXE\HERCULES.TYP or file C:\ANOPP\EXE\IBMCORE.TYP does not exist.

- If an incorrect Graphics Board ID is installed on the system, then the SETUP command (see page 4-13) must be used to change the Graphics Board ID.

- If a valid Graphics Board ID has been entered, then the following menu will be displayed:

```
                PLOT MENU
-------------------------------------------------------------
CODE            PLOT TYPE
----            ---------

 P              PNLT .VS. RECEPTION TIME
 N              NARROWBAND SPL .VS. FREQUENCY
 B              BROADBAND SPL .VS. FREQUENCY
 O              OVERALL SPL .VS. RECEPTION TIME
 A              A-WT SPL .VS. RECEPTION TIME
 E              EXIT PLOT PROGRAM

ENTER PLOT CODE
```

- CODE: P If the user enters code P then the following prompts are displayed:

```
PNLT .VS. TIME
-----------------------------------------
OBSERVER COORDINATES:
NUMBER          X          Y          Z
   1            x1         y1         z1
   .
   .
   N            xn         yn         zn
-----------------------------------------

ENTER OBSERVER NUMBER (INTEGER)     (See note 1)
ENTER PLOT TITLE
(1-50 CHARACTERS)       (See note 2)
```

- CODE: N If the user enters code N, then the following prompts are displayed:

```
NARROWBAND SPL .VS. FREQUENCY
-----------------------------------------
OBSERVER COORDINATES:
NUMBER          X          Y          Z
  1             x1         y1         z1
  .
  .
  N             xn         yn         zn
-----------------------------------------

ENTER OBSERVER NUMBER (INTEGER)     (See note 1)
ENTER RECEPTION TIME (REAL)     (See note 3)
ENTER PLOT TITLE
(1-50 CHARACTERS)     (See note 2)
```

- CODE: B If the user enters code B, then the following prompts are displayed:

```
BROADBAND SPL .VS. FREQUENCY
-------------------------------------------
OBSERVER COORDINATES:
NUMBER          X          Y          Z
  1             x1         y1         z1
  .
  .
  N             xn         yn         zn
-------------------------------------------

ENTER OBSERVER NUMBER (INTEGER)     (See note 1)
ENTER RECEPTION TIME (REAL)     (See note 3)
ENTER PLOT TITLE
(1-50 CHARACTERS)     (See note 2)
```

- CODE: O If the user enters code O, then the following prompts are displayed:

```
OASPL .VS. RECEPTION TIME
-----------------------------------------------
OBSERVER COORDINATES:
NUMBER          X          Y          Z
  1             x1         y1         z1
  .
  .
  N             xn         yn         zn
-----------------------------------------------

ENTER OBSERVER NUMBER (INTEGER)     (See note 1)
ENTER PLOT TITLE
(1-50 CHARACTERS)     (See note 2)
```

CODE: A If the user enters code A, then the following prompts are displayed:

```
A-WEIGHTED SPL .VS. RECEPTION TIME
-------------------------------------------
OBSERVER COORDINATES:
NUMBER          X          Y          Z
  1             x1         y1         z1
  .
  .
  N             xn         yn         zn
-------------------------------------------

ENTER OBSERVER NUMBER (INTEGER)     (See note 1)
ENTER PLOT TITLE
(1-50 CHARACTERS)     (See note 2)
```

- CODE: E The Plot menu will be displayed after each plot until Code E (EXIT) is entered.

- The user must press the ENTER key after each response. After a plot has been displayed, the user must press the ENTER key to continue. A prompt asking if another plot of the same type is to be created is displayed. If Y is entered, then the user is asked to enter an observer number. If N is entered, the Plot Menu is displayed.

- Note 1: ENTER OBSERVER NUMBER (INTEGER)
  The postprocessor displays the observer coordinates (x,y,z), for each observer that was entered in the Flight Path Preprocessor (see page 5-39). The user must type one of the listed observer numbers and press the ENTER key.

- Note 2: ENTER PLOT TITLE (1-50 CHARACTERS)
  The user types a 1-50 character plot title and presses the ENTER key. The title will be displayed at the top of the plot screen. If no title is desired, then the user presses only the ENTER key.

- Note 3: ENTER RECEPTION TIME (REAL)
  The user enters the desired reception time and presses the ENTER key. The reception time must include a decimal point. If the reception time is not valid for the prediction run, then a message which contains the high and low reception time boundaries is displayed and the user is asked to enter another observer number and/or reception time.

Step 9 Backup of Input/Output Data

- Command: BKUP

  This command allows the user to back up input and output files to diskettes. The system displays the following prompt:

- YOU WILL NEED UP TO 6 EMPTY FORMATTED DISKETTES. DO YOU WISH TO CONTINUE WITH BACKUP?
  PLEASE PRESS Y OR N (DO NOT PRESS "ENTER"):

  If the user has prepared the required diskettes, then Y is pressed (the ENTER key is not pressed) and the backup continues with the output file reduction. All unit files listed under the SMALL command will be reduced to their smallest size. The system displays:

- DO YOU WISH TO BACKUP PROPELLER BLADE GEOMETRY FILES?
  PLEASE PRESS Y OR N (DO NOT PRESS "ENTER"):

  If Y is pressed, then the system displays:

- PLACE EMPTY FORMATTED DISKETTE IN DRIVE A: STRIKE A KEY WHEN READY...

  The user places an empty formatted diskette in Drive A and presses any key. The system will copy the following files to Drive A:

BLM.UNT	BLM.ANT	BLM.PAR		IBL.UNT	IBL.ANT	IBL.PAR
RBA.UNT	RBA.ANT	RBA.PAR	or	IBA.UNT	IBA.ANT	IBA.PAR
RBS.UNT	RBS.ANT	RBS.PAR		IBS.UNT	IBS.ANT	IBS.PAR

GRID.UNT	<propeller id>.DEF or <propeller id>.IDF
GEOM.UNT	(blade geometry input file)

  The system then displays:

- CAN THESE FILES BE REMOVED FROM THIS ANOPP SUBDIRECTORY?
  PLEASE PRESS Y OR N (DO NOT PRESS "ENTER"):

If the user presses Y, then the files will be deleted from the current subdirectory. The system displays:

- DO YOU WISH TO BACK UP ALL OTHER ANOPP DATA FILES?
  PLEASE PRESS Y OR N (DO NOT PRESS "ENTER"):

  If the user presses Y, then the system displays:

- PLACE EMPTY FORMATTED DISKETTE IN DRIVE A: Strike a key when ready...

  The user places an empty formatted diskette in Drive A and presses any key. The system will copy any remaining .UNT, .PAR and .ANT files to Drive A (blade geometry files will not be copied at this time).

  The system then displays:

- CAN THESE FILES BE REMOVED FROM THIS ANOPP SUBDIRECTORY?
  PLEASE PRESS Y OR N (DO NOT PRESS "ENTER"):

  If the user presses Y, then the files will be deleted from the current subdirectory.

  The system displays:

- ***WARNING: IF RESULTS ARE BACKED UP TO DISKETTE, ONLY THE FOLLOWING FILES WILL BE SAVED:

RESULTS.PRP	RESULTS.SPN
RESULTS.SFO	RESULTS.GEO
RESULTS.LEV	RESULTS.EFF

  *** ALL OTHER RESULTS FILES MAY BE DELETED.
  *** PLEASE HAVE AN EMPTY FORMATTED DISKETTE READY BEFORE REQUESTING BKUP.
  DO YOU WISH TO BACKUP RESULTS FILES?
  PLEASE PRESS Y OR N (DO NOT PRESS "ENTER"):

  If the user presses Y, then the results files listed above will be backed up to Drive A. The user places an empty diskette in Drive A and presses any key after each prompt displayed by the DOS backup command. The system then displays:

- MAY ALL RESULTS FILES BE REMOVED FROM THIS ANOPP SUBDIRECTORY?
  PLEASE PRESS Y OR N (DO NOT PRESS "ENTER"):
  If the user presses Y, then all results files will be deleted from the current directory.

- Plot files are not backed up to diskette. They can be recreated by executing the LEVELS command.

Optional Commands

- Command: RUN
  This command will execute individual functional modules chosen by the user. A program file [fm].EXE must be installed on directory C:\

ANOPP\EXE for each functional module, before it is executed.
EXAMPLE: C:\ANOPP\EXE\ABS.EXE must exist before executing functional module ABS.)

To execute a functional module, type RUN and the functional module name and press ENTER. A prompt will be displayed when the program terminates.
EXAMPLE: To run functional module GEO.

Type Press ENTER	RUN GEO

The printed output from the functional module will be stored in file RESULTS.[FM].
EXAMPLE: Printed output from the execution of BLM will be found in file RESULTS.BLM.

- Command: PAS

This command will run 13 of the modules in the ANOPP-PAS system in the following order: RBS, RBA, BLM, PRP, PLD, SPN, ATM, ABS, SFO, GEO, PRT, LEV and EFF. The functional module files (.EXE), the user parameter files (.PAR), the alternate names files (.ANT) and all required data base files for each module must be installed on the appropriate directories. Your system must also have at least 2 MB of free space on the fixed disk after all of the required files are installed. The user types PAS and presses ENTER to run the system. The system run may take 35 hours or longer to complete. All data unit files that have their names listed in the subsection below will be reduced to their minimum size at the end of the function module executions.

- Commands: REDUCE and SMALL

These commands will reduce the size of data files. Files that contain Data Units (files with .UNT extension) can contain fragmented free space after several functional module runs. To reduce the size of any .UNT file to the minimum required number of bytes, type REDUCE and the Unit file name without the .UNT extension.
(EXAMPLE: To reduce unit file PRT.UNT:

Type Press ENTER	REDUCE PRT PRT.UNT will be reduced to its minimum size

Command SMALL will reduce any or all of the following files, if they exist on the current directory:

ATM.UNT	SFIELD.UNT	IBL.UNT
GEO.UNT	PRT.UNT	IBA.UNT
GRID.UNT	PRP.UNT	IBS.UNT
GEOM.UNT	RBS.UNT	PRO.UNT
PLD.UNT	OBSRV.UNT	PTE.UNT
FLI.UNT	BLM.UNT	
RBATMP.UNT	RBA.UNT	
SPN.UNT		

To use this command, type SMALL and press ENTER.

- Command: SETUP

  This command allows the user to change the Graphics Board ID file, which informs the Plot Postprocessor which graphics board, if any, is installed on the system:

  The following menu is displayed:

  GRAPHIC PLOTS ARE AVAILABLE USING ONE OF THE FOLLOWING GRAPHICS BOARDS:

ID CODE	GRAPHICS BOARD
H	HERCULES MONOCHROME GRAPHICS BOARD
I	IBM COLOR GRAPHICS BOARD/OR IBM EXTENDED GRAPHICS BOARD
N	NO GRAPHICS BOARD/OR GRAPHICS BOARD NOT LISTED

  ENTER GRAPHICS ID CODE

  The user enters the ID CODE and presses the ENTER key. If N is entered, then the Plot Postprocessor cannot be executed.

- Command: MPAS

  This command allows the user to run multiple ANOPP-PAS predictions on separate subdirectories of ANOPP, each of which contain blade geometry unit files. The blade geometry unit files are created by the Blade Geometry Procedure (command GEOM). Once the Blade Geometry Procedure for a particular propeller has been run on one subdirectory, the user can create another subdirectory for that propeller and copy the following blade geometry unit files to the new subdirectory:

GEOM.UNT		RBA.UNT		IBA.UNT
GRID.UNT	and	RBS.UNT	or	IBS.UNT
		BLM.UNT		IBL.UNT

  EXAMPLE: If the Blade Geometry Procedure was run with the Improved Blade Geometry Option, on subdirectory SDIR1, and the user wishes to create a new subdirectory SDIR2 for predictions using the same blade geometry, then the following commands would be used:

Type and Press ENTER after each line	MD ANOPP\SDIR2 COPY \ANOPP\SDIR1\GEOM.UNT \ANOPP\SDIR2 COPY \ANOPP\SDIR1\GRID.UNT \ANOPP\SDIR2 COPY \ANOPP\SDIR1\IBS.UNT \ANOPP\SDIR2 COPY \ANOPP\SDIR1\IBA.UNT \ANOPP\SDIR2 COPY \ANOPP\SDIR1\IBL.UNT \ANOPP\SDIR2

- The user must run the Performance, Noise, Flight Path, and Propagation Preprocessors, using the INPUT command, on each subdirectory.

- To execute the MPAS command:

Type Press ENTER	MPAS name1 [name2 name3 . . .]

where name1, name2, and name3 . . . represent the subdirectory names. The command MPAS and the directory names must be separated by a space or comma. There is no limit to the number of different subdirectory predictions that can be run with this command, but the entire command line must not exceed 127 characters.

EXAMPLE: To execute prediction runs on SDIR1 and SDIR2:

Type Press ENTER	MPAS SDIR1 SDIR2

- THE MPAS command computes the blade pitch by matching the power coefficient to measure power on each subdirectory.

PREPROCESSORS

Description of 9 Data Preprocessors

There are 9 data entry programs, called preprocessors, in the ANOPP-PAS. There are five procedure proprocessors which display prompts and read values for all data items required by each of the five prediction procedures. There are four executive preprocessors, which create, edit or list specific named data items used by the prediction procedures.

Procedure Preprocessors

IBGPREP interactively reads the data items required by the Blade Geometry Procedure (functional modules RBS, RBA and BLM or IBS, IBA and IBL).

PRFPREP interactively reads the data items required by the Performance Procedure (functional modules PRP and PLD).

SNSPREP interactively reads the data items required by the Noise Procedure (functional modules SPN and PTE).

FLPPREP interactively reads the data items required by the Flight Path Procedure (functional modules ATM, ABS, SFO and GEO).

PLEPREP interactively reads all data items required by the Propagation Procedure (functional modules PRT, PRO, LEV and EFF).

Executive Preprocessors

The IBM-PC version of ANOPP-PAS contains four executive preprocessors, one for each type of input data available to the functional modules.

The User Parameter Editor (UPFEDT) provides the following options:

1. Create a new user parameter file.
2. Edit an existing parameter file.
3. Print a directory listing of all parameter files installed on the system.
4. Print a user parameter file dump to the screen display.
5. Print a user parameter file dump to a listings file.
6. Exit the program.

The Alternate Names Table Editor (ANTEDT) provides the following options:

1. Create a new alternate names table.
2. Edit an existing alternate names table.
3. Print a directory listing of all alternate names table files that are installed on system.
4. Print an alternate names table to the screen display.

5. Print an alternate names table to a listings file.

6. Exit the program.

The Member Manager Editor (MMEDT) provides the following options:

1. Enter a member in a data unit.

2. Print a member to the display screen.

3. Print a member to a listings file.

4. Print a catalog listing of all members on a unit to the screen display.

5. Print a catalog listing of all members on a unit to a listings file.

6. Print a directory listing of all units installed on the system.

7. Edit the records of an existing member.

0. Exit the program.

The Table Manager Editor (TMEDT) provides the following options:

1. To enter a data table on a data unit.

2. Print a data table to the display screen.

3. Print a data table to a listing file.

4. Print a catalog listing of all members on a unit to the screen.

5. Print a catalog listing of all members on a unit to a listings file.

6. Print a directory listing of all units installed on the system.

0. Exit the program.

This section provides documentation for these preprocessors, explains the options available on each, and gives examples of how they are used. The symbol <cr> is used to represent 'press the ENTER key' in this documentation.

Blade Geometry Preprocessor (IBGPREP)

IBGPREP is a BLADE GEOMETRY PREPROCESSOR for interactively entering the input data required by the original Blade Geometry functional modules, RBA, RBS, and BLM, or the Improved Blade Geometry modules, IBS, IBA, and IBL.

File IBG.PAR will be created. If the original Blade Geometry modules will be used, then IBG.PAR will be copied to files: RBS.PAR, RBA.PAR and BLM.PAR. If the Improved Blade Geometry modules will be used, then IBG.PAR will be copied to files: IBS.PAR, IBA.PAR, and IBL.PAR.

Files IBGPREP.EXE and MAKE.BAT must be installed on directory C:\ANOPP\EXE. The propeller blade geometry input file, <propeller id>.DEF (for the original Blade Geometry modules) or <propeller id>.IDF (for the Improved Blade Geometry modules) must exist on directory C:\ANOPP. (The <propeller id> is the identification name (1-8 characters) chosen by the user for a propeller. All predictions made for that propeller will be made on directory C:\ANOPP\<propeller id>.)

To create a directory for a new propeller and run the BLADE GEOMETRY PREPROCESSOR, the user types:

MAKE <propeller id>

Example: If the user wishes to run predictions with a new propeller, BLADEB, then the user must create a blade geometry input file named BLADEB.DEF (if the original Blade Geometry modules are to be used) or BLADEB.IDF (if the Improved Blade Geometry modules are to be used) and place it on directory C:\ANOPP (see PART 1 for description of file) and type:

MAKE BLADEB <cr>

A directory C:\ANOPP\BLADEB will be created and will be made the current directory, and the blade geometry preprocessor will be run.

The program displays the following prompt:

```
DO YOU WISH TO USE THE IMPROVED BLADE GEOMETRY MODULES?
PLEASE PRESS Y OR N (DO NOT PRESS ENTER):
```

If the user wishes to use the Improved Blade Geometry modules, IBS, IBA, and IBL, then Y is pressed. If the user wishes to use the original Blade Geometry modules, RBS, RBA, and BLM, then N is pressed.

PART 1

If a current blade geometry file exists, then the program displays the blade name or description. If the file does not exist, then the program prints an error message and execution of the preprocessor is terminated.
If the file exists, then program asks the user to verify the blade geometry description:

```
DO YOU WISH INPUT BLADE GEOMETRY FOR
  A DIFFERENT BLADE ?
```

The user enters "Y" and presses the ENTER key, if a different blade is to be used, or presses only the ENTER key if the current blade will be used.
If Y is entered, then the program prints an exit message and execution of the preprocessor is terminated.

```
**********************************************************************
**********************************************************************
**                                                                  **
** Example: If current blade geometry file, BLADEB.DEF,             **
**          contains the blade geometry for BLADEB and              **
**          exists on C:\ANOPP, then the following prompts          **
**          will be displayed:                                      **
**                                                                  **
**         (displayed on terminal)              (typed by user)     **
**    -------------------------------           ---------------     **
**    THE INPUT FILE ON THIS DIRECTORY                              **
**     CONTAINS BLADE GEOMETRY FOR -                                **
**          BLADEB                                                  **
**    DO YOU WISH INPUT BLADE GEOMETRY FOR                          **
**     A DIFFERENT BLADE ?                                          **
**    ENTER "Y" OR PRESS "ENTER" FOR NO=>          <cr>            **
**                                                                  **
**********************************************************************
**********************************************************************
```

BLADE GEOMETRY INPUT FILE FORMAT FOR IMPROVED BLADE GEOMETRY MODULES: IBS, IBA, AND IBL

If the Improved Blade Geometry modules, IBS, IBA, and IBL are to be used, then the blade geometry input file must be named <propeller id>.IDF (where <propeller id> is any name choosen by the user), and must have the following format:

LINE	DESCRIPTION OF DATA
1	BLADE NAME OR DESCRIPTION (1-80 CHARACTERS)
2	NUMBER OF AIRFOIL SECTIONS ON THE BLADE (INTEGER)
3	A SET OF NUMBERS OF DIFFERENT CROSS SECTIONS (INTEGER) (IE. , IF LINE 2 IS: 8 AND LINE 3 IS: 1, 1, 1, 2, 3, THEN THERE ARE 8 DIFFERENT SPANWISE STATIONS, BUT AT THE 4TH AND 5TH SPANWISE STATIONS, THE CROSS SECTIONS HAVE THE SAME (X,Y) COORDINATES AND AT THE 6TH, 7TH, AND 8TH SPANWISE STATIONS, THE CROSS SECTIONS HAVE THE SAME (X,Y) COORDINATES.)
4	SPANWISE COORDINATE, (RE BLADE LENGTH), (REAL), LEADING EDGE ABSCISSA (RE BLADE LENGTH), (REAL), LEADING EDGE ORDINATE (RE BLADE LENGTH), (REAL), CHORD LENGTH, (RE BLADE LENGTH), (REAL), LEADING EDGE RADIUS (RE CHORD LENGTH) (REAL), BLADE TWIST ANGLE MEASURED POSITIVE CLOCKWISE LOOKING OUTBOARD, IN DEGREES (REAL), NUMBER OF (X,Y) PAIRS ON UPPER SURFACE (INTEGER), NUMBER OF (X,Y) PAIRS ON LOWER SURFACE OF FIRST AIRFOIL SECTION (INTEGER).
5	FIRST (X,Y) PAIR ON UPPER SURFACE OF FIRST AIRFOIL SECTION (REAL)
6	SECOND (X,Y) PAIR ON UPPER SURFACE OF FIRST AIRFOIL SECTION (REAL)
.	.
.	.
.	.
N+4	NTH (X,Y) PAIR ON UPPER SURFACE OF FIRST AIRFOIL SECTION (REAL)
N+5	FIRST(X,Y) PAIR ON LOWER SURFACE OF FIRST AIRFOIL SECTION (REAL)

```
N+6      SECOND (X,Y) PAIR ON LOWER SURFACE OF
                FIRST AIRFOIL SECTION ( REAL )
.          .
.          .
.          .
N+4+M    MTH (X,Y) PAIR ON LOWER SURFACE OF FIRST
                AIRFOIL SECTION (REAL)
.          .
.          .
.          .
```

(LINES 4 THROUGH N+4+M MAY REPEAT FOR EACH AIRFOIL SECTION. IF THERE ARE IDENTICAL CROSS SECTIONS, THE SAME FORMAT OF RECORD THREE REPEATS FOR EACH OF THE SAME CROSS SECTIONS AND THE (X,Y) PAIRS ARE ENTERED ONLY ONCE. NOTE : M AND N MAY BE UNEQUAL AND MAY BE DIFFERENT FOR EACH SECTION.)

(Each line may end with an optional <space>$)

Text file C:\ANOPP\EXE\GEOM.IDF can be examined as an example of a Blade Geometry input file for the Improved Blade Geometry modules.

BLADE GEOMETRY INPUT FILE FORMAT FOR ORIGINAL BLADE GEOMETRY MODULES: RBS, RBA, AND BLM

If the original Blade Geometry modules, RBS, RBA, and BLM are to be used, then the blade geometry input file must be named <propeller id>.DEF (where <propeller id> is any name choosen by the user), and must have the following format:

LINE	DESCRIPTION OF DATA
1	BLADE NAME OR DESCRIPTION (1-80 CHARACTERS)
2	NUMBER OF AIRFOIL SECTIONS ON THE BLADE (INTEGER)

```
3       SPANWISE COORDINATE (REAL),
        LEADING EDGE ABSCISSA (REAL),
        LEADING EDGE ORDINATE (REAL),
        CHORD LENGTH (REAL),
        ( THE FIRST FOUR ENTRIES ON LINE 2 MUST BE
          IN THE SAME UNITS AS THE SYSTEM OFUNITS
          INDICATOR LISTED ABOVE)
        LEADING EDGE RADIUS (RE CHORD LENGTH) (REAL),
        BLADE TWIST ANGLE MEASURED POSITIVE CLOCKWISE
          LOOKING OUTBOARD, IN DEGREES (REAL),
        NUMBER OF (X,Y) PAIRS ON UPPER SURFACE (INTEGER),
        NUMBER OF (X,Y) PAIRS ON LOWER SURFACE
          OF FIRST AIRFOIL SECTION (INTEGER).
4       FIRST (X,Y) PAIR ON UPPER SURFACE OF
                FIRST AIRFOIL SECTION (REAL)
5       SECOND (X,Y) PAIR ON UPPER SURFACE OF
                FIRST AIRFOIL SECTION (REAL)
.         .
.         .
.         .
N+3     NTH (X,Y) PAIR ON UPPER SURFACE OF FIRST
                AIRFOIL SECTION ( REAL )
N+4     FIRST(X,Y) PAIR ON LOWER SURFACE OF
                FIRST AIRFOIL SECTION ( REAL )
N+5     SECOND (X,Y) PAIR ON LOWER SURFACE OF
                FIRST AIRFOIL SECTION ( REAL )
.         .
.         .
.         .
N+3+M   MTH (X,Y) PAIR ON LOWER SURFACE OF FIRST
                AIRFOIL SECTION (REAL)
.         .
.         .
.         .

  ( LINES 2 THROUGH N+3+M REPEAT FOR EACH AIRFOIL SECTION.
    NOTE : M AND N MAY BE UNEQUAL AND MAY BE DIFFERENT FOR
    EACH SECTION. )

 ( Each line may end with an optional <space>$ )
```

Text file C:\ANOPP\EXE\GEOM.DEF can be examined as an example of a Blade Geometry input file for the original Blade Geometry modules.

PART 2

The program displays the description and default setting, for all single value input data required by each of the blade geometry functional modules. (If the data is listed as (REAL), then the entry must include a decimal point. If the data is listed as (INTEGER), then the entry may NOT include a decimal point.) The user is prompted to enter the value or press "ENTER" only, if the default setting is to be used, for the following data:

```
**********************************************
ENTER SYSTEM OF UNITS INDICATOR:
   "S" FOR SI UNITS
   "E" FOR ENGLISH UNITS
OR PRESS "ENTER" ONLY FOR DEFAULT (SI)=>
```

If SI units are chosen, then the following is displayed:

```
**********************************************
ENTER BLADE LENGTH IN METERS (REAL)
   OR PRESS "ENTER" ONLY FOR
   DEFAULT (1.016 METERS)=>
```

If ENGLISH units are chosen, then the following is displayed:

```
**********************************************
ENTER BLADE LENGTH IN INCHES (REAL)
   OR PRESS "ENTER" ONLY FOR
   DEFAULT (40 INCHES)=>
```

If a value is entered for blade length, then that value is displayed on the screen.

```
**************************************************************************
**************************************************************************
**                                                                      **
**  Example:                                                            **
**         ENTER BLADE LENGTH IN INCHES (REAL)                          **
**          OR PRESS "ENTER" ONLY FOR                                   **
**          DEFAULT (40 INCHES)=>                                       **
**                                    (displayed on terminal)           **
**                                                                      **
**  To enter a value of 38 inches, the user types 38.D0                 **
**  and presses the ENTER key:                                          **
**                                                                      **
**                               => 38.D0  <cr>                         **
**                                                                      **
**  The program echos back the following value:                         **
**                                                                      **
**                   38.000000000000000000                              **
**                                                                      **
**************************************************************************
**************************************************************************
```

The program then asks the user to enter a print directive:

```
*********************************************
ENTER PRINT FLAG (INTEGER)
    =0, NO PRINT DESIRED
    =1, INPUT PRINT ONLY
    =2, OUTPUT PRINT ONLY
    =3,BOTH INPUT AND OUTPUT PRINT
OR PRESS "ENTER" ONLY FOR DEFAULT(3)=>
```

```
*********************************************
If the user wishes to have both input and output data printed
to results files: RESULTS.RBS, RESULTS.RBA, and RESULTS.BLM,
or RESULTS.IBS, RESULTS.IBA, and RESULTS.IBL,then only the
ENTER key is pressed.  These files will be created
when functional modules, RBS, RBA, and BLM, or IBS, IBA,
and IBL are executed.
```

If the Improved Blade Geometry modules, IBS, IBA, and IBL are to be used, the program then asks the user to enter the following:

```
*********************************************
ENTER RATIO OF SPECIFIC HEAT (REAL)'
   OR PRESS "ENTER" ONLY FOR
   DEFAULT (1.400) =>
*********************************************'
  ENTER COMPRESSIBILITY CORRECTION INDICATOR'
  FOR PRESSURE COEFFICIENTS:'
      "K" FOR KARMAN-TSIEN COMPRESSIBILITY CORRECTION'
      "G" FOR GLAUERT COMPRESSIBILITY CORRECTION'
    "N" FOR NO CORRECTION'
   OR PRESS "ENTER" ONLY FOR DEFAULT (KARMAN-TSIEN)=>'
  *********************************************'
  ENTER COMPRESSIBILITY CORRECTION INDICATOR'
  FOR LIFT COEFFICIENTS:'
    "G" FOR GLAUERT COMPRESSIBILITY CORRECTION'
    "N" FOR NO CORRECTION'
   OR PRESS "ENTER" ONLY FOR DEFAULT (NO CORRECTION)'
  *********************************************'
  ENTER MODEL INDICATOR:'
   "F" FOR FLAT PLATE DESCRIPTION'
   "L" FOR FULL LAMINAR AND TURBULENT FLOW DESCRIPTION'
   OR PRESS "ENTER" ONLY FOR DEFAULT (L)=>'
  *********************************************'
```

PART 3

If a set of chordwise blade stations exists on file GRID.UNT, then the program displays those values.
EXAMPLE:

```
CURRENT CHORDWISE STATIONS:
*************************************************************
UNIT MEMBER GRID(XI2) - CHORDWISE STATIONS-
 0.000   0.100   0.200   0.300   0.400   0.500   0.600
 0.700   0.800   0.900   1.000
*************************************************************
```

If the set of chordwise stations does not exist on file GRID.UNT, then the program displays folloving default values:

```
DEFAULT CHORDWISE STATIONS:
*************************************************************
UNIT MEMBER GRID(XI2) - CHORDWISE STATIONS-
 0.000   0.050   0.100   0.150   0.200   0.250   0.300
 0.350   0.400   0.450   0.500   0.550   0.600   0.650
 0.700   0.750   0.800   0.850   0.900   0.950   1.000

*************************************************************
```

After either the current or default values are printed to the screen, the folloving prompt is displayed:

```
*****************************************************
DO YOU WISH TO CHANGE THESE CHORDWISE STATIONS?
 ENTER "Y" OR PRESS "ENTER" FOR NO=>
```

If the user wishes to use the printed values (either the current values or default values), then only the ENTER key is pressed. If the values are to be changed, then Y and the ENTER key are pressed and the folloving prompt appears:

```
*****************************************************
   ENTER THE NUMBER OF CHORDWISE STATIONS (INTEGER)
      (MAXIMUM OF 25) =>
```

The program will then prompt the user to enter each of the chordvise stations.

```
**********************************************************************
**********************************************************************
**                                                                  **
**  Example: (interactively enter chordwise stations)               **
**                                                                  **
**         (displayed on terminal)                (typed by user)   **
**    -------------------------------          ---------------      **
**    ENTER THE NUMBER OF CHORDWISE STATIONS (INTEGER)              **
**      (MAXIMUM OF 25) =>                        7        <cr>     **
**                                                                  **
**             ENTER POSITION 1 (REAL)  =>       0.000    <cr>     **
**             ENTER POSITION 2 (REAL)  =>       0.200    <cr>     **
**             ENTER POSITION 3 (REAL)  =>       0.250    <cr>     **
**             ENTER POSITION 4 (REAL)  =>       0.500    <cr>     **
**             ENTER POSITION 5 (REAL)  =>       0.700    <cr>     **
**             ENTER POSITION 6 (REAL)  =>       0.750    <cr>     **
**             ENTER POSITION 7 (REAL)  =>       1.000    <cr>     **
**                                                                  **
**********************************************************************
**********************************************************************
```

PART 4

If a set of Mach numbers exists on file RBA.UNT, (if the original Blade Geometry modules are to be used) or on file IBA.UNT (if the Improved Blade Geometry modules are to be used), then the program displays those values.

EXAMPLES:

If the original Blade Geometry modules are to used:

```
CURRENT MACH NUMBERS:
************************************************
UNIT MEMBER RBA(MACH) - MACH NUMBERS
 0.100   0.200  0.500   0.900
************************************************
```

If the Improved Blade Geometry modules are to used:

```
CURRENT MACH NUMBERS:
************************************************
UNIT MEMBER IBA(MACH) - MACH NUMBERS
 0.100   0.200  0.500   0.900
************************************************
```

If the set of Mach numbers does not exist, then the program displays one of the following sets of default values:

If the original Blade Geometry modules are to used:

```
************************************************
UNIT MEMBER RBA(MACH) - MACH NUMBERS
 0.100   0.300   0.500   0.700  0.900
************************************************
```

If the Improved Blade Geometry modules are to used:

```
************************************************
UNIT MEMBER IBA(MACH) - MACH NUMBERS
 0.100   0.300   0.500   0.700  0.900
************************************************
```

After either the current or default values are printed to the screen, the following prompt is displayed:

```
************************************************
DO YOU WISH TO CHANGE THESE MACH NUMBERS ?
 ENTER "Y" OR PRESS "ENTER" FOR NO=>
```

If the user wishes to use the printed values (either the current values or default values), then only the ENTER key is pressed. If the values are to be changed, then Y and the ENTER key are pressed and the following prompt appears:

```
************************************************
 ENTER NUMBER OF MACH NUMBERS (INTEGER)
  ( MAXIMUM OF 10) =>
************************************************
```

The number of MACH NUMBERS to be entered should be typed by the user. The program will then prompt the user to enter each MACH NUMBER.

```
**********************************************************************
**********************************************************************
**                                                                  **
**  Example:    (interactively enter 5 MACH NUMBERS)                **
**                                                                  **
**         (displayed on terminal)          (typed by user)         **
**    ------------------------------     ----------------           **
**      ENTER NUMBER OF MACH NUMBERS (INTEGER)                      **
**       ( MAXIMUM OF 10)                =>        5   <cr>        **
**            ENTER POSITION 1 (REAL)    =>    0.100   <cr>        **
**            ENTER POSITION 2 (REAL)    =>    0.200   <cr>        **
**            ENTER POSITION 3 (REAL)    =>    0.300   <cr>        **
**            ENTER POSITION 4 (REAL)    =>    0.500   <cr>        **
**            ENTER POSITION 5 (REAL)    =>    0.700   <cr>        **
**                                                                  **
**********************************************************************
**********************************************************************
```

PART 5

If a set of ANGLES OF ATTACK exists on file RBA.UNT,
(if the original Blade Geometry modules are to be used)
or on file IBA.UNT (if the Improved Blade Geometry modules
are to be used), then the program displays those values.

EXAMPLE:

If the original Blade Geometry modules are to
used:

```
CURRENT ANGLES OF ATTACK:
************************************************
UNIT MEMBER RBA(ALPHA) - ANGLES OF ATTACK
 1.000   3.000  5.000
************************************************
```

If the Improved Blade Geometry modules are to
used:

```
CURRENT ANGLES OF ATTACK:
************************************************
UNIT MEMBER IBA(ALPHA) - ANGLES OF ATTACK
 1.000   3.000  5.000
************************************************
```

If the set of ANGLES OF ATTACK does not exist, then the program displays one of the following sets of default values:

If the original Blade Geometry modules are to used:

```
**************************************************
UNIT MEMBER RBA(ALPHA) - ANGLES OF ATTACK
   1.000   3.000   6.000
**************************************************
```

If the Improved Blade Geometry modules are to used:

```
**************************************************
UNIT MEMBER IBA(ALPHA) - ANGLES OF ATTACK
   1.000   3.000   6.000
**************************************************
```

After either the current or default values are printed to the screen, the following prompt is displayed:

```
**************************************************
DO YOU WISH TO CHANGE THESE ANGLES OF ATTACK ?
 ENTER "Y" OR PRESS "ENTER" FOR NO=>
```

If the user wishes to use the printed values (either the current values or default values), then only the ENTER key is pressed. If the values are to be changed, then Y and the ENTER key are pressed and the following prompt appears:

```
**************************************************
 ENTER NUMBER OF ANGLES OF ATTACK (INTEGER)
  ( MAXIMUM OF 10) =>
**************************************************
```

The number of ANGLES OF ATTACK to be entered should be typed by the user. The program will then prompt the user to enter each ANGLE OF ATTACK.

```
**********************************************************************
**********************************************************************
**                                                                  **
**  Example: (interactively enter 5 ANGLES OF ATTACK )              **
**                                                                  **
**         (displayed on terminal)              (typed by user)     **
**    --------------------------------        ----------------      **
**     ENTER NUMBER OF ANGLES OF ATTACK(INTEGER)                     **
**      ( MAXIMUM OF 10 ) =>                    5                   **
**                                                                  **
**             ENTER POSITION 1          =>    -4.000   <cr>       **
**             ENTER POSITION 2          =>    -2.000   <cr>       **
**             ENTER POSITION 3          =>     0.000   <cr>       **
**             ENTER POSITION 4          =>     2.000   <cr>       **
**             ENTER POSITION 5          =>     4.000   <cr>       **
**                                                                  **
**********************************************************************
**********************************************************************
```

The following exit message will be displayed:

```
   ************************************
   *                                  *
   *   BLADE GEOMETRY INPUT COMPLETED *
   *                                  *
   ************************************
```

Performance Preprocessor (PRFPREP)

PRFPREP is a PERFORMANCE MODULE PREPROCESSOR for interactively entering the input data required by the performance functional modules, PRP and PLD.
User Parameter file PRF.PAR will be created and will be copied to files PRP.PAR, and PLD.PAR. Alternate names file PLD.ANT will be created.

The user must make C:\ANOPP\<propeller id> the current directory. (If it is not the current directory, then the user types: CHANGE <propeller id> and presses ENTER.)

<propeller id> is the identification name given to a propeller (1-8 characters) and is also the name of the subdirectory used for predictions with that propeller (see MAKE command in Section IV.)

File PRFPREP.EXE and INPUT.BAT must be installed on directory C:\ANOPP\EXE.
To run PRFPREP the user types:

INPUT <cr>

The system will ask the user if each preprocessor is to be run. The following prompt is displayed:

DO YOU WISH TO RUN THE PERFORMANCE PREPROCESSOR?
PLEASE PRESS Y OR N (DO NOT PRESS "ENTER"):

The user types Y to run the preprocessor.

PART 1

The program displays the following description and default setting, in parenthesis, for the single value input data required by performance modules PRP and PLD. The user is prompted to enter the value or press the ENTER key only, if the default setting is to be used. (If the data is listed as (REAL), then the entry must include a decimal point. If the data is listed as (INTEGER), then the entry may NOT include a decimal point.)

```
   ********************************************************
     ENTER PROPELLER ANGLE OF ATTACK IN DEGREES (REAL)
     OR PRESS "ENTER" ONLY FOR DEFAULT (4.30) =>

   ********************************************************
     ENTER MACH NUMBER OF BLADE TIP (REAL)
     OR PRESS "ENTER" ONLY FOR DEFAULT (0.86434) =>

   ********************************************************
     ENTER INFLOW MACH NUMBER (REAL)
     OR PRESS "ENTER" ONLY FOR DEFAULT (0.12327) =>

   ********************************************************
     ENTER NUMBER OF PROPELLER BLADES (INTEGER)
     OR PRESS "ENTER" ONLY FOR DEFAULT (2) =>

   ********************************************************
```

```
**********************************************************************
**********************************************************************
**                                                                  **
**  Example:                                                        **
**       ENTER INFLOW MACH NUMBER (REAL)                            **
**       OR PRESS "ENTER" ONLY FOR DEFAULT  ( 0.12327) =>           **
**                                           (displayed             **
**                                            on terminal)          **
**                                                                  **
**  To enter a value of 0.4, the user                               **
**  types 0.4D0 and the presses the ENTER key:                      **
**                                                                  **
**                                  => 0.4D0  <cr>                  **
**                                                                  **
**  To enter the default value of 0.12327, the user                 **
**  presses the ENTER key:                                          **
**                                                                  **
**                                  =>   <cr>                       **
**********************************************************************
**********************************************************************
```

PART 2

The program then displays the following prompt:

```
 ******************************************************
 DO YOU WISH TO COMPUTE OR ENTER BLADE PITCH VALUE?
  ENTER "C" FOR COMPUTE OR "E" FOR ENTER=>
 ******************************************************
```

The user must press "E" and the ENTER key, if an exact value for the the blade pitch is to be entered. The user must press "C" and the ENTER key, if the blade pitch is to be computed from an initial guess and convergence of the power coefficient to measured power, using the secant method.
If the user chooses E, then the program displays the following description and default setting, in parenthesis, for the blade pitch input data. The user is prompted to enter the value or press "ENTER" only, if the default setting is to be used.

```
 ******************************************************
   ENTER BLADE PITCH SETTING AT THE ROOT (RADIANS)
     OR PRESS "ENTER" ONLY FOR DEFAULT  (0.0) =>
 ******************************************************
```

If the user chooses C, then the program displays the following description and default setting, in parenthesis, for all single value input data required for computation of the blade pitch. The user is prompted to enter the value or press "ENTER" only, if the default setting is to be used, for each of the following values:

```
 ******************************************************
 ENTER INITIAL GUESS FOR PROPELLER PITCH
  AT 3/4 SPAN IN DEGREES (REAL) OR PRESS
  "ENTER"  FOR DEFAULT (20.8 DEGREES)=>

 ENTER BLADE TWIST FROM ROOT TO 3/4 SPAN
 IN DEGREES (REAL) OR PRESS "ENTER" ONLY
 FOR DEFAULT (13.5 DEGREES)=>

   ENTER POWER COEFFICIENT (REAL)
    OR PRESS "ENTER" ONLY FOR
    DEFAULT  (0.47408D-1) =>
```

```
**********************************************************************
**********************************************************************
**                                                                  **
**  Example:                                                        **
**     ENTER INITIAL GUESS FOR PROPELLER PITCH ( displayed          **
**      AT 3/4 SPAN IN DEGREES (REAL) OR PRESS  on terminal)        **
**      "ENTER"  FOR DEFAULT (20.8 DEGREES)=>                       **
**  To enter a value of 20.0, the user                              **
**  types 20.0d0 and presses the ENTER key:                         **
**                                                                  **
**                                  => 20.0D0  <cr>                 **
**                                                                  **
**  To enter default value of 20.8, the user                        **
**  presses only the ENTER key:                                     **
**                                                                  **
**                                  =>   <cr>                       **
**********************************************************************
**********************************************************************
```

PART 3

The program displays the description and current value for the print directive (if it was entered in the blade geometry preprocessor) and asks the user if the value is to be changed.

If the value was not found or if the user requested to change the value, then the program displays the following description and default setting, in parenthesis, for the print directive. The user enters the value or presses "ENTER" only, if the default setting is to be used.

```
 ******************************************************
    ENTER PRINT FLAG (INTEGER)
          =0, NO PRINT DESIRED
          =1, INPUT PRINT ONLY
          =2, OUTPUT PRINT ONLY
          =3, BOTH INPUT AND OUTPUT PRINT
    OR PRESS "ENTER" ONLY FOR DEFAULT (3)  =>
******************************************************
```

If the user chooses 1, 2, or 3, then any requested input and any printed output for functional module PRP will be located in file RESULTS.PRP, on the current subdirectory. This file will be created when the functional module is executed. Only the input data for functional module PLD will be printed to file RESULTS.PLD.

PART 4

If a set of spanvise blade stations exists on file GRID.UNT then the program displays those values.

```
*********************************************************
*********************************************************
** EXAMPLE:                                            **
**                                                     **
**   CURRENT SPANWISE STATIONS:                        **
**  ***********************************************    **
**   UNIT MEMBER  GRID(XI1) - SPANWISE STATIONS-       **
**   DEFAULT GRID:                                     **
**     0.400   0.500   0.600   0.700   0.750           **
**                                                     **
**     0.800   0.850   0.900   0.950   0.990           **
**                                                     **
*********************************************************
*********************************************************
```

If the set of spanvise stations does not exist on file GRID.UNT, then the program displays the folloving default values:

```
DEFAULT SPANWISE STATIONS:
*************************************************************
UNIT MEMBER  GRID(XI1) - SPANWISE STATIONS-
 0.400   0.500   0.600   0.700   0.750

 0.800   0.850   0.900   0.950   0.990
*************************************************************
```

After the current or default values are printed to the screen, the folloving prompt is displayed:

```
*************************************************
 DO YOU WISH TO CHANGE THESE SPANWISE STATIONS?
  ENTER "Y" OR PRESS "ENTER" FOR NO =>
*************************************************
```

If the user vishes to use the printed values (either the current values or the default values), then only the ENTER key is pressed. If the values are to be changed, then Y and the ENTER key are pressed and the folloving prompt appears:

```
*************************************************************
 ENTER NUMBER OF SPANWISE STATIONS  (INTEGER)
   (MAXIMUM OF 10)  =>
*************************************************************
```

The program will then prompt the user to enter each of the spanwise stations.

```
**********************************************************************
**********************************************************************
**                                                                  **
**  Example: (interactively enter 7 spanwise stations  )            **
**                                                                  **
**         (displayed on terminal)              (typed by user)     **
**    --------------------------------        ----------------      **
**    ENTER NUMBER OF SPANWISE STATIONS (INTEGER)                   **
**     (MAXIMUM OF 10)                  =>      7   <cr>            **
**             ENTER POSITION 1         =>      0.000   <cr>       **
**             ENTER POSITION 2         =>      0.200   <cr>       **
**             ENTER POSITION 3         =>      0.250   <cr>       **
**             ENTER POSITION 4         =>      0.500   <cr>       **
**             ENTER POSITION 5         =>      0.700   <cr>       **
**             ENTER POSITION 6         =>      0.750   <cr>       **
**             ENTER POSITION 7         =>      0.900   <cr>       **
**                                                                  **
**********************************************************************
**********************************************************************
```

PART 5

If a set of BLADE INPLANE STATIONS exists on file GRID.UNT, then the program displays those values.

```
**********************************************************************
**********************************************************************
**  EXAMPLE:                                                        **
**                                                                  **
**  CURRENT BLADE INPLANE STATIONS:                                 **
**  ***********************************************             **
**  UNIT MEMBER GRID(PSI)- BLADE INPLANE STATIONS                   **
**  0.0000   0.2500   0.3750  0.5000   0.6250   0.7500              **
**  0.8750   1.0000                                                 **
**                                                                  **
**********************************************************************
**********************************************************************
```

If the set of BLADE INPLANE STATIONS does not exist, then the program displays the following default values:

```
DEFAULT BLADE INPLANE STATIONS:
************************************************
UNIT MEMBER GRID(PSI)- BLADE INPLANE STATIONS
0.0000   0.0625   0.1250   0.1875   0.2500   0.3125   0.3750
0.4375   0.5000   0.5625   0.6250   0.6875   0.7500   0.8125
0.8750   0.9375   1.0000
************************************************
```

After either the current or default values are printed to the screen, the following prompt is displayed:

```
************************************************
DO YOU WISH TO CHANGE THESE INPLANE STATIONS?
 ENTER "Y" OR PRESS "ENTER" FOR NO=>
```

If the user wishes to use the printed values (either the current values or default values), then only the enter key is pressed. If the values are to be changed, then Y and the ENTER key are pressed and the following prompt appears:

```
************************************************
 ENTER NUMBER OF BLADE INPLANE STATIONS (INTEGER)
   (MAXIMUM OF 25)  =>
****************************************
```

The number of BLADE INPLANE STATIONS to be entered should be typed by the user. The program will then prompt the user to enter each of the BLADE INPLANE STATIONS.

```
**********************************************************************
**********************************************************************
**                                                                  **
**  Example:    (interactively enter 3 BLADE INPLANE STATIONS)      **
**                                                                  **
**         (displayed on terminal)              (typed by user)     **
**    ---------------------------------        ---------------     **
**      ENTER NUMBER OF BLADE INPLANE STATIONS                      **
**      (INTEGER) (MAXIMUM OF 25)       =>         3    <cr>       **
**             ENTER POSITION 1 (REAL)  =>     0.000    <cr>       **
**             ENTER POSITION 2 (REAL)  =>     0.100    <cr>       **
**             ENTER POSITION 3 (REAL)  =>     0.200    <cr>       **
**                                                                  **
**********************************************************************
**********************************************************************
```

PART 6

The program then verifies that the unit members, required by functional modules PRP and PLD, exist and do not exceed maximum numbers. Any errors or missing files will cause one or more of the following messages to be displayed:

```
** REQUIRED FILE GRID.UNT CONTAINING
       UNIT MEMBER GRID(XI1)DOES NOT EXIST
      ------------------------------------------------
      ( GRID(XI1) is created by execution of module RBS
        or it is created by the performance prepocessor.)

** REQUIRED FILE GRID.UNT CONTAINING
       UNIT MEMBER GRID(XI2) DOES NOT EXIST
      ------------------------------------------------
      ( GRID(XI2) is created by the blade geometry preprocessor.)

** REQUIRED FILE GRID.UNT CONTAINING
       UNIT MEMBER GRID(PSI) DOES NOT EXIST
      ------------------------------------------------
      ( GRID(PSI) is created by the performance prepocessor.)

*** ERROR- NUMBER OF CHORDWISE
       STATIONS EXCEEDS 25 **

*** ERROR- NUMBER OF SPANWISE
       STATIONS EXCEEDS 10 **

*** ERROR- NUMBER OF INPLANE
       STATIONS EXCEEDS 25 **

*** ERROR- PRODUCT OF CHORDWISE, SPANWISE, AND
       INPLANE STATIONS EXCEEDS 3600    **
```

If the original Blade Geometry modules: RBS, RBA, and BLM are to be used, then the following messages may be displayed:

```
** REQUIRED FILE RBS.UNT CONTAINING UNIT MEMBER
       RBS(SLPSPN) DOES NOT EXIST ON THIS DIRECTORY
      ------------------------------------------------
      ( RBS(SLPSPN) is created by execution of module RBS.)
```

```
** REQUIRED FILE RBS.UNT CONTAINING UNIT MEMBER
      RBS(SLPCRD) DOES NOT EXIST ON THIS DIRECTORY
     ------------------------------------------------
     ( RBS(SLPCRD) is created by execution of module RBS.)

** REQUIRED FILE RBS.UNT CONTAINING UNIT MEMBER
      RBS(SPAN) DOES NOT EXIST ON THIS DIRECTORY
     ------------------------------------------------
     ( RBS(SPAN) is created by execution of module RBS.)

** REQUIRED FILE BLM.UNT CONTAINING UNIT MEMBER
      BLM(LIFTDRAG) DOES NOT EXIST ON THIS DIRECTORY
     ------------------------------------------------
     ( BLM(LIFTDRAG) is created by execution of module BLM.)

** REQUIRED FILE RBA.UNT CONTAINING UNIT MEMBER
      RBA(MACH) DOES NOT EXIST ON THIS DIRECTORY
     ------------------------------------------------
     ( RBA(MACH) is created by the blade geometry preprocessor.)
```

If the Improved Blade Geometry modules: IBS, IBA, and IBL are to be used, then the following messages may be displayed:

```
** REQUIRED FILE IBS.UNT CONTAINING UNIT MEMBER
      IBS(SLPSPN) DOES NOT EXIST ON THIS DIRECTORY
     ------------------------------------------------
     ( IBS(SLPSPN) is created by execution of module IBS.)

** REQUIRED FILE IBS.UNT CONTAINING UNIT MEMBER
      IBS(SLPCRD) DOES NOT EXIST ON THIS DIRECTORY
     ------------------------------------------------
     ( IBS(SLPCRD) is created by execution of module IBS.)

** REQUIRED FILE IBS.UNT CONTAINING UNIT MEMBER
      IBS(SPAN) DOES NOT EXIST ON THIS DIRECTORY
     ------------------------------------------------
     ( IBS(SPAN) is created by execution of module IBS.)

** REQUIRED FILE IBA.UNT CONTAINING UNIT MEMBER
      IBA(LIFT) DOES NOT EXIST ON THIS DIRECTORY
     ------------------------------------------------
     ( IBA(LIFT) is created by execution of module IBA.)

** REQUIRED FILE IBL.UNT CONTAINING UNIT MEMBER
      IBL(DRAG) DOES NOT EXIST ON THIS DIRECTORY
     ------------------------------------------------
     (IBL(DRAG) is created by execution of module IBL.
```

```
** REQUIRED FILE IBA.UNT CONTAINING UNIT MEMBER
       IBA(MACH) DOES NOT EXIST ON THIS DIRECTORY
      ---------------------------------------------
      ( IBA(MACH) is created by the blade geometry preprocessor.)
```

The following exit message will be displayed:

```
     ************************************
     *                                  *
     *  PERFORMANCE INPUT COMPLETED     *
     *                                  *
     ************************************
   Execution suspended: PRESS "ENTER" TO CONTINUE
```

The user must press the ENTER key to continue with the next preprocessor prompt.

Noise Preprocessor (SNSPREP)

SNSPREP is a NOISE MODULE PREPROCESSOR for interactively entering the input data required by the noise functional modules SPN and PTE.
User Parameter files SPN.PAR and PTE.PAR will be created.

The user must make C:\ANOPP\<propeller id> the current directory. (If it is not the current directory, the user types: CHANGE <propeller id> and presses ENTER.)

<propeller id> is the identification name given to a propeller (1-8 characters) and is also the name of the subdirectory used for predictions with that propeller (see MAKE command in Section IV.)

File SNSPREP.EXE and INPUT.BAT must be installed on directory C:\ANOPP\EXE.
To run SNSPREP the user types:

INPUT <cr>

The system will ask the user if each preprocessor is to be run. The following prompt is displayed:

```
DO YOU WISH TO RUN THE NOISE PREPROCESSOR?
PLEASE PRESS Y OR N (DO NOT PRESS "ENTER"):
```

The user types Y to run the preprocessor.

PART 1

The program displays prompts for the single value input data required by the noise module. If the value has been entered for the performance modules, then the program displays the description and given value and asks if the value is to be changed. If the value is to be changed or if the value was was not found, then the description and default setting, in parenthesis, are displayed. The user is prompted to enter the value or press the ENTER key only, if the default setting is to be used. (If the data is listed as (REAL), then the entry must include a decimal point. If the data is listed as (INTEGER), then the entry may NOT include a decimal point.) The program displays prompts for the following input data:

```
      ********************************************
         SYSTEM OF UNITS INDICATOR:
            "S" FOR SI UNITS
            "E" FOR ENGLISH UNITS
         DEFAULT (SI)
      ********************************************
      ********************************************
If SI units entered:
         BLADE LENGTH IN METERS (REAL)
         DEFAULT (1.016 METERS)
      ********************************************
If ENGLISH units entered:
         BLADE LENGTH IN INCHES (REAL)
         DEFAULT (40. INCHES)
      ********************************************
      ********************************************
         NUMBER OF PROPELLER BLADES (INTEGER)
         DEFAULT (2)
      ********************************************
         MACH NUMBER OF BLADE TIP (REAL)
         DEFAULT  (0.86434)
      ********************************************
         INFLOW MACH NUMBER (REAL)
         DEFAULT  ( 0.12327)
      ********************************************
         BLADE PITCH SETTING AT THE ROOT
         (REAL) DEFAULT (0.0 RADIANS)
      ********************************************
If SI units entered:
         DENSITY IN KG/M**3 (REAL)
         DEFAULT (1.164 KG/M**3)
      ********************************************
If ENGLISH units entered:
         DENSITY IN SLUGS/FT**3 (REAL)
         DEFAULT (2.3766E-3 SLUGS/FT**3)
      ********************************************
```

```
      ********************************************
         PRINT FLAG (INTEGER)
         =0, NO PRINT DESIRED
         =1, INPUT PRINT ONLY
         =2, OUTPUT PRINT ONLY
         =3, BOTH INPUT AND OUTPUT PRINT
       DEFAULT (3)
       ********************************************
       ********************************************
If SI units entered:
       AMBIENT SPEED OF SOUND IN METERS/SEC
        DEFAULT (342.2 M/S)
       ********************************************
If ENGLISH units entered:
       AMBIENT SPEED OF SOUND IN FEET/SEC
        DEFAULT (1122.7034 FT/S)
       ********************************************
       ********************************************
       BLADE LOADING SELECTOR:
            "S" FOR STEADY LOADING
            "T" FOR TIME DEPENDENT LOADING
       DEFAULT (TIME DEPENDENT)
       ********************************************
       COMPUTATIONAL METHOD SELECTOR (INTEGER)
          1, FULL BLADE FORMULATION
          3, COMPACT CHORD APPROXIMATION
          4, COMPACT SOURCE APPROXIMATION
       DEFAULT  ( 1, FULL BLADE FORMULATION)
       ********************************************

**********************************************************************
**********************************************************************
**                                                                  **
**  Example 1: ( 4 propeller blades were entered in                 **
**               performance preprocessor and will                  **
**               be changed to 2 for noise module)                  **
**                                                                  **
**       NUMBER OF PROPELLER BLADES = 4                             **
**       DO YOU WISH TO CHANGE NUMBER OF BLADES?                    **
**       ENTER "Y" OR PRESS "ENTER" FOR NO=>        Y <cr>          **
**                                                                  **
**       ENTER NUMBER OF PROPELLER BLADES (INTEGER)                 **
**       OR PRESS "ENTER" ONLY FOR                                  **
**       DEFAULT  ( 2) =>                    <cr>                   **
**                                                                  **
**       The number of blades will be 2.                            **
**                                                                  **
**********************************************************************
**********************************************************************
```

```
**********************************************************************
**********************************************************************
**                                                                  **
**  Example 2:                                                      **
**       ENTER AMBIENT SPEED OF SOUND IN METERS/SEC                 **
**       OR PRESS "ENTER" ONLY FOR                                  **
**       DEFAULT  ( 342.2 M/S) =>              ( displayed          **
**                                            on terminal)          **
**    (If the user has chosen ENGLISH units, then the prompt        **
**         and default setting will be shown in FT/SEC.)            **
**                                                                  **
**  To enter a value of 325.0, the user                             **
**  types 325.D0 and the presses ENTER key:                         **
**                                                                  **
**                                  => 325.D0  <cr>                 **
**                                                                  **
**  To enter default value of 342.2, the user                       **
**  presses only the ENTER key:                                     **
**                                                                  **
**                                  =>   <cr>                       **
**********************************************************************
**********************************************************************
```

PART 2

If a set of OBSERVER POLAR DIRECTIVITY ANGLES exists on file SFIELD.UNT, then the program displays those values.

```
**********************************************************************
**********************************************************************
**  EXAMPLE:                                                        **
**                                                                  **
**  CURRENT OBSERVER POLAR DIRECTIVITY ANGLES :                     **
**  ******************************************                      **
**   UNIT MEMBER  SFIELD(THETA) - OBSERVER POLAR                    **
**        DIRECTIVITY ANGLES :                                      **
**   1.0000   50.0000   90.000  110.0000   150.0000                  **
**                                                                  **
**********************************************************************
**********************************************************************
```

If the set of OBSERVER POLAR DIRECTIVITY ANGLES does not exist, then the program displays following default values:

```
DEFAULT OBSERVER POLAR DIRECTIVITY ANGLES:
************************************************************
 UNIT MEMBER  SFIELD(THETA) - OBSERVER POLAR
                          DIRECTIVITY ANGLES :
   1.0000    30.0000    50.0000    70.0000    90.0000
 110.0000   130.0000   150.0000   179.0000
************************************************************
```

After either the current or default values are printed to the screen, the following prompt is displayed:

```
**************************************************
DO YOU WISH TO CHANGE THESE DIRECTIVITY ANGLES?
 ENTER "Y" OR PRESS "ENTER" FOR NO=>
```

If the user wishes to use the printed values (either the current values or default values), then only the enter key is pressed. If the values are to be changed, then Y and the ENTER key are pressed and the following prompt appears:

```
**************************************************
 ENTER NUMBER OF OBSERVER POLAR DIRECTIVITY ANGLES
 (INTEGER)   (MAXIMUM OF 25)  =>
**************************************************
```

The number of OBSERVER POLAR DIRECTIVITY ANGLES to be entered should be typed by the user. The program will then prompt the user to enter each of the OBSERVER POLAR DIRECTIVITY ANGLES.

```
**********************************************************************
**********************************************************************
**                                                                  **
**  Example:    (interactively enter 5 OBSERVER POLAR               **
**                                     DIRECTIVITY ANGLES )         **
**                                                                  **
**         (displayed on terminal)              (typed by user)     **
**    ---------------------------------        ---------------     **
**      ENTER NUMBER OF OBSERVER POLAR DIRECTIVITY ANGLES           **
**      (INTEGER) (MAXIMUM OF 25)        =>          5    <cr>     **
**             ENTER POSITION 1 (REAL)   =>      1.000    <cr>     **
**             ENTER POSITION 2 (REAL)   =>     50.000    <cr>     **
**             ENTER POSITION 3 (REAL)   =>     90.000    <cr>     **
**             ENTER POSITION 4 (REAL)   =>    110.000    <cr>     **
**             ENTER POSITION 5 (REAL)   =>    150.000    <cr>     **
**                                                                  **
**********************************************************************
**********************************************************************
```

PART 3

If a set of OBSERVER AZIMUTHAL DIRECTIVITY ANGLES exists on file SFIELD.UNT, then the program displays those values.

```
**********************************************************************
**********************************************************************
**  EXAMPLE:                                                        **
**                                                                  **
**  CURRENT OBSERVER AZIMUTHAL DIRECTIVITY ANGLES :                 **
**  ********************************************                    **
**   UNIT MEMBER  SFIELD(PHI) - OBSERVER AZIMUTHAL                  **
**        DIRECTIVITY ANGLES :                                      **
**   0.0000   50.0000                                               **
**                                                                  **
**********************************************************************
**********************************************************************
```

If the set of OBSERVER AZIMUTHAL DIRECTIVITY ANGLES does not exist, then the program displays the following default values:

```
DEFAULT OBSERVER AZIMUTHAL DIRECTIVITY ANGLES:
**************************************************************
 UNIT MEMBER  SFIELD(PHI) - OBSERVER AZIMUTHAL
                          DIRECTIVITY ANGLES:
   0.0000
**************************************************
```

After either the current or default values are printed to the screen, the following prompt is displayed:

```
**************************************************
DO YOU WISH TO CHANGE THESE DIRECTIVITY ANGLES?
 ENTER "Y" OR PRESS "ENTER" FOR NO=>
```

If the user wishes to use the printed values (either the current values or default values), then only the enter key is pressed. If the values are to be changed, then Y and the ENTER key are pressed and the following prompt appears:

```
**************************************************
 ENTER NUMBER OF OBSERVER AZIMUTHAL DIRECTIVITY ANGLES
 (INTEGER)   (MAXIMUM OF 25)  =>
****************************************
```

The number of OBSERVER AZIMUTHAL DIRECTIVITY ANGLES to be entered should be typed by the user. The program will then prompt the user to enter each of the OBSERVER AZIMUTHAL DIRECTIVITY ANGLES.

```
**********************************************************************
**********************************************************************
**                                                                  **
**  Example:    (interactively enter 2 OBSERVER AZIMUTHAL           **
**                                     DIRECTIVITY ANGLES )         **
**                                                                  **
**         (displayed on terminal)              (typed by user)     **
**    ---------------------------------        ---------------     **
**      ENTER NUMBER OF OBSERVER AZIMUTHAL DIRECTIVITY ANGLES       **
**      (INTEGER) (MAXIMUM OF 25)        =>          2   <cr>       **
**             ENTER POSITION 1 (REAL)   =>      0.000   <cr>       **
**             ENTER POSITION 2 (REAL)   =>     30.000   <cr>       **
**                                                                  **
**********************************************************************
**********************************************************************
```

PART 4

The program then verifies that the unit members required by functional modules SPN and PTE exist and do not exceed maximum numbers. Any missing files or errors cause one or more of the following messages to be displayed:

```
** REQUIRED FILE GRID.UNT CONTAINING
       UNIT MEMBER GRID(XI1)DOES NOT EXIST
      ------------------------------------------------
      ( GRID(XI1) is created by execution of module RBS or IBS
        or it is created by the performance preprocessor.)

** REQUIRED FILE GRID.UNT CONTAINING
       UNIT MEMBER GRID(XI2) DOES NOT EXIST
      ------------------------------------------------
      ( GRID(XI2) is created by the blade geometry preprocessor.)
```

```
          *** ERROR- NUMBER OF CHORDWISE
                     STATIONS EXCEEDS 25 **

          *** ERROR- NUMBER OF SPANWISE
                     STATIONS EXCEEDS 10 **

** REQUIRED FILE PLD.UNT CONTAINING
       UNIT MEMBER PLD(LOADS) DOES NOT EXIST
    ------------------------------------------------
      ( PLD(LOADS) is created by execution of module PLD
        and is required as input to module SPN.)

          *** ERROR- SIZE OF PLD(LOADS) TABLE
                     TOO LARGE RERUN MODULE
                     PLD WITH IN RANGE INPUT **

** REQUIRED FILE PRP.UNT CONTAINING
       UNIT MEMBER PRP(PERFORM) DOES NOT EXIST
    ------------------------------------------------
      ( PRP(PERFORM) is created by execution of module PRP
        and is required as input to module PTE.)

** REQUIRED FILE SFIELD.UNT CONTAINING
       UNIT MEMBER SFIELD(THETA) DOES NOT EXIST
      ------------------------------------------------
      ( SFIELD(THETA) is created by the noise preprocessor.)

** REQUIRED FILE SFIELD.UNT CONTAINING
       UNIT MEMBER SFIELD(PHI) DOES NOT EXIST
      ------------------------------------------------
      ( SFIELD(PHI) is created by the noise preprocessor.)

If the original Blade Geometry modules: RBS, RBA, and
BLM are to be used, then the following messages may be displayed:

** REQUIRED FILE RBS.UNT CONTAINING UNIT MEMBER
       RBS(SLPSPN) DOES NOT EXIST ON THIS DIRECTORY
      ------------------------------------------------
      ( RBS(SLPSPN) is created by execution of module RBS.)

** REQUIRED FILE RBS.UNT CONTAINING UNIT MEMBER
       RBS(SHAPE) DOES NOT EXIST ON THIS DIRECTORY
     ------------------------------------------------
      ( RBS(SHAPE) is created by execution of module RBS.)

** REQUIRED FILE RBS.UNT CONTAINING UNIT MEMBER
       RBS(SLPCRD) DOES NOT EXIST ON THIS DIRECTORY
      ------------------------------------------------
      ( RBS(SLPCRD) is created by execution of module RBS.)
```

If the Improved Blade Geometry modules: IBS, IBA, and IBL are to be used, then the following messages may be displayed:

```
** REQUIRED FILE IBS.UNT CONTAINING UNIT MEMBER
       IBS(SLPSPN) DOES NOT EXIST ON THIS DIRECTORY
      -----------------------------------------------
      ( IBS(SLPSPN) is created by execution of module IBS.)

** REQUIRED FILE IBS.UNT CONTAINING UNIT MEMBER
       IBS(SHAPE) DOES NOT EXIST ON THIS DIRECTORY
     -----------------------------------------------
      ( IBS(SHAPE) is created by execution of module IBS.)

** REQUIRED FILE IBS.UNT CONTAINING UNIT MEMBER
       IBS(SLPCRD) DOES NOT EXIST ON THIS DIRECTORY
      -----------------------------------------------
      ( IBS(SLPCRD) is created by execution of module IBS.)
```

The following exit message will be displayed:

```
    ************************************
    *                                  *
    *  NOISE INPUT COMPLETED           *
    *                                  *
    ************************************
 Execution suspended : PRESS "ENTER" TO CONTINUE
```

The user must press the ENTER key to continue with the next preprocessor prompt.

Flight Path Preprocessor (FLPPREP)

FLPPREP is a FLIGHT PATH MODULE PREPROCESSOR for interactively entering the input data required by the FLIGHT PATH functional modules ATM, ABS, SFO, and GEO.
User Parameter file FLP.PAR will be created and will be copied to files ATM.PAR, ABS.PAR, SFO.PAR and GEO.PAR.

The user must make C:\ANOPP\<propeller id> the current directory. (If it is not the current directory, the user types: CHANGE <propeller id> and presses ENTER.)

<propeller id> is the identification name given to a propeller (1-8 characters) and is also the name of the subdirectory used for predictions with that propeller (see MAKE command in Section IV.)

File FLPPREP.EXE and INPUT.BAT must be installed on directory C:\ANOPP\EXE.

To run FLPPREP the user types:

INPUT <cr>

The system will ask the user if each preprocessor is to be run. The following prompt is displayed:

DO YOU WISH TO RUN THE FLIGHT PATH PREPROCESSOR?
PLEASE PRESS Y OR N (DO NOT PRESS "ENTER"):

The user types Y, if the preprocessor is to be run.

PART 1

The system displays the following:

```
.....................................................
THERE ARE 3 OPTIONS FOR ATMOSPHERIC TABLES:

 1. STANDARD ATMOSPHERE (PRESSSURE:2116.22 LB/FT**2)
      EXISTING TABLES WILL BE USED

 2. STANDARD ATMOSPHERE + 10 DEGREES CENTIGRADE
      (PRESSSURE:2116.22 LB/FT**2) EXISTING
       TABLES WILL BE USED

 3. TABLES WILL BE CREATED FROM ATMOSPHERIC
       PROFILE INPUT BY USER
.....................................................
 DO YOU WISH TO USE STANDARD ATMOSPHERE (OPTION 1)?
 PLEASE PRESS Y OR N (DO NOT PRESS "ENTER"):
```

If Y is pressed, then the STANDARD ATMOSPHERE tables will be used.
If N is pressed, then the following is displayed:

```
.....................................................
 DO YOU WISH TO USE STANDARD ATMOSPHERE + 10 DEGREES
 CENTIGRADE (OPTION 2)?
 PLEASE PRESS Y OR N (DO NOT PRESS "ENTER"):
```

If Y is pressed, then the STANDARD ATMOSPHERE + 10 DEGREES CENTIGRADE tables will be used.
If N is pressed, then the following is displayed:

```
.............................................
 OPTION 3 (USER INPUT) WILL BE USED
```

The user will be asked to enter the atmospheric profile.

PART 2

The program displays the description and current values for the type of units and the print directive (entered or verified in the noise preprocessor) and asks the user if each value is to be changed.

If the value was not found or if the user requested to change the value, then the program displays the description and default setting, in parenthesis, for the type of units and/or the print directive. The user enters the value or presses "ENTER" only, if the default setting is to be used.

If system of units indicator exists, then the following prompt is displayed:

```
      SI UNITS WILL BE USED (If units entered as SI)
or
      ENGLISH UNITS WILL BE USED(If units entered as ENGLISH)
      DO YOU WISH TO CHANGE THE TYPE OF UNITS ?
      ENTER "Y" OR PRESS "ENTER" FOR NO =>
```

If systems of units do not exist or Y was entered, then the following prompt is displayed:

```
      ********************************************
      ENTER SYSTEM OF UNITS INDICATOR:
            "S" FOR SI UNITS
            "E" FOR ENGLISH UNITS
      OR PRESS "ENTER" ONLY FOR DEFAULT (SI)=>
      ********************************************
```

The user must press S or E and the ENTER key (or only the ENTER key if the default (SI) is to be used.)

```
*****************************************************
*****************************************************
```

If the print directive exists (in this example it equals 0), then the following prompt is displayed:

```
      IPRINT = 0 NO PRINT DESIRED
      DO YOU WISH TO CHANGE THE PRINT DIRECTIVE ?
      ENTER "Y" OR PRESS "ENTER" FOR NO =>
```

If print directive does not exist or Y was entered, then the following prompt is displayed:

```
**************************************************
   ENTER PRINT FLAG (INTEGER)
         =0, NO PRINT DESIRED
         =1, INPUT PRINT ONLY
         =2, OUTPUT PRINT ONLY
         =3, BOTH INPUT AND OUTPUT PRINT
   OR PRESS "ENTER" ONLY FOR DEFAULT (3)  =>
**************************************************
```

If the user chooses 1,2, or 3, then any requested input and any printed output for flight path functional modules will be located in files RESULTS.ATM, RESULTS.ABS, RESULTS.SFO and RESULTS.GEO, on the current directory. These files will be created when the functional modules are executed.

PART 3

If the STARDARD ATMOSPHERIC TABLES are not to be used, and if an ATMOSPHERIC PROFILE exists on file ATM.UNT, then the program displays those values. If SI units are to be used, then all displays include the following units:

ALTITUDE	TEMPERATURE	REL. HUMIDITY
(METERS)	(KELVIN)	(PERCENT)

If ENGLISH units are to be used, then all displays include the following units:

ALTITUDE	TEMPERATURE	REL. HUMIDITY
(FEET)	(RANKIN)	(PERCENT)

(SI units will be used in the following example.)

```
**********************************************************************
**********************************************************************
**  EXAMPLE:                                                        **
**                                                                  **
**  CURRENT PROFILE FOR ATMOSPHERIC TEMPERATURE                     **
**   AND RELATIVE HUMIDITY:                                         **
**  ************************************************              **
**   UNIT MEMBER  ATM(IN) - ATMOSPHERIC PROFILE                     **
**                                                                  **
**   ALTITUDE    TEMPERATURE  REL. HUMIDITY                         **
**   (METERS)      (KELVIN)     (PERCENT)                           **
**      0.00       280.0          70.                              **
**    500.00       270.0          70.                              **
**   1000.00       268.0          70.                              **
**                                                                  **
**********************************************************************
**********************************************************************
```

If the ATMOSPHERIC PROFILE does not exist, then the program displays the following default values:

```
  DEFAULT PROFILE FOR ATMOSPHERIC TEMPERATURE
  AND RELATIVE HUMIDITY:
 ************************************************
  UNIT MEMBER  ATM(IN) - ATMOSPHERIC PROFILE

   ALTITUDE    TEMPERATURE  REL. HUMIDITY
   (METERS)      (KELVIN)     (PERCENT)
      0.00       288.15         70.
    304.80       286.17         70.
```

After either the current or default values are printed to the screen, the following prompt is displayed:

```
**************************************************
DO YOU WISH TO CHANGE THESE values?
 ENTER "Y" OR PRESS "ENTER" FOR NO=>
```

If the user wishes to use the printed values (either the current values or default values), then only the ENTER key is pressed. If the ATMOSPHERIC TABLES for that profile exist on the current directory, then a flag is set and modules ATM and ABS will not be executed.
If the values are to be changed, then Y and the ENTER key are pressed and the following prompt appears:

```
************************************************
 ENTER NUMBER OF ALTITUDES FOR ATMOSPHERIC
 PROFILE (INTEGER)   (MAXIMUM OF 25)  =>
************************************
 The number of ALTITUDES to be entered should be typed
 by the user.  The program will then display the
 following prompt:

ENTER ATMOSPHERIC PROFILE (REAL) FOR EACH ALTITUDE
 (separate the values by comma or space and press
  "ENTER" after each set of values)

   ALTITUDE    TEMPERATURE  REL. HUMIDITY
   (METERS)      (KELVIN)     (PERCENT)
-----------------------------------------
=>
```

The user must enter each set of altitude, temperature, and relative humidity on a single line and press the ENTER key at the end of each line.

```
**************************************************************************
**************************************************************************
**  EXAMPLE:                                                            **
**                                                                      **
**  ************************************************                    **
**  ENTER NUMBER OF ALTITUDES FOR ATMOSPHERIC                           **
**   PROFILE (INTEGER)   (MAXIMUM OF 25)  =>         3                  **
**  ************************************                                **
**                                                                      **
**  ENTER ATMOSPHERIC PROFILE (REAL) FOR EACH ALTITUDE                  **
**  (separate the values by comma or space and press                    **
**   "ENTER" after each set of values)                                  **
**                                                                      **
**   ALTITUDE    TEMPERATURE  REL. HUMIDITY                             **
**   (METERS)      (KELVIN)     (PERCENT)                               **
**  ----------------------------------------                            **
**  =>    0.00       280.0          70.    <cr>                        **
**  =>  500.00       270.0          70.    <cr>                        **
**  => 1000.00       268.0          70.    <cr>                        **
**                                                                      **
**************************************************************************
**************************************************************************
```

After all values for the atmospheric profile have been entered, then the values are printed to the screen and the user is asked to verify the entry.

```
 ATMOSPHERIC PROFILE IS ENTERED AS:
 ************************************************
   ALTITUDE     TEMPERATURE  REL. HUMIDITY
   (METERS)       (KELVIN)     (PERCENT)
      0.00         280.0           70.
    500.0          270.0           70.
   1000.00         268.0           70.
 ************************************************
 DO YOU WISH TO CHANGE THESE values?
 ENTER "Y" OR PRESS "ENTER" FOR NO=>
```

If the user enters Y , then the program prompts the user to enter a new profile.

PART 4

If the STANDARD ATMOSPHERIC TABLES or the current ATMOSPHERIC TABLES are not used, then the program displays prompts for the single value input data required by module ATM. The description and default setting, in parenthesis, are displayed for each value. The user is prompted to enter the value or press the ENTER key only, if the default setting is to be used. (If the data is listed as (REAL), then the entry must include a decimal point. If the data is listed as (INTEGER), then the entry may NOT include a decimal point.) The program displays prompts for the following input data:

```
*****************************************************
If SI units entered:
            ENTER GROUND LEVEL ALTITUDE IN METERS
              OR PRESS "ENTER" ONLY FOR
              DEFAULT (0.0 METERS)=>

If ENGLISH units entered:
            ENTER GROUND LEVEL ALTITUDE IN FEET
              OR PRESS "ENTER" ONLY FOR
              DEFAULT (0.0 FEET)=>
```

```
**************************************************
If SI units entered:
            ENTER ATMOSPHERIC PRESSURE AT GROUND LEVEL
             IN N/M**2 OR PRESS "ENTER" ONLY FOR
              DEFAULT (101325.0 N/M**2)=>

If ENGLISH units entered:
            ENTER ATMOSPHERIC PRESSURE AT GROUND LEVEL
             IN LBF/FT**2 OR PRESS "ENTER" ONLY FOR
              DEFAULT (2116.2167 LBF/FT**2)=>

**************************************************
        ENTER NUMBER OF ALTITUDES FOR OUTPUT
         ATMOSPHERIC FUNCTION OR PRESS "ENTER"
         ONLY FOR DEFAULT (10) =>

**************************************************
If SI units entered:
            ENTER ALTITUDE INCREMENT FOR OUTPUT
              IN METERS OR PRESS "ENTER" ONLY FOR
              DEFAULT (100.0 METERS)=>

If ENGLISH units entered:
            ENTER ALTITUDE INCREMENT FOR OUTPUT
              IN FEET OR PRESS "ENTER" ONLY FOR
              DEFAULT (328.0839 FEET)=>
**************************************************

**********************************************************************
**********************************************************************
**                                                                  **
**  Example :                                                       **
**             ENTER ALTITUDE INCREMENT FOR OUTPUT                  **
**               IN METERS OR PRESS "ENTER" ONLY FOR                **
**               DEFAULT (100.0 METERS)=>       ( displayed       **
**                                             on terminal)         **
**   (If the user has chosen ENGLISH units, then the prompt         **
**        and default setting will be shown in FEET.)               **
**                                                                  **
**  To enter a value of 200.0, the user                             **
**  types 200.D0 and the presses the ENTER key:                     **
**                                                                  **
**                                  => 200.D0  <cr>                 **
**                                                                  **
**  To enter the default value of 100.0, the user                   **
**  presses only the ENTER key:                                     **
**                                                                  **
**                                  =>   <cr>                       **
**********************************************************************
**********************************************************************
```

PART 5

The program displays prompts for the single value input data required by modules SFO and GEO. The description and default setting, in parenthesis, are displayed for each value.
The user is prompted to enter the value or press the ENTER key only, if the default setting is to be used. (If the data is listed as (REAL), then the entry must include a decimal point. If the data is listed as (INTEGER), then the entry may NOT include a decimal point.) The program displays prompts for the following input data:

```
******************************************************
If SI units entered:
            ENTER AIRCRAFT ALTITUDE AT ORIGIN
              IN METERS OR PRESS "ENTER" ONLY FOR
              DEFAULT (211.5 METERS)=>

If ENGLISH units entered:
            ENTER AIRCRAFT ALTITUDE AT ORIGIN
              IN FEET OR PRESS "ENTER" ONLY FOR
              DEFAULT (693.93 FEET)=>

******************************************************
******************************************************
If SI units entered:
            ENTER FORWARD AIRCRAFT VELOCITY IN METERS
              PER SECOND OR PRESS "ENTER" ONLY FOR
              DEFAULT (42.18 M/SEC)=>

If ENGLISH units entered:
            ENTER FORWARD AIRCRAFT VELOCITY IN FEET
              PER SECOND OR PRESS "ENTER" ONLY FOR
              DEFAULT (138.22 FT/SEC)=>
******************************************************
******************************************************

       ENTER INCLINATION OF FLIGHT VECTOR IN
         DEGREES OR PRESS "ENTER" ONLY FOR
         DEFAULT (7.0 DEGREES)=>

******************************************************
       ENTER ANGLE OF ATTACK IN DEGREES
         OR PRESS "ENTER" ONLY FOR
         DEFAULT (4.3 DEGREES)=>
******************************************************
```

```
**********************************************************************
**********************************************************************
**                                                                  **
**  Example 2:                                                      **
**      ENTER AIRCRAFT ALTITUDE AT ORIGIN                           **
**       IN METERS OR PRESS "ENTER" ONLY FOR                        **
**       DEFAULT (211.5 METERS)=>              ( displayed          **
**                                             on terminal)         **
**    (If the user has chosen ENGLISH units, then the prompt        **
**         and default setting will be shown in FEET.)              **
**                                                                  **
**  To enter a value of 200.0, the user                             **
**  types 200.D0 and the presses the ENTER key:                     **
**                                                                  **
**                                    => 200.D0  <cr>               **
**                                                                  **
**  To enter the default value of 211.5,  the user                  **
**  presses only the ENTER key:                                     **
**                                                                  **
**                                    =>   <cr>                     **
**********************************************************************
**********************************************************************
```

PART 6

If OBSERVER COORDINATES exists on file OBSERV.UNT, then the the program displays those values.

```
**********************************************************************
**********************************************************************
**  EXAMPLE:                                                        **
**                                                                  **
**  CURRENT X,Y,Z COORDINATES FOR EACH OBSERVER:                    **
**  ********************************************                    **
**   UNIT MEMBER  OBSERV(COORD) - X,Y,Z COORDINATES                 **
**                                FOR EACH OBSERVER:                **
**                                                                  **
**   OBSERVER NUMBER      X          Y          Z                   **
**         1             0.0        0.0        0.0                 **
**         2             0.0        0.0        1.2                 **
**                                                                  **
**********************************************************************
**********************************************************************
```

If the set of OBSERVER COORDINATES does not exist, then the program displays the following default values:

```
   DEFAULT X,Y,Z COORDINATES FOR EACH OBSERVER:
  ***********************************************
   UNIT MEMBER  OBSERV(COORD) - X,Y,Z COORDINATES
                                FOR EACH OBSERVER:

   OBSERVER NUMBER       X          Y          Z
         1             0.0        0.0        0.0
         2             0.0        0.0        1.2
         3          -610.0        0.0        0.0
         4          -610.0        0.0        1.2
```

After either the current or default values are printed to the screen, the following prompt is displayed:

```
*****************************************************
DO YOU WISH TO CHANGE THESE values?
 ENTER "Y" OR PRESS "ENTER" FOR NO=>
```

If the user wishes to use the printed values (either the current values or default values), then only the enter key is pressed. If the values are to be changed, then Y and the ENTER key are pressed and the following prompt appears:

```
***********************************************************
 ENTER NUMBER OF OBSERVERS (INTEGER) (MAXIMUM OF 25) =>
```

The number of OBSERVERS to be entered should be typed by the user. The program will then display the following prompt:

```
ENTER X,Y,Z COORDINATES (REAL) FOR EACH OBSERVER
 (separate the values by comma or space and press
  "ENTER" after each set of X,Y,Z values)

OBSERVER NUMBER        X         Y         Z
OBSERVER   1 =>
```

The user must enter each set of X,Y,Z coordinates on a single line, and press the ENTER key at the end of each line.

```
**********************************************************************
**********************************************************************
**  EXAMPLE:                                                        **
**                                                                  **
**  ************************************************                **
**  ENTER NUMBER OF OBSERVERS (INTEGER) (MAXIMUM OF 25)=> 2          **
**  ****************************************                        **
**                                                                  **
**  ENTER X,Y,Z COORDINATES (REAL) FOR EACH OBSERVER                **
**  (separate the values by a comma or space and press              **
**   "ENTER" after each set of X,Y,Z values)                        **
**                                                                  **
**   OBSERVER NUMBER       X         Y         Z                    **
**   OBSERVER  1 =>        0.0       0.0       0.0     <cr>         **
**   OBSERVER  2 =>        0.0       0.0       1.2     <cr>         **
**                                                                  **
**********************************************************************
**********************************************************************
```

After all values for the observer coordinates have been entered, then the values are printed to the screen and the user is asked to verify the entry.

```
  X,Y,Z OBSERVER COORDINATES ARE ENTERED AS:
  **********************************************

   OBSERVER NUMBER       X          Y          Z
        1              0.0        0.0        0.0
        2              0.0        0.0        1.2
 ***********************************************
 DO YOU WISH TO CHANGE THESE values?
 ENTER "Y" OR PRESS "ENTER" FOR NO=>
```

If the user enters Y , then the program prompts the user to enter a new set of observer coordinates.

PART 7

If 1/3 OCTAVE BAND FREQUENCIES exists on file SFIELD.UNT, then the program displays the number of frequencies and the starting and ending frequency.

```
**************************************************************************
**************************************************************************
**  EXAMPLE:                                                            **
**                                                                      **
**  CURRENT 1/3 OCTAVE BAND FREQUENCIES:                                **
**  NUMBER OF FREQUENCIES = 24                                          **
**  STARTING FREQUENCY = 0.2500000D+02                                  **
**  ENDING FREQUENCY = 0.5000000D+04                                    **
**                                                                      **
**************************************************************************
**************************************************************************
```

If the set of 1/3 BAND OCTAVE FREQUENCIES does not exist, then the program displays the following default STANDARD FREQUENCIES:

```
  ************************************************
  DEFAULT STANDARD 1/3 OCTAVE BAND FREQUENCIES:
  NUMBER OF FREQUENCIES = 24
  STARTING FREQUENCY = 0.5000000D+02
  ENDING FREQUENCY = 0.1000000D+05
```

After either the current or default frequencies are printed to the screen, the following prompt is displayed:

```
************************************************
DO YOU WISH TO CHANGE THESE values?
 ENTER "Y" OR PRESS "ENTER" FOR NO=>
```

If the user wishes to use the printed values (either the current values or default values), then only the enter key is pressed. If the values are to be changed, then Y and the ENTER key are pressed and the following prompt appears:

```
 ENTER STARTING FREQUENCY (GREATER
  THAN OR EQUAL TO 10.0) =>

 ENTER ENDING FREQUENCY (LESS
  THAN OR EQUAL TO 100000.0) =>
```

The starting and ending frequencies should be typed by the user. The program will enter all of the 1/3 band octave frequencies from the following list that are within the starting and ending frequency range. (If the starting or ending frequency is not in this list, then the frequency, which is closest to but less than the value, will be used.)

1/3 OCTAVE BAND FREQUENCIES:

10.0	100.0	1000.0	10000.0	100000.0
12.5	125.0	1250.0	12500.0	
16.0	160.0	1600.0	16000.0	
20.0	200.0	2000.0	20000.0	
25.0	250.0	2500.0	25000.0	
31.5	315.0	3150.0	31500.0	
40.0	400.0	4000.0	40000.0	
50.0	500.0	5000.0	50000.0	
63.0	630.0	6300.0	63000.0	
80.0	800.0	8000.0	80000.0	

PART 8

The program then verifies that the unit files, required by functional modules ATM, ABS, SFO and GEO, exist. Any missing files cause one or more of the following messages to be displayed:

```
** REQUIRED FILE ATM.UNT CONTAINING
        UNIT MEMBER ATM(TMOD) DOES NOT EXIST
    -------------------------------------------------
    ( ATM(TMOD) is created by execution of module ATM.)

** REQUIRED FILE ATM.UNT CONTAINING
        UNIT MEMBER ATM(AAC) DOES NOT EXIST
    -------------------------------------------------
    ( ATM(AAC) is created by execution of module ABS.)
```

```
** REQUIRED FILE SFIELD.UNT CONTAINING
       UNIT MEMBER SFIELD(FREQ) DOES NOT EXIST
    ------------------------------------------------
    ( SFIELD(FREQ)  is created by the flight path preprocessor.)


** REQUIRED FILE OBSERV.UNT CONTAINING
       UNIT MEMBER OBSERV(COORD) DOES NOT EXIST
    ------------------------------------------------
    ( OBSERV(COORD) is created by the flight path preprocessor.)
```

The following exit message will be displayed:

```
    ************************************
    *                                  *
    *  FLIGHT PATH INPUT COMPLETED     *
    *                                  *
    ************************************
  Execution suspended:  PRESS "ENTER" TO CONTINUE
```

The user must press the ENTER key to continue with next preprocessor prompt.

Propagation Preprocessor (PLEPREP)

PLEPREP is a PROPAGATION PREPROCESSOR for interactively entering the input data required by the functional modules PRT, PRO, LEV and EFF.
User Parameter file PLE.PAR will be created and copied to files PRT.PAR, PRO.PAR, LEV.PAR, and EFF.PAR. Alternate names table files PRT.ANT and/or PRO.ANT will be created.

The user must make C:\ANOPP\<propeller id> the current directory. (If it is not the current directory, the user types: CHANGE <propeller id> and presses ENTER.)

<propeller id> is the identification name given to a propeller (1-8 characters) and is also the name of the subdirectory used for predictions with that propeller (see MAKE command in Section IV.)

File PLEPREP.EXE and INPUT.BAT must be installed on directory C:\ANOPP\EXE.
To run PLEPREP the user types:

INPUT <cr>

The system will ask the user if each preprocessor is to be run. The following prompt is displayed:

DO YOU WISH TO RUN THE PROPAGATION PREPROCESSOR?
PLEASE PRESS Y OR N (DO NOT PRESS "ENTER"):

The user types Y to run the preprocessor and the following prompt is displayed:

```
**********************************************************
  CHOOSE FROM THE FOLLOWING TYPE CODES:
  TYPE CODE   NOISE LEVEL COMPUTATION OPTION
  ---------   ------------------------------
      1       NARROW BAND NOISE DATA TO BE SUMMED
      2       1/3 OCTAVE BAND NOISE DATA TO BE SUMMED
      3       BOTH NARROW BAND NOISE DATA TO BE SUMMED
              AND 1/3 OCTAVE BAND NOISE DATA TO BE SUMMED
 ENTER (INTEGER) TYPE CODE=>
```

The type code determines if functional modules PRT and PRO will be executed.
If code 1 is entered and the output from SPN exists, then functional module PRT will be run.
If code 2 is entered and the output from PTE exists, then functional module PRO will be run.
If code 3 is entered and the output from SPN and PTE exists, then both functional modules PRT and PRO will be run.

PART 1

The program displays prompts for the single value input data required by the propagation modules. If the values have been entered for the noise modules, then the program displays the description and given value and asks if the value is to be changed. If the value is to be changed or if the value was was not found, then the description and default setting, in parenthesis, are displayed. The user is prompted to enter the value or press the ENTER key only, if the default setting is to be used. (If the data is listed as (REAL), then the entry must include a decimal point. If the data is listed as (INTEGER), then the entry may NOT include a decimal point.) The program displays prompts for the following input data:

```
      ********************************************
If SI units entered:
         BLADE LENGTH IN METERS (REAL)
         DEFAULT (1.016 METERS)=>
      ********************************************
If ENGLISH units entered:
         BLADE LENGTH IN INCHES (REAL)
         DEFAULT (40. INCHES)=>
      ********************************************
      ********************************************
         NUMBER OF PROPELLER BLADES (INTEGER)
         DEFAULT (2)
      ********************************************
      ********************************************
If SI units entered:
       AMBIENT SPEED OF SOUND IN METERS/SEC
        (REAL) DEFAULT (342.2 M/S)
      ********************************************
If ENGLISH units entered:
       AMBIENT SPEED OF SOUND IN FEET/SEC
        (REAL) DEFAULT (1122.7034 FT/S)
      ********************************************
      ********************************************
         PRINT FLAG (INTEGER)
         =0, NO PRINT DESIRED
         =1, INPUT PRINT ONLY
         =2, OUTPUT PRINT ONLY
         =3, BOTH INPUT AND OUTPUT PRINT
       DEFAULT (3)
       ********************************************
        NUMBER OF PROPELLERS (MAXIMUM OF 4)
        (INTEGER) DEFAULT (1)
       ********************************************
```

```
**********************************************************************
**********************************************************************
**                                                                  **
**  Example 1: ( 4 propeller blades were entered in                 **
**               the noise preprocessor and will                    **
**               be changed to 2 )                                  **
**                                                                  **
**       NUMBER OF PROPELLER BLADES = 4                             **
**       DO YOU WISH TO CHANGE NUMBER OF BLADES?                    **
**       ENTER "Y" OR PRESS "ENTER" FOR NO=>          Y <cr>        **
**                                                                  **
**       ENTER NUMBER OF PROPELLER BLADES (INTEGER)                 **
**       OR PRESS "ENTER" ONLY FOR                                  **
**       DEFAULT  ( 2) =>                        <cr>               **
**                                                                  **
**         2 blades will be entered.                                **
**                                                                  **
**********************************************************************
**********************************************************************

**********************************************************************
**********************************************************************
**                                                                  **
**  Example 2:                                                      **
**       ENTER AMBIENT SPEED OF SOUND IN METERS/SEC                 **
**       (REAL) OR PRESS "ENTER" ONLY FOR                           **
**       DEFAULT  ( 342.2 M/S) =>                ( displayed        **
**                                              on terminal)        **
**   (If the user has chosen ENGLISH units, then the prompt         **
**        and default setting will be shown in FT/SEC.)             **
**                                                                  **
**  To enter a value of 325.0, the user                             **
**  types 325.D0 and the presses the ENTER key:                     **
**                                                                  **
**                                  => 325.D0  <cr>                 **
**                                                                  **
**  To enter the default value of 342.2, the user                   **
**  presses only the ENTER key:                                     **
**                                                                  **
**                                  =>   <cr>                       **
**********************************************************************
**********************************************************************
```

PART 2

The program displays the following prompt:

```
        DO YOU WISH TO INCLUDE GROUND EFFECTS ?
         ENTER "Y" OR PRESS "ENTER" FOR NO =>
         ******************************************
```

If ground effects are to be included, then the user presses Y and the ENTER key. If ground effects are not to be used, then only the ENTER key is pressed.
If ground effects are to be included then the following prompt is displayed:

```
ENTER TYPE OF SURFACE TO BE USED IN
CALCULATING GROUND EFFECTS:

ENTER "S" FOR SOFT, OR "H" FOR HARD
OR PRESS "ENTER" ONLY FOR DEFAULT (S)=>
*******************************************
```

The user presses H and the ENTER key, if hard surface is to be used.
The user presses S and the ENTER key, or only the ENTER key, if soft surface is to be used.
The program then displays the following prompt:

```
DO YOU WISH TO INCLUDE EFFECTS OF
 ATMOSPHERIC ABSORPTION?
 ENTER "Y" OR PRESS "ENTER" FOR NO =>
```

If atmospheric absorption effects are to be included, then the user presses Y and the ENTER key. If atmospheric absorption effects are not to be used, then only the ENTER key is pressed.

```
**********************************************************************
**********************************************************************
**                                                                  **
**  Example :  (Ground effects with hard surface, and               **
**               atmospheric absorption are to be included)         **
**                                                                  **
**      ********************************************              **
**      DO YOU WISH TO INCLUDE GROUND EFFECTS ?                     **
**       ENTER "Y" OR PRESS "ENTER" FOR NO =>       Y  <cr>        **
**      ********************************************              **
**       ENTER TYPE OF SURFACE TO BE USED IN                        **
**       CALCULATING GROUND EFFECTS:                                **
**                                                                  **
**       ENTER "S" FOR SOFT, OR "H" FOR HARD                        **
**        OR PRESS "ENTER" ONLY FOR DEFAULT (S)=>   H  <cr>         **
**      ********************************************              **
**      DO YOU WISH TO INCLUDE EFFECTS OF                           **
**       ATMOSPHERIC ABSORPTION?                                    **
**       ENTER "Y" OR PRESS "ENTER" FOR NO =>       Y  <cr>        **
**                                                                  **
**                                                                  **
**********************************************************************
**********************************************************************
```

PART 3

The program then verifies that the unit files required by functional modules PRT, LEV, and EFF exist. Any missing files cause one or more of the following messages to be displayed:

```
** REQUIRED FILE ATM.UNT CONTAINING
        UNIT MEMBER ATM(TMOD) DOES NOT EXIST
    ------------------------------------------------
    ( ATM(TMOD) is created by execution of module ATM.)

** REQUIRED FILE ATM.UNT CONTAINING
        UNIT MEMBER ATM(AAC) DOES NOT EXIST
    ------------------------------------------------
    ( ATM(AAC) is created by execution of module ABS.)

** REQUIRED FILE SFIELD.UNT CONTAINING
        UNIT MEMBER SFIELD(FREQ) DOES NOT EXIST
    ------------------------------------------------
    ( SFIELD(FREQ)  is created by the flight path preprocessor.)

** REQUIRED FILE OBSERV.UNT CONTAINING
        UNIT MEMBER OBSERV(COORD) DOES NOT EXIST
    ------------------------------------------------
    ( OBSERV(COORD) is created by the flight path preprocessor.)

** REQUIRED FILE GEO.UNT CONTAINING
        UNIT MEMBER GEO(BODY) DOES NOT EXIST
    ------------------------------------------------
    ( GEO(BODY) is created by execution of module GEO.)

** REQUIRED FILE SPN.UNT CONTAINING
        UNIT MEMBER SPN(XXX001) DOES NOT EXIST
    ------------------------------------------------
    ( SPN(XXX001) is created by execution of module SPN.)

** REQUIRED FILE PTE.UNT CONTAINING
        UNIT MEMBER PTE(XXX001) DOES NOT EXIST
    ------------------------------------------------
    ( PTE(XXX001) is created by execution of module PTE.)
```

The following exit message will be displayed:

```
    ************************************
    *                                  *
    *  PROPAGATION INPUT COMPLETED     *
    *                                  *
    ************************************
  Execution suspended : PRESS "ENTER" TO CONTINUE
```

The user must press the ENTER key to continue with next preprocessor prompt.

User Parameter File Editor (UPFEDT)

UPFEDT IS A USER PARAMETER FILE PREPROCESSOR FOR INTERACTIVELY ENTERING USER PARAMETER NAMES AND VALUES REQUIRED BY ANOPP FUNCTIONAL MODULES

The current directory must contain all user parameter files which are to be edited. All output will be created on this directory.

File UPFEDT.EXE must be installed on directory C:\ANOPP\EXE.
To run UPFEDT the user types:

PARAM <cr>

The following menu will appear on the screen:

```
**************************************************************
*       MENU - USER PARAMETER FILE ACCESS                    *
*  CHOOSE ONE OF THE FOLLOWING OPTIONS :                     *
*                                                            *
*  1 - CREATE A NEW USER PARAMETER FILE                      *
*  2 - EDIT AN EXISTING USER PARAMETER FILE                  *
*  3 - DIRECTORY LISTING OF USER PARAMETER FILES             *
*  4 - PRINT USER PARAMETER LISTING TO SCREEN                *
*  5 - PRINT USER PARAMETER LISTING TO FILE                  *
*  6 - EXIT                                                  *
*                                                            *
*  ENTER 1, 2, 3, 4, 5, 0R 6 :                               *
   =>
**************************************************************
```

(NOTE: the ENTER key must be pressed after each user response.)

OPTION 1

If the user wishes to create a new file, 1 is entered and the following prompt appears on the screen:

```
**************************************************************
       ENTER USER PARAMETER FILE NAME
       1-8 CHARACTERS - NO EXTENSION
      =>
```

The user enters the name of the USER PARAMETER (PARAM) File.
Extension .PAR will be added by the preprocessor.
(EXAMPLE: If user enters:

=> TABLE1

then external file TABLE1.PAR will be created. If the PARAM File already exists on the current directory, the user is prompted:

```
***** FILE EXISTS DO YOU WISH TO OVERWRITE FILE? *****
      ENTER "YES" OR "NO"   :
       =>
*************************************************************
```

If the user types NO, then the program redisplays the main menu.
If the user types YES, then the old version of the file is deleted.

After a PARAM File is created the preprocessor will continue to prompt for user parameter names and values until NULL is entered when prompted for user parameter name.

```
*****************************************************
   ENTER "NULL" TO STOP OR-
   USER PARAMETER NAME-1 TO 8 CHARACTERS:
   =>
```

The user must enter the user parameter name or NULL (to complete the PARAM file and return to the main menu).
The preprocessor then prompts the user:

```
*****************************************************
   CHOOSE FROM THE FOLLOWING TYPE CODES:'
   TYPE CODE          PARAMETER VALUE TYPE'
     I                INTEGER'
     R                REAL SINGLE PRECISION'
     C                COMPLEX SINGLE PRECISION'
     L                LOGICAL'
     A                STRING OF  CHARACTERS'
 *****************************************************
 ENTER TYPE CODE :
```

The user must enter the alphabetic code for the parameter type and is prompted to enter the number of elements in the user parameter array.

```
*****************************************************
 ENTER NUMBER OF ELEMENTS IN PARAM:
```

The user enters the number of elements in the user parameter array. If the user parameter is a single value, then it is an array of one element and 1 is entered. If the user parameter is a complex array (of one or more elements), each real and imaginary number pair is counted as one element. The user is then prompted to enter each element of the user parameter.

EXAMPLE: If user wishes to enter five user parameters: ALPHA with a value of 808, BETA with a value of 1.1D2 ,GAMMA ,a complex array with two elements, (-1.23, 1.0) and (3.33D4, 2.4), FLAG with a value of .TRUE., and CARD with a value of 'THIS IS A TEST', the following prompts and responses would be displayed and entered:

```
*****************************************************
   ENTER "NULL" TO STOP OR-
   USER PARAMETER NAME-1 TO 8 CHARACTERS:
   => ALPHA

*****************************************************
   CHOOSE FROM THE FOLLOWING TYPE CODES:'
   TYPE CODE          PARAMETER VALUE TYPE'
     I                INTEGER'
     R                REAL SINGLE PRECISION'
     C                COMPLEX SINGLE PRECISION'
     L                LOGICAL'
     A                STRING OF  CHARACTERS'
 *****************************************************
 ENTER TYPE CODE : I

*****************************************************
 ENTER NUMBER OF ELEMENTS IN PARAM:1

*****************************************************
ENTER ALPHA      : 808
*****************************************************

  ENTER "NULL" TO STOP OR-
   USER PARAMETER NAME-1 TO 8 CHARACTERS:
   => BETA

*****************************************************
   CHOOSE FROM THE FOLLOWING TYPE CODES:'
   TYPE CODE          PARAMETER VALUE TYPE'
     I                INTEGER'
     R                REAL SINGLE PRECISION'
     C                COMPLEX SINGLE PRECISION'
     L                LOGICAL'
     A                STRING OF  CHARACTERS'
 *****************************************************
 ENTER TYPE CODE :R

*****************************************************
 ENTER NUMBER OF ELEMENTS IN PARAM:1
*****************************************************
 ENTER BETA     : 1.1D2
```


C-2

```
**************************************************
   ENTER "NULL" TO STOP OR-
   USER PARAMETER NAME-1 TO 8 CHARACTERS:
   => GAMMA

**************************************************
   CHOOSE FROM THE FOLLOWING TYPE CODES:'
   TYPE CODE          PARAMETER VALUE TYPE'
     I                INTEGER'
     R                REAL SINGLE PRECISION'
     C                COMPLEX SINGLE PRECISION'
     L                LOGICAL'
     A                STRING OF  CHARACTERS'
 **************************************************
 ENTER TYPE CODE :C

**************************************************
 ENTER NUMBER OF ELEMENTS IN PARAM:2

**************************************************
 ENTER GAMMA(1)     :
  ENTER REAL PART OF COMPLEX NUMBER:-1.23

  ENTER IMAGINARY PART:1.0

 ENTER GAMMA(2)     :
  ENTER REAL PART OF COMPLEX NUMBER:3.33D4

  ENTER IMAGINARY PART:2.4

**************************************************
   ENTER "NULL" TO STOP OR-
   USER PARAMETER NAME-1 TO 8 CHARACTERS:
   => FLAG

**************************************************
   CHOOSE FROM THE FOLLOWING TYPE CODES:'
   TYPE CODE          PARAMETER VALUE TYPE'
     I                INTEGER'
     R                REAL SINGLE PRECISION'
     C                COMPLEX SINGLE PRECISION'
     L                LOGICAL'
     A                STRING OF  CHARACTERS'
 **************************************************
 ENTER TYPE CODE :L

**************************************************
 ENTER NUMBER OF ELEMENTS IN PARAM:1

**************************************************
 ENTER FLAG     : .TRUE.
```

```
*****************************************************
   ENTER "NULL" TO STOP OR-
   USER PARAMETER NAME-1 TO 8 CHARACTERS:
   => CARD
*****************************************************
   CHOOSE FROM THE FOLLOWING TYPE CODES:'
   TYPE CODE          PARAMETER VALUE TYPE'
     I                INTEGER'
     R                REAL SINGLE PRECISION'
     C                COMPLEX SINGLE PRECISION'
     L                LOGICAL'
     A                STRING OF  CHARACTERS'
 *****************************************************
 ENTER TYPE CODE :A

*****************************************************
 ENTER NUMBER OF CHARACTERS :14

*****************************************************
 ENTER NUMBER OF ELEMENTS IN PARAM:1

*****************************************************
 ENTER CARD     :THIS IS A TEST

*****************************************************
   ENTER "NULL" TO STOP OR-
   USER PARAMETER NAME-1 TO 8 CHARACTERS:
   => NULL
*****************************************************
```

The user parameter table is completed and the main menu is redisplayed.

OPTION 2

If the user wishes to add or change a user parameter in an existing PARAM file, 2 is entered as a menu option and the following is displayed:

```
ENTER USER PARAMETER FILE NAME
1-8 CHARACTERS - NO EXTENSION
=>
```

If the file does not exist, this message is displayed:

```
*****FILE DOES NOT EXIST*****
```

and the program redisplays the main menu. If the file does exist, then the preprocessor will continue to prompt the user and read user parameter names, types and values until NULL is typed to complete the PARAM file and return to the main menu.
(See example in OPTION 1)

OPTION 3

If the user wishes to display a list of all existing PARAM files on the current directory, 3 is entered as a menu option and the following is displayed:

```
**********************************************************
*  ENTER DIRECTORY DOS CALL-                             *
*                                                        *
*  DIR  *.PAR <cr>                                       *
Execution suspended : -enter <cr> to return to menu
**********************************************************
```

If the user wishes to display the names of all PARAM Files on the current directory, the following is entered:

```
DIR *.PAR
```

A list of all PARAM Files on the user's current directory will be displayed. The user must press the ENTER key to return to the main menu.

OPTION 4

If the user wishes to list all USER PARAMETERS in an existing PARAM File on the monitor screen, 4 is entered as a menu option and the following is displayed:

```
ENTER USER PARAMETER FILE NAME
1-8 CHARACTERS - NO EXTENSION
=>
```

If the file does not exist, this message is displayed:

```
*****FILE DOES NOT EXIST*****
```

and the program redisplays the main menu. If the file does exist, then the preprocessor will list the PARAM on the monitor screen.

EXAMPLE: If the user wished a listing of PARAM File TABLE1.PAR created in the example shown under OPTION 1, the following would be entered and displayed:

```
ENTER USER PARAMETER FILE NAME
1-8 CHARACTERS - NO EXTENSION
=> TABLE1
```

The following would be displayed on the screen:

```
        USER PARAMETER FILE - TABLE1  .PAR

                USER  PARAMETER  TABLE  DUMP
 ------------------------------------------------------------
                    TYPE
  NAME      TYPE    CODE    ELEMENT      VALUE
 ------------------------------------------------------------
 ALPHA        I       1     ( 1)                  808
 BETA        RS       2     ( 1)  0.11000000000000E+03
 CARD         A      -14    ( 1)  THIS IS A TEST
 FLAG         L       6     ( 1)  T
 GAMMA        C       4     ( 1)  -0.1230000000000E+01  0.1000000000000E+01
                            ( 2)   0.3330000000000E+05  0.2400000000000E+01
**************************************************************************
ENTER <cr> TO RETURN TO MENU:
```

The user must press the ENTER key to return to the main menu.

OPTION 5

If the user wishes to list all USER PARAMETERS in an existing PARAM File to a listings file, 5 is entered as a menu option and the following is displayed:

```
ENTER USER PARAMETER FILE NAME
1-8 CHARACTERS - NO EXTENSION
=>
```

If the file does not exist, this message is displayed:

```
****FILE DOES NOT EXIST*****
```

and the program redisplays the main menu. If the file does exist, then the preprocessor will display the following prompt:

```
** ENTER FILE NAME FOR USER PARAMETER LISTING **
=>
```

The user must type the name of the file that is to be created and contain the PARAM listing. If an extension is desired, it must be entered. If a file with the same name exists, it will be deleted when the new file is created.

EXAMPLE: If the user wishes to create a listings file, TABLE1.LST which contains the USER PARAMETERS in PARAM File TABLE1.PAR, the following would be entered and displayed:

```
      ************************************************************
      ENTER USER PARAMETER FILE NAME
      1-8 CHARACTERS - NO EXTENSION
      =>  TABLE1

      ************************************************************
      ** ENTER FILE NAME FOR USER PARAMETER LISTING **
      =>  TABLE1.LST

 *** LISTING FILE COMPLETED******
```

The preprocessor then redisplays the main menu. To examine the listings file the user must exit the UPFEDT program (OPTION 6) and use a DOS file editor to access the file. File TABLE1.LST would contain the following:

```
         USER PARAMETER FILE - TABLE1  .PAR

                 USER  PARAMETER  TABLE  DUMP
 ------------------------------------------------------------
                      TYPE
   NAME       TYPE     CODE    ELEMENT      VALUE
 ------------------------------------------------------------
 ALPHA          I         1     (  1)                  808
 BETA          RS         2     (  1)   0.11000000000000E+03
 CARD           A       -14     (  1)  THIS IS A TEST
 FLAG           L         6     (  1)  T
 GAMMA          C         4     (  1)  -0.1230000000000E+01  0.1000000000000E+01
                                (  2)   0.3330000000000E+05  0.2400000000000E+01
************************************************************************
```

OPTION 6

If the user wishes to exit the UPFEDT program, 6 is entered and the following is displayed:

```
**************************************************************
*                                                            *
*          USER PARAMETER FILE EDITING COMPLETED             *
*                                                            *
**************************************************************
```

The program is completed and a prompt should be displayed by DOS.

Alternate Names File Editor (ANTEDT)

ANTEDT IS AN ALTERNATE NAME TABLE PREPROCESSOR FOR INTERACTIVELY ENTERING ALTERNATE NAMES FOR NAMED DATA ITEMS REQUIRED BY ANOPP FUNCTIONAL MODULES

The current directory must contain all alternate names files which are to be edited. All output will be created on this directory.

File ANTEDT.EXE must be installed on directory C:\ANOPP\EXE.
To run ANTEDT the user types:

ALT <cr>

The following menu will appear on the screen:

```
**************************************************************
*       MENU - ALTERNATE NAMES FILE ACCESS                   *
*  CHOOSE ONE OF THE FOLLOWING OPTIONS :                     *
*                                                            *
*  1 - CREATE A NEW ALTERNATE NAMES FILE                     *
*  2 - EDIT AN EXISTING ALTERNATE NAMES FILE                 *
*  3 - DIRECTORY LISTING OF ALTERNATE NAMES FILES            *
*  4 - PRINT ALTERNATE NAMES LISTING TO SCREEN               *
*  5 - PRINT ALTERNATE NAMES LISTING TO FILE                 *
*  6 - EXIT                                                  *
*                                                            *
*  ENTER 1, 2, 3, 4, 5, 0R 6 :                               *
   =>
**************************************************************
```

(NOTE: the ENTER key must be pressed after each user response.)

OPTION 1

If the user types to create a new file, 1 is entered and the following prompt appears on the screen:

```
**************************************************************
       ENTER ALTERNATE NAMES FILE NAME
       1-8 CHARACTERS - NO EXTENSION
      =>
```

The user enters the name of the Alternate Names Table (ANT) File. Extension .ANT will be added by the preprocessor.

EXAMPLE: If user enters:

```
=>   TABLE1
```

then external file TABLE1.ANT will be created. If the ANT File already exists on the current directory, the user is prompted:

```
**** FILE EXISTS DO YOU WISH TO OVERWRITE FILE?  *****
     ENTER  "YES" OR "NO"   :
      =>
************************************************************
```

If the user types NO, then the program redisplays the main menu. If the user types YES, then the old version of the file is deleted.

After an ANT File is created the preprocessor will continue to prompt the user and read REFERENCE NAMES (names of named data items referenced by ANOPP functional modules) and ALTERNATE NAMES (names of user created data items to be used in place of REFERENCED NAMES) until NULL is typed to complete the ANT file and return to the main menu.

```
ENTER REFERENCE NAME OR "NULL" TO END FILE AND
RETURN TO MENU =>

ENTER ALTERNATE NAME=>
```

EXAMPLE: If user wishes to enter two alternate names- BETA for reference name ALPHA and ZZ for reference name X, the following prompts and responses would be displayed and entered:

```
ENTER REFERENCE NAME OR "NULL" TO END FILE AND
RETURN TO MENU => ALPHA

ENTER ALTERNATE NAME=> BETA

ENTER REFERENCE NAME OR "NULL" TO END FILE AND
RETURN TO MENU => X

ENTER ALTERNATE NAME=> ZZ

ENTER REFERENCE NAME OR "NULL" TO END FILE AND
RETURN TO MENU => NULL
```

The ANT File with containing two Alternate Names would be completed and the preprocessor would redisplay the main menu.

OPTION 2

If the user wishes to add or change an alternate name in an existing ANT file, 2 is entered as a menu option and the following is displayed:

```
ENTER ALTERNATE NAMES FILE NAME
1-8 CHARACTERS - NO EXTENSION
=>
```

If the file does not exist, this message is displayed:

```
****FILE DOES NOT EXIST*****
```

and the program redisplays the main menu. If the file does exist, then the preprocessor will continue to prompt the user (and read REFERENCE NAMES (names of named data items referenced by ANOPP functional modules) and ALTERNATE NAMES (names of user created data items to be used in place of REFERENCED NAMES) until NULL is typed to complete the ANT file and return to the main menu.

```
ENTER REFERENCE NAME OR "NULL" TO END FILE AND
RETURN TO MENU =>

ENTER ALTERNATE NAME=>
```

(See example in OPTION 1)

OPTION 3

If the user wishes to display a list of all existing ANT files on the current directory, 3 is entered as a menu option and the following is displayed:

```
************************************************************
*  ENTER DIRECTORY DOS CALL-                               *
*                                                          *
*  DIR  *.ANT <cr>                                         *
Execution suspended : -enter <cr> to return to menu
************************************************************
```

If the user wishes to display the names of all ANT Files on the current directory, the following is entered:

```
DIR *.ANT
```

A list of all ANT Files on the user's current directory will be displayed. The user must press the ENTER key to return to the main menu.

OPTION 4

If the user wishes to list all alternate names in an existing ANT File on the monitor screen, 4 is entered as a menu option and the following is displayed:

```
ENTER ALTERNATE NAMES FILE NAME
1-8 CHARACTERS - NO EXTENSION
=>
```

If the file does not, exist this message is displayed:

```
****FILE DOES NOT EXIST*****
```

and the program redisplays the main menu. If the file does exist, then the preprocessor will list the ANT on the monitor screen.

EXAMPLE: If the user wished a listing of ANT File TABLE1.ANT created in the example shown under OPTION 1, the following would be entered and displayed:

```
ENTER ALTERNATE NAMES FILE NAME
1-8 CHARACTERS - NO EXTENSION
=> TABLE1


     ALTERNATE NAMES FILE - TABLE1    .ANT
                       ALTERNATE NAMES TABLE DUMP
-------------------------------------------------------------------

       FILE : TABLE1     .ANT

                 BETA       -IS ALTERNATE NAME OF -ALPHA
                 ZZ         -IS ALTERNATE NAME OF -X
*******************************************************************
ENTER <cr> TO RETURN TO MENU:
```

The user must press the ENTER key to return to the main menu.

OPTION 5

If the user wishes to list all alternate names in an existing ANT File to a listings file, 5 is entered as a menu option and the following is displayed:

```
ENTER ALTERNATE NAMES FILE NAME
1-8 CHARACTERS - NO EXTENSION
=>
```

If the file does not exist, this message is displayed:

```
****FILE DOES NOT EXIST*****
```

and the program redisplays the main menu. If the file does exist, then the preprocessor will display the following prompt:

```
** ENTER FILE NAME FOR ALTERNATE NAMES LISTING **
   =>
```

The user must type the name of the file that is to be created and contain the ANT listing. If an extension is desired, it must be entered. If a file with the same name exists, it will be deleted when the new file is created.

EXAMPLE: If the user wishes to create a listings file, TABLE1.LST which contains the alternate names in ANT File TABLE1.ANT, the following would be entered and displayed:

```
*************************************************************
 ENTER ALTERNATE NAMES FILE NAME
 1-8 CHARACTERS - NO EXTENSION
 =>  TABLE1

*************************************************************
** ENTER FILE NAME FOR ALTERNATE NAMES LISTING **
=>  TABLE1.LST

*** LISTING FILE COMPLETED******
```

The preprocessor then redisplays the main menu. To examine the listings file, the user must exit the ANTEDT program (OPTION 6) and use a DOS file editor to access the file. File TABLE1.LST would contain the following data:

```
     ALTERNATE NAMES FILE - TABLE1    .ANT
                     ALTERNATE NAMES TABLE DUMP
-------------------------------------------------------------------
       FILE : TABLE1     .ANT

                BETA       -IS ALTERNATE NAME OF -ALPHA
                ZZ         -IS ALTERNATE NAME OF -X
```

OPTION 6

If the user wishes to exit the ANTEDT program, 6 is entered and the following is displayed:

```
**************************************************************
*                                                            *
*          ALTERNATE NAMES FILE EDITING COMPLETED            *
*                                                            *
**************************************************************
```

The program is completed and a prompt should be displayed by DOS.

Member Manager Editor (MMEDT)

MMEDT IS A MEMBER MANAGER PREPROCESSOR FOR INTERACTIVELY ENTERING, EDITING OR LISTING UNIT MEMBERS REQUIRED BY ANOPP FUNCTIONAL MODULES

The current directory must contain all unit member files which are to be edited. All output will be created on this directory.

File MMEDT.EXE must be installed on directory C:\ANOPP\EXE.
To run MMEDT the user types:

UNIT <cr>

The following menu will appear on the screen:

```
******************************************************
   CHOOSE FROM THE FOLLOWING OPTION CODES:
    OPTION CODE       OPTION
         1        ENTER MEMBER
         2        PRINT MEMBER TO SCREEN
         3        PRINT MEMBER TO FILE
         4        PRINT CATALOG OF UNIT MEMBERS TO SCREEN
         5        PRINT CATALOG OF UNIT MEMBERS TO FILE
         6        DIRECTORY LISTING OF UNITS
         7        EDIT MEMBER RECORDS
         0        EXIT PROGRAM
******************************************************
ENTER OPTION CODE=>
```

(NOTE: the ENTER key must be pressed after each user response.)

OPTION 1

If the user types to create a new MEMBER, 1 is entered and the following prompt appears on the screen:

(Note: If a MEMBER with the same UNIT and MEMBER name exists, it will be deleted when the new MEMBER is created.)

```
************************************************
ENTER UNIT NAME: 1-8 CHARACTERS=>
```

The user enters the name of the UNIT which will contain the MEMBER. Extension .UNT will be added by the preprocessor.
(EXAMPLE: If user enters:

=> UNIT1

then external file UNIT1.UNT will be created.

The user is then prompted to enter the MEMBER name:

```
**************************************************
ENTER MEMBER NAME: 1-8 CHARACTERS =>
```

The user enters the MEMBER name and is prompted to enter the format specifications of the MEMBER:

```
**********************************************
CHOOSE FROM THE FOLLOWING FORMAT TYPES:
   FORMAT CODE       FORMAT TYPE
      0               UNFORMATTED
      1               FIXED FORMAT
      2               VARIABLE FORMAT
      3               CARD IMAGE
**********************************************
ENTER FORMAT TYPE CODE=>
**********************************************
```

The user enters the format type code and the following prompt is displayed:

```
**************************************************
 DO YOU WISH TO SEE AN EXPLANATION AND EXAMPLE OF
   THE CHOSEN TYPE OF FORMAT ENTRY?
    ENTER "Y" OR "N" =>
**************************************************
```

If the chosen format type is fixed (each record in the MEMBER has the same number, types and sequence of elements) and if Y is entered by the user, the following is displayed:

```
**************************************************
 FORMAT IS FIXED TYPE
 You will be prompted to enter the type code of
 each set of elements and the number of
 elements in that set.
  TYPE CODE         ELEMENT VALUE TYPE
      I               INTEGER
      R               REAL SINGLE PRECISION
      C               COMPLEX SINGLE PRECISION
      L               LOGICAL
      A               STRING OF n CHARACTERS
 **********************************************
   FORMAT SPECIFICATION EXAMPLE:
   " 3I, 4RS, 1L"   would be entered as follows:
   ENTER TYPE CODE OR "E" TO END FIXED PART => I
   ENTER NUMBER OF ELEMENTS IN SET          => 3
```

```
   ENTER TYPE CODE OR "E" TO END FIXED PART => R
   ENTER NUMBER OF ELEMENTS IN SET          => 4
   ENTER TYPE CODE OR "E" TO END FIXED PART => L
   ENTER NUMBER OF ELEMENTS IN SET          => 1
   ENTER TYPE CODE OR "E" TO END FIXED PART => E
 ************************************************
    enter return to continue:
```

If the chosen format type is variable (each record in the MEMBER may have an undetermined number of repetitions for the last specified sequence of elements) and if Y is entered by the user, the following is displayed:

```
*****************************************************
 FORMAT IS VARIABLE TYPE
 You will be prompted to first enter the fixed
 part of the format and then prompted for the
 variable part of the the format.  In each part
 you will be prompted to enter the type code of
 each set of elements and the number of elements
 in that set.
        TYPE CODE          ELEMENT VALUE TYPE
             I                INTEGER
             R                REAL SINGLE PRECISION
             C                COMPLEX SINGLE PRECISION
             L                LOGICAL
             A                STRING OF n CHARACTERS
 *****************************************************

 FORMAT SPECIFICATION EXAMPLE:
************************************************
  ("*" will precede the variable part)
  " 4A8,*(2RS,1C)" would be entered as follows:

  ENTER TYPE CODE OR "E" TO END FIXED PART   => A
  ENTER NUMBER OF CHARACTERS PER STRING      => 8
  ENTER NUMBER OF ELEMENTS IN SET            => 4
  ENTER TYPE CODE OR "E" TO END FIXED PART   => E
  ENTER TYPE CODE OR "E" TO END VARIABLE PART=> R
  ENTER NUMBER OF ELEMENTS IN SET            => 2
  ENTER TYPE CODE OR "E" TO END VARIABLE PART=> C
  ENTER NUMBER OF ELEMENTS IN SET            => 1
  ENTER TYPE CODE OR "E" TO END VARIABLE PART=> E
************************************************
  enter return to continue:
```

If the chosen format type is variable, the user will be asked if there is a fixed part to the format (a sequence of elements that do not repeat) before the variable part:

```
IS THERE A FIXED FORMAT PART BEFORE THE
 START OF THE VARIABLE FORMAT?
  ENTER "Y" OR "N"  =>
```

If the user enters Y, then the following prompts are displayed:

```
******START FIXED FORMAT *******

**************************************************
CHOOSE FROM THE FOLLOWING TYPE CODES:
TYPE CODE          ELEMENT VALUE TYPE
    I                INTEGER
    R                REAL SINGLE PRECISION
    C                COMPLEX SINGLE PRECISION
    L                LOGICAL
    A                STRING OF n CHARACTERS
**************************************************
ENTER TYPE CODE OF SET OR "E" TO END FIXED PART=>
**************************************************
ENTER NUMBER OF ELEMENTS IN SET =>
**************************************************
```

The prompts to enter element type and number will continue until user enters E to end the fixed part of format specifications. The following is displayed:

```
*****START VARIABLE FORMAT PART*******
**************************************************
  CHOOSE FROM THE FOLLOWING TYPE CODES:
    TYPE CODE          ELEMENT VALUE TYPE
       I                 INTEGER
       R                 REAL SINGLE PRECISION
       C                 COMPLEX SINGLE PRECISION
       L                 LOGICAL
       A                 STRING OF n CHARACTERS
**************************************************
  ENTER TYPE CODE  OR "E" TO END VARIABLE PART=>
**********************************************
  ENTER NUMBER OF ELEMENTS IN SET =>
**********************************************
```

The prompts to enter format element type and number will continue until the user enters E to end the variable part and complete the format.

When the format specification has been completed, the following is displayed:

```
**********************************************
ENTER MAXIMUM NUMBER OF RECORDS TO BE READ
 OR ENTER 0 FOR DEFAULT (10,000) =>
**********************************************
```

The user enters the maximum number of records that the MEMBER can have, or enters 0 for a maximum of 10000. The following is displayed:

```
**********************************************
ENTER ACTUAL NUMBER OF RECORDS TO BE READ=>
**********************************************
```

The user enters the actual number of records to be entered in the MEMBER. The user is prompted to enter each element of each record by type.

EXAMPLE: The prompts and responses to enter the following UNIT MEMBER will be listed:

```
UNIT name:                  UNIT1
MEMBER name:                MEM1
Maximum number of records:  10,000
Actual number of records:   2
Format specifications:      2I,1L,C (Each record will consist of 2
                                     integer values,followed by 1
                                     logical and 1 complex value)
Record one:                 2, 5, .FALSE., (2.5, 1.0)
Record two:                 1, 0, .TRUE.,  (1.5D4, 0.0)

**************************************************
ENTER UNIT NAME: 1-8 CHARACTERS=> UNIT1

**************************************************
ENTER MEMBER NAME: 1-8 CHARACTERS => MEM1
```

```
**********************************************
CHOOSE FROM THE FOLLOWING FORMAT TYPES:
   FORMAT CODE        FORMAT TYPE
      0                UNFORMATTED
      1                FIXED FORMAT
      2                VARIABLE FORMAT
      3                CARD IMAGE
**********************************************
ENTER FORMAT TYPE CODE=> 1
**********************************************
 DO YOU WISH TO SEE AN EXPLANATION AND EXAMPLE OF
   THE chosen TYPE OF FORMAT ENTRY?
    ENTER "Y" OR "N" => N
**********************************************


******START FIXED FORMAT *******

**********************************************
CHOOSE FROM THE FOLLOWING TYPE CODES:
TYPE CODE          ELEMENT VALUE TYPE
    I                INTEGER
    R                REAL SINGLE PRECISION
    C                COMPLEX SINGLE PRECISION
    L                LOGICAL
    A                STRING OF n CHARACTERS
**********************************************
ENTER TYPE CODE OF SET OR "E" TO END FIXED PART=> I
**********************************************
ENTER NUMBER OF ELEMENTS IN SET => 2

**********************************************
CHOOSE FROM THE FOLLOWING TYPE CODES:
TYPE CODE          ELEMENT VALUE TYPE
    I                INTEGER
    R                REAL SINGLE PRECISION
    C                COMPLEX SINGLE PRECISION
    L                LOGICAL
    A                STRING OF n CHARACTERS

**********************************************
ENTER TYPE CODE OF SET OR "E" TO END FIXED PART=> L
**********************************************
ENTER NUMBER OF ELEMENTS IN SET => 1
```

```
**************************************************
CHOOSE FROM THE FOLLOWING TYPE CODES:
TYPE CODE          ELEMENT VALUE TYPE
    I                INTEGER
    R                REAL SINGLE PRECISION
    C                COMPLEX SINGLE PRECISION
    L                LOGICAL
    A                STRING OF n CHARACTERS
**************************************************
ENTER TYPE CODE OF SET OR "E" TO END FIXED PART=> C
**************************************************
ENTER NUMBER OF ELEMENTS IN SET => 1
**************************************************
CHOOSE FROM THE FOLLOWING TYPE CODES:
TYPE CODE          ELEMENT VALUE TYPE

    I                INTEGER
    R                REAL SINGLE PRECISION
    C                COMPLEX SINGLE PRECISION
    L                LOGICAL
    A                STRING OF n CHARACTERS
**************************************************
ENTER TYPE CODE OF SET OR "E" TO END FIXED PART=>E


**************************************************
ENTER MAXIMUM NUMBER OF RECORDS TO BE READ
 OR ENTER 0 FOR DEFAULT (10,000) => 0
**********************************************
ENTER ACTUAL NUMBER OF RECORDS TO BE READ=> 2
**********************************************
RECORD    1:

ENTER INTEGER (1): 2
ENTER INTEGER (2): 5
ENTER LOGICAL   :  .FALSE.
ENTER REAL PART OF COMPLEX NUMBER: 2.5
ENTER IMAGINARY PART:  1.0

RECORD    2:

ENTER INTEGER (1): 1
ENTER INTEGER (2): 0
ENTER LOGICAL   :  .TRUE.
ENTER REAL PART OF COMPLEX NUMBER: 1.5D4
ENTER IMAGINARY PART:  0.0
```

OPTION 2

If the user wishes to print a MEMBER listing to the monitor screen, 2 is entered as a menu option and the following is displayed on the screen:

```
**************************************************
ENTER UNIT NAME: 1-8 CHARACTERS=>
```

The user enters the name of the UNIT which will contain the MEMBER. The user is then prompted to enter the MEMBER name:

```
**************************************************
ENTER MEMBER NAME: 1-8 CHARACTERS =>
```

The user enters the MEMBER name and is asked if a full listing of records is to be displayed.

```
**************************************************
DO YOU WISH A FULL LISTING OF RECORDS?
 ENTER "Y" OR "N" =>
**********************************************
```

The user enters Y, if the variable values are to be listed with the MEMBER description, and N, if only a summary of the MEMBER is to be given. The user is then asked if a format other than the one that was created with the MEMBER is to be used when listing the MEMBER. (Unformatted records are listed in Hexadecimal unless a format in entered.)

```
**************************************************
DO YOU WISH TO ENTER A FORMAT?
  ENTER "Y" OR "N" =>
```

If Y is entered, the user is prompted to enter format specifications as described in OPTION 1. The user is then asked if all records are to be listed:

```
**********************************************
DO YOU WISH TO LIST ALL RECORDS ?
  ENTER "Y" OR "N" =>
**********************************************
```

If N is entered, the user is asked to enter the number of the first and last record to be entered:

```
**************************************************
ENTER NUMBER OF FIRST RECORD
  TO BE LISTED =>

**********************************************
ENTER NUMBER OF LAST RECORD
  TO BE LISTED =>
**********************************************
```

The user enters the desired record numbers.

EXAMPLE: If the user wishes to list a MEMBER description with variable values to the screen for the MEMBER described in OPTION 1, the following prompts and responses would be displayed and entered:

```
**************************************************
ENTER UNIT NAME: 1-8 CHARACTERS=> UNIT1

**************************************************
ENTER MEMBER NAME: 1-8 CHARACTERS=>MEM1

**************************************************
  DO YOU WISH A FULL LISTING OF RECORDS?
   ENTER "Y" OR "N" => Y
**************************************************
DO YOU WISH TO ENTER A FORMAT?
  ENTER "Y" OR "N" =>N
**************************************************
DO YOU WISH TO LIST ALL RECORDS ?
  ENTER "Y" OR "N" => Y
**************************************************
```

The following will be displayed on the monitor screen:

```
  MEMBER NAME = MEM1
       MAXIMUM NO. OF RECORDS =  10000     DATE CREATED = 12/09/86
       CURRENT NO. OF RECORDS =      2     TIME CREATED = 13:14
       MAXIMUM RECORD LENGTH  =      5
       FORMAT= 2I,1L,1C$

RECORD    WORD
    1       1                        2                        5  F
    1       4 (0.25000000000000E+01,    0.10000000000000E+01)
    2       1                        1                        0  T
    2       4 (0.15000000000000E+05,    0.00000000000000E+00)
```

OPTION 3

If the user wishes to print a MEMBER listing to a DOS file, 3 is entered as a menu option and the following is displayed on the screen:

```
**************************************************
ENTER UNIT NAME: 1-8 CHARACTERS=>
```

The user enters the name of the UNIT that contains the MEMBER and the following is displayed:

```
**************************************************
ENTER MEMBER NAME: 1-8 CHARACTERS=>
```

The user enters the MEMBER name and the following is displayed:

```
**************************************************
ENTER LISTING FILE NAME=>
```

The user must type the name of the file that is to be created and contain the MEMBER listing. If an extension is desired, it must be entered. If a file with the same name exists, it will be deleted when the new file is created.
The following is displayed:

```
**************************************************
DO YOU WANT A 132 COLUMN LISTING?
  ENTER "Y" FOR 132 COLUMNS  OR
  "N" FOR 80 COLUMN DEFAULT=>
```

The user enters Y for a full 132 column wide listing or N for an 80 column wide listing in the DOS file and the following is displayed:

```
**************************************************
  DO YOU WISH A FULL LISTING OF RECORDS?
   ENTER "Y" OR "N" =>
```

The user enters Y if the record values are to be listed with the MEMBER description and N if only the MEMBER description is to be listed.

EXAMPLE: If the user wished to list a MEMBER description without record values to an 80 column wide DOS file, MEM1.LST, for the MEMBER described in OPTION 1, the following prompts and responses would be displayed:

```
**************************************************
ENTER UNIT NAME: 1-8 CHARACTERS=> UNIT1

**************************************************
ENTER MEMBER NAME: 1-8 CHARACTERS=>MEM1

**************************************************
ENTER LISTING FILE NAME=> MEM1.LST

**************************************************
DO YOU WANT A 132 COLUMN LISTING?
  ENTER "Y" FOR 132 COLUMNS  OR
  "N" FOR 80 COLUMN DEFAULT=> N

**************************************************
  DO YOU WISH A FULL LISTING OF RECORDS?
   ENTER "Y" OR "N" => N
**************************************************
```

The preprocessor then redisplays the main menu. To examine the listings file, the user must exit the MMEDT program (OPTION 0) and use a DOS file editor or a print command to access the file. MEM1.LST would contain the following data:

```
 MEMBER NAME = MEM1
      MAXIMUM NO. OF RECORDS =  10000     DATE CREATED = 12/09/86
      CURRENT NO. OF RECORDS =      2     TIME CREATED = 13:14
      MAXIMUM RECORD LENGTH  =      5
      FORMAT= 2I,1L,1C6
```

OPTION 4

If the user wishes to print a catalog listing of all MEMBERS in a data UNIT to the monitor screen, 4 is entered as a menu option and the following is displayed:

```
**************************************************
ENTER UNIT NAME: 1-8 CHARACTERS=>
```

The user enters the UNIT name and the following prompt is displayed:

```
************************************************
  DO YOU WISH A FULL LISTING OF MEMBERS?
   ENTER "Y" OR "N" =>
```

If the user types Y, a summary of all MEMBERS in that UNIT are listed on the monitor screen. If the user types N, only MEMBER names are listed.

EXAMPLE: If the user wished to display a full catalog listing of MEMBERS in the UNIT UNIT1 (described in example in OPTION 1), the following prompts and responses would be displayed:

```
*****************************************************
ENTER UNIT NAME: 1-8 CHARACTERS=> UNIT1

*****************************************************

 UNIT NAME = UNIT1        EXTERNAL FILE NAME = UNIT1      .UNT

 MEMBER NAME = MEM1
     MAXIMUM NO. OF RECORDS =  10000      DATE CREATED = 12/09/86
     CURRENT NO. OF RECORDS =      2      TIME CREATED = 13:14
     MAXIMUM RECORD LENGTH  =      5
     FORMAT= 2I,1L,1C$
```

OPTION 5

If the user wishes to print a catalog listing of all MEMBERS (a TABLE is a special one record MEMBER) in a data UNIT to a DOS file, 5 is entered as a menu option and the following is displayed:

```
*****************************************************
ENTER UNIT NAME: 1-8 CHARACTERS=>
```

The user enters the name of the UNIT that contains the MEMBER and the following is displayed:

```
*****************************************************
ENTER LISTING FILE NAME=>
```

The user must type the name of the file that is to be created and contain the catalog listing. If an extension is desired, it must be entered. If a file with the same name exists, it will be deleted when the new file is created.
The following is displayed:

```
**********************************************
DO YOU WANT A 132 COLUMN LISTING?
  ENTER "Y" FOR 132 COLUMNS  OR
  "N" FOR 80 COLUMN DEFAULT=>
```

The user enters Y for a full 132 column wide listing or N for an 80 column wide listing in the DOS file. The following is displayed:

```
**********************************************
  DO YOU WISH A FULL LISTING OF MEMBERS?
   ENTER "Y" OR "N" =>
```

If the user types Y, a summary of all MEMBERS in that UNIT are listed. If the user types N, only MEMBER names are listed. The preprocessor then redisplays the main menu. To examine the catalog listing file, the user must exit the MMEDT program (OPTION 0) and use a DOS file editor or a print commmand to access the file. The DOS file listing would have the same format as the catalog listing example in OPTION 4.

OPTION 6

If the user wishes to display a list of all existing DOS files which contain UNITS, 6 is entered as a menu option and the following is displayed:

```
      ************************************************************
      *  ENTER DIRECTORY DOS CALL-                               *
      *                                                          *
      *  DIR  *.UNT <cr>                                         *
      Execution suspended:  -enter <cr> to return to menu
      ************************************************************
```

If the user wishes to display the names of all UNIT files on the current directory, the following is entered:

```
  DIR *.UNT
```

A list of all UNIT files on the user's current directory will be displayed. The user must press the ENTER key to return to the main menu.

OPTION 7

If the user wishes to enter or change records in an existing MEMBER, 7 is entered as a menu option, and the user is prompted to enter the UNIT and MEMBER names of the UNIT MEMBER to be edited:

```
**************************************************
ENTER UNIT NAME: 1-8 CHARACTERS=>

**************************************************
ENTER MEMBER NAME: 1-8 CHARACTERS =>
```

The user enters the name of the UNIT which will contain the MEMBER, and the MEMBER name. The user is then asked if the number of maximum records on the MEMBER is to be altered:

```
DO YOU WISH TO ALTER MAXIMUM NUMBER OF RECORDS ON MEMBER
    ENTER "Y" or "N" =>
```

If Y is entered, the user is asked to enter the new value:

```
ENTER MAXIMUM NUMBER OF RECORDS OR 0 FOR DEFAULT
=>
```

The preprocessor will continue to prompt the user to enter whether records are to be inserted between the current records on the MEMBER or are to replace particular current records, until directive 3, (QUIT) is entered. The insertion and replacement of records must occur in numerical sequence according to the record numbers. (i.e. The record number entered, for inserting after or replacing a record, must be greater than the last record number entered after a directive option.)
The following prompts are displayed:

```
CHOOSE FROM THE FOLLOWING:
  1 - INSERT RECORDS
  2 - REPLACE RECORDS
  3 - QUIT
ENTER DIRECTIVE NUMBER =>
```

[DIRECTIVE 1 - INSERT RECORDS]

If the user wishes to insert records between currently existing records, 1 is entered and the following prompt is displayed:

```
ENTER RECORD NUMBER AFTER WHICH RECORDS
  ARE TO BE INSERTED =>
```

The user must enter the Original record number (Record number before any editing was done to the MEMBER), after which the new records are to be inserted. If the records are to be inserted before the first record, then 0 is entered. The following is displayed:

```
ENTER NUMBER OF RECORDS TO BE INSERTED=>
```

The user must enter the number of records to be inserted, and the following is displayed:

```
  CHOOSE FROM THE FOLLOWING:
  1 - COPY FROM CURRENT MEMBER
  2 - COPY FROM A DIFFERENT MEMBER
  3 - ENTER RECORDS FROM TERMINAL
ENTER OPTION NUMBER=>
```

The user enters from where the records are to be copied. If they are to be copied from records in the same MEMBER being edited, then option 1 is entered. If they are to be copied from another MEMBER, then option 2 is entered. If the records are to be entered from the terminal, then option 3 is entered and the user is prompted to enter each record element by type, according to the format specification. If the records in the MEMBER are unformatted, then the user is prompted to enter the element type before entering each set of elements. If Option 2 is entered, the following prompts are displayed:

```
ENTER UNIT NAME(1-8 CHARACTERS-no extension)=>

ENTER MEMBER NAME (1-8 CHARACTERS)=>
```

The user enters the names of the UNIT and MEMBER from which the records are to be copied. The user is then asked to enter the record number of the first record to be copied:

```
ENTER NUMBER OF FIRST RECORD TO BE COPIED=>
```

The user enters the record number and the required records are inserted.

[DIRECTIVE 2 - REPLACE RECORDS]

If directive 2 is entered, then records are to be replaced on the MEMBER. The options are the same as in directive 1 - inserting records.

```
EXAMPLE: The following MEMBER will be changed using OPTION 7:

 UNIT NAME = UNIT1          EXTERNAL FILE NAME = UNIT1    .UNT

  MEMBER NAME = MEM1
      MAXIMUM NO. OF RECORDS =       1      DATE CREATED = 12/09/86
      CURRENT NO. OF RECORDS =       1      TIME CREATED = 10:04
      MAXIMUM RECORD LENGTH  =       6
      FORMAT= 2C,1L,1A3$

RECORD    WORD
     1       1 (0.11000000000000E+01,    0.20000000000000E+01)
     1       2 (0.13000000000000E+01,    0.10000000000000E+01)  T  abc


The updated MEMBER will have the following records:

RECORD    WORD
     1       1 (0.11000000000000E+01,    0.20000000000000E+01)
     1       2 (0.13000000000000E+01,    0.10000000000000E+01)  T  abc
     2       1 (0.11000000000000E+01,    0.20000000000000E+01)
     2       2 (0.13000000000000E+01,    0.10000000000000E+01)  T  abc
     3       1 (0.00000000000000E+00,    0.10000000000000E+01)
     3       2 (0.30000000000000E+01,    0.00000000000000E+00)  F  new

OPTION 7 would be selected and the following prompts and responses
would be displayed and entered:
**************************************************
ENTER UNIT NAME: 1-8 CHARACTERS=>  UNIT1

**************************************************
ENTER MEMBER NAME: 1-8 CHARACTERS =>  MEM1

DO YOU WISH TO ALTER MAXIMUM NUMBER OF RECORDS ON MEMBER
    ENTER "Y" or "N" => Y

If Y is entered, the user is asked to enter the new value:

ENTER MAXIMUM NUMBER OF RECORDS OR 0 FOR DEFAULT
=>    3

CHOOSE FROM THE FOLLOWING:
  1 - INSERT RECORDS
  2 - REPLACE RECORDS
  3 - QUIT
ENTER DIRECTIVE NUMBER => 1

ENTER RECORD NUMBER AFTER WHICH RECORDS
  ARE TO BE INSERTED => 0
```

```
ENTER NUMBER OF RECORDS TO BE INSERTED=>1

  CHOOSE FROM THE FOLLOWING:
  1 - COPY FROM CURRENT MEMBER
  2 - COPY FROM A DIFFERENT MEMBER
  3 - ENTER RECORDS FROM TERMINAL
ENTER OPTION NUMBER=>1

ENTER NUMBER OF FIRST RECORD TO BE COPIED=>1

CHOOSE FROM THE FOLLOWING:
  1 - INSERT RECORDS
  2 - REPLACE RECORDS
  3 - QUIT
ENTER DIRECTIVE NUMBER => 1

ENTER RECORD NUMBER AFTER WHICH RECORDS
  ARE TO BE INSERTED => 1

ENTER NUMBER OF RECORDS TO BE INSERTED=>1

  CHOOSE FROM THE FOLLOWING:
  1 - COPY FROM CURRENT MEMBER
  2 - COPY FROM A DIFFERENT MEMBER
  3 - ENTER RECORDS FROM TERMINAL
ENTER OPTION NUMBER=>3

ENTER COMPLEX NUMBER(1):
ENTER REAL PART=> 0.D0
ENTER IMAGINARY PART=> 0.1D1
ENTER COMPLEX NUMBER(2):
ENTER REAL PART=> 0.3D1
ENTER IMAGINARY PART=> 0.D0
ENTER LOGICAL=> FALSE
ENTER 3 CHARACTERS=> new
```

OPTION 0

If the user wishes to exit the MMEDT program, 0 is entered and the following is displayed:

```
**********************************************
*                                            *
*        EXITING MEMBER MANAGER EDITOR       *
*                                            *
**********************************************
```

The program is completed and a prompt should be displayed by DOS.

Table Manager Editor (TMEDT)

TMEDT IS A TABLE MANAGER PREPROCESSOR FOR INTERACTIVELY ENTERING, EDITING OR LISTING TYPE ONE TABLES REQUIRED BY ANOPP FUNCTIONAL MODULES

The current directory must contain all unit member files which are to be edited. All output will be created on this directory.

File TMEDT.EXE must be installed on directory C:\ANOPP\EXE.
To run TMEDT the user types:

TABLE <cr>

The following menu will appear on the screen:

```
**************************************************
  CHOOSE FROM THE FOLLOWING OPTION CODES:
   OPTION CODE       OPTION
        1         ENTER TABLE
        2         PRINT TABLE TO SCREEN
        3         PRINT TABLE TO FILE
        4         PRINT CATALOG OF UNIT MEMBERS TO SCREEN
        5         PRINT CATALOG OF UNIT MEMBERS TO FILE
        6         DIRECTORY LISTING OF UNITS
        0         EXIT PROGRAM
**************************************************
ENTER OPTION CODE=>
```

(NOTE: the ENTER key must be pressed after each user response.)

OPTION 1

If the user wishes to create a new type one table, 1 is entered and the following prompt appears on the screen:

(Note: a table with the same unit and table name exists, it will be deleted when the new table is created.)

```
**************************************************
ENTER UNIT NAME: 1-8 CHARACTERS=>
```

The user enters the name of the UNIT which will contain the table. Extension .UNT will be added by the preprocessor.

```
EXAMPLE: If user enters:
                      =>   TABLE1
then external file TABLE1.UNT will be created.
*********************************************************
The user is then prompted to enter the table name:
*************************************************
ENTER TABLE NAME: 1-8 CHARACTERS=>

*************************************************
The user enters the table name and the user is prompted to enter
the number of interpolation procedures that are permitted on this
table:

 TYPE ONE TABLE
 ENTER NUMBER OF INTERPOLATION PROCEDURES
   PERMITTED ON THIS TABLE=>


The user enters the number of procedures, from the chart below,
  to be used and is prompted to enter the type codes for those
  procedures:

***********************************************
 CHOOSE FROM THE FOLLOWING  TYPES:
   CODE      INTERPOLATION TYPE
     0        NO INTERPOLATION
     1        LINEAR INTERPOLATION
     2        CUBIC SPLINE INTERPOLATION
***********************************************
 ENTER INTERPOLATION CODE=>

( ENTER INTERPOLATION CODE=>) -displayed if 2 or more procedures
      are to be used

( ENTER INTERPOLATION CODE=>) -displayed if 3 procedures are to
      be used

The user is then prompted to enter the dependent variable type:

***********************************************
 CHOOSE FROM THE FOLLOWING  TYPES:
   CODE      DEPENDENT VARIABLE TYPE
     I           INTEGER
     R           REAL
     C           COMPLEX
***********************************************
 ENTER TYPE CODE FOR DEPENDENT VARIABLES=>
```

The user is then prompted to enter the number of independent variables:

```
*********************************************
 ENTER NUMBER OF INDEPENDENT VARIABLES=>
```

The user enters the number of independent variables (maximum of four) and the program lists the independent variable number and prompts for the following information for each independent variable:

```
***INDEPENDENT VARIABLE  #  ** (where # is the actual number)
*********************************************
 CHOOSE FROM THE FOLLOWING  TYPES:
  CODE     INDEPENDENT VARIABLE TYPE
   O           ORDERED POSITION
   I           INTEGER
   R           REAL
*********************************************
ENTER INDEPENDENT VARIABLE TYPE CODE=>
```

The user enters the type code for a particular independent variable. All independent and dependent variable types within a table do not have to be the same type. The user is then prompted to enter the number values in a particular independent variable:

```
*********************************************
ENTER NUMBER OF VALUES=>
*********************************************
```

The user enters the number of values. All independent variables do not have to have the same number of values. The user is then prompted to enter the extrapolation types, for values above the upper bound and values below the lower bound, for the particular independent variable.

```
*****************************************
CHOOSE FROM THE FOLLOWING  TYPES:
 CODE      EXTRAPOLATION TYPE
  0        NO EXTRAPOLATION
  1        CLOSEST VALUE
  2        LINEAR EXTRAPOLATION
*****************************************
 ENTER CODE FOR EXTRAPOLATION TYPE IF
 DESIRED VALUE IS BEYOND UPPER BOUND=>
*****************************************
 ENTER CODE FOR EXTRAPOLATION TYPE IF
 DESIRED VALUE IS BELOW LOWER BOUND=>
*****************************************
```

The user enters the extrapolation types and is prompted to enter each value by type, for a particular independent variable:

EXAMPLE: If the user has entered that there are two values of type integer for a particular independent variable, then the following prompts would be displayed:

```
ENTER INTEGER (1) =>

ENTER INTEGER (2) =>
```

The above sequence will be repeated until all independent variables have been entered. The program then displays prompts to enter all dependent varibles by type. (The number of dependent variables is equal to the product of the number of values for each independent variable. EXAMPLE: If there are three independent variables and independent variable one has three values, independent variable two has two values, and independent variable three has four values, then the number of dependent values would be 3 x 2 x 4 = 24 values. The user would be prompted to enter each of the twenty four dependent variable values.)

EXAMPLE: If the dependent variable type is REAL, then the following prompts will be displayed:

```
DEPENDENT VARIABLES

ENTER REAL NUMBER (1) =>

ENTER REAL NUMBER (2) =>

ENTER REAL NUMBER (3) =>

        .
        .
        .
```

until all dependent values have been entered.

The table is then completed and the main menu is redisplayed.

EXAMPLE: The prompts and responses to enter the following table will be listed:

```
Unit name:                  UNI
Table name:                 DMS
Interpolation procedures
 allowed on table:          No interpolation
                            Linear interpolation
Dependent variable type:    INTEGER
Number of independent
 variables:                 2
Independent variable 1
----------------------
 Type:                      REAL
 Number of values:          3
 Extrapolation (1):         no extrapolation allowed
 Extrapolation (2):         use table value closest to specified value
 Values:                    1.5, 2.0, 4.5
Independent variable 2
----------------------
 Type:                      INTEGER
 Number of values:          2
 Extrapolation (1):         no extrapolation allowed
 Extrapolation (2):         use table value closest to specified value
 Values:                    5, 10
Dependent variable values: 3, 5, 7, 8, 9, 10

*Note*
Extrapolation(1) is the procedure to be used if the specified value
for the independent variable falls after the last table value for the
independent variable.
Extrapolation(2) is the procedure to be used if the specified value
for the independent variable falls before the first table value for
the independent variable.

************************************************
ENTER UNIT NAME: 1-8 CHARACTERS=> UNI

************************************************
ENTER TABLE NAME: 1-8 CHARACTERS=> DMS

************************************************

 TYPE ONE TABLE
 ENTER NUMBER OF INTERPOLATION PROCEDURES
   PERMITTED ON THIS TABLE=> 2
```

```
*********************************************
 CHOOSE FROM THE FOLLOWING  TYPES:
   CODE      INTERPOLATION TYPE
     0        NO INTERPOLATION
     1        LINEAR INTERPOLATION
     2        CUBIC SPLINE INTERPOLATION
*********************************************
 ENTER INTERPOLATION CODE=> 0

 ENTER INTERPOLATION CODE=> 1

*********************************************
 CHOOSE FROM THE FOLLOWING  TYPES:
   CODE      DEPENDENT VARIABLE TYPE
     I           INTEGER
     R           REAL
     C           COMPLEX
*********************************************
 ENTER TYPE CODE FOR DEPENDENT VARIABLES=> I

*********************************************
  ENTER NUMBER OF INDEPENDENT VARIABLES=> 2

 *** INDEPENDENT VARIABLE  1 **
*********************************************
 CHOOSE FROM THE FOLLOWING  TYPES:
  CODE     INDEPENDENT VARIABLE TYPE
   0           ORDERED POSITION
   I           INTEGER
   R           REAL
*********************************************
ENTER INDEPENDENT VARIABLE TYPE CODE=>R

*********************************************
ENTER NUMBER OF VALUES=> 3

*********************************************
CHOOSE FROM THE FOLLOWING  TYPES:
 CODE      EXTRAPOLATION TYPE
  0        NO EXTRAPOLATION
  1        CLOSEST VALUE
  2        LINEAR EXTRAPOLATION
*********************************************
 ENTER CODE FOR EXTRAPOLATION TYPE IF
 DESIRED VALUE IS BEYOND UPPER BOUND=>0
*********************************************
 ENTER CODE FOR EXTRAPOLATION TYPE IF
 DESIRED VALUE IS BELOW LOWER BOUND=> 1
*********************************************

ENTER REAL NUMBER (1) => 1.5
ENTER REAL NUMBER (1) => 2.0
ENTER REAL NUMBER (1) => 4.5
```

```
*** INDEPENDENT VARIABLE  2 **
*********************************************
 CHOOSE FROM THE FOLLOWING  TYPES:
  CODE      INDEPENDENT VARIABLE TYPE
   O            ORDERED POSITION
   I            INTEGER
   R            REAL
*********************************************
ENTER INDEPENDENT VARIABLE TYPE CODE=>I

*********************************************
ENTER NUMBER OF VALUES=> 2
*********************************************
 CHOOSE FROM THE FOLLOWING  TYPES:
  CODE       EXTRAPOLATION TYPE
   0         NO EXTRAPOLATION
   1         CLOSEST VALUE
   2         LINEAR EXTRAPOLATION
*********************************************
 ENTER CODE FOR EXTRAPOLATION TYPE IF
 DESIRED VALUE IS BEYOND UPPER BOUND=>0
*********************************************
 ENTER CODE FOR EXTRAPOLATION TYPE IF
 DESIRED VALUE IS BELOW LOWER BOUND=> 1
*********************************************

ENTER INTEGER (1) => 5

ENTER INTEGER (2) => 10

DEPENDENT VARIABLES

ENTER INTEGER (1) =>3

ENTER INTEGER (1) =>5

ENTER INTEGER (1) =>7

ENTER INTEGER (1) =>8

ENTER INTEGER (1) =>9

ENTER INTEGER (1) =>10
```

OPTION 2

If the user wishes to print a table listing to the monitor screen, 2 is entered as a menu option and the following is displayed on the screen:

```
**************************************************
ENTER UNIT NAME: 1-8 CHARACTERS=>
```

The user enters the name of the unit that contains the table and the following is displayed:

```
**************************************************
ENTER TABLE NAME: 1-8 CHARACTERS=>
```

The user enters the table name and the following is displayed:

```
**********************************************
  DO YOU WISH A FULL LISTING OF RECORDS?
   ENTER "Y" OR "N" =>
**********************************************
```

The user enters Y, if the variable values are to be listed with the table description, and N, if only the table description is to be listed.

EXAMPLE: If the user wishes to list a table description with variable values to the screen for the table described in OPTION 1, the following prompts and responses will be displayed and entered:

```
**************************************************
ENTER UNIT NAME: 1-8 CHARACTERS=> UNI

**************************************************
ENTER TABLE NAME: 1-8 CHARACTERS=>DMS

**********************************************
  DO YOU WISH A FULL LISTING OF RECORDS?
   ENTER "Y" OR "N" => Y
**********************************************
```

The following listing will be displayed:

```
UNIT NAME = UNI          EXTERNAL FILE NAME = UNI      .UNT

MEMBER NAME = DMS          DATE CREATED = 12/08/86     TIME CREATED =  8:49
                    NO. INDEPENDENT VARIABLES =  2     TABLE LENGTH =    31
                    INTERPOLATION PROCEDURES =
                                                 NO INTERPOLATION (0)
                                                 LINEAR INTERPOLATION (1)
    INDEPENDENT VARIABLE 1

         FORMAT TYPE                  NO. VARIABLES
    REAL SINGLE (2)                         3

         EXTRAPOLATION BEYOND                EXTRAPOLATION BEYOND
         LARGEST VALUE                       SMALLEST VALUE

         EXTRAPOLATION NOT ALLOWED (0)       CLOSEST VARIABLE VALUE (1)

         VALUES

 POSITION
    1    0.150000000000000E+01    0.200000000000000E+01    0.450000000000000E+01

   INDEPENDENT VARIABLE 2
       FORMAT TYPE                NO. VARIABLES
   INTEGER (1)                           2

         EXTRAPOLATION BEYOND                EXTRAPOLATION BEYOND
         LARGEST VALUE                       SMALLEST VALUE

         EXTRAPOLATION NOT ALLOWED (0)       CLOSEST VARIABLE VALUE (1)

         VALUES

POSITION
     1                    5                   10

    DEPENDENT VARIABLE

         FORMAT TYPE                  NO. VARIABLES
    INTEGER (1)                             6

         VALUES

POSITION
     1              3              5              7
     4              8              9             10

-----------------------------------------------------------------
```

OPTION 3

If the user wishes to print a table listing to a DOS file, 3 is entered as a menu option and the following is displayed on the screen:

```
**************************************************
ENTER UNIT NAME: 1-8 CHARACTERS=>
```

The user enters the name of the unit that contains the table and the following is displayed:

```
**************************************************
ENTER TABLE NAME: 1-8 CHARACTERS=>
```

The user enters the table name and the follwing is displayed:

```
**************************************************
ENTER LISTING FILE NAME=>
```

The user must type the name of the file that is to be created and contain the table listing. If an extension is desired, it must be entered. If a file with the same name exists, it will be deleted when the new file is created. The following is displayed:

```
**************************************************
DO YOU WANT A 132 COLUMN LISTING?
  ENTER "Y" FOR 132 COLUMNS  OR
  "N" FOR 80 COLUMN DEFAULT=>
```

The user enters Y for a full 132 column wide listing or N for an 80 column wide listing in the DOS file and the following is displayed:

```
**************************************************
  DO YOU WISH A FULL LISTING OF RECORDS?
   ENTER "Y" OR "N" =>
```

The user enters Y, if the variable values are to be listed with the table description, and N, if only the table description is to be listed.

EXAMPLE: If the user wished to list a table description without variable values to an 80 column wide DOS file, DMS.LST, for the table described in OPTION 1, the following prompts and responses would be displayed:

```
**************************************************
ENTER UNIT NAME: 1-8 CHARACTERS=> UNI

**************************************************
ENTER TABLE NAME: 1-8 CHARACTERS=>DMS

**************************************************
ENTER LISTING FILE NAME=> DMS.LST

**************************************************
DO YOU WANT A 132 COLUMN LISTING?
  ENTER "Y" FOR 132 COLUMNS  OR
  "N" FOR 80 COLUMN DEFAULT=> N

**************************************************
  DO YOU WISH A FULL LISTING OF RECORDS?
   ENTER "Y" OR "N" => N
```

The preprocessor then redisplays the main menu. To examine the listings file the user must exit the TMEDT program (OPTION 0) and use a DOS file editor or a print command to access the file. DMS.LST would contain the following data:

```
UNIT NAME = UNI          EXTERNAL FILE NAME = UNI      .UNT

MEMBER NAME = DSM        DATE CREATED = 12/08/86   TIME CREATED =  8:49
               NO. INDEPENDENT VARIABLES =  2     TABLE LENGTH =    31
               INTERPOLATION PROCEDURES =
                                        NO INTERPOLATION (0)
                                        LINEAR INTERPOLATION (1)
```

OPTION 4

If the user wishes to print a catalog listing of all members (a table is a special one record member) in a data unit to the monitor screen, 4 is entered as a menu option and the following is displayed:

```
************************************************
ENTER UNIT NAME: 1-8 CHARACTERS=>
```

The user enters the unit name and a summary of all members in the unit is listed on the monitor screen.

EXAMPLE: If the user wishes to display a catalog listing of members in the unit UNI (desribed in example in OPTION 1), the following prompts and responses would be displayed:

```
************************************************
ENTER UNIT NAME: 1-8 CHARACTERS=> UNI


 UNIT NAME = UNI          EXTERNAL FILE NAME = UNI     .UNT

 MEMBER NAME = DSM
      MAXIMUM NO. OF RECORDS =      1     DATE CREATED = 12/08/86
      CURRENT NO. OF RECORDS =      1     TIME CREATED =  8:49
      MAXIMUM RECORD LENGTH  =     31
      FORMAT = UNFORMATTED
```

OPTION 5

If the user wishes to print a catalog listing of all members (a table is a special one record member) in a data unit to a DOS file, 5 is entered as a menu option and the following is displayed:

```
************************************************
ENTER UNIT NAME: 1-8 CHARACTERS=>
```

The user enters the name of the unit that contains the table and the following is displayed:

```
************************************************
ENTER LISTING FILE NAME=>
```

The user must type the name of the file that is to be created and will contain the catalog listing. If an extension is desired, it must be entered. If a file with the same name exists, it will be deleted when the new file is created. The following is displayed:

```
**********************************************
DO YOU WANT A 132 COLUMN LISTING?
  ENTER "Y" FOR 132 COLUMNS  OR
  "N" FOR 80 COLUMN DEFAULT=>
```

The user enters Y for a full 132 column wide listing or N for an 80 column wide listing in the DOS file. The preprocessor then redisplays the main menu. To examine the catalog listing file the user must exit the TMEDT program (OPTION 0) and use a DOS file editor or a print commmand to access the file. The DOS file listing would have the same format as the catalog listing example in OPTION 4.

OPTION 6

If the user wishes to display a list of all existing DOS files which contain units, 6 is entered as a menu option and the following is displayed:

```
*************************************************************
*  ENTER DIRECTORY DOS CALL-                                *
*                                                           *
*  DIR  *.UNT <cr>                                          *
Execution suspended:  -enter <cr> to return to menu
*************************************************************
```

If the user wishes to display the names of all unit files on the current directory, the following is entered:

```
DIR *.UNT
```

A list of all unit files on the user's current directory will be displayed. The user must press the ENTER key to return to the main menu.

OPTION 0

If the user wishes to exit the TMEDT program, 0 is entered and the following is displayed:

```
***********************************************
*                                             *
*        EXITING TABLE MANAGER EDITOR         *
*                                             *
***********************************************
```

The program is completed and a prompt should be displayed by DOS.

ANOPP-PAS PROCEDURES AND FUNCTIONAL MODULES

Overview

This section describes the five ANOPP-PAS prediction procedures and the eighteen functional modules which they control. It provides instructions for their use, documents the input data required for execution, and describes the resulting output. Each procedure is executed by typing a single command, and each controls the execution of one or more functional modules. Single functional modules may be executed using the RUN command as documented in Section IV.

Prediction Procedures

The prediction procedures create or verify the existence of input required by specific functional modules and call those functional modules into execution.

The Blade Geometry Procedure is called by command GEOM and controls the execution of:

RBS		IBS
RBA	or	IBA
BLM		IBL

The Performance Procedure is called by command PERF and controls the execution of:

PRP
PLD

The Noise Procedure is called by command NOISE and controls the execution of:

SPN
PTE

The Flight Path Procedure is called by command FPATH and controls the execution of:

ATM
ABS
SFO
GEO

The Propagation Procedure is called by command LEVELS and controls the execution of:

PRT
PRO
LEV
EFF

Functional Modules

The functional modules described in this section have internal documentation that is maintained in the FORTRAN source code and on the Documentation Diskette. For ease of use, it is arranged in accordance with the format described in figure 3.

The ANOPP Theoretical Manual, volume 1 (ref. 4) and volume 3 (ref. 1) document the prediction methods for the functional modules.

```
PURPOSE - short description of the functional module (1 - 2 sentences)

AUTHOR - initials and level number, such as L01/00/00

INPUT
  USER PARAMETERS
    Name1 - description - default value
    .
    .
    .
    Namen - description - default value
  MEMBERS AND TABLES
    DATA UNIT(DATA MEMBER) - short description of data requirement

OUTPUT
  SYSTEM PARAMETERS
    Name - description
  USER PARAMETER - same as for INPUT
  MEMBERS AND TABLES
    DATA UNIT(DATA MEMBER) - short description of data requirements

DATA BASE STRUCTURES
  DATA UNIT(DATA MEMBER) - complete description of data and required format

ERRORS
  NON-FATAL - description of errors that are possible within the functional
              module.
  FATAL - functional modules are requested to not use fatal errors.  In the
          event this is not followed they must be documented in this section.

INPUT SIZE RESTRICTIONS - describes maximum number of entires that are
                          allowed for input data.
LDS REQUIREMENTS - describes the amount of local dynamic storage required by
                   this module and maximum allocations.
GDS REQUIREMENTS - describes the amount of global dynamic storage required for
                   this module and maximum allocations.
```

Figure 3.- ANOPP functional module prologue format.

PURPOSE - TO FORMULATE A FUNCTIONAL REPRESENTATION OF THE BLADE SURFACE SUITABLE FOR AEROACOUSTIC AND AERODYNAMIC CALCULATIONS

AUTHOR - CBF(L03/02/00)

INPUT

USER PARAMETERS

		DEFAULT
B	- BLADE LENGTH MEASURED FROM AXIS TO TIP (RS), M (FT)	1.
IUNITS	- INPUT UNITS FLAG (A) SI, INPUT IN SI UNITS ENGLISH, INPUT IN ENGLISH UNITS	SI
IPRINT	- PRINT FLAG (I) =0, NO PRINT DESIRED =1, INPUT PRINT ONLY =2, OUTPUT PRINT ONLY =3, BOTH INPUT AND OUTPUT PRINT	3
ZSLOPE	- BOUNDARY CONDITION OPTION (L) =.TRUE., ZERO SLOPE =.FALSE., ZERO CURVATURE	F
WEIGHT	- WEIGHTING OPTION (L) =.TRUE., WEIGHTING FACTORS ARE INCLUDED =.FALSE., WEIGHTING FACTORS ARE NOT INCLUDED	F

DATA BASE UNIT MEMBERS
(DESCRIBED UNDER DATA BASE STRUCTURES)
GRID(XI2)
GEOM (BLADE)

OUTPUT

USER PARAMETERS - NONE

SYSTEM PARAMETERS

NERR	- SYSTEM ERROR FLAG =.TRUE., AN ERROR WAS ENCOUNTERED DURING MODULE EXECUTION =.FALSE., NO ERRORS OCCURRED

DATA BASE UNIT MEMBERS
(DESCRIBED UNDER DATA BASE STRUCTURES)
RBS(SHAPE)
RBS(SPAN)
RBS(BLD)
RBS(SLPSPN)
RBS(SLPCRD)
GRID(XI1)

DATA BASE STRUCTURES

GRID(XI1) ONE-RECORD OUTPUT MEMBER IN RS FORMAT CONTAINING THE ARRAY OF SPANWISE STATIONS, RE B. SPANWISE STATIONS ARE THE SPANWISE COORDINATES FROM UNIT MEMBER, GEOM(BLADE) DIVIDED BY THE USER PARAMETER, B.

GRID(XI2) ONE-RECORD INPUT MEMBER IN RS FORMAT CONTAINING THE ARRAY OF CHORDWISE STATIONS BETWEEN 0 AND 1, RADIANS

GEOM(BLADE) MULTI-RECORD UNFORMATTED MEMBER CONTAINING SPANWISE FUNCTIONS OF THE BLADE IF WEIGHTING IS INCLUDED, (IE., IF PARAMETER WEIGHT IS .TRUE.) A WEIGHTING FACTOR SHOULD BE INCLUDED AFTER EACH (X,Y) PAIR. IF NO WEIGHTING FACTOR IS INCLUDED FOR A GIVEN (X,Y) PAIR, IT IS ASSUMED TO BE ONE.

RECORD	FORMAT	DESCRIPTION
1	I	NUMBER OF AIRFOIL SECTIONS ON THE BLADE
2	6RS,2I	SPANWISE COORDINATE, LEADING EDGE ABSCISSA, LEADING EDGE ORDINATE, CHORD LENGTH, LEADING EDGE RADIUS (RE CHORD LENGTH), BLADE TWIST ANGLE MEASURED POSITIVE CLOCKWISE LOOKING OUTBOARD, IN DEGREES, NUMBER OF (X,Y) PAIRS ON UPPER SURFACE, AND NUMBER OF (X,Y) PAIRS ON LOWER SURFACE OF FIRST AIRFOIL SECTION. FIRST FOUR WORDS ARE IN THE SAME UNITS AS GIVEN BY PARAMETER, IUNITS.
3	RS	FIRST (X,Y) PAIR ON UPPER SURFACE OF FIRST AIRFOIL SECTION (AND IF PARAMETER WEIGHT IS .TRUE., A WEIGHTING FACTOR)

4	RS	SECOND (X,Y) PAIR ON UPPER SURFACE OF FIRST AIRFOIL SECTION (AND IF PARAMETER WEIGHT IS .TRUE., A WEIGHTING FACTOR)
.	.	
.	.	
.	.	
N+2	RS	NTH (X,Y) PAIR ON UPPER SURFACE OF FIRST AIRFOIL SECTION (AND IF PARAMETER WEIGHT IS .TRUE., A WEIGHTING FACTOR)
N+3	RS	FIRST(X,Y) PAIR ON LOWER SURFACE OF FIRST AIRFOIL SECTION (AND IF PARAMETER WEIGHT IS .TRUE., A WEIGHTING FACTOR)
N+4	RS	SECOND (X,Y) PAIR ON LOWER SURFACE OF FIRST AIRFOIL SECTION (AND IF PARAMETER WEIGHT IS .TRUE., A WEIGHTING FACTOR)
.	.	
.	.	
.	.	
N+2+M	RS	MTH (X,Y) PAIR ON LOWER SURFACE OF FIRST AIRFOIL SECTION (AND IF PARAMETER WEIGHT IS .TRUE., A WEIGHTING FACTOR)
.	.	
.	.	
.	.	

(RECORDS 2 THROUGH N+2+M REPEAT FOR EACH AIRFOIL SECTION. NOTE : M AND N MAY BE UNEQUAL AND MAY BE DIFFERENT FOR EACH SECTION.)

RBS(SHAPE) TYPE ONE DATA TABLE CONTAINING FUNCTIONS OF SPAN AND CHORD OF THE BLADE. FIRST DIMENSION IS SPANWISE STATION, RE B. SECOND DIMENSION IS CHORDWISE STATION, IN RADIANS. THIRD DIMENSION IS ORDERED POSITION WITH THE FOLLOWING THREE SETS OF DATA :
(1) BLADE SURFACE ABSCISSAS, RE B
(2) BLADE SURFACE ORDINATES, RE B
(3) BLADE SURFACE ELLIPTIC RADIAL COORDINATES, IN RADIANS

RBS(SPAN) TYPE ONE DATA TABLE CONTAINING FUNCTIONS OF SPAN. FIRST DIMENSION IS SPANWISE STATIONS. SPANWISE STATIONS ARE THE SPANWISE COORDINATES, NORMALIZED.
SECOND DIMENSION IS ORDERED POSITION IN THE FOLLOWING ORDER :
(1) LEADING EDGE ABSCISSAS, RE B
(2) LEADING EDGE ORDINATES, RE B

(3) JOUKOWSKI TRANSFORMATION PARAMETERS, RE B
(4) CHORD LENGTHS, RE B
(5) ELLIPTIC SYSTEM ORIGIN ABSCISSAS, RE B
(6) ELLIPTIC SYSTEM ORIGIN ORDINATES, RE B
(7) BLADE ELLIPTIC AXIS TWIST ANGLE MEASURED POSITIVE CLOCKWISE LOOKING OUTWARD, IN RADIANS
(8) BLADE CROSS-SECTIONAL AREA, RE B 2

RBS(BLD) ONE RECORD MEMBER IN 3RS FORMAT CONTAINING THE BLADE VOLUME, RE B 3, THE BLADE ASPECT RATIO, AND THE BLADE ACTIVITY FACTOR

RBS(SLPSPN) MULTI-RECORD MEMBER IN RS FORMAT CONTAINING THE SLOPE MATRIX IN THE SPANWISE DIRECTION. A RECORD ON THE MEMBER REPRESENTS A COLUMN IN THE MATRIX. THE NUMBER OF RECORDS EQUALS THE NUMBER OF WORDS PER RECORD ON THE MEMBER. (IE, A SQUARE MATRIX)

RBS(SLPCRD) MULTI-RECORD MEMBER IN RS FORMAT CONTAINING THE SLOPE MATRIX IN THE CHORDWISE DIRECTION. A RECORD ON THE MEMBER REPRESENTS A COLUMN IN THE MATRIX. THE NUMBER OF RECORDS EQUALS THE NUMBER OF WORDS PER RECORD ON THE MEMBER. (IE, A SQUARE MATRIX)

ERRORS

NON-FATAL

1. INSUFFICIENT LOCAL DYNAMIC STORAGE.
2. UNABLE TO BUILD SPECIFIED OUTPUT TABLE.
3. MEMBER MANAGER ERROR OCCURRED ON SPECIFIED UNIT MEMBER.
4. NUMBER OF SPANWISE OR CHORDWISE STATIONS IS .LT. 3.
5. SPECIFIED UNIT MEMBER NOT AVAILABLE.
6. ARRAY OF INDEPENDENT VARIABLE VALUES NOT IN INCREASING ORDER.
7. SINGULAR MATRIX WHEN SOLVING SYSTEM OF REAL LINEAR EQUATIONS IN SUBPROGRAM GELIM.
8. X OR Y VALUE ON UNIT MEMBER GEOM(BLADE) IS .LT. 0 OR .GT. 1. (WARNING ONLY)
9. DETERMINANT OF MATRIX IS ZERO.
10. NUMBER OF CHORDWISE STATIONS EXCEEDS THE NUMBER OF (X,Y) PAIRS ON A PARTICULAR AIRFOIL SECTION.
11. (X,Y) PAIRS, TRANSFORMED TO THE ELLIPTIC COORDINATE SYSTEM DO NO LIE BETWEEN THE FIRST AND LAST CHORDWISE STATIONS.

FATAL - NONE

INPUT DATA SIZE RESTRICTIONS

	MAXIMUM NUMBER
NUMBER OF AIRFOILS ON BLADE	10
NUMBER OF CHORDWISE STATIONS	25

LDS REQUIREMENTS

(Maximum Allocation of LDS - 5190)

```
LENGTH = ( NCROSS * 25 ) + ( NPAIR * 3 ) + ( NXY * (2) )
       + NWT + ( NCROSS ** 2 ) * 2 ) + ( ( NXI2 ** 2 ) * 3 )
       + ( NXI2 * NCROSS * 4 ) + NXI2 * ( 13 + LARGE )
```

WHERE

- NCROSS = NUMBER OF AIRFOILS ON THE BLADE (IE., NUMBER OF SPANWISE STATIONS)
- NXI2 = NUMBER OF CHORDWISE STATIONS
- NPAIR = TOTAL NUMBER OF (X,Y) PAIRS ON THE BLADE SURFACE
- NXY = NPAIR * 3, IF WEIGHTING FACTORS ARE INCLUDED
  NPAIR * 2, IF NO WEIGHTING IS USED
- NWT = NPAIR, IF WEIGHTING FACTORS ARE INCLUDED
  0, IF NO WEIGHTING IS USED
- LARGE = MAXIMUM NUMBER OF (X,Y) PAIRS ON AN AIRFOIL SECTION

GDS REQUIREMENTS

(Maximum Allocation of GDS - 3000)

SUFFICIENT ALLOCATION FOR THE FOLLOWING TABLES :

- RBS(SHAPE)
- RBS(SPAN)

PURPOSE - COMPUTES THE PRESSURE FORCES ACTING ON THE UPPER AND LOWER SURFACES OF A TWO-DIMENSIONAL AIRFOIL FOR SPECIFIED ANGLE OF ATTACK AND MACH NUMBER VALUES

AUTHOR - CBF(L03/02/00)

INPUT

		DEFAULT
USER PARAMETERS		
EPSLON	- CRITERION FOR STOPPING ITERATION (RS)	.001
NORDER	- ORDER OF FOURIER SERIES (I) (VALUE SHOULD BE IN RANGE OF 1 .LE. NORDER .LE. (NXI2-1)/4, WHERE NXI2 = NUMBER OF VALUES ON GRID(XI2))	5
IPRINT	- PRINT FLAG (I) =0, NO PRINT DESIRED =1, INPUT PRINT ONLY =2, OUTPUT PRINT ONLY =3, BOTH INPUT AND OUTPUT PRINT	3
RBAPRES	- THREE-LETTER CODE, YYY, USED TO FORM TABLE UNIT MEMBER NAME RBA(YYYNNN) WHERE NNN IS THE INDEX INTO THE MACH NUMBER ARRAY AND RBA(YYYNNN) REPRESENTS THE TABLES OF LOCAL PRESSURE COEFFICIENTS	PRS

DATA BASE UNIT MEMBERS
(DESCRIBED UNDER DATA BASE STRUCTURES)
- RBA(MACH)
- RBA(ALPHA)
- RBS(SPAN)
- RBS(SHAPE)
- RBS(SLPSPN)
- RBS(SLPCRD)

OUTPUT

SYSTEM PARAMETERS

NERR	- SYSTEM ERROR FLAG	
	=.TRUE.,	AN ERROR WAS ENCOUNTERED DURING MODULE EXECUTION
	=.FALSE.,	NO ERRORS OCCURRED

DATA BASE UNIT MEMBERS
(DESCRIBED UNDER DATA BASE STRUCTURES)

- RBATMP(LIFT)
- RBA(AEROCTR)
- RBA(MOMENT)
- RBA(YYYNNN) NOTE : MEMBER NAME YYYNNN IS FORMED FROM USER PARAMETER RBAPRES AND FROM THE INDEX INTO THE MACH NUMBER ARRAY FOR THE MACH NUMBER ASSOCIATED WITH THE TABLE. CONTAINS THE LOCAL PRESSURE COEFFICIENTS.

DATA BASE STRUCTURES (NOTE: B REFERS TO BLADE LENGTH)

- RBA(MACH) ONE-RECORD INPUT MEMBER IN RS FORMAT CONTAINING RANGE OF MACH NUMBERS
- RBA(ALPHA) ONE-RECORD INPUT MEMBER IN RS FORMAT CONTAINING THE RANGE OF ANGLES OF ATTACK, IN DEGREES
- RBS(SLPSPN) MULTI-RECORD INPUT MEMBER IN RS FORMAT CONTAINING THE SLOPE MATRIX ASSOCIATED WITH THE SPANWISE DIRECTION (EACH RECORD CORRESPONDS TO A COLUMN IN THE MATRIX)
- RBS(SLPCRD) MULTI-RECORD INPUT MEMBER IN RS FORMAT CONTAINING THE SLOPE MATRIX ASSOCIATED WITH THE CHORDWISE DIRECTION (EACH RECORD CORRESPONDS TO A COLUMN IN THE MATRIX)
- RBS(SPAN) TYPE ONE DATA TABLE CONTAINING FUNCTIONS OF SPANWISE DIRECTION OF THE BLADE. FIRST DIMENSION IS SPANWISE STATIONS, RE B. SECOND DIMENSION IS ORDERED POSITION IN THE FOLLOWING ORDER:
  (1) LEADING EDGE ABSCISSAS, RE B
  (2) LEADING EDGE ORDINATES, RE B
  (3) JOUKOWSKI TRANSFORMATION PARAMETERS, RE B
  (4) CHORD LENGTHS, RE B
  (5) ELLIPTIC SYSTEM ORIGIN ABSCISSAS, RE B
  (NOT USED IN THIS MODULE)
  (6) ELLIPTIC SYSTEM ORIGIN ORDINATES, RE B
  (NOT USED IN THIS MODULE)
  (7) BLADE TWIST ANGLE MEASURED POSITIVE CLOCKWISE LOOKING OUTBOARD, IN RADIANS
  (8) BLADE SECTION AREA, RE B 2
  (NOT USED IN THIS MODULE)

RBS(SHAPE) TYPE ONE DATA TABLE CONTAINING FUNCTIONS OF SPAN AND CHORD OF THE BLADE. FIRST DIMENSION IS SPANWISE STATIONS, RE B. SECOND DIMENSION IS CHORDWISE STATIONS, IN RADIANS. THIRD DIMENSION IS ORDERED POSITION WITH THE FOLLOWING SETS OF DATA :
(1) BLADE SURFACE ABSCISSAS, RE B (NOT USED IN THIS MODULE)
(2) BLADE SURFACE ORDINATES, RE B (NOT USED IN THIS MODULE)
(3) BLADE SURFACE ELLIPTIC RADIAL COORDINATES, IN RADIANS

RBATMP(LIFT) FOUR-DIMENSIONAL TYPE ONE DATA TABLE OF COEFFICIENTS. FIRST DIMENSION IS SPANWISE STATIONS, RE B. SECOND DIMENSION IS ANGLE OF ATTACK, IN DEGREES.
THIRD DIMENSION IS MACH NUMBER. FOURTH DIMENSION IS ORDERED POSITION IN THE FOLLOWING ORDER :
(1) SECTION LIFT COEFFICIENTS
(2) VALUE FOR STAGNATION POINT CHORDWISE STATION, RADIANS

RBA(AEROCTR) TWO-DIMENSIONAL TYPE ONE DATA TABLE CONTAINING THE COORDINATES OF THE AERODYNAMIC CENTER. FIRST DIMENSION IS SPANWISE STATION, RE B. SECOND DIMENSION IS ORDERED POSITION IN THE FOLLOWING ORDER :
(1) ABSCISSA OF AERODYNAMIC CENTER, RE B
(2) ORDINATE OF AERODYNAMIC CENTER, RE B

RBA(MOMENT) THREE-DIMENSIONAL TYPE ONE DATA TABLE CONTAINING THE SECTION MOMENT COEFFICIENT ABOUT THE AERODYNAMIC CENTER. FIRST DIMENSION IS SPANWISE STATION, RE B. SECOND DIMENSION IS ANGLE OF ATTACK, IN DEGREES. THIRD DIMENSION IS MACH NUMBER.

RBA(YYYNNN) THREE-DIMENSIONAL TYPE ONE DATA TABLE CONTAINING THE LOCAL PRESSURE COEFFICIENTS. THE THREE DIMENSIONS ARE: (1) SPANWISE STATION, RE B, (2) CHORDWISE STATION, IN RADIANS, AND (3) ANGLE OF ATTACK, IN DEGREES.
THERE WILL BE ONE TABLE FOR EACH MACH NUMBER, WHERE NNN IS THE NUMBER OF THE MACH NUMBER THAT IS ASSOCIATED WITH THE TABLE AND YYY IS TAKEN FROM USER PARAMETER RBAPRES.

RBA(INCOMP) THREE-DIMENSIONAL TYPE ONE DATA TABLE CONTAINING THE INCOMPRESSIBLE LOCAL PRESSURE COEFFICIENTS (SAME DIMENSIONS AS RBA(YYYNNN))

ERRORS

NON-FATAL

1. INSUFFICIENT LOCAL DYNAMIC STORAGE.
2. SPECIFIED UNIT MEMBER NOT AVAILABLE.
3. UNABLE TO INTERPOLATE SPECIFIED TABLE FOR GIVEN INDEPENDENT VARIABLE VALUES.
4. UNABLE TO BUILD SPECIFIED OUTPUT TABLE.
5. PARAMETER VALUE OUT OF RANGE. DEFAULT VALUE WILL BE USED.
6. ARRAY OF INDEPENDENT VARIABLE VALUES IN SUBPROGRAM SPLS NOT IN INCREASING ORDER.
7. INPUT TO SUBPROGRAM SPLS IS DEFINED INCORRECTLY.
8. MEMBER MANAGER ERROR OCCURRED ON SPECIFIED UNIT MEMBER.

FATAL - NONE

INPUT DATA SIZE RESTRICTIONS	MAXIMUM NUMBER OF ENTRIES
NUMBER OF AIRFOILS ON BLADE	10
NUMBER OF CHORDWISE STATIONS	25
NUMBER OF ANGLES OF ATTACK	10
NUMBER OF MACH NUMBERS	10

LDS REQUIREMENTS

(Maximum Allocation of LDS - 5190)

LDS REQUIREMENTS

LENGTH = (NXI2 * 20) + NXI1 * (12 + 2 * NXI2 + 4 * NALPHA * NMACH + 2 * NALPHA * NXI2) + NMACH + 3 * NALPHA

WHERE

- NXI1 = NUMBER OF SPANWISE STATIONS
- NXI2 = NUMBER OF CHORDWISE STATIONS
- NALPHA = NUMBER OF ANGLES OF ATTACK
- NMACH = NUMBER OF MACH NUMBERS

GDS REQUIREMENTS

(Maximum Allocation of GDS - 3000)

SUFFICIENT ALLOCATION FOR THE FOLLOWING TABLES :

- RBS(SHAPE)
- RBS(SPAN)
- SCRATCH(ERROR)
- SCRATCH(PSI)

Blade Section Boundary Layer Module (BLM)

PURPOSE - TO COMPUTE THE SKIN FRICTION COEFFICIENTS, BOUNDARY LAYER THICKNESSES AT THE TRAILING EDGE, AND SECTION DRAG COEFFICIENTS FOR A ROTATING BLADE

AUTHOR - DSW(L03/02/00)

INPUT

USER PARAMETERS

		DEFAULT
RINF	REYNOLD'S NUMBER BASED ON BLADE LENGTH, SOUND SPEED, AND KINEMATIC VISCOSITY (RS)	2.3296E7
HSEP	VALUE OF TURBULENT MODIFIED SHAPE FACTOR AT SEPARATION (RS)	0.761
IPRINT	PRINT FLAG (I) =0, NO PRINT DESIRED =1, INPUT PRINT ONLY =2, OUTPUT PRINT ONLY =3, BOTH INPUT AND OUTPUT PRINT	3
BLMFRCT	THREE LETTER CODE, YYY, USED TO FORM TABLE NAME BLM(YYYNNN), WHERE NNN IS THE INDEX INTO THE MACH NUMBER ARRAY	FRC

UNIT MEMBERS (SEE DESCRIPTION UNDER DATA BASE STRUCTURES)

BLADE	(TRIP)	(OPTIONAL)
RBA	(ALPHA)	
RBA	(MACH)	
RBA	(INCOMP)	
RBATMP	(LIFT)	
RBS	(SHAPE)	
RBS	(SPAN)	
RBS	(SLPCRD)	
RBS	(SLPSPN)	

OUTPUT

USER PARAMETERS - NONE

SYSTEM PARAMETERS

NERR SYSTEM ERROR FLAG
=.TRUE., ERROR OCCURRED DURING MODULE EXECUTION
=.FALSE., NO ERRORS OCCURRED

UNIT MEMBERS (SEE DESCRIPTION UNDER DATA BASE STRUCTURES)

BLM	(LIFTDRAG)
BLM	(THICK)
BLM	(YYYNNN)
SCRATCH	(METRIC)

DATA BASE STRUCTURES

BLADE	(TRIP)	MULTI-RECORD UNIT MEMBER IN 2RS FORMAT CONTAINING THE CHORDWISE COORDINATE, IN RADIANS, OF THE UPPER AND LOWER SURFACE BOUNDARY LAYER TRIP LOCATIONS. EACH RECORD CORRESPONDS TO THE SPANWISE STATION INPUT TO THE RBS MODULE ON UNIT MEMBER GEOM(BLADE).
BLM	(LIFTDRAG)	FOUR DIMENSIONAL TYPE 1 TABLE OF FORCE COEFFICIENTS. FIRST DIMENSION IS SPANWISE STATIONS, RE B. SECOND DIMENSION IS ANGLE OF ATTACK, DEGREES. THIRD DIMENSION IS MACH NUMBER. FOURTH DIMENSION IS ORDERED POSITION IN THE FOLLOWING ORDER: (1) SECTION LIFT COEFFICIENTS (2) SECTION DRAG COEFFICIENTS
BLM	(THICK)	FOUR DIMENSIONAL TYPE 1 TABLE OF BOUNDARY LAYER THICKNESSES AT THE TRAILING EDGE, RE C. FIRST DIMENSION IS SPANWISE STATIONS, RE B. SECOND DIMENSION IS ORDERED POSITION IN THE FOLLOWING ORDER: (1) UPPER SURFACE DISPLACEMENT THICKNESS (2) LOWER SURFACE DISPLACEMENT THICKNESS (3) UPPER SURFACE MOMENTUM THICKNESS (4) LOWER SURFACE MOMENTUM THICKNESS THIRD DIMENSION IS ANGLE OF ATTACK, DEGREES. FOURTH DIMENSION IS MACH NUMBER.
BLM	(YYYNNN)	THREE DIMENSIONAL TYPE 1 TABLE OF SKIN FRICTION COEFFICIENTS. THE THREE DIMENSIONS ARE: (1) SPANWISE STATIONS, RE B, (2) CHORDWISE STATIONS, RADIANS, AND (3) ANGLE OF ATTACK, DEGREES. THERE IS ONE TABLE FOR EACH MACH NUMBER VALUE ON RBA(MACH).
RBA	(ALPHA)	ONE RECORD UNIT MEMBER IN RS FORMAT CONTAINING THE VALUES OF ANGLE OF ATTACK, DEGREES.
RBA	(MACH)	ONE RECORD UNIT MEMBER IN RS FORMAT CONTAINING THE VALUES OF MACH NUMBER
RBA	(INCOMP)	THREE DIMENSIONAL TYPE 1 TABLE OF LOCAL INCOMPRESSIBLE PRESSURE COEFFICIENTS. THE THREE DIMENSIONS ARE: (1) SPANWISE STATIONS, RE B, (2) CHORDWISE STATIONS, RADIANS, AND (3) ANGLE OF ATTACK, DEGREES.

RBATMP	(LIFT)	FOUR DIMENSIONAL TYPE 1 TABLE OF COEFFICIENTS. FIRST DIMENSION IS SPANWISE STATIONS, RE B, SECOND DIMENSION IS ANGLE OF ATTACK, DEGREES. THIRD DIMENSION IS MACH NUMBER. FOURTH DIMENSION IS ORDERED POSITION IN THE FOLLOWING ORDER: (1) SECTION LIFT COEFFICIENTS (2) LEADING EDGE STAGNATION POINT CHORDWISE STATION, RADIANS
RBS	(SHAPE)	THREE DIMENSIONAL TYPE 1 TABLE OF BLADE SHAPE DATA. FIRST DIMENSION IS SPANWISE STATIONS, RE B. SECOND DIMENSION IS CHORDWISE STATIONS, RADIANS. THIRD DIMENSION IS ORDERED POSITION IN THE FOLLOWING ORDER: (1) BLADE SURFACE ABSCISSA, RE B (2) BLADE SURFACE ORDINATE, RE B (3) BLADE SURFACE ELLIPTIC RADIAL COORDINATES, RADIANS
RBS	(SPAN)	TWO DIMENSIONAL TYPE 1 TABLE OF SPANWISE DATA. FIRST DIMENSION IS SPANWISE STATIONS, RE B. SECOND DIMENSION IS ORDERED POSITION IN THE FOLLOWING ORDER: (1) LEADING EDGE ABSCISSAS, RE B (2) LEADING EDGE ORDINATES, RE B (3) JOUKOWSKI TRANSFORMATION PARAMETERS, RE B (4) CHORD LENGTHS, RE B (5) ELLIPTIC SYSTEM ORIGIN ABSCISSAS, RE B (6) ELLIPTIC SYSTEM ORIGIN ORDINATES, RE B (7) BLADE TWIST ANGLE MEASURED POSITIVE CLOCKWISE LOOKING OUTWARD, RADIANS (8) BLADE SECTION AREAS, RE B 2
RBS	(SLPCRD)	MULTI-RECORD UNIT MEMBER IN RS FORMAT CONTAINING THE SLOPE MATRIX ASSOCIATED WITH THE CHORDWISE DIRECTION (EACH RECORD CORRESPONDS TO A COLUMN IN THE MATRIX)
RBS	(SLPSPN)	MULTI-RECORD UNIT MEMBER IN RS FORMAT CONTAINING THE SLOPE MATRIX ASSOCIATED WITH THE SPANWISE DIRECTION (EACH RECORD CORRESPONDS TO A COLUMN IN THE MATRIX)
SCRATCH	(METRIC)	TWO DIMENSIONAL TYPE 1 TABLE OF THE METRIC COEFFICIENT FOR ARC LENGTH. THE DIMENSIONS ARE (1) SPANWISE STATIONS, RE B AND (2) CHORDWISE STATIONS, RADIANS.

ERRORS

NON-FATAL

1. INSUFFICIENT LOCAL DYNAMIC STORAGE.
2. MEMBER MANAGER ERROR OCCURRED ON SPECIFIED UNIT MEMBER.
3. REQUIRED UNIT MEMBER NOT AVAILABLE.
4. UNABLE TO INTERPOLATE SPECIFIED TABLE.
5. INCORRECT NUMBER OF VALUES ON BLADE(TRIP).
6. UNABLE TO BUILD SPECIFIED TABLE.
7. ERROR OCCURRED IN DIFFERENTIAL EQUATION SOLVER.
8. BOUNDARY LAYER SEPARATION OCCURRED.
9. VELOCITY WITHIN BOUNDARY LAYER EXCEEDS LOCAL VELOCITY OUTSIDE BOUNDARY LAYER.

FATAL - NONE

INPUT DATA SIZE RESTRICTIONS

	MAXIMUM NUMBER OF ENTRIES
NUMBER OF AIRFOILS ON BLADE	10
NUMBER OF CHORDWISE STATIONS	25
NUMBER OF ANGLES OF ATTACK	10
NUMBER OF MACH NUMBERS	10

LDS REQUIREMENTS

(Maximum Allocation of LDS - 5190)

```
LENGTH = NXI1 * NALPHA * NMACH * 6 + NXI1 * NXI2 * NALPHA
       + NXI1 * NXI2 + NXI1 * 5 + NXI2 + NALPHA + NMACH
       + (NXI2+1) * ( 5 + 2 * (NXI2+1) )
```

WHERE

- NXI1 = NUMBER OF SPANWISE STATIONS
- NXI2 = NUMBER OF CHORDWISE STATIONS
- NALPHA = NUMBER OF ANGLES OF ATTACK
- NMACH = NUMBER OF MACH NUMBERS

GDS REQUIREMENTS

(Maximum Allocation of GDS - 3000)

SUFFICIENT ALLOCATION FOR THE FOLLOWING TABLES

```
RBA     (INCOMP  )
RBATMP (LIFT    )
RBS     (SHAPE   )
RBS     (SPAN    )
SCRATCH(METRIC  )
```

```
PURPOSE - TO FORMULATE A FUNCTIONAL REPRESENTATION OF THE BLADE
          SURFACE SUITABLE FOR AEROACOUSTIC AND AERODYNAMIC
          CALCULATIONS

AUTHOR  - CBF(L03/02/00)

INPUT
  USER PARAMETERS                                          DEFAULT
    IUNITS - INPUT UNITS FLAG (CHARACTER)                    SI
             SI,      INPUT IN SI UNITS
             ENGLISH, INPUT IN ENGLISH UNITS
    IPRINT - PRINT FLAG (INTEGER)                            3
             =0, NO PRINT DESIRED
             =1, INPUT PRINT ONLY
             =2, OUTPUT PRINT ONLY
             =3, BOTH INPUT AND OUTPUT PRINT
    ZSLOPE - BOUNDARY CONDITION OPTION (LOGICAL)             F
             =.TRUE.,  ZERO SLOPE
             =.FALSE., ZERO CURVATURE
    WEIGHT - WEIGHTING OPTION (LOGICAL)                      F
             =.TRUE.,  WEIGHTING FACTORS ARE INCLUDED
             =.FALSE., WEIGHTING FACTORS ARE NOT
                       INCLUDED

  DATA BASE UNIT MEMBERS
  (DESCRIBED UNDER DATA BASE STRUCTURES)
   GRID(XI2)
   GEOM (IBLADE)

OUTPUT
  USER PARAMETERS - NONE

  SYSTEM PARAMETERS
   NERR   - SYSTEM ERROR FLAG
            =.TRUE.,  AN ERROR WAS ENCOUNTERED DURING
                      MODULE EXECUTION
            =.FALSE., NO ERRORS OCCURRED

  DATA BASE UNIT MEMBERS
  (DESCRIBED UNDER DATA BASE STRUCTURES)
   IBS(SHAPE)
   IBS(SPAN)
   IBS(BLD)
   IBS(SLPSPN)
   IBS(SLPCRD)
   IBS(THK)
   GRID(XI1)
```

DATA BASE STRUCTURES

GRID(XI1) ONE-RECORD OUTPUT MEMBER IN REAL FORMAT CONTAINING THE ARRAY OF SPANWISE STATIONS, RE B. SPANWISE STATIONS ARE THE SPANWISE COORDINATES FROM UNIT MEMBER, GEOM(IBLADE)

GRID(XI2) ONE-RECORD INPUT MEMBER IN REAL FORMAT CONTAINING THE ARRAY OF CHORDWISE STATIONS BETWEEN 0 AND 1, RADIANS

GEOM(IBLADE) MULTI-RECORD UNFORMATTED MEMBER CONTAINING SPANWISE FUNCTIONS OF THE BLADE IF WEIGHTING IS INCLUDED, (IE., IF PARAMETER WEIGHT IS .TRUE.) A WEIGHTING FACTOR SHOULD BE INCLUDED AFTER EACH (X,Y) PAIR. IF NO WEIGHTING FACTOR IS INCLUDED FOR A GIVEN (X,Y) PAIR, IT IS ASSUMED TO BE ONE.

RECORD	FORMAT	DESCRIPTION
1	I	NUMBER OF AIRFOIL SECTIONS ON THE BLADE
2	I	A SET OF NUMBERS OF DIFFERENT CROSS SECTIONS (IE. , IF RECORD 1 IS 8 AND RECORD 2 IS 1, 1, 1, 2, 3. THIS MEANS THAT THERE ARE 8 DIFFERENT SPANWISE STATIONS. HOWEVER, AT THE 4TH AND 5TH SPANWISE STATIONS, THE CROSS SECTIONS HAVE THE SAME (X,Y) COORDINATES AND THE SAME FOR THE 6TH, 7TH, AND 8TH SPANWISE STATIONS.)
3	6RS,2I	SPANWISE COORDINATE, LEADING EDGE ABSCISSA, LEADING EDGE ORDINATE, CHORD LENGTH, LEADING EDGE RADIUS (RE CHORD LENGTH), BLADE TWIST ANGLE MEASURED POSITIVE CLOCKWISE LOOKING OUTBOARD, IN DEGREES, NUMBER OF (X,Y) PAIRS ON UPPER SURFACE, AND NUMBER OF (X,Y) PAIRS ON LOWER SURFACE OF FIRST AIRFOIL SECTION. FIRST FOUR WORDS ARE IN THE SAME UNITS AS GIVEN BY PARAMETER, IUNITS.
4	RS	FIRST (X,Y) PAIR ON UPPER SURFACE OF FIRST AIRFOIL SECTION (AND IF PARAMETER WEIGHT IS .TRUE., A WEIGHTING FACTOR)
5	RS	SECOND (X,Y) PAIR ON UPPER SURFACE OF FIRST AIRFOIL SECTION (AND IF PARAMETER WEIGHT IS .TRUE., A WEIGHTING FACTOR)
.	.	
.	.	
.	.	
N+3	RS	NTH (X,Y) PAIR ON UPPER SURFACE OF FIRST AIRFOIL SECTION (AND IF PARAMETER WEIGHT IS .TRUE., A WEIGHTING FACTOR)

N+4	RS	FIRST(X,Y) PAIR ON LOWER SURFACE OF FIRST AIRFOIL SECTION (AND IF PARAMETER WEIGHT IS .TRUE., A WEIGHTING FACTOR)
N+5	RS	SECOND (X,Y) PAIR ON LOWER SURFACE OF FIRST AIRFOIL SECTION (AND IF PARAMETER WEIGHT IS .TRUE., A WEIGHTING FACTOR)
.	.	
.	.	
.	.	
N+3+M	RS	MTH (X,Y) PAIR ON LOWER SURFACE OF FIRST AIRFOIL SECTION (AND IF PARAMETER WEIGHT IS .TRUE., A WEIGHTING FACTOR)
.	.	
.	.	
.	.	

(RECORDS 3 THROUGH N+3+M REPEAT FOR EACH AIRFOIL SECTION. IF THERE ARE IDENTICAL CROSS SECTIONS, THE SAME FORMAT OF RECORD THREE REPEATS FOR EACH OF THE SAME CROSS SECTIONS AND THE (X,Y) PAIRS ARE ENTERED ONLY ONCE. NOTE : M AND N MAY BE UNEQUAL AND MAY BE DIFFERENT FOR EACH SECTION.)

IBS(SHAPE) TYPE ONE DATA TABLE CONTAINING FUNCTIONS OF SPAN AND CHORD OF THE BLADE. FIRST DIMENSION IS SPANWISE STATION, RE B. SECOND DIMENSION IS CHORDWISE STATION, IN RADIANS. THIRD DIMENSION IS ORDERED POSITION WITH THE FOLLOWING THREE SETS OF DATA :

(1) BLADE SURFACE ABSCISSAS, RE B
(2) BLADE SURFACE ORDINATES, RE B
(3) BLADE SURFACE ELLIPTIC RADIAL COORDINATES, IN RADIANS

IBS(SPAN) TYPE ONE DATA TABLE CONTAINING FUNCTIONS OF SPAN. FIRST DIMENSION IS SPANWISE STATIONS. SPANWISE STATIONS ARE THE SPANWISE COORDINATES, NORMALIZED.
SECOND DIMENSION IS ORDERED POSITION IN THE FOLLOWING ORDER :

(1) LEADING EDGE ABSCISSAS, RE B
(2) LEADING EDGE ORDINATES, RE B
(3) JOUKOWSKI TRANSFORMATION PARAMETERS, RE B
(4) CHORD LENGTHS, RE B
(5) ELLIPTIC SYSTEM ORIGIN ABSCISSAS, RE B
(6) ELLIPTIC SYSTEM ORIGIN ORDINATES, RE B
(7) BLADE ELLIPTIC AXIS TWIST ANGLE MEASURED POSITIVE CLOCKWISE LOOKING OUTWARD, IN RADIANS
(8) BLADE CROSS-SECTIONAL AREA, RE B 2

IBS(BLD) ONE RECORD MEMBER IN 3RS FORMAT CONTAINING THE BLADE VOLUME, RE B 3, THE BLADE ASPECT RATIO, AND THE BLADE ACTIVITY FACTOR

IBS(SLPSPN) MULTI-RECORD MEMBER IN RS FORMAT CONTAINING THE SLOPE MATRIX IN THE SPANWISE DIRECTION. A RECORD ON THE MEMBER REPRESENTS A COLUMN IN THE MATRIX. THE NUMBER OF RECORDS EQUALS THE NUMBER OF WORDS PER RECORD ON THE MEMBER. (IE, A SQUARE MATRIX)

IBS(SLPCRD) MULTI-RECORD MEMBER IN RS FORMAT CONTAINING THE SLOPE MATRIX IN THE CHORDWISE DIRECTION. A RECORD ON THE MEMBER REPRESENTS A COLUMN IN THE MATRIX. THE NUMBER OF RECORDS EQUALS THE NUMBER OF WORDS PER RECORD ON THE MEMBER. (IE, A SQUARE MATRIX)

IBS(THK) TYPE ONE DATA TABLE CONTAINING FUNCTIONS OF SPAN. FIRST DIMENSION IS SPANWISE STATIONS. SECOND DIMENSION IS THE FOLLOWING ORDER:
(1) MAXIMUM THICKNESS LOCATIONS, RE CHORD LENGTH
(2) TRAILING EDGE THICKNESS, RE CHORD LENGTH

ERRORS

NON-FATAL

1. INSUFFICIENT LOCAL DYNAMIC STORAGE.
2. UNABLE TO BUILD SPECIFIED OUTPUT TABLE.
3. MEMBER MANAGER ERROR OCCURRED ON SPECIFIED UNIT MEMBER.
4. NUMBER OF SPANWISE OR CHORDWISE STATIONS IS .LT. 3.
5. SPECIFIED UNIT MEMBER NOT AVAILABLE.
6. ARRAY OF INDEPENDENT VARIABLE VALUES NOT IN INCREASING ORDER.
7. SINGULAR MATRIX WHEN SOLVING SYSTEM OF REAL LINEAR EQUATIONS IN SUBPROGRAM GELIM.
8. X OR Y VALUE ON UNIT MEMBER GEOM(BLADE) IS .LT. 0 OR .GT. 1. (WARNING ONLY)
9. DETERMINANT OF MATRIX IS ZERO.
10. NUMBER OF CHORDWISE STATIONS EXCEEDS THE NUMBER OF (X,Y) PAIRS ON A PARTICULAR AIRFOIL SECTION.
11. (X,Y) PAIRS, TRANSFORMED TO THE ELLIPTIC COORDINATE SYSTEM DO NO LIE BETWEEN THE FIRST AND LAST CHORDWISE STATIONS.

FATAL - NONE

INPUT DATA SIZE RESTRICTIONS

	MAXIMUM NUMBER
NUMBER OF AIRFOILS ON BLADE	10
NUMBER OF CHORDWISE STATIONS	25

LDS REQUIREMENTS

(Maximum Allocation of LDS - 5190)

LENGTH = (NCROSS * 25) + (NPAIR * 3) + (NXY * (2))
+ NWT + (NCROSS ** 2) * 2) + ((NXI2 ** 2) * 3)
+ (NXI2 * NCROSS * 4) + NXI2 * (13 + LARGE)

WHERE

NCROSS = NUMBER OF AIRFOILS ON THE BLADE (IE., NUMBER OF SPANWISE STATIONS)

NXI2 = NUMBER OF CHORDWISE STATIONS

NPAIR = TOTAL NUMBER OF (X,Y) PAIRS ON THE BLADE SURFACE

NXY = NPAIR 3, IF WEIGHTING FACTORS ARE INCLUDED
NPAIR 2, IF NO WEIGHTING IS USED

NWT = NPAIR, IF WEIGHTING FACTORS ARE INCLUDED
0, IF NO WEIGHTING IS USED

LARGE = MAXIMUM NUMBER OF (X,Y) PAIRS ON AN AIRFOIL SECTION

GDS REQUIREMENTS

(Maximum Allocation of GDS - 3000)

SUFFICIENT ALLOCATION FOR THE FOLLOWING TABLES :

IBS(SHAPE)

IBS(SPAN)

Improved Blade Section Aerodynamics Module (IBA)

PURPOSE - COMPUTES THE PRESSURE FORCES ACTING ON THE UPPER AND LOWER SURFACES OF A TWO-DIMENSIONAL AIRFOIL FOR SPECIFIED ANGLE OF ATTACK AND MACH NUMBER VALUES

AUTHOR - CBF(L03/02/00)

INPUT		DEFAULT
USER PARAMETERS		
EPSLON	- CRITERION FOR STOPPING ITERATION (REAL)	.001
NORDER	- ORDER OF FOURIER SERIES (INTEGER) (VALUE SHOULD BE IN RANGE OF 1 .LE. NORDER .LE. (NXI2-1)/4, WHERE NXI2 = NUMBER OF VALUES ON GRID(XI2))	5
IPRINT	- PRINT FLAG (INTEGER) =0, NO PRINT DESIRED =1, INPUT PRINT ONLY =2, OUTPUT PRINT ONLY =3, BOTH INPUT AND OUTPUT PRINT	3
IBAPRES	- THREE-LETTER CODE, YYY, USED TO FORM TABLE UNIT MEMBER NAME IBA(YYYNNN) WHERE NNN IS THE INDEX INTO THE MACH NUMBER ARRAY AND IBA(YYYNNN) REPRESENTS THE TABLES OF LOCAL PRESSURE COEFFICIENTS (CHARACTER)	PRS
ICL	- COMPRESSIBILITY CORRECTION OF LIFT COEFFICIENTS FLAG (INTEGER) =0, NO CORRECTION =1, GLAUERT'S CORRECTION	0
ICP	- COMPRESSIBILITY CORRECTION OF PRESSURE COEFFICIENTS FLAG (INTEGER) =0, NO CORRECTION =1, GLAUERT'S CORRECTION =2, KARMAN TSIEN CORRECTION	2
GAMMA	- RATIO OF SPECIFIC HEAT (REAL)	1.4
RINF	- REYNOLDS NUMBER BASED ON BLADE LENGTH, SPEED OF SOUND, KINEMATIC VISCOSITY (REAL)	2.3296E7

DATA BASE UNIT MEMBERS
(DESCRIBED UNDER DATA BASE STRUCTURES)

- IBA(MACH)
- IBA(ALPHA)
- IBS(SPAN)
- IBS(SHAPE)
- IBS(SLPSPN)
- IBS(SLPCRD)

```
OUTPUT
  SYSTEM PARAMETERS
    NERR   - SYSTEM ERROR FLAG
             =.TRUE.,  AN ERROR WAS ENCOUNTERED DURING MODULE
                       EXECUTION
             =.FALSE., NO ERRORS OCCURRED

  DATA BASE UNIT MEMBERS
  (DESCRIBED UNDER DATA BASE STRUCTURES)
    IBS(STAG)
    IBA(LIFT)
    IBA(AEROCTR)
    IBA(MOMENT)
    IBA(YYYNNN)  NOTE : MEMBER NAME YYYNNN IS FORMED FROM USER
                 PARAMETER IBAPRES AND FROM THE INDEX INTO THE
                 MACH NUMBER ARRAY FOR THE MACH NUMBER
                 ASSOCIATED WITH THE TABLE.  CONTAINS THE LOCAL
                 PRESSURE COEFFICIENTS.

DATA BASE STRUCTURES (NOTE: B REFERS TO BLADE LENGTH)
  IBA(MACH)      ONE-RECORD INPUT MEMBER IN  RS FORMAT
                 CONTAINING RANGE OF MACH NUMBERS
  IBA(ALPHA)     ONE-RECORD INPUT MEMBER IN  RS FORMAT
                 CONTAINING THE RANGE OF ANGLES OF ATTACK, IN
                 DEGREES
  IBA(LIFT)      THREE-DIMENSIONAL TYPE ONE DATA TABLE OF COEF-
                 FICIENTS. FIRST DIMENSION IS SPANWISE STATIONS,
                 DEGREES. THIRD DIMENSION IS MACH NUMBER
  IBA(STAG)      THREE-DIMENSIONAL TYPE ONE DATA TABLE OF
                 STAGNATION POINT CHORDWISE STATIONS, RADIANS.
                 FIRST DIMENSION IS SPANWISE STATIONS, RE B.
                 SECOND DIMENSION IS ANGLE OF ATTACK, IN DEGREES.
                 THIRD DIMENSION IS MACH NUMBER.
  IBS(SLPSPN)    MULTI-RECORD INPUT MEMBER IN  RS FORMAT
                 CONTAINING THE SLOPE MATRIX ASSOCIATED
                 WITH THE SPANWISE DIRECTION ( EACH RECORD
                 CORRESPONDS TO A COLUMN IN THE MATRIX )
  IBS(SLPCRD)    MULTI-RECORD INPUT MEMBER IN  RS FORMAT
                 CONTAINING THE SLOPE MATRIX ASSOCIATED
                 WITH THE CHORDWISE DIRECTION ( EACH RECORD
                 CORRESPONDS TO A COLUMN IN THE MATRIX )
  IBS(SPAN)      TYPE ONE DATA TABLE CONTAINING FUNCTIONS OF
                 SPANWISE DIRECTION OF THE BLADE.  FIRST
                 DIMENSION IS SPANWISE STATIONS, RE B.  SECOND
                 DIMENSION IS ORDERED POSITION IN THE
                 FOLLOWING ORDER:
                 (1) LEADING EDGE ABSCISSAS, RE B
                 (2) LEADING EDGE ORDINATES, RE B
                 (3) JOUKOWSKI TRANSFORMATION PARAMETERS, RE B
                 (4) CHORD LENGTHS, RE B
                 (5) ELLIPTIC SYSTEM ORIGIN ABSCISSAS, RE B
                     ( NOT USED IN THIS MODULE )
```

(6) ELLIPTIC SYSTEM ORIGIN ORDINATES, RE B
(NOT USED IN THIS MODULE)
(7) BLADE TWIST ANGLE MEASURED POSITIVE CLOCKWISE LOOKING OUTBOARD, IN RADIANS
(8) BLADE SECTION AREA, RE B 2
(NOT USED IN THIS MODULE)

IBS(SHAPE) TYPE ONE DATA TABLE CONTAINING FUNCTIONS OF SPAN AND CHORD OF THE BLADE. FIRST DIMENSION IS SPANWISE STATIONS, RE B. SECOND DIMENSION IS CHORDWISE STATIONS, IN RADIANS. THIRD DIMENSION IS ORDERED POSITION WITH THE FOLLOWING SETS OF DATA :
(1) BLADE SURFACE ABSCISSAS, RE B (NOT USED IN THIS MODULE)
(2) BLADE SURFACE ORDINATES, RE B (NOT USED IN THIS MODULE)
(3) BLADE SURFACE ELLIPTIC RADIAL COORDINATES, IN RADIANS

IBA(AEROCTR) TWO-DIMENSIONAL TYPE ONE DATA TABLE CONTAINING THE COORDINATES OF THE AERODYNAMIC CENTER. FIRST DIMENSION IS SPANWISE STATION, RE B. SECOND DIMENSION IS ORDERED POSITION IN THE FOLLOWING ORDER :
(1) ABSCISSA OF AERODYNAMIC CENTER, RE B
(2) ORDINATE OF AERODYNAMIC CENTER, RE B

IBA(MOMENT) THREE-DIMENSIONAL TYPE ONE DATA TABLE CONTAINING THE SECTION MOMENT COEFFICIENT ABOUT THE AERODYNAMIC CENTER. FIRST DIMENSION IS SPANWISE STATION, RE B. SECOND DIMENSION IS ANGLE OF ATTACK, IN DEGREES. THIRD DIMENSION IS MACH NUMBER.

IBA(YYYNNN) THREE-DIMENSIONAL TYPE ONE DATA TABLE CONTAINING THE LOCAL PRESSURE COEFFICIENTS. THE THREE DIMENSIONS ARE: (1) SPANWISE STATION, RE B, (2) CHORDWISE STATION, IN RADIANS, AND (3) ANGLE OF ATTACK, IN DEGREES.
THERE WILL BE ONE TABLE FOR EACH MACH NUMBER, WHERE NNN IS THE NUMBER OF THE MACH NUMBER THAT IS ASSOCIATED WITH THE TABLE AND YYY IS TAKEN FROM USER PARAMETER IBAPRES.

IBA(INCOMP) THREE-DIMENSIONAL TYPE ONE DATA TABLE CONTAINING THE INCOMPRESSIBLE LOCAL PRESSURE COEFFICIENTS (SAME DIMENSIONS AS IBA(YYYNNN))

ERRORS

NON-FATAL

1. INSUFFICIENT LOCAL DYNAMIC STORAGE.
2. SPECIFIED UNIT MEMBER NOT AVAILABLE.
3. UNABLE TO INTERPOLATE SPECIFIED TABLE FOR GIVEN INDEPENDENT VARIABLE VALUES.
4. UNABLE TO BUILD SPECIFIED OUTPUT TABLE.
5. PARAMETER VALUE OUT OF RANGE. DEFAULT VALUE WILL BE USED.
6. ARRAY OF INDEPENDENT VARIABLE VALUES IN SUBPROGRAM SPLS NOT IN INCREASING ORDER.
7. INPUT TO SUBPROGRAM SPLS IS DEFINED INCORRECTLY.
8. MEMBER MANAGER ERROR OCCURRED ON SPECIFIED UNIT MEMBER.

FATAL - NONE

INPUT DATA SIZE RESTRICTIONS

	MAXIMUM NUMBER OF ENTRIES
NUMBER OF AIRFOILS ON BLADE	10
NUMBER OF CHORDWISE STATIONS	25
NUMBER OF ANGLES OF ATTACK	10
NUMBER OF MACH NUMBERS	10

LDS REQUIREMENTS

(Maximum Allocation of LDS - 5190)

LDS REQUIREMENTS

LENGTH = (NXI1 * 13) + (NXI2 * 5) + (NORDER * NXI2 * 15) + (NXI1 * NXI2 * 2 * (1 + NALPHA)) - (10 * NORDER) + (3 * NXI1 * NALPH * NMACH) + (2 * NALPHA) + 92

WHERE

NXI1 = NUMBER OF SPANWISE STATIONS
NXI2 = NUMBER OF CHORDWISE STATIONS
NALPHA = NUMBER OF ANGLES OF ATTACK
NMACH = NUMBER OF MACH NUMBERS

GDS REQUIREMENTS

(Maximum Allocation of GDS - 3000)

SUFFICIENT ALLOCATION FOR THE FOLLOWING TABLES :

IBS(SHAPE)
IBS(SPAN)
SCRATCH(ERROR)
SCRATCH(PSI)

Improved Blade Section Boundary Layer Module (IBL)

PURPOSE - TO COMPUTE THE SKIN FRICTION COEFFICIENTS, BOUNDARY LAYER THICKNESSES AT THE TRAILING EDGE, AND SECTION DRAG COEFFICIENTS FOR A ROTATING BLADE

AUTHOR - DSW(L03/02/00)

INPUT

USER PARAMETERS

		DEFAULT
RINF	REYNOLD'S NUMBER BASED ON BLADE LENGTH, SOUND SPEED, AND KINEMATIC VISCOSITY (REAL)	2.3296E7
HSEP	VALUE OF TURBULENT MODIFIED SHAPE FACTOR AT SEPARATION (REAL)	0.761
IPRINT	PRINT FLAG (INTEGER) =0, NO PRINT DESIRED =1, INPUT PRINT ONLY =2, OUTPUT PRINT ONLY =3, BOTH INPUT AND OUTPUT PRINT	3
IBLFRCT	THREE LETTER CODE, YYY, USED TO FORM TABLE NAME IBL(YYYNNN), WHERE NNN IS THE INDEX INTO THE MACH NUMBER ARRAY (CHARACTER)	FRC
FLAT	MODEL OPTION FLAG = .TRUE., ZERO PRESSURE GRADIENT FLAT PLATE MODEL USED = .FALSE., FULL LAMINAR AND TURBULENT FLOW MODEL USED	F

UNIT MEMBERS (SEE DESCRIPTION UNDER DATA BASE STRUCTURES)

BLADE	(TRIP)	(OPTIONAL)
IBA	(ALPHA)	
IBA	(MACH)	
IBA	(INCOMP)	
IBA	(STAG)	
IBS	(SHAPE)	
IBS	(SPAN)	
IBS	(SLPCRD)	
IBS	(SLPSPN)	

OUTPUT

USER PARAMETERS - NONE

SYSTEM PARAMETERS

NERR	SYSTEM ERROR FLAG =.TRUE., ERROR OCCURRED DURING MODULE EXECUTION =.FALSE., NO ERRORS OCCURRED

UNIT MEMBERS (SEE DESCRIPTION UNDER DATA BASE STRUCTURES)

IBL (DRAG)
IBL (THICK)
IBL (YYYNNN)
SCRATCH(METRIC)

DATA BASE STRUCTURES

BLADE (TRIP)	MULTI-RECORD UNIT MEMBER IN 2RS FORMAT CONTAINING THE CHORDWISE COORDINATE, IN RADIANS, OF THE UPPER AND LOWER SURFACE BOUNDARY LAYER TRIP LOCATIONS. EACH RECORD CORRESPONDS TO THE SPANWISE STATION INPUT TO THE IBS MODULE ON UNIT MEMBER GEOM(BLADE).
IBL (DRAG)	THREE DIMENSIONAL TYPE 1 TABLE OF DRAG COEFFICIENTS. FIRST DIMENSION IS SPANWISE STATIONS, RE B. SECOND DIMENSION IS ANGLE OF ATTACK, DEGREES. THIRD DIMENSION IS MACH NUMBER.
IBL (THICK)	FOUR DIMENSIONAL TYPE 1 TABLE OF BOUNDARY LAYER THICKNESSES AT THE TRAILING EDGE, RE C. FIRST DIMENSION IS SPANWISE STATIONS, RE B. SECOND DIMENSION IS ORDERED POSITION IN THE FOLLOWING ORDER: (1) UPPER SURFACE DISPLACEMENT THICKNESS (2) LOWER SURFACE DISPLACEMENT THICKNESS (3) UPPER SURFACE MOMENTUM THICKNESS (4) LOWER SURFACE MOMENTUM THICKNESS THIRD DIMENSION IS ANGLE OF ATTACK, DEGREES. FOURTH DIMENSION IS MACH NUMBER.

IBL	(YYYNNN)	THREE DIMENSIONAL TYPE 1 TABLE OF SKIN FRICTION COEFFICIENTS. THE THREE DIMENSIONS ARE: (1) SPANWISE STATIONS, RE B, (2) CHORDWISE STATIONS, RADIANS, AND (3) ANGLE OF ATTACK, DEGREES. THERE IS ONE TABLE FOR EACH MACH NUMBER VALUE ON IBA(MACH).
IBA	(ALPHA)	ONE RECORD UNIT MEMBER IN RS FORMAT CONTAINING THE VALUES OF ANGLE OF ATTACK, DEGREES.
IBA	(MACH)	ONE RECORD UNIT MEMBER IN RS FORMAT CONTAINING THE VALUES OF MACH NUMBER
IBA	(INCOMP)	THREE DIMENSIONAL TYPE 1 TABLE OF LOCAL INCOMPRESSIBLE PRESSURE COEFFICIENTS. THE THREE DIMENSIONS ARE: (1) SPANWISE STATIONS, RE B, (2) CHORDWISE STATIONS, RADIANS, AND (3) ANGLE OF ATTACK, DEGREES.
IBS	(SHAPE)	THREE DIMENSIONAL TYPE 1 TABLE OF BLADE SHAPE DATA. FIRST DIMENSION IS SPANWISE STATIONS, RE B. SECOND DIMENSION IS CHORDWISE STATIONS, RADIANS. THIRD DIMENSION IS ORDERED POSITION IN THE FOLLOWING ORDER: (1) BLADE SURFACE ABSCISSA, RE B (2) BLADE SURFACE ORDINATE, RE B (3) BLADE SURFACE ELLIPTIC RADIAL COORDINATES, RADIANS
IBA	(STAG)	THREE-DIMENSIONAL TYPE ONE DATA TABLE OF STAGNATION POINT CHORDWISE STATIONS, RADIANS. FIRST DIMENSION IS SPANWISE STATIONS, RE B. SECOND DIMENSION IS ANGLE OF ATTACK, IN DEGREES. THIRD DIMENSION IS MACH NUMBER.
IBS	(SPAN)	TWO DIMENSIONAL TYPE 1 TABLE OF SPANWISE DATA. FIRST DIMENSION IS SPANWISE STATIONS, RE B. SECOND DIMENSION IS ORDERED POSITION IN THE FOLLOWING ORDER: (1) LEADING EDGE ABSCISSAS, RE B (2) LEADING EDGE ORDINATES, RE B (3) JOUKOWSKI TRANSFORMATION PARAMETERS, RE B (4) CHORD LENGTHS, RE B (5) ELLIPTIC SYSTEM ORIGIN ABSCISSAS, RE B (6) ELLIPTIC SYSTEM ORIGIN ORDINATES, RE B (7) BLADE TWIST ANGLE MEASURED POSITIVE CLOCKWISE LOOKING OUTWARD, RADIANS (8) BLADE SECTION AREAS, RE B 2

IBS	(SLPCRD)	MULTI-RECORD UNIT MEMBER IN RS FORMAT CONTAINING THE SLOPE MATRIX ASSOCIATED WITH THE CHORDWISE DIRECTION (EACH RECORD CORRESPONDS TO A COLUMN IN THE MATRIX)
IBS~	(SLPSPN)	MULTI-RECORD UNIT MEMBER IN RS FORMAT CONTAINING THE SLOPE MATRIX ASSOCIATED WITH THE SPANWISE DIRECTION (EACH RECORD CORRESPONDS TO A COLUMN IN THE MATRIX)
SCRATCH	(METRIC)	TWO DIMENSIONAL TYPE 1 TABLE OF THE METRIC COEFFICIENT FOR ARC LENGTH. THE DIMENSIONS ARE (1) SPANWISE STATIONS, RE B AND (2) CHORDWISE STATIONS, RADIANS.

ERRORS

NON-FATAL

1. INSUFFICIENT LOCAL DYNAMIC STORAGE.
2. MEMBER MANAGER ERROR OCCURRED ON SPECIFIED UNIT MEMBER.
3. REQUIRED UNIT MEMBER NOT AVAILABLE.
4. UNABLE TO INTERPOLATE SPECIFIED TABLE.
5. INCORRECT NUMBER OF VALUES ON BLADE(TRIP).
6. UNABLE TO BUILD SPECIFIED TABLE.
7. ERROR OCCURRED IN DIFFERENTIAL EQUATION SOLVER.
8. BOUNDARY LAYER SEPARATION OCCURRED.
9. VELOCITY WITHIN BOUNDARY LAYER EXCEEDS LOCAL VELOCITY OUTSIDE BOUNDARY LAYER.

FATAL - NONE

INPUT DATA SIZE RESTRICTIONS

	MAXIMUM NUMBER OF ENTRIES
NUMBER OF AIRFOILS ON BLADE	10
NUMBER OF CHORDWISE STATIONS	25
NUMBER OF ANGLES OF ATTACK	10
NUMBER OF MACH NUMBERS	10

LDS REQUIREMENTS

(Maximum Allocation of LDS - 5190)

$$LENGTH = NXI1 * NALPHA * NMACH * 6 + NXI1 * NXI2 * NALPHA + NXI1 * NXI2 + NXI1 * 5 + NXI2 + NALPHA + NMACH + (NXI2+1) * (5 + 2 * (NXI2+1))$$

WHERE

- NXI1 = NUMBER OF SPANWISE STATIONS
- NXI2 = NUMBER OF CHORDWISE STATIONS
- NALPHA = NUMBER OF ANGLES OF ATTACK
- NMACH = NUMBER OF MACH NUMBERS

```
GDS REQUIREMENTS
                (Maximum Allocation of GDS - 3000)

  SUFFICIENT ALLOCATION FOR THE FOLLOWING TABLES
    IBA    (INCOMP  )
    IBA    (STAG    )
    IBS    (SHAPE   )
    IBS    (SPAN    )
    SCRATCH(METRIC  )
```

Propeller Performance Module (PRP)

PURPOSE - PRP COMPUTES THE INDUCED VELOCITY FIELD, THRUST, TORQUE, AND EFFICIENCY FOR A GIVEN PROPELLER UNDER SPECIFIED OPERATING CONDITIONS

AUTHOR - WKB(L03/02/00)

INPUT

USER PARAMETERS

			DEFAULT
ALPHAP	PROPELLER ANGLE OF ATTACK (RADIANS)	RS	0.0
MACHRF	REFERENCE MACH NUMBER	RS	.8
THETAR	BLADE PITCH SETTING AT THE ROOT (RADIANS)	RS	0.
EPSILON	ERROR CRITERION FOR STOPPING THE INTERATION ON THE INDUCED ANGULAR AND AXIAL VELOCITIES	RS	.01
TIPLIM	RATIO OF SOLIDITY TO TIP RELIEF FACTOR AT BLADE TIP	RS	0.
OPTION	METHODOLOGY OPTION =0, BLADE ELEMENT-MOMENTUM THEORY =1, PRANDTL TIP RELIEF CORRECTION	I	1
UNIFORM	INFLOW OPTION =.TRUE. , UNIFORM INFLOW SPECIFIED BY MZ =.FALSE., RADIALLY VARYING INFLOW SPECIFIED BY RBF(FLOW)	L	.T.
IMPROV	IMPROV OPTION (LOGICAL) =.TRUE. , USE UNIT MEMBERS CREATED BY MODULES IBA, IBS, AND IBL =.FALSE., USE UNIT MEMBERS CREATED BY MODULES RBA, RBS, AND BLM		.F.
MZ	INFLOW MACH NUMBER (IF UNIFORM IS .TRUE.)	RS	0.5
NBLADE	NUMBER OF BLADES COMPRISING PROPELLER	I	4
IPRINT	PRINT FLAG =0, NO PRINT =1, PRINT INPUT ONLY =2, PRINT OUTPUT ONLY =3, PRINT BOTH INPUT AND OUTPUT	I	3

UNIT MEMBERS

THE FOLLOWING ARE REQUIRED FOR ALL CASES :

RBF(FLOW)	TYPE 1 TABLE OF FLOW FIELD (IF UNIFORM IS .FALSE.)
GRID(PSI)	ONE-RECORD MEMBER IN *RS FORMAT CONTAINING VALUES OF THE BLADE INPLANE ANGLES
GRID(XI1)	ONE-RECORD MEMBER CONTAINING SPANWISE STATIONS TO BE USED IN CREATING THE TABLE PRP(PERFORM)

THE FOLLOWING ARE REQUIRED FOR IMPROV = .FALSE. :

RBS(SPAN)	TYPE 1 TABLE CONTAINING THE SPANWISE FUNCTIONS OF THE BLADE
RBS(SLPSPN)	TYPE 1 TABLE CONTAINING THE SLOPE MATRIX FOR THE SPANWISE FUNCTIONS OF THE BLADE
BLM(LIFTDRAG)	TYPE 1 TABLE OF LIFT AND DRAG COEFFICIENTS

THE FOLLOWING ARE REQUIRED FOR IMPROV = .TRUE. :

IBS(SPAN)	TYPE 1 TABLE CONTAINING THE SPANWISE FUNCTIONS OF THE BLADE
IBS(SLPSPN)	TYPE 1 TABLE CONTAINING THE SLOPE MATRIX FOR THE SPANWISE FUNCTIONS OF THE BLADE
IBA(LIFT)	TYPE 1 TABLE OF SECTION LIFT COEFFICIENTS
IBL(DRAG)	TYPE 1 TABLE OF SECTION DRAG COEFFICIENTS

OUTPUT

USER PARAMETERS

CP	POWER COEFFICIENT
CT	THRUST COEFFICIENT
JRATIO	ADVANCE RATIO
PROPEFF	PROPELLER EFFICIENCY

UNIT MEMBERS

PRP(PERFORM)	TYPE 1 TABLE OF PROPELLER PERFORMANCE DATA

DATA BASE STRUCTURES

GRID(XI1) - 1 RECORD MEMBER IN *RS FORMAT CONTAINING VALUES OF THE SPANWISE STATIONS (RE BLADE LENGTH, B)

GRID(PSI) - 1 RECORD MEMBER IN *RS FORMAT CONTAINING VALUES OF THE BLADE INPLANE STATIONS BETWEEN 0 AND 1

RBS(SPAN) - TYPE 1 TABLE CONTAINING FUNCTIONS OF SPAN

OR IBS(SPAN) OF THE BLADE. FIRST DIMENSION IS SPANWISE STATIONS THAT ARE NORMALIZED BASED ON THE SPAN LENGTH, B, OF THE BLADE. THE SECOND DIMENSION IS ORDERED POSITION AS FOLLOWS:

1. LEADING EDGE ABSCISSAS, RE B
2. LEADING EDGE ORDINATES, RE B
3. JOUKOWSKI TRANSFORMATION PARAMETERS, RE B
4. CHORD LENGTHS, RE B

5. ELLIPTIC SYSTEM ORIGIN ABSCISSAS, RE B
6. ELLIPTIC SYSTEM ORIGIN ORDINATES, RE B
7. BLADE ELLIPTIC AXIS TWIST ANGLE MEASURED CLOCKWISE LOOKING OUTWARD, IN RADIANS

RBS(SLPSPN) - MULTI-RECORD MEMBER IN *RS FORMAT
OR IBS(SLPSPN) CONTAINING THE SLOPE MATRIX IN THE SPANWISE DIRECTION. EACH RECORD CONTAINS ONE COLUMN OF THE MATRIX.

RBF(FLOW) - TYPE 1 TABLE OF THE FLOW FIELD AS A FUNCTION OF (1) SPANWISE STATIONS, RE B, (2) INPLANE ANGLE, IN RADIANS, AND (3) ORDERED POSITION. THE ORDERED POSITIONS ARE THE FOLLOWING THREE COMPONENTS OF THE FLOW FIELD VECTOR :

1. COMPONENT ON AXIS OF ROTATION, RE SPEED OF SOUND
2. COMPONENT ALONG THE SPAN OF THE BLADE, RE SPEED OF SOUND
3. COMPONENT TO COMPLETE RIGHT-HAND COORDINATE SYSTEM, RE SPEED OF SOUND

IBA(LIFT) - TYPE 1 TABLE OF SECTION LIFT COEFFICIENTS AS A FUNCTION OF (1) SPANWISE STATION, RE B, (2) ANGLE OF ATTACK, DEGREES, (3) MACH NUMBER

IBL(DRAG) - TYPE 1 TABLE OF SECTION DRAG COEFFICIENTS AS A FUNCTION OF (1) SPANWISE STATION, RE B, (2) ANGLE OF ATTACK, DEGREES, (3) MACH NUMBER

BLM(LIFTDRAG) - TYPE 1 TABLE OF SECTION LIFT AND DRAG COEFFICIENTS AS A FUNCTION OF (1) SPANWISE STATION, RE B, (2) ANGLE OF ATTACK, DEGREES, (3) MACH NUMBER, AND (4) ORDERED POSITION. THE ORDERED POSITIONS ARE:

1. SECTION LIFT COEFFICIENT
2. SECTION DRAG COEFFICIENT

PRP(PERFORM) - TYPE 1 TABLE OF PROPELLER PERFORMANCE DATA AS A FUNCTION OF (1) SPANWISE STATIONS, RE B, (2) INPLANE ANGLE, RADIANS, AND (3) ORDERED POSITION. THE ORDERED POSITIONS ARE:

1. LOCAL ANGLE OF ATTACK, RADIANS
2. LOCAL MACH NUMBER
3. RESULTANT VELOCITY IN THE THRUST DIRECTION, RE B*OMEGA

4. RESULTANT VELOCITY OF FLUID IN THE DISK PLANE, RE B*OMEGA
5. INDUCED AXIAL VELOCITY AT THE DISK PLANE, RE B*OMEGA
6. INDUCED ANGULAR VELOCITY AT THE DISK PLANE, RE OMEGA
7. INFLOW ANGLE, RADIANS

ERRORS

NON-FATAL

1. INSUFFICIENT LOCAL DYNAMIC STORAGE AVAILABLE
2. UNABLE TO BUILD TABLE PRP(PERFORM)
3. UNABLE TO INTERPOLATE TABLE SPECIFIED
4. UNABLE TO INVERT THE JACOBIAN MATRIX
5. UNABLE TO INTEGRATE EQUATION SPECIFIED
6. MEMBER MANAGER ERROR OCCURRED ON SPECIFIED UNIT MEMBER
7. INDEPENDENT VARIABLE ARRAY NOT IN INCREASING ORDER IN SUBPROGRAM SPLS
8. UNABLE TO ITERATE AND FIND A SOLUTION FOR INDUCED AXIAL VELOCITY AND INDUCED ANGULAR VELOCITY

FATAL - NONE

INPUT DATA SIZE RESTRICTIONS	MAXIMUM NUMBER OF ENTRIES
NUMBER OF AIRFOILS ON BLADE (SPANWISE STATIONS)	10
NUMBER OF BLADE INPLANE ANGLE VALUES	25

LDS REQUIREMENTS

(Maximum Allocation of LDS - 6190)

LENGTH = 13 * N + (NXI * (7 * NPSI + 1))

WHERE

NXI = NUMBER OF SPANWISE STATIONS
NPSI= NUMBER OF BLADE INPLANE ANGLE VALUES
N = NUMBER OF SPANWISE STATIONS PLUS ONE

GDS REQUIREMENTS

(Maximum Allocation of GDS - 2000)

SUFFICIENT GDS IS REQUIRED FOR THE FOLLOWING TABLES:

RBS(SPAN) OR IBS(SPAN)
RBF(FLOW)
BLM(LIFTDRAG) OR IBA(LIFT) AND IBL(DRAG)
PRP(PERFORM)

Propeller Loads Module (PLD)

PURPOSE - CALCULATES THE LOADS AT SPECIFIED SURFACE POINTS AND FOR SPECIFIED TIMES

AUTHOR - WKB(L03/02/00)

INPUT

USER PARAMETERS		DEFAULT
MACHRF	MACH NUMBER OF BLADE TIP (REAL)	.8
IMPROV	IMPROV OPTION (LOGICAL) = .TRUE. , USE UNIT MEMBERS CREATED BY MODULES IBA, IBS, AND IBL = .FALSE., USE UNIT MEMBERS CREATED BY MODULES RBA, RBS, AND BLM	.F.
NBLADE	NUMBER OF BLADES COMPRISING PROPELLER (INTEGER)	4
IPRINT	PRINT FLAG (INTEGER) =0, NO PRINT =1, PRINT INPUT ONLY =2, PRINT OUTPUT ONLY =3, PRINT BOTH INPUT AND OUTPUT	3
THE FOLLOWING ARE REQUIRED FOR IMPROV = .FALSE. :		
BLMFRCT	3 CHARACTER NAME TO BE USED TO FORM "YYY" PART OF BLM(YYYXXX) UNIT MEMBER NAME	FRC
RBAPRES	3 CHARACTER NAME TO BE USED TO FORM "ZZZ" PART OF RBA(ZZZXXX) UNIT MEMBER NAME	PRS
THE FOLLOWING ARE REQUIRED FOR IMPROV = .TRUE. :		
IBLFRCT	3 CHARACTER NAME TO BE USED TO FORM "YYY" PART OF IBL(YYYXXX) UNIT MEMBER NAME	FRC
IBAPRES	3 CHARACTER NAME TO BE USED TO FORM "ZZZ" PART OF IBA(ZZZXXX) UNIT MEMBER NAME	PRS

UNIT MEMBERS

THE FOLLOWING ARE REQUIRED FOR ALL CASES:

PRP(PERFORM)	TYPE 1 TABLE OF PROPELLER PERFORMANCE DATA
GRID(XI1)	UNIT MEMBER OF SPANWISE STATIONS
GRID(XI2)	UNIT MEMBER OF CHORDWISE STATIONS
GRID(PSI)	UNIT MEMBER OF INPLANE STATIONS

THE FOLLOWING ARE REQUIRED FOR IMPROV = .FALSE. :

RBS(SPAN)	TYPE 1 TABLE OF BLADE GEOMETRIC FUNCTIONS IN THE SPAN DIRECTION
BLM(LIFTDRAG)	TYPE 1 TABLE OF LIFT COEFFICIENTS AND DRAG COEFFICIENTS
BLM(YYYXXX)	TYPE 1 TABLES OF SKIN FRICTION COEFFICIENTS (THE "XXX" IN THE MEMBER NAME ARE INTEGER NUMBERS THAT RUN FROM 01 TO THE NUMBER OF MACH NUMBERS GIVEN IN UNIT MEMBER RBA(MACH) - THE YYY IS OBTAINED FROM USER PARAMETER BLMFRCT)
RBA(ZZZXXX)	TYPE 1 TABLES OF LOCAL PRESSURE COEFFICIENTS (THE "XXX" IN THE MEMBER NAME ARE INTEGER NUMBERS THAT RUN FROM 01 TO THE NUMBER 0F MACH NUMBERS GIVEN IN UNIT MEMBER RBA(MACH) - THE ZZZ IS OBTAINED FROM USER PARAMETER RBAPRES)
RBA(MACH)	UNIT MEMBER OF MACH NUMBERS ASSOCIATED WITH THE UNIT MEMBERS RBA(YYYXXX) AND RBA(ZZZXXX)

THE FOLLOWING ARE REQUIRED FOR IMPROV = .TRUE. :

IBS(SPAN)	TYPE 1 TABLE OF BLADE GEOMETRIC FUNCTIONS IN THE SPAN DIRECTION
IBA(LIFT)	TYPE 1 TABLE OF LIFT COEFFICIENTS
IBL(DRAG)	TYPE 1 TABLE OF DRAG COEFFICIENTS
IBL(YYYXXX)	TYPE 1 TABLES OF SKIN FRICTION COEFFICIENTS (THE "XXX" IN THE MEMBER NAME ARE INTEGER NUMBERS THAT RUN FROM 01 TO THE NUMBER OF MACH NUMBERS GIVEN IN UNIT MEMBER RBA(MACH) - THE YYY IS OBTAINED FROM USER PARAMETER BLMFRCT)
IBA(ZZZXXX)	TYPE 1 TABLES OF LOCAL PRESSURE COEFFICIENTS (THE "XXX" IN THE MEMBER NAME ARE INTEGER NUMBERS THAT RUN FROM 01 TO THE NUMBER 0F MACH NUMBERS GIVEN IN UNIT MEMBER RBA(MACH) - THE ZZZ IS OBTAINED FROM USER PARAMETER RBAPRES)
IBA(MACH)	UNIT MEMBER OF MACH NUMBERS ASSOCIATED WITH THE UNIT MEMBERS RBA(YYYXXX) AND RBA(ZZZXXX)

OUTPUT

SYSTEM PARAMETERS

NERR	=.FALSE., IF NO ERRORS =.TRUE. , IF AN ERROR OCCURS

UNIT MEMBERS

PLD(LOADS) TYPE 1 TABLE OF PRESSURE LOADING AND SKIN FRICTION LOADING

PLD(FORCES) TYPE 1 TABLE OF SECTION AERODYNAMIC FORCES

PLD(INPLANE) TYPE 1 TABLE OF NET FORCES AND THRUSTS IN THE INPLANE DIRECTION

DATA BASE STRUCTURES

GRID(XI1) - 1 RECORD MEMBER IN *RS FORMAT CONTAINING VALUES OF THE SPANWISE STATIONS, RE B

GRID(XI2) - 1 RECORD MEMBER IN *RS FORMAT CONTAINING VALUES OF THE CHORDWISE STATIONS BETWEEN 0 AND 1, RADIANS

GRID(PSI) - 1 RECORD MEMBER IN *RS FORMAT CONTAINING VALUES OF THE INPLANE STATIONS BETWEEN 0 AND 1, RADIANS (MUST CONTAIN AT LEAST 4 VALUES, INCLUDING 0 AND 1)

RBS(SPAN) - TYPE 1 TABLE OF FUNCTIONS OF SPANWISE STATIONS WITH INDEPENDENT VARIABLES (1) SPANWISE STATIONS, RE B, AND (2) ORDERED POSITION. THE ORDERED POSITIONS ARE:

OR IBS(SPAN)

1. LEADING EDGE ABSCISSA, RE B
2. LEADING EDGE ORDINATE, RE B
3. JOUKOWSKI TRANSFORMATION PARAMETER, RE B
4. CHORD LENGTH, RE B
5. ELLIPTIC SYSTEM ORIGIN ABSCISSA, RE B
6. ELLIPTIC SYSTEM ORIGIN ORDINATE, RE B
7. BLADE TWIST ANGLE, RADIANS
8. BLADE SECTION AREA, RE B**2

PRP(PERFORM) - TYPE 1 TABLE OF PROPELLER PERFORMANCE DATA AS A FUNCTION OF (1) SPANWISE STATIONS, RE B, (2) INPLANE ANGLE, RADIANS, AND (3) ORDERED POSITION. THE ORDERED POSITIONS ARE :

1. LOCAL ANGLE OF ATTACK, RADIANS
2. LOCAL MACH NUMBER
3. RESULTANT VELOCITY IN THE THRUST DIRECTION OF FLUID IN THE DISK PLANE, RE B*OMEGA
4. RESULTANT VELOCITY OF FLUID IN THE DISK PLANE, RE B*OMEGA

5. INDUCED AXIAL VELOCITY AT THE DISK PLANE, RE B*OMEGA
6. INDUCED ANGULAR VELOCITY AT THE DISK PLANE, RE OMEGA
7. INFLOW ANGLE, RADIANS

ONLY THE FIRST, SECOND AND SEVENTH WORDS ARE USED.

BLM(LIFTDRAG) - TYPE 1 TABLE OF SECTION LIFT AND DRAG COEFFICIENTS AS A FUNCTION OF (1) SPANWISE STATION, RE B, (2) ANGLE OF ATTACK, DEGREES (3) MACH NUMBER, AND (4) ORDERED POSITION.
THE ORDERED POSITIONS ARE:
1. SECTION LIFT COEFFICIENT
2. SECTION DRAG COEFFICIENT

IBA(LIFT) - TYPE 1 TABLE OF SECTION LIFT COEFFICIENTS AS A FUNCTION OF (1) SPANWISE STATION, RE B, (2) ANGLE OF ATTACK, DEGREES (3) MACH NUMBER
2. SECTION DRAG COEFFICIENT

IBL(DRAG) - TYPE 1 TABLE OF SECTION DRAG COEFFICIENTS AS A FUNCTION OF (1) SPANWISE STATION, RE B, (2) ANGLE OF ATTACK, DEGREES (3) MACH NUMBER

RBA(MACH) - 1 RECORD MEMBER IN *RS FORMAT CONTAINING THE MACH NUMBERS ASSOCIATED WITH THE UNIT MEMBERS RBA(ZZZXXX) AND RBA(YYYXXX) (FOR EXAMPLE, THE FIRST MACH NUMBER GIVEN IN RBA(MACH) IS ASSOCIATED WITH THE UNIT MEMBERS RBA(YYY001) AND RBA(ZZZ001), THE SECOND WITH RBA(YYY002) AND RBA(ZZZ002),ETC.)

IBA(MACH) - 1 RECORD MEMBER IN *RS FORMAT CONTAINING THE MACH NUMBERS ASSOCIATED WITH THE UNIT MEMBERS IBA(ZZZXXX) AND IBA(YYYXXX) (FOR EXAMPLE, THE FIRST MACH NUMBER GIVEN IN IBA(MACH) IS ASSOCIATED WITH THE UNIT MEMBERS IBA(YYY001) AND IBA(ZZZ001), THE SECOND WITH IBA(YYY002) AND IBA(ZZZ002),ETC.)

BLM(YYYXXX) - TYPE 1 TABLES OF SKIN FRICTION
OR IBL(YYYXXX) COEFFICIENTS AS A FUNCTION OF (1) SPANWISE STATIONS, RE B, (2) CHORDWISE STATIONS, RADIANS, AND (3) ANGLE OF ATTACK, DEGREES.

RBA(ZZZXXX) - TYPE 1 TABLES OF LOCAL PRESSURE
OR IBA(ZZZXXX) COEFFICIENTS AS A FUNCTION OF (1) SPANWISE STATIONS, RE B, (2) CHORDWISE STATIONS, RADIANS, AND (3) ANGLE OF ATTACK, DEGREES.

PLD(LOADS) - TYPE 1 TABLE OF PRESSURE LOADING AND SKIN FRICTION LOADING AS A FUNCTION OF (1) SPANWISE STATIONS, RE B, (2) CHORDWISE STATIONS, RADIANS, (3) INPLANE STATIONS, RADIANS, AND (4) ORDERED POSITION. THE ORDERED POSITIONS ARE:

1. PRESSURE LOADING, RE (B*OMEGA)**2
2. SKIN FRICTION LOADING, RE (B*OMEGA)**2

PLD(FORCES) - TYPE 1 TABLE OF FORCES AS A FUNCTION OF (1) SPANWISE STATIONS, RE B, (2) INPLANE STATIONS, RADIANS, AND (3) ORDERED POSITION. THE ORDERED POSITIONS ARE:

1. SECTION AERODYNAMIC FORCE PARALLEL TO THRUST AXIS, RE B*(B*OMEGA)**2
2. SECTION AERODYNAMIC FORCE IN THE DIRECTION OF INCREASING INPLANE ANGLE, RE B*(B*OMEGA)**2

PLD(INPLANE) - TYPE 1 TABLE OF NET THRUST ON BLADE, NET INPLANE FORCE, EFFECTIVE BLADE RADIUS WHERE NET THRUSTS ACTS, AND EFFECTIVE BLADE RADIUS WHERE THE INPLANE FORCE ACTS AS A FUNCTION OF (1) INPLANE STATIONS, RADIANS AND (2) ORDERED POSITION. THE ORDERED POSITIONS ARE:

1. NET THRUST ON BLADE, RE (B**2)*(B*OMEGA)**2
2. NET INPLANE FORCE, RE (B**2)*(B*OMEGA)**2
3. EFFECTIVE BLADE RADIUS WHERE NET THRUST ACTS, RE B
4. EFFECTIVE BLADE RADIUS WHERE NET INPLANE FORCE ACTS, RE B

ERRORS

NON-FATAL

1. INPUT PARAMETER VALUE OUT OF RANGE
2. REQUIRED UNIT MEMBER NOT AVAILABLE
3. UNABLE TO COMPUTE INTEGRALS FOR NET THRUST AND TORQUE COEFFICIENTS
4. UNABLE TO GET SUFFICIENT LDS
5. UNABLE TO INTERPOLATE SPECIFIED TABLE
6. UNABLE TO BUILD SPECIFIED TABLES

INPUT DATA SIZE RESTRICTIONS	MAXIMUM NUMBER OF ENTRIES
NUMBER OF AIRFOILS ON BLADE (SPANWISE STATIONS)	10
NUMBER OF CHORDWISE STATIONS	25
NUMBER OF MACH NUMBERS	10
NUMBER OF INPLANE ANGLES (AZIMUTH STATIONS)	25

*** RESTRICTION: NXI1*NXI2*NPSI must be less than 3601
*** RESTRICTION: NXI1* NPSI must be less than 171

LDS REQUIREMENTS

(Maximum Allocation of LDS - 690)

LENGTH = NMACH + NXI2 + NPSI + 2*NXI1*NPSI
+ 4*NPSI + 11 * NXI1

WHERE
NMACH - NUMBER OF MACH NUMBERS
NXI1 - NUMBER OF SPANWISE STATIONS
NXI2 - NUMBER OF CHORDWISE STATIONS
NPSI - NUMBER OF AZIMUTH STATIONS

GDS REQUIREMENTS

(Maximum Allocation of GDS - 7500)

SUFFICIENT GDS IS REQUIRED TO STORE THE FOLLOWING TABLES

RBS(SPAN) OR IBS(SPAN)
RBF(FLOW)
BLM(LIFTDRAG) OR IBA(LIFT) AND IBL(DRAG)
RBA(YYYXXX) OR IBA(YYYXXX) - 2 TABLES AT ONE TIME
RBA(ZZZXXX) OR IBA(ZZZXXX) - 2 TABLES AT ONE TIME

Subsonic Propeller Noise Module (SPN)

PURPOSE - TO CALCULATE THE PERIODIC ACOUSTIC PRESSURE SIGNATURE AND SPECTRUM OF A PROPELLER WITH SUBSONIC TIP SPEED

AUTHOR - SLP(L03/02/00)

INPUT

USER PARAMETERS

		DEFAULT SI UNITS
Z	AIRCRAFT ALTITUDE (REAL), M (FT) (REQUIRED IF IATM=1)	0.0
B	BLADE LENGTH FROM AXIS TO TIP (REAL), M (FT)	1.0
CA	AMBIENT SPEED OF SOUND (REAL), M/S (FT/S) (REQUIRED IF IATM=0)	340.294
RHOA	AMBIENT DENSITY (REAL), KG/M**3 (SLUGS/FT**3) (REQUIRED IF IATM=0)	1.225
ERRTOL	ERROR TOLERANCE FOR ROOT TO RETARDED TIME EQUATION (INTEGER) (DEFAULT IS RECOMMENDED)	1.E-8
GLIM	TOLERANCE FOR F(X)=0 IN NEWTON'S METHOD (INTEGER) (DEFAULT IS RECOMMENDED)	1.E-5
IATM	ATMOSPHERIC CONDITIONS SELECTOR (INTEGER) 0 - CA AND RHOA ARE INPUT PARAMETERS 1 - ATM(TMOD) HAS AMBIENT CONDITIONS	0
IDPDT	BLADE LOADING SELECTOR (INTEGER) 0 - BLADE LOADING IS STEADY 1 - BLADE LOADING IS TIME DEPENDENT	0
IOUT	MEMBER OUTPUT OPTION (INTEGER) 0 - NO OUTPUT MEMBER (REQUIRES INPUT UNIT MEMBER OBSERV(COORD)) 1 - OUTPUT TABLE SPN(XXXNNN) IS PRODUCED FOR FARFIELD NOISE ANALYSIS (REQUIRES INPUT UNIT MEMBERS SFIELD(THETA) AND SFIELD(PHI)) 2 - OUTPUT UNIT MEMBER SPN(FFT) IS PRODUCED FOR NEARFIELD NOISE ANALYSIS (REQUIRES INPUT UNIT MEMBER OBSERV(COORD))	0
IMPROV	IMPROVE OPTION (LOGICAL) =.TRUE ,USE UNIT MEMBERS CREATED BY MODULES IBA, IBS,AND IBL =.FALSE ,USE UNIT MEMBERS CREATED BY MODULES RBA, RBS, AND BLM	.F.
PROTIME	THREE LETTER ID (XXX) USED TO FORM OUTPUT TABLE NAME SPN(XXXNNN) (CHARACTER)	XXX
PRONUM	INTEGER (NNN : 0<NNN<1000) USED TO FORM OUTPUT TABLE NAME SPN(XXXNNN) (INTEGER)	1

Subsonic Propeller Noise Module (SPN)

IPRINT	PRINT OPTION (INTEGER) 0 - NO PRINT DESIRED 1 - INPUT PRINT ONLY 2 - OUTPUT PRINT ONLY 3 - INPUT AND OUTPUT PRINT 4 - FULL PRINT + PLOT DATA	3
ITMAX	MAXIMUM NO. OF ITERATIONS FOR NEWTON'S METHOD (INTEGER) (DEFAULT IS RECOMMENDED)	10
IUNITS	SYSTEM OF UNITS INDICATOR (CHARACTER SI SI UNITS ENGLISH ENGLISH UNITS	SI
NBLADE	NUMBER OF BLADES ON PROPELLER (INTEGER)	4
NHARM	NUMBER OF HARMONICS PRINTED (INTEGER)	16
NTIME	NUMBER OF TIME POINTS USED FOR TIME SIGNATURE 4 < 2*NHARM < NTIME (INTEGER)	100
METHOD	COMPUTATIONAL METHOD SELECTOR (INTEGER) 1 - FULL BLADE FORMULATION 2 - MEAN SURFACE APPROXIMATION (*NOTE: METHOD 2 IS NOT AVAILABLE ON IBM-PC VERSION) 3 - COMPACT CHORD APPROXIMATION 4 - COMPACT SOURCE APPROXIMATION	1
MACHRF	REFERENCE MACH NUMBER (REAL)	0.8
PSI0	INITIAL AZIMUTH ANGLE OF BLADE (REAL), RADIANS	0.0
THETAR	ROOT PITCH CHANGE (REAL), RADIANS	0.0
RX	SOURCE RADIUS FOR OUTPUT (REAL), RE B (REQUIRED IF IOUT=1)	10.0
MZ	AIRCRAFT MACH NUMBER (REAL)	0.3
IPRES	PLOT FILE SELECTOR (INTEGER) 1 - WRITE TOTAL NOISE TO PLOT FILE 2 - WRITE LOADING NOISE TO PLOT FILE 3 - WRITE THICKNESS NOISE TO PLOT FILE	1
PLTID	LABLE FOR PLOT FILE (CHARACTER)	SPN

TABLES AND MEMBERS

NOTE: THE FOLLOWING INPUT UNIT(MEMBER) NAMES ARE FULLY DESCRIBED UNDER DATA STRUCTURES

THE FOLLOWING ARE REQUIRED FOR ALL VALUES OF METHOD:

ATM (TMOD)	ATMOSPHERIC PROPERTIES TABLE (REQUIRED IF IATM=1)
OBSERV(COORD)	UNIT MEMBER OF OBSERVER POSITIONS (REQUIRED IF IOUT=0 OR 2)
SFIELD(THETA)	UNIT MEMBER OF POLAR DIRECTIVITY ANGLES TO OBSERVER (REQUIRED IF IOUT=1)
SFIELD(PHI)	UNIT MEMBER OF AZIMUTHAL DIRECTIVITY ANGLES TO OBSERVER (REQUIRED IF IOUT=1)

```
(*NOTE: METHOD 2 IS NOT AVAILABLE ON IBM-PC VERSION )

        THE FOLLOWING ARE REQUIRED FOR METHOD=1 OR 2, AND IF
         IMPROV = .FALSE.:

        GRID  (XI1    )  UNIT MEMBER OF SPANWISE STATIONS
        GRID  (XI2    )  UNIT MEMBER OF CHORDWISE STATIONS
        RBS   (SHAPE  )  BLADE SHAPE TABLE
        RBS   (SLPSPN)   UNIT MEMBER OF SPANWISE SLOPE MATRIX
        RBS   (SLPCRD)   UNIT MEMBER OF CHORDWISE SLOPE MATRIX
        PLD   (LOADS  )  BLADE LOADING TABLE

        THE FOLLOWING ARE REQUIRED FOR METHOD=1 OR 2, AND IF
         IMPROV = .TRUE.:
        GRID  (XI1    )  UNIT MEMBER OF SPANWISE STATIONS
        GRID  (XI2    )  UNIT MEMBER OF CHORDWISE STATIONS
        IBS   (SHAPE  )  BLADE SHAPE TABLE
        IBS   (SLPSPN)   UNIT MEMBER OF SPANWISE SLOPE MATRIX
        IBS   (SLPCRD)   UNIT MEMBER OF CHORDWISE SLOPE MATRIX
        PLD   (LOADS  )  BLADE LOADING TABLE

        THE FOLLOWING ARE REQUIRED FOR METHOD=3, AND IF
         IMPROV = .FALSE.:

        GRID  (XI1    )  UNIT MEMBER OF SPANWISE STATIONS
        RBS   (SPAN   )  BLADE SECTION DATA TABLE
        RBS   (SLPSPN)   UNIT MEMBER OF SPANWISE SLOPE MATRIX
        PLD   (FORCES)   BLADE SECTION FORCE DATA

        THE FOLLOWING ARE REQUIRED FOR METHOD=3, AND IF
         IMPROV = .TRUE.:

        GRID  (XI1    )  UNIT MEMBER OF SPANWISE STATIONS
        IBS   (SPAN   )  BLADE SECTION DATA TABLE
        IBS   (SLPSPN)   UNIT MEMBER OF SPANWISE SLOPE MATRIX
        PLD   (FORCES)   BLADE SECTION FORCE DATA

        THE FOLLOWING ARE REQUIRED FOR METHOD=4, AND IF
         IMPROV = .FALSE.:

        RBS   (BLD    )  UNIT MEMBER OF BLADE DATA
        PLD   (INPLANE)  BLADE FORCE DATA

        THE FOLLOWING ARE REQUIRED FOR METHOD=4, AND IF
         IMPROV = .TRUE.

        IBS   (BLD    )  UNIT MEMBER OF BLADE DATA
        PLD   (INPLANE)  BLADE FORCE DATA
```

```
OUTPUT
  USER PARAMETERS
    RS              RADIAL DISTANCE FROM SOURCE TO OBSERVER
                    (RS), M (FT) (PROVIDED IF IOUT=1)
  TABLES AND MEMBERS

    SCRATCH( TME )  TEMPORARY TIME SIGNATURE TABLE
    SPN( FFT )      PRESSURE SPECTRUM OUTPUT MEMBER
                    (PROVIDED IF IOUT=2)
    SPN( XXXNNN )   PRESSURE SPECTRUM OUTPUT TABLE
                    (PROVIDED IF IOUT=1)

  SYSTEM PARAMETER
    NERR        SYSTEM ERROR PARAMETER
                .TRUE. - ERROR OCCURED DURING MODULE EXECUTION
                .FALSE.- NO ERROR

DATA BASE STRUCTURES
  ATM( TMOD )   - TYPE 1 DATA TABLE OF ATMOSPHERIC MODEL WITH
                  NINE FUNCTIONS OF ALTITUDE.
                  VARIABLES IN THE FOLLOWING ORDER:
                  (PRESSURE, DENSITY, TEMP., SOUND SPEED,
                   AVE. SOUND SPEED, ABSOLUTE HUMIDITY,
                   COEF. OF VISCOSITY, COEF. OF THERMAL
                   CONDUCTIVITY, CHARACTERISTIC IMPEDANCE)

  GRID( XI1 )   - ONE RECORD MEMBER OF SPANWISE STATIONS
                  TO BE USED TO DEFINE THE BLADE SHAPE

  GRID( XI2 )   - ONE RECORD MEMBER OF CHORDWISE STATIONS
                  BETWEEN 0 AND 1
  RBS( SHAPE )  - TYPE ONE DATA TABLE CONTAINING FUNCTIONS OF
OR IBS( SHAPE )   SPAN AND CHORD OF THE BLADE.  FIRST DIMENSION
                  IS SPANWISE STATIONS.  SECOND DIMENSION IS
                  CHORDWISE STATIONS.  THIRD DIMENSION IS ORDERED
                  POSITION WITH THE FOLLOWING THREE SETS OF DATA:
                  (1) BLADE SURFACE ABSCISSAS, RE B
                  (2) BLADE SURFACE ORDINATES, RE B
                  (3) BLADE SURFACE ELLIPTIC RADIAL COORDINATES,
                      IN RADIANS

  RBS(SLPSPN)   - MULTI-RECORD MEMBER IN *RS FORMAT CONTAINING
OR IBS(SLPSPN)    THE SLOPE MATRIX IN THE SPANWISE DIRECTION.
                  A RECORD ON THE MEMBER REPRESENTS A COLUMN
                  IN THE MATRIX.  THE NUMBER OF RECORDS EQUALS
                  THE NUMBER OF WORDS PER RECORD ON THE MEMBER.
                  ( I.E., A SQUARE MATRIX )
```

RBS(SLPCRD) - MULTI-RECORD MEMBER IN *RS FORMAT CONTAINING
OR IBS(SLPCRD) THE SLOPE MATRIX IN THE CHORDWISE DIRECTION. A RECORD ON THE MEMBER REPRESENTS A COLUMN IN THE MATRIX. THE NUMBER OF RECORDS EQUALS THE NUMBER OF WORDS PER RECORD ON THE MEMBER. (IE, A SQUARE MATRIX)

RBS(SPAN) - TYPE ONE DATA TABLE CONTAINING FUNCTIONS OF
OR IBS(SPAN) SPAN. FIRST DIMENSION IS SPANWISE STATIONS, RE B. SECOND DIMENSION IS ORDERED POSITION IN THE FOLLOWING ORDER:
(1) LEADING EDGE ABSCISSAS, RE B
(2) LEADING EDGE ORDINATES, RE B
(3) JOUKOWSKI TRANSFORMATION PARAMETERS, RE B
(4) CHORD LENGTHS, RE B
(5) ELLIPTIC SYSTEM ORIGIN ABSCISSAS, RE B
(6) ELLIPTIC SYSTEM ORIGIN ORDINATES, RE B
(7) BLADE ELLIPTIC AXIS TWIST ANGLE MEASURED POSITIVE CLOCKWISE LOOKING OUTWARD, IN RADIANS
(8) BLADE CROSS-SECTIONAL AREA, RE B**2

RBS (BLD) - ONE RECORD MEMBER IN 3RS FORMAT CONTAINING
OR IBS (BLD) THE BLADE VOLUME, RE B**3, THE BLADE ASPECT RATIO, AND THE BLADE ACTIVITY FACTOR

PLD(LOADS) - TYPE 1 TABLE OF PRESSURE LOADING AND SKIN FRICTION LOADING AS A FUNCTION OF (1) SPANWISE STATIONS, RE B, (2) CHORDWISE STATIONS, RADIANS, (3) INPLANE STATIONS, RADIANS, AND (4) ORDERED POSITION. THE ORDERED POSITIONS ARE:
1. PRESSURE LOADING, RE RHO*(B*OMEGA)**2
2. SKIN FRICTION LOADING, RE RHO*(B*OMEGA)**2

SCRATCH(TME) -TYPE ONE DATA TABLE CONTAINING TIME SIGNATURE PRODUCED BY ONE REVOLUTION OF A SINGLE BLADE. FIRST IND. VARIABLE IS TIME. SECOND IND. VARIABLE IS ORDERED POSITION AND CORRESPONDS TO OBSERVER NO. IN OUTPUT LISTING. THE DEPENDENT VARIABLE IS NONDIMENSIONAL PRESSURE.

SPN(FFT) - UNFORMATTED UNIT MEMBER CONTAINING SPECTRAL DATA FOR EACH OBSERVER. FIRST RECORD HAS BLADE PASSAGE FREQ. AND NO. OF HARMONICS. SECOND RECORD HAS NONDIMENSIONAL OBSERVER COORDINATES WRT THE AIRCRAFT-FIXED RECTANGULAR COORDINATE SYSTEM. THIRD RECORD HAS NS VALUES OF COMPLEX ACOUSTIC PRESSURE. RECORDS 2 AND 3 WILL BE REPEATED FOR EACH OF THE NOBS OBSERVER LOCATIONS.

SPN(XXXNNN) - TYPE 1 TABLE OF MEAN SQ. PRESSURE AS A FUNCTION OF FREQUENCY, POLAR DIRECTIVITY ANGLE AND AZIMUTH ANGLE. USER PARAMETERS PROTIME AND PRONUM SUPPLY XXX AND NNN.

SFIELD(PHI) - ONE RECORD MEMBER OF OBSERVER AZIMUTH ANGLES IN DEGREES. FORMAT=4H*RS$

SFIELD(THETA) ONE RECORD MEMBER OF OBSERVER POLAR DIRECTIVITY ANGLES IN DEGREES. FORMAT=4H*RS$

OBSERV(COORD)-MULTI-RECORD MEMBER WITH FORMAT= 3RS , CONTAINING ONE RECORD FOR EACH OBSERVER WITH VALUES OF THE X,Y,Z COORDINATES IN THE AIRCRAFT-FIXED FRAME .

PLD (FORCES) - TYPE ONE TABLE OF FORCES AS A FUNCTION OF (1) SPANWISE STATIONS, RE B, (2) INPLANE STATIONS, RADIANS, AND (3) ORDERED POSITION. THE ORDERED POSITIONS ARE:
(1) SECTION AERODYNAMIC FORCE PARALLEL TO THRUST AXIS, RE RHO*B*(B*OMEGA)**2
(2) SECTION AERODYNAMIC FORCE IN THE DIRECTION OF INCREASING INPLANE ANGLE, RE RHO*B*(B*OMEGA)**2

PLD(INPLANE)- TYPE 1 TABLE OF NFT THRUST ON BLADE, NET INPLANE FORCE, EFFECTIVE BLADE RADIUS WHERE NET THRUSTS ACTS, AND EFFECTIVE BLADE RADIUS WHERE THE INPLANE FORCE ACTS AS A FUNCTION OF (1) INPLANE STATIONS, RADIANS AND (2) ORDERED POSITION. THE ORDERED POSITIONS ARE:

1. NET THRUST ON BLADE, RE RHO*(B**2)*(B*OMEGA)**2
2. NET INPLANE FORCE, RE RHO*(B**2)*(B*OMEGA)**2
3. EFFECTIVE BLADE RADIUS WHERE NET THRUST ACTS, RE B
4. EFFECTIVE BLADE RADIUS WHERE NET INPLANE FORCE ACTS, RE B

ERRORS

NON-FATAL

1. REQUIRED UNIT MEMBER NOT AVAILABLE.
3. MEMBER MANAGER ERROR OCCURRED ON SPECIFIED UNIT MEMBER.
4. UNABLE TO INTERPOLATE TABLE.
5. UNABLE TO BUILD TABLE.
6. ERROR IN SPLINE INTEGRATION ROUTINE.
7. ERROR IN NEWTON'S METHOD.
8. OBSERVER DEFINED ON PROPELLER CENTERLINE.

FATAL - NONE

INPUT DATA SIZE RESTRICTIONS	MAXIMUM NUMBER OF ENTRIES
NUMBER OF AIRFOILS ON BLADE (SPANWISE STATIONS)	10
NUMBER OF CHORDWISE STATIONS	25
NUMBER OF PTS. IN TIME HISTORY	300

LDS REQUIREMENTS

(Maximum Allocation of LDS - 690)

GDS REQUIREMENTS

(Maximum Allocation of GDS - 7500)

LENGTH MUST BE SUFFICIENT TO HOLD PLD(LOADS) TABLE

Propeller Trailing Edge Module (PTE)

PURPOSE - TO PREDICT THE BROADBAND AND HARMONIC NOISE DUE TO THE INTERACTION OF THE BLADE TURBULENT BOUNDARY LAYER WITH THE TRAILING EDGE

AUTHOR - DSW(L03/02/00)

INPUT

USER PARAMETERS		DEFAULT
B	BLADE LENGTH (REAL), M (FT)	1.
MACHC	TURBULENCE CONVECTION MACH NUMBER (REAL), RE M	0.8
MACHRF	REFERENCE MACH NUMBER (REAL)	0.5
NBLADE	NUMBER OF BLADES (INTEGER)	1
NTIME	NUMBER OF TIME POINTS TO AVERAGE (INTEGER)	16
Z	AIRCRAFT ALTITUDE (REAL), M (FT)	0.
THETAR	BLADE ROOT PITCH ANGLE (REAL), RAD	0.
IUNITS	UNIT OPTION FLAG (CHARACTER) = SI - SI UNITS = ENGLISH - ENGLISH UNITS	SI
IPRINT	PRINT OPTION FLAG (INTEGER) =0 - NO PRINT =1 - INPUT PRINT ONLY =2 - OUTPUT PRINT ONLY =3 - BOTH INPUT AND OUTPUT PRINT	3
THEORY	METHODOLOGY OPTION (LOGICAL) =.TRUE. - USE THEORETICAL PREDICTION =.FALSE. - USE EMPIRICAL PREDICTION	.F.
DIROPT	DIRECTIVITY FUNCTION OPTION FLAG (INTEGER) =1 - FULL THEORETICAL DIRECTIVITY =2 - HIGH FREQUENCY APPROXIMATION =3 - LOW FREQUENCY APPROXIMATION =4 - DIRECTIVITY FUNCTION BY M. L. FINK (ONLY VALID WHEN THEORY = .FALSE.)	2
SPCOPT	SPECTRUM FUNCTION OPTION FLAG (INTEGER) =1 - FLAT PLATE SURFACE SPECTRUM =2 - AIRFOIL SURFACE SPECTRUM =3 - EMPIRICAL SPECTRUM BY M. L. FINK	3
IDPDT	BLADE LOADING SELECTOR (INTEGER) =0 - BLADE LOADING IS STEADY =1 - BLADE LOADING IS TIME DEPENDENT	0
IOUT	MEMBER OUTPUT OPTION (INTEGER) =0 - NO OUTPUT UNIT MEMBER (REQUIRES INPUT UNIT MEMBER OBSERV(COORD) =1 - OUTPUT TABLE PTE(XXXNNN) IS PRODUCED (REQUIRES UNIT MEMBER SFIELD(THETA) AND SFIELD(PHI)) =2 - OUTPUT UNIT MEMBER PTE(BBN) IS PRODUCED (REQUIRES INPUT MEMBER OBSERV(COORD)	0
PROTIM	THREE LETTER ID(XXX)USED TO FORM OUTPUT TABLE NAME PTE(XXXNNN) (CHARACTER)	XXX
PRONUM	INTEGER (NNN : 0<NNN<1000) USED TO FORM OUTPUT TABLE NAME PTE(XXXNNN) (INTEGER)	1

```
    RX       OBSERVER RADIUS FOR OUTPUT (REAL), RE B        10.
             ( REQUIRED IF IOUT = 1 )
    IMPROV   IMPROVE OPTION FLAG (LOGICAL)                   F
             = .TRUE.  , UNIT MEMBER IBS AND IBL ARE USED
             = .FALSE. , UNIT MEMBER RBS AND BLM ARE USED

(NOTE:  THEORETICAL PREDICTION ( THEORY=.TRUE. ) AND FULL
 THEORETICAL DIRECTIVITY ( DIROPT = 1 ) HAVE NOT BEEN
 INSTALLED)

  UNIT MEMBERS ( DESCRIBED UNDER DATA BASE STRUCTURES )
    GRID   (XI1     )  UNIT MEMBER OF SPANWISE STATIONS
    SFIELD(FREQ     )  UNIT MEMBER OF 1/3 OCTAVE BAND CENTER
                       FREQUENCIES
    RBS    (SPAN    )  TYPE 1 TABLE OF BLADE GEOMETRIC FUNCTIONS
                       IN THE SPANWISE DIRECTION (USED WHEN IMPROV
                       EQUAL TO FALSE )
    IBS    (SPAN    )  TYPE 1 TABLE OF BLADE GEOMETRIC FUNCTIONS
                       IN THE SPANWISE DIRECTION (USED WHEN IMPROV
                       EQUAL TO TRUE)
    RBS    (SHAPE   )  TYPE 1 TABLE OF BLADE COORDINATES
                       ( USED WHEN IMPROV EQUAL TO FALSE )
    IBS    (SHAPE   )  TYPE 1 TABLE OF BLADE COORDINATES
                       ( USED WHEN IMPROV EQUAL TO TRUE )
    OBSERV(COORD    )  UNIT MEMBER OF OBSERVER COORDINATES
    SFIELD(PHI      )  UNIT MEMBER OF OBSERVER AZIMUTHAL ANGLES
                       ( WHEN IOUT = 1 )
    SFIELD(THETA    )  UNIT MEMBER OF OBSERVER POLAR DIRECTIVITY
                       ANGLES ( WHEN IOUT = 1 )
    PRP    (PERFORM)   TYPE 1 TABLE OF PROPELLER PERFORMANCE DATA
    BLM    (THICK   )  TYPE 1 TABLE OF BOUNDARY LAYER THICKNESSES
                       ( USED WHEN IMPROV EQUAL TO FALSE )
    IBL    (THICK   )  TYPE 1 TABLE OF BOUNDARY LAYER THICKNESSES
                       ( USED WHEN IMPROV EQUAL TO TRUE )
    ATM    (TMOD    )  TYPE 1 TABLE OF ATMOSPHERIC DATA

OUTPUT
  USER PARAMETERS - NONE

  SYSTEM PARAMETERS
    NERR     = .TRUE.  - IF AN ERROR OCCURS
             = .FALSE. - IF NO ERRORS

  UNIT MEMBERS ( DESCRIBED UNDER DATA BASE STRUCTURES )
    PTE    (BBAND )  UNIT MEMBER OF 1/3 OCTAVE BAND BROADBAND
                     NOISE
    PTE    (XXXNNN)  PRESSURE SPECTRUM OUTPUT TABLE (PROVIDED
                      IF IOUT = 1 )
```

DATA BASE STRUCTURES

	GRID (XI1)	1 RECORD MEMBER IN RS FORMAT CONTAINING VALUES OF THE SPANWISE STATIONS, RE B
	SFIELD(FREQ)	1 RECORD MEMBER IN RS FORMAT CONTAINING VALUES OF THE 1/3 OCTAVE BAND FREQUENCIES, HZ
	SFIELD(THETA)	ONE RECORD MEMBER IN RS FORMAT CONTAINING OBSERVER POLAR DIRECTIVITY ANGLES IN DEGREES.
	SFIELD(PHI)	ONE RECORD MEMBER IN RS FORMAT CONTAINING OBSERVER AZIMUTHAL ANGLES IN DEGREES.
	RBS (SPAN)	TYPE 1 DATA TABLE CONTAINING FUNCTIONS OF
OR	IBS (SPAN)	SPAN. FIRST DIMENSION IS SPANWISE STATIONS, RE B. SECOND DIMENSION IS ORDERED POSITION IN THE FOLLOWING ORDER: (1) LEADING EDGE ABSCISSA, RE B (2) LEADING EDGE ORDINATE, RE B (3) JOUKOWSKI TRANSFORMATION PARAMETER, RE B (4) CHORD LENGTH, RE B (5) ELLIPTIC SYSTEM ORIGIN ABSCISSA, RE B (6) ELLIPTIC SYSTEM ORIGIN ORDINATE, RE B (7) BLADE ELLIPTIC AXIS TWIST ANGLE MEASURED POSITIVE CLOCKWISE LOOKING OUTWARD, RAD (8) BLADE SECTION AREA, RE B 2
	RBS (SHAPE)	TYPE 1 DATA TABLE CONTAINING FUNCTIONS OF
OR	IBS (SHAPE)	SPAN AND CHORD. FIRST DIMENSION IS SPANWISE STATIONS, RE B. SECOND DIMENSION IS CHORDWISE STATIONS, RAD. THIRD DIMENSION IS ORDERED POSITION IN THE FOLLOWING ORDER: (1) BLADE SURFACE ABSCISSA, RE B (2) BLADE SURFACE ORDINATE, RE B (3) BLADE SURFACE ELLIPTIC RADIAL COORDINATE, RAD
	OBSERV(COORD)	MULTI-RECORD MEMBER IN 3RS FORMAT CONTAINING THE (X,Y,Z) COORDINATES OF THE OBSERVERS RELATIVE TO THE PROPELLER HUB, M (FT)
	PRP (PERFORM)	TYPE 1 TABLE OF PROPELLER PERFORMANCE DATA AS A FUNCTION OF (1) SPANWISE STATIONS, RE B, (2) INPLANE ANGLE, RADIANS, AND (3) ORDERED POSITION IN THE FOLLOWING ORDER: (1) LOCAL ANGLE OF ATTACK, RAD (2) LOCAL MACH NUMBER (3) RESULTANT VELOCITY IN THE THRUST DIRECTION, RE B OMEGA (4) RESULTANT VELOCITY IN THE DISK PLANE, RE B OMEGA (5) INDUCED AXIAL VELOCITY AT THE DISK PLANE, RE B OMEGA (6) INDUCED ANGULAR VELOCITY AT THE DISK PLANE, RE OMEGA (7) INFLOW ANGLE, RAD

BLM (THICK) TYPE 1 TABLE OF BOUNDARY LAYER THICKNESS
OR IBL (THICK) DATA AS A FUNCTION OF (1) SPANWISE STATIONS, RE B, (2) ORDERED POSITION IN THE FOLLOWING ORDER:
(1) UPPER SURFACE DISPLACEMENT THICKNESS, RE CHORD
(2) LOWER SURFACE DISPLACEMENT THICKNESS, RE CHORD
(3) UPPER SURFACE MOMENTUM THICKNESS, RE CHORD
(4) LOWER SURFACE MOMENTUM THICKNESS, RE CHORD,
(3) ANGLE OF ATTACK, DEGREES, AND (4) MACH NUMBER

ATM (TMOD) TYPE 1 TABLE OF ATMOSPHERIC PROPERTIES IN DIMENSIONLESS FORM AS GENERATED BY THE ATM MODULE. FIRST DIMENSION IS DIMENSIONLESS ALTITUDE. SECOND DIMENSION IS ORDERED POSITION IN THE FOLLOWING ORDER:
(1) PRESSURE
(2) DENSITY
(3) TEMPERATURE
(4) SPEED OF SOUND
(5) AVERAGE SPEED OF SOUND
(6) HUMIDITY
(7) COEFFICIENT OF VISCOSITY
(8) COEFFICIENT OF THERMAL CONDUCTIVITY
(9) CHARACTERISTIC IMPEDENCE

PTE (BBAND) UNFORMATTED UNIT MEMBER CONTAINING THE AVERAGED 1/3 OCTAVE BAND BROADBAND TRAILING EDGE NOISE WITH THE FOLLOWING RECORDS:

RECORD	FORMAT	DESCRIPTION
1	RS	NFREQ VALUES OF THE 1/3 OCTAVE BAND CENTER FREQUENCY VALUES IN HERTZ
2	3RS	COORDINATES OF THE FIRST OBSERVER POSITION
3	RS	NFREQ VALUES OF THE DIMENSIONLESS MEAN-SQUARE ACOUSTIC PRESSURE FOR FIRST OBSERVER
.		
.		
.		
2*NOBS	3RS	COORDINATES OF THE LAST OBSERVER POSITION
2*NOBS+1	RS	NFREQ VALUES OF THE DIMENSIONLESS MEAN-SQUARE ACOUSTIC PRESSURE FOR LAST OBSERVER

PTE (XXXNNN) TYPE ONE TABLE AS A FUNCTION OF FREQUENCY, POLAR DIRECTIVITY ANGLE, AND AZIMUTHAL ANGLE. USER PARAMETER PROTIME AND PRONUM SUPPLY XXX AND NNN

ERRORS

NON-FATAL

1. INPUT PARAMETER VALUE OUT OF RANGE.
2. REQUIRED UNIT MEMBER NOT AVAILABLE.
3. UNABLE TO GET SUFFICIENT LDS.
4. MEMBER MANAGER ERROR.
5. UNABLE TO INTERPOLATE SPECIFIED TABLES.

FATAL - NONE

INPUT DATA RESTRICTIONS	MAXIMUM NUMBER OF ENTRIES
NUMBER OF AIRFOILS ON BLADE (SPANWISE STATIONS)	10

LDS REQUIREMENTS

(Maximum Allocation of LDS - 5190)

LENGTH = 6*NXI1 + 3*NOBS + NFREQ + NFREQ*NOBS

WHERE

NXI1 = NUMBER OF SPANWISE STATIONS
NOBS = NUMBER OF OBSERVERS
NFREQ = NUMBER OF 1/3 OCTAVE BAND FREQUENCIES

GDS REQUIREMENTS

(Maximum Allocation of GDS - 3000)

SUFFICIENT GDS IS REQUIRED TO STORE THE FOLLOWING TABLES

RBS	(SPAN)	OR	IBS	(SPAN)
RBS	(SHAPE)	OR	IBS	(SHAPE)
PRP	(PERFORM)			
BLM	(THICK)	OR	IBL	(THICK)
ATM	(TMOD)			

```
          Atmospheric Module(ATM)

PURPOSE - BUILD TABLE OF ATMOSPHERIC MODEL DATA AS FUNCTIONS
          OF ALTITUDE

AUTHOR  - SWP(L03/02/00)

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     SI
                =SI     , INPUTS ARE IN SI
                          UNITS
                =ENGLISH, 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 )
```

Atmospheric Module(ATM)

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

LDS REQUIREMENTS

(Maximum Allocation of LDS - 6190)

GDS REQUIREMENTS

(Maximum Allocation of GDS - 2000)

C-3

Atmospheric Absorption Module (ABS)

PURPOSE. COMPUTE AVERAGE ABSORPTION/WAVELENGTH AS FUNCTION OF ALTITUDE AND FREQUENCY AND BUILD TABLE

AUTHOR SWP(L03/02/00)

INPUT

USER PARAMETERS		DEFAULT IN SI UNITS
IUNITS	INPUT UNITS CODE =SI , INPUTS IN SI UNITS =ENGLISH, INPUTS IN ENGLISH UNITS	SI
ABSINT	NUMBER OF INTEGRATION STEPS	5
IPRINT	INTEGER PRINT OPTION 0 NO PRINT DESIRED 1 INPUT PRINT ONLY 2 OUTPUT PRINT ONLY 3 INPUT AND OUTPUT PRINT DESIRED	3
SAE	METHOD OPTION (L) .TRUE. - SAE ARP 866 METHOD .FALSE. - ANSI STANDARD METHOD	F
MEMBERS	DESCRIBED UNDER DATA BASE STRUCTURES SFIELD(FREQ) NOTE: MEMBERS ARE SPECIFIED BY UNIT(MEMBER) NAME	
TABLES	DESCRIBED UNDER DATA BASE STRUCTURES ATM(TMOD)	

OUTPUT

SYSTEM PARAMETER		
NERR	EXECUTIVE SYSTEM ERROR INDICATOR (L) .TRUE. - ERROR ENCOUNTERED DURING MODULE EXECUTION .FALSE. - NO ERROR ENCOUNTERED	
TABLES	DESCRIBED UNDER DATA BASE STRUCTURES ATM(AAC)	

DATA BASE STRUCTURES

ATM(TMOD) TWO-DIMENSIONAL TYPE ONE DATA TABLE OF ATMOSPHERIC MODEL OF NINE FUNCTIONS

INDEPENDENT VARIABLES
1. ALTITUDE
2. ORDERED POSITION

DEPENDENT VARIABLES
WITH VALUES IN DIMENSIONLESS UNITS
ARRANGED IN FOLLOWING ORDER -
1. PRESSURE
2. DENSITY

3. TEMPERATURE
4. SPEED OF SOUND
5. AVERAGE SPEED OF SOUND
6. HUMIDITY
7. COEFFICIENT OF VISCOSITY
8. COEFFICIENT OF THERMAL CONDUCTIVITY
9. CHARACTERISTIC IMPEDANCE

LINEAR INTERPOLATION AND CLOSEST VALUE EXTRAPOLATION

ATM(AAC) TYPE ONE DATA TABLE OF ATMOSPHERIC ABSORPTION COEFFICIENT AS FUNCTION OF ALTITUDE AND FREQUENCIES. NO EXTRAPOLATION AND LINEAR INTERPOLATION

SFIELD(FREQ) 1 RECORD MEMBER IN *RS FORMAT CONTAINING VALUES OF 1/3 OCTAVE BAND CENTER FREQUENCIES IN HERTZ

ERRORS

NON-FATAL

1. UNABLE TO INTERPOLATE TABLE ATM(TMOD)
2. UNABLE TO BUILD TABLE ATM(AAC)
3. INSUFFICIENT SPACE IN LDS FOR REQUIRED ALLOCATION

FATAL - NONE

INPUT DATA SIZE RESTRICTIONS	MAXIMUM NUMBER OF ENTRIES
NUMBER OF FREQUENCIES	24

LDS REQUIREMENTS

(Maximum Allocation of LDS - 6190)

GDS REQUIREMENTS

(Maximum Allocation of GDS - 2000)

Steady Flyover Module (SFO)

PURPOSE - A QUICK METHOD OF PRODUCING FLIGHT DYNAMICS DATA IN THE CASE OF A STEADY STATE FLYOVER. ONE RECORD OF TRAJECTORY DATA IS WRITTEN TO A MEMBER AT EACH TIME STEP.

AUTHOR - SLP(L03/02/00)

INPUT USER PARAMETERS		DEFAULT SI UNITS
IUNITS	SYSTEM OF UNITS SI SI ENGLISH ENGLISH	SI
NJO	NO. OF TIME STEPS ALREADY COMPLETED	0
IPRINT	INTEGER PRINT OPTION 0 NO PRINT 1 INPUT PRINT ONLY 2 OUTPUT PRINT ONLY 3 INPUT AND OUTPUT PRINT	3
IOUT	OUTPUT MEMBER OPTION 0 NO OUTPUT MEMBER 1 OUTPUT IN ORIGINAL UNITS	0
J	INITIAL STEP NUMBER	1
TSTEP	TIME INTERVAL BETWEEN STEPS (SEC)	0.5
Z1	ALTITUDE AT BRAKE RELEASE (M) (FT)	0.
ENGNAM	ENGINE IDENTIFIER NAME (FORMAT 3HXXX)	EN1
TT	INITIAL TIME (SEC)	0.0
VA	AIRCRAFT VELOCITY (M/S) (FT/S)	0.0
XA	INITIAL DISTANCE FROM ORIGIN (M) (FT)	0.0
YA	LATERAL DISTANCE FROM ORIGIN (M) (FT)	0.0
ZA	INITIAL ALTITUDE (ZA=Z1 IMPLIES GROUND LEVEL) (M) (FT)	0.0
THW	INCLINATION OF FLIGHT VECTOR WRT HORIZONTAL (THW>0 IMPLIES CLIMB) (DEG)	0.
PLG	INITIAL LANDING GEAR POSITION UP , RETRACTED DOWN, EXTENDED	UP
TLG	TIME AT WHICH LANDING GEAR POSITION WAS (OR WILL BE) RESET (SEC)	0.
JF	FINAL STEP NO. LIMIT	50
TF	FINAL TIME LIMIT (SEC)	100.

XF	FINAL DISTANCE LIMIT (M) (FT)	10000.
ZF	FINAL ALTITUDE LIMIT (IGNORED IF THW=0) (M) (FT)	1000.
ALPHA	ANGLE-OF-ATTACK (DEG)	0.
DELTA	FLAP SETTING (DEG)	0.
THROT	POWER SETTING	1.0

TABLES/MEMBERS
- ATM(TMOD) *IF IOUT=1
- FLI(PATH) *OPTIONAL
- FLI(FLIXXX) *OPTIONAL

OUTPUT

USER PARAMETERS

NJO	NO. OF STEPS COMPLETED
J	NEXT STEP NUMBER
PLG	FINAL LANDING GEAR POSITION
TT	ACTUAL FINAL TIME
XA	ACTUAL FINAL DISTANCE
ZA	ACTUAL FINAL ALTITUDE

MEMBERS
- FLI(PATH) *IF IOUT=1
- FLI(FLIXXX) *IF IOUT=1

SYSTEM PARAMETER

NERR — SYSTEM ERROR PARAMETER
.TRUE. - ERROR ENCOUNTERED DURING EXECUTION
.FALSE.- NO ERROR

DATA BASE STRUCTURES

ATM(TMOD) - TYPE 1 DATA TABLE OF ATMOSPHERIC MODEL OF NINE FUNCTIONS OF ALTITUDE IN FOLLOWING ORDER: (PRESSURE, DENSITY, TEMPERATURE, SOUND SPEED, AVERAGE SOUND SPEED, ABSOLUTE HUMIDITY, VISCOSITY, THERMAL CONDUCTIVITY, CHARACTERISTIC IMPEDANCE)

FLI(PATH) - OUTPUT MEMBER IN *RS FORMAT CONTAINING THE FOLLOWING TRAJECTORY DATA IN EACH OF THE NJO RECORDS: (TIME, AIRCRAFT POSITION(X, Y, Z), EULER ANGLES FROM VEHICLE-CARRIED TO BODY AXES AND EULER ANGLES FROM BODY TO WIND AXES) ORIGINAL DIMENSIONAL UNITS ARE USED. INPUT MEMBER OF THIS FORM IS EXPECTED IF NJO IS NONZERO ON INPUT.

Steady Flyover Module (SFO)

FLI(FLIXXX) - OUTPUT MEMBER IN 6RS,A4,2RS FORMAT AND
DATA IN THE FOLLOWING ORDER:
(TIME,MACH NUMBER,POWER SETTING,SPEED OF SOUND,
DENSITY,VISCOSITY,LANDING GEAR INDICATOR,
FLAP SETTING AND HUMIDITY)
(LANDING GEAR POSITION IS EITHER DOWN OR
UP .)
ENGNAM REPLACES XXX IN MEMBER NAME.
INPUT MEMBER OF THIS FORM EXPECTED IF
NJO IS NONZERO ON INPUT.

ERRORS
NON-FATAL
1. INPUTS INVALID
2. MEMBER MANAGER ERROR ON OPENING DATA UNITS
FATAL
1. TABLE NOT DEFINED TO DATA BASE MANAGER

LDS REQUIREMENTS
(Maximum Allocation of LDS - 4190)

GDS REQUIREMENTS
(Maximum Allocation of GDS - 4000)

Geometry Module (GEO)

PURPOSE - TO CALCULATE THE SOURCE TO OBSERVER GEOMETRY

AUTHOR REG(L03/02/00)

INPUT

USER PARAMETERS

		DEFAULT SI UNITS
AW	REFERENCE AREA OF THE AIRCRAFT (EXAMPLE, WING AREA), M**2 (FT**2)	1.00
CTK	CHARACTERISTIC TIME CONSTANT	1.00
DELDB	LIMITING NOISE LEVEL, DOWN FROM THE PEAK(DB)	20.
MASSAC	REFERENCE MASS OF THE AIRCRAFT, KG (SLUGS)	416.8
START	INITIAL FLIGHT TIME TO BE CONSIDERED, S	0.
STOP	FINAL FLIGHT TIME TO BE CONSIDERED, S	9999.
DELT	RECEPTION TIME INCREMENT, S	.5
DELTH	MAXIMUM POLAR DIRECTIVITY ANGLE LIMIT, DEG	10.
ICOORD	OUTPUT OPTION 0 NEITHER BODY NOR WIND OUTPUT REQUIRED 1 BODY AXES OUTPUT REQUIRED 2 WIND AXES OUTPUT REQUIRED 3 BOTH WIND AND BODY AXES OUTPUT REQUIRED	3
DIRECT	=.FALSE., INTERPOLATE FROM FLI(PATH) OBSERVER RECEPTION TIMES AND GEOMETRY BASED ON USER PARAMETERS START, STOP, DELTH, AND DELT =.TRUE. , CALCULATE OBSERVER RECEPTION TIMES AND GEOMETRY BASED ON THE EXACT FLIGHT TIMES AS READ FROM UNIT MEMBER FLI(PATH)	.FALSE.
IPRINT	OUTPUT PRINT OPTION 0 NO PRINT DESIRED 1 INPUT PRINT ONLY 2 OUTPUT PRINT ONLY 3 BOTH INPUT AND OUTPUT PRINT	3
IUNITS	UNITS FLAG FOR BOTH INPUT AND OUTPUT SI METRIC UNITS ENGLISH ENGLISH UNITS	SI

```
 MEMBERS
    FLI(PATH)
    GEO(SOURCE)     - NEEDED IF OTHER THAN BODY AND WIND AXES
                      SYSTEMS ARE TO BE OUTPUT.
    OBSERV(COORD)   - GEOMETRY OF OBSERVERS


OUTPUT
  USER PARAMETERS
    GEOERR      GEOMETRY ERROR FLAG
                0    NO OUTPUT WITHIN RANGE
                1    SUCCESSFUL COMPLETION

  MEMBERS
    GEO(BODY) - GEOMETRY ASSOCIATED WITH BODY AXIS SYSTEM
    GEO(WIND) - GEOMETRY ASSOCIATED WITH WIND AXIS SYSTEM
    GEO(XXXX) - GEOMETRY ASSOCIATED WITH USER SUPPLIED SOURCE
                COORDINATE SYSTEM(S).  THE MEMBER NAME(S) XXXX
                IS GIVEN BY THE SECOND ELEMENT OF EACH RECORD
                FROM UNIT MEMBER GEO(SOURCE).  THE NUMBER OF
                RECORDS IN GEO(SOURCE) DETERMINES THE NUMBER OF
                GEO(XXXX) UNIT MEMBERS.  IT IS IMPORTANT THAT
                THE SECOND ELEMENT OF EACH RECORD IN
                GEO(SOURCE) BE UNIQUE TO ALLOW EACH COORDINATE
                SYSTEM TO HAVE ITS OWN MEMBER NAME ON UNIT GEO.

TEMPORARY
  MEMBERS
    SCRATCH(FLITAB) - TABLE OF FLIGHT DATA CREATED AND USED
                      BY GEO
    SCRATCH(FLIGHT) - TEMPORARY MEMBER CONTAINING OUTPUT DATA
                      PRIOR TO ARRANGING IT ONTO THE OUTPUT
                      DATA MEMBERS


DATA STRUCTURES
 ATM(TMOD)  - TYPE 1 DATA TABLE OF ATMOSPHERIC MODEL CONTAINING
              EIGHT FUNCTIONS OF ALTITUDE.  DEPENDENT VARIABLES
              ARE IN THE FOLLOWING ORDER BY ORDERED POSITION
                1.  ATMOSPHERIC PRESSURE
                2.  DENSITY
                3.  TEMPERATURE
                4.  SPEED OF SOUND
                5.  AVERAGE SPEED OF SOUND
                6.  RELATIVE HUMIDITY
                7.  COEFFICIENT OF VISCOSITY
                8.  COEFFICIENT OF THERMAL CONDUCTIVITY
                9.  CHARACTERISTIC IMPEDANCE
```

FLI(PATH) - INPUT MEMBER IN 10RS FORMAT, EACH RECORD CONTAINS THE A/C TRAJECTORY DATA IN THE FOLLOWING ORDER

- WORD 1 - FLIGHT TIME
- WORD 2 - A/C X LOCATION (EARTH FIXED)
- WORD 3 - Y LOCATION (EARTH FIXED)
- WORD 4 - Z LOCATION (EARTH FIXED)
- WORD 5 - PSI EULER ANGLE (EARTH FIXED - BODY)
- WORD 6 - TH EULER ANGLE (EARTH FIXED - BODY)
- WORD 7 - PHI EULER ANGLE (EARTH FIXED - BODY)
- WORD 8 - PSI EULER ANGLE (BODY - WIND)
- WORD 9 - TH EULER ANGLE (BODY - WIND)
- WORD 10 - PHI EULER ANGLE (BODY - WIND)

GEO(SOURCE)- INPUT MEMBER CONTAINING DESCRIPTIONS OF SOURCE COORDINATE SYSTEMS FOR WHICH OUTPUT IS REQUIRED THE MEMBER CONSISTS OF RECORDS IN I,A8,6RS,A80 FORMAT WHERE

- ELEMENT 1 - SOURCE COORDINATE INDEX
- ELEMENT 2 - SOURCE COORDINATE NAME (THIS NAME WILL BE THE MEMBER NAME XXXX USED ON UNIT MEMBER GEO(XXXX))
- ELEMENT 3 - X OFFSET FROM THE BODY ORIGIN
- ELEMENT 4 - Y OFFSET FROM THE BODY ORIGIN
- ELEMENT 5 - Z OFFSET FROM THE BODY ORIGIN
- ELEMENT 6 - PSI EULER ANGLE(BODY - SOURCE)
- ELEMENT 7 - TH EULER ANGLE(BODY - SOURCE)
- ELEMENT 8 - PHI EULER ANGLE(BODY - SOURCE)
- ELEMENT 9 - SOURCE COORDINATE DESCRIPTION

GEO(XXXX) -
THE FORMAT OF GEO(XXXX) IS AS FOLLOWS:

RECORD	WORD	DESCRIPTION
1		RECORD FORMAT IS I,3RS,I,RS
	1	OBSERVER INDEX FOR FIRST OBSERVER
	2	X COORDINATE OF OBSERVER
	3	Y COORDINATE OF OBSERVER
	4	Z COORDINATE OF OBSERVER
	5	NUMBER OF RECEPTION TIMES ASSOCIATED WITH THIS OBSERVER (ASSUME VALUE IS N)
	6	OBSERVER'S HEIGHT
2		RECORD FORMAT IS *RS
	1	
	.	RECEPTION TIMES FOR CURRENT OBSERVER
	.	INDEX
	N	

RECORDS 3 THROUGH N+2 CONTAIN GEOMETRY DATA FOR EACH RECEPTION TIME. RECORD 3 CONTAINS GEOMETRY DATA FOR THE FIRST RECEPTION TIME, RECORD 4 FOR THE SECOND RECEPTION TIME,... RECORD N+2 FOR THE N TH RECEPTION TIME.

3		RECORD FORMAT IS *RS
	1	DISTANCE OF SOURCE FROM OBSERVER
	2	EMISSION TIME
	3	DIRECTIVITY ANGLE
	4	ELEVATION ANGLE
	5	AZIMUTH ANGLE
4		REPEAT OF RECORD 3 FOR SECOND RECEPTION TIME
.	.	
.	.	
N+3		RECORD FORMAT IS *RS
	1	OBSERVER INDEX FOR SECOND OBSERVER
	2	X COORDINATE OF OBSERVER
	3	Y COORDINATE OF OBSERVER
	Y	Z COORDINATE OF OBSERVER
	5	NUMBER OF RECEPTION TIMES ASSOCIATED WITH THIS OBSERVER (ASSUME VALUE IS M)
N+4		RECORD FORMAT IS *RS
	1	
	.	RECEPTION TIMES FOR CURRENT OBSERVER
	.	INDEX
	M	

RECORD N+5 THROUGH RECORD N+M+4 CONTAIN GEOMETRY DATA FOR EACH RECEPTION TIME STORED IN THE SAME MANNER AS DESCRIBED ABOVE IN RECORDS 3 THROUGH N+2.

THE PATTERN AS SEEN IN RECORDS 1 THROUGH N+2 AND RECORDS N+3 THROUGH N+M+4 CONTINUES FOR ALL OBSERVERS

OBSERV(COORD) - MULTI-RECORD MEMBER WITH FORMAT = 4H3RS$, CONTAINING ONE RECORD FOR EACH OBSERVER WITH VALUES OF THE X, Y, Z COORDINATES

GEO(BODY) - GEOMETRY ASSOCIATED WITH BODY AXES - SAME FORMAT AS FOR GEO(XXXX)

GEO(WIND) - GEOMETRY ASSOCIATED WITH WIND AXES - SAME FORMAT AS FOR GEO(XXXX)

SCRATCH(FLITAB)-TYPE ONE DATA TABLE OF NINE FUNCTIONS OF FLIGHT TIME ARRANGED IN THE FOLLOWING ORDER

1. FLIGHT TIME
2. A/C X (EARTH FIXED)
3. A/C Y (EARTH FIXED)
4. A/C Z (EARTH FIXED)
5. PSI EULER ANGLE (EARTH FIXED - BODY)
6. TH EULER ANGLE (EARTH FIXED - BODY)
7. PHI EULER ANGLE (EARTH FIXED - BODY)
8. PSI EULER ANGLE (BODY - WIND)
9. TH EULER ANGLE (BODY - WIND)
10. PHI EULER ANGLE (BODY - WIND)

SCRATCH(FLIGHT)-TEMPORARY DATA UNIT(DATA MEMBER) TO HOLD OUTPUT DATA PRIOR TO TRANSLATION INTO OUTPUT COORDINATE SYSTEMS. THE MEMBER CONSISTS OF RECORDS IN 11RS FORMAT IN THE FOLLOWING ORDER

1. RECEIVED TIME
2. FLIGHT TIME
3. A/C X (EARTH FIXED)
4. A/C Y (EARTH FIXED)
5. A/C Z (EARTH FIXED)
6. PSI EULER ANGLE (EARTH FIXED - BODY)
7. TH EULER ANGLE (EARTH FIXED - BODY)
8. PHI EULER ANGLE (EARTH FIXED - BODY)
9. PSI EULER ANGLE (BODY - WIND)
10. TH EULER ANGLE (BODY - WIND)
11. PHI EULER ANGLE (BODY - WIND)

ERRORS

NON-FATAL

1. INSUFFICIENT LDS FOR EXECUTION
2. MORE THAN 18 SOURCE COORDINATE SYSTEMS DEFINED
3. UNABLE TO INTERPOLATE TABLE ATM(TMOD)
4. USER PARAMETER VALUE OUT OF RANGE, DEFAULT TO BE USED
5. NO FLIGHT TIMES THAT CORRESPOND TO USER SUPPLIED INPUT PARAMETERS START AND STOP

LDS REQUIREMENTS

(Maximum Allocation of LDS - 6190)

LENGTH = 8*(NSOURCE+2) + 3*NUMOBS + MAXTIMES + 10*NTIMES

WHERE

NSOURCE	=	NUMBER OF COORDINATE SYSTEMS
NUMOBS	=	NUMBER OF OBSERVERS
MAXTIMES	=	MAXIMUM NUMBER OF TIMES FOR ANY OBSERVER
NTIMES	=	NUMBER OF FLIGHT PATH TIMES

GDS REQUIREMENTS

(Maximum Allocation of GDS - 2000)

ALLOCATION REQUIRED FOR TABLES ATM(TMOD) AND SCRATCH(FLITAB)

Tone Propagation Module (PRT)

PURPOSE - PRT IS THE ENTRY POINT FOR THE TONE PROPAGATION MODULE. PRT TAKES NOISE DATA WHICH IS GENERATED BY THE NOISE MODULES IN THE SOURCE FRAME OF REFERENCE AND APPLIES ALL OF THE APPROPRIATE COMPUTATIONS TO TRANSFER IT TO THE OBSERVER FRAME OF REFERENCE.

AUTHOR - REG(L03/02/00)

INPUT

USER PARAMETERS

IPRINT	OUTPUT PRINT OPTION CODE (INTEGER) =0 NO PRINTED OUTPUT =1 PRINT INPUT DATA ONLY =2 PRINT OUTPUT DATA ONLY =3 PRINT BOTH INPUT AND OUTPUT DATA (DEFAULT)
IOUT	=1 PRINT OUTPUT AS SOUND PRESSURE LEVEL, SPL, IN DECIBELS (INTEGER) =2 PRINT OUTPUT AS DIMENSIONLESS MEAN-SQUARE PRESSURE (DEFAULT) =3 PRINT OUTPUT IN BOTH FORMS
DELF	FREQUENCY BANDWIDTH (RS) (DEFAULT IS 10. HZ)
SIGMA	SPECIFIC FLOW RESISTANCE OF THE GROUND, KG/(S M**3) OR (SLUG/(S FT**3)) (REAL SG) (DEFAULT IS 2.5 10**5 KG/(S M**3))
IUNITS	=SI , INPUTS ARE IN SI UNITS (DEFAULT) =ENGLISH, INPUTS ARE IN ENGLISH UNITS
RS	SOURCE RADIUS, M (FT) (REAL SINGLE) (DEFAULT IS 1.0 M) USER SHOULD NOTE THAT RS VALUE MUST BE THE SAME USED BY THE NOISE MODULES IN GENERATING THE NOISE TABLES.
SURFACE	TYPE OF SURFACE TO BE USED IN CALCULATING GROUND EFFECTS =SOFT, NON HARD GROUND SURFACE (DEFAULT) =HARD, HARD GROUND SURFACE
COH	INCOHERENCE COEFFICIENT - DEFAULT IS .01
ABSORP	=.TRUE. INCLUDE THE EFFECTS OF ATMOSPHERIC ABSORPTION =.FALSE. DO NOT INCLUDE THE EFFECTS OF ATMOSPHERIC ABSORPTION (DEFAULT)
GROUND	=.TRUE. INCLUDE GROUND EFFECTS =.FALSE. DO NOT INCLUDE GROUND EFFECTS (DEFAULT)

PRTTIME THREE LETTER ID (XXX) TO APPEND TO FLI TO DEFINE THE UNIT MEMBER FLI(FLIXXX) THAT WAS GENERATED BY THE FLIGHT MODULE CONTAINING THE SOURCE TIMES TO BE USED BY THE NOISE MODULES. THIS ID ALSO BECOMES THE FIRST THREE CHARACTERS OF EACH TABLE (ONE TABLE FOR EACH SOURCE TIME) MEMBER THAT IS GENERATED BY A NOISE MODULE. FOR EXAMPLE, IF THE VALUE OF PARAMETER PRTTIME IS F01, THEN THE SOURCE TIMES ARE WRITTEN ON UNIT MEMBER FLI(FLIF01) BY THE FLIGHT MODULE. IF THERE ARE 20 SOURCE TIMES ON UNIT MEMBER FLI(FLIF01), THEN THERE 20 TABLES GENERATED FOR EACH NOISE MODULE CALLED AND THE MEMBER NAMES FOR THESE TABLES ARE THE SAME FOR ALL NOISE SOURCES WITH ONLY THE UNIT NAME BEING DIFFERENT. FOR THIS EXAMPLE, IF NOISE MODULE SPN IS CALLED, THE TABLES ARE ON UNIT MEMBERS SPN(F01001), SPN(F01002), ... SPN(F01020), DEFAULT IS XXX.

DATA BASE UNITS AND MEMBERS

ATM(TMOD) TYPE 1 TABLE CONTAINING ATMOSPHERIC PROPERTIES (SEE MODULE ATM)
- INDEPENDENT VARIABLES
  1. ALTITUDE
  2. ORDERED POSITION
- DEPENDENT VARIABLES IN THE FOLLOWING ORDERED POSITION
  1. PRESSURE
  2. DENSITY
  3. TEMPERATURE
  4. SPEED OF SOUND
  5. AVERAGE SPEED OF SOUND
  6. HUMIDITY
  7. COEFFICIENT OF VISCOSITY
  8. COEFFICIENT OF THERMAL CONDUCTIVITY
  9. CHARACTERISTIC IMPEDANCE (RHO*C)

ATM(AAC) TYPE 1 TABLE CONTAINING ATMOSPHERIC ABSORPTION COEFFICIENTS (SEE MODULE ABS)
- INDEPENDENT VARIABLES
  1. ALTITUDE
  2. FREQUENCY

GEO(GEOM) GEOMETRY DATA FOR ALL OBSERVERS RELATIVE TO ONE NOISE SOURCE COORDINATE SYSTEM SEE DESCRIPTION IN DATA BASE STRUCTURES. (SEE MODULE GEO)

FLI(FLIXXX) FLIGHT DATA CONTAINING SOURCE TIMES USED BY THE NOISE MODULES. THE XXX IS REPLACED BY THE ID FOUND IN PARAMETER PRTTIME. (SEE DESCRIPTION IN DATA BASE STRUCTURES)

YYYYYY(XXXNNN) TYPE 1 TABLE CONTAINING DIMENSIONLESS MEAN SQUARE PRESSURE. THE YYYYYY IS THE UNIT NAME ASSOCIATED WITH A NOISE MODULE (E.G., FAN OR CORE). THE XXX IS THE ID FOUND IN PARAMETER PRTTIME. THE NNN IS A COUNTER STARTING AT 001 AND CONTINUING FOR THE NUMBER OF SOURCE TIMES. THERE IS ONE TABLE PER SOURCE TIME. IF THERE ARE TWENTY SOURCE TIMES, THEN THERE ARE TWENTY TABLES FOR EACH NOISE MODULE.

INDEPENDENT VARIABLES

1. FREQUENCY, HZ
2. POLAR DIRECTIVITY ANGLE, DEG
3. AZIMUTHAL DIRECTIVITY ANGLE, DEG

OUTPUT

USER PARAMETERS

NERR =.TRUE. , ERROR ENCOUNTERED, PRT TERMINATED ABNORMALLY
=.FALSE., NO ERRORS ENCOUNTERED, PRT TERMINATED SUCCESSFULLY

DATA BASE UNITS AND MEMBERS

PRT(PRES) FREQUENCY AND DIMENSIONLESS MEAN SQUARE PRESSURE AT THE OBSERVER AS A FUNCTION OF TIME. (SEE DESCRIPTION IN DATA BASE STRUCTURES.)

DATA BASE STRUCTURES

THE FORMAT OF GEO(GEOM) IS AS FOLLOWS:

RECORD	WORD	DESCRIPTION
1		RECORD FORMAT IS I,3RS,I,RS
	1	OBSERVER INDEX FOR FIRST OBSERVER
	2	X COORDINATE OF OBSERVER
	3	Y COORDINATE OF OBSERVER
	4	Z COORDINATE OF OBSERVER
	5	NUMBER OF RECEPTION TIMES ASSOCIATED WITH THIS OBSERVER (ASSUME VALUE IS N)
	6	OBSERVER'S HEIGHT
2		RECORD FORMAT IS *RS
	1	
	.	RECEPTION TIMES FOR CURRENT OBSERVER
	.	INDEX
	N	

RECORDS 3 THROUGH N+2 CONTAIN GEOMETRY DATA FOR EACH RECEPTION TIME. RECORD 3 CONTAINS GEOMETRY DATA FOR THE FIRST RECEPTION TIME, RECORD 4 FOR THE SECOND RECEPTION TIME,... RECORD N+2 FOR THE N TH RECEPTION TIME.

3		RECORD FORMAT IS *RS
	1	DISTANCE OF SOURCE FROM OBSERVER
	2	EMISSION TIME, SEC
	3	DIRECTIVITY ANGLE, DEG
	4	ELEVATION ANGLE, DEG
	5	AZIMUTH ANGLE, DEG
4		REPEAT OF RECORD 3 FOR SECOND RECEPTION TIME
.	.	
.	.	
N+3		RECORD FORMAT IS I,3RS,I,RS
	1	OBSERVER INDEX FOR SECOND OBSERVER
	2	X COORDINATE OF OBSERVER
	3	Y COORDINATE OF OBSERVER
	4	Z COORDINATE OF OBSERVER
	5	NUMBER OF RECEPTION TIMES ASSOCIATED WITH THIS OBSERVER (ASSUME VALUE IS M)
N+4		RECORD FORMAT IS *RS
	1	
	.	RECEPTION TIMES FOR CURRENT OBSERVER
	.	INDEX
	M	

RECORD N+5 THROUGH RECORD N+M+4 CONTAIN GEOMETRY DATA FOR EACH RECEPTION TIME STORED IN THE SAME MANNER AS DESCRIBED ABOVE IN RECORDS 3 THROUGH N+2.

THE PATTERN AS SEEN IN RECORDS 1 THROUGH N+2 AND RECORDS N+3 THROUGH N+M+4 CONTINUES FOR ALL OBSERVERS

THE FORMAT OF FLI(FLIXXX) IS AS FOLLOWS:

RECORD	WORD	DESCRIPTION
1		RECORD FORMAT IS *RS
	1	VALUE OF FIRST SOURCE TIME
	2	MACH NUMBER
	3	POWER SETTING
	4	AMBIENT SPEED OF SOUND
	5	DENSITY
	6	VISCOSITY
	7	LANDING GEAR INDICATOR
	8	FLAP SETTING
	9	HUMIDITY
2		RECORD FORMAT IS *RS
	.	SAME INFORMATION AS IN RECORD 1 EXCEPT FOR THE SECOND SOURCE TIME
3	.	
.	.	
.	.	
.	.	

Tone Propagation Module (PRT)

THE FORMAT OF PRT(PRES) IS AS FOLLOWS:

RECORD	WORD	DESCRIPTION
1		RECORD FORMAT IS I,A8
	1	NUMBER OF NOISE SOURCES PROPAGATED TO THE OBSERVERS, = 1
	2	MODULE NAME OF NOISE SOURCE PROPAGATED TO THE OBSERVERS
2		RECORD FORMAT IS 2I,2RS
	1	OBSERVER INDEX FOR THE FIRST OBSERVER
	2	NUMBER OF RECEPTION TIMES ASSOCIATED WITH THIS OBSERVER (ASSUME VALUE IS N)
	3	AIR DENSITY AT THE OBSERVER (RE RHO_R)
	4	SPEED OF SOUND AT THE OBSERVER (RE C_R)
3		RECORD FORMAT IS *RS
	1	
	.	RECEPTION TIMES FOR CURRENT OBSERVER
	.	INDEX
	N	
4		RECORD FORMAT IS *RS
	1	OBSERVED FREQUENCY FOR THE FIRST HARMONIC NUMBER FOR THE FIRST TIME
	2	OBSERVED FREQUENCY FOR THE SECOND HARMONIC NUMBER FOR THE FIRST TIME
	.	
	.	
	NF	OBSERVED FREQUENCY FOR THE LAST HARMONIC NUMBER TO BE CONSIDERED FOR THE FIRST TIME
5	1	DIMENSIONLESS MEAN SQUARE PRESSURE FOR THE FIRST FREQUENCY AND THE FIRST RECEPTION TIME
	2	DIMENSIONLESS MEAN SQUARE PRESSURE FOR THE SECOND FREQUENCY AND THE FIRST RECEPTION TIME
	.	
	.	
	NF	DIMENSIONLESS MEAN SQUARE PRESSURE FOR THE LAST FREQUENCY AND THE FIRST RECEPTION TIME

6	SAME AS RECORD 4 FOR SECOND RECEPTION TIME
7	SAME AS RECORD 5 FOR SECOND RECEPTION TIME
.	
.	
.	
2*N+4	SAME AS RECORD 2 BUT DATA IS FOR SECOND OBSERVER
2*N+5	SAME AS RECORD 3 BUT DATA IS FOR SECOND OBSERVER

RECORDS 2 THROUGH 2*N+3 REPEAT FOR ALL OBSERVERS. THE VALUE OF N DIFFERS FOR EACH OBSERVER.

ERRORS

NON-FATAL

FUNCTIONAL MODULE ERRORS

1. REQUIRED UNIT MEMBER NOT AVAILABLE.
2. INSUFFICIENT LDS DYNAMIC STORAGE.
3. UNIT MEMBER NOT OF CORRECT FORMAT.
4. MEMBER MANAGER ERROR OCCURRED ON READING OR OPENING A UNIT MEMBER.
5. UNABLE TO INTERPOLATE TABLE ---- FOR VALUES -, -, -, -
6. ERROR ENCOUNTERED IN BUILDING TABLE.

FATAL - NONE

LDS REQUIREMENTS

(Maximum Allocation of LDS - 6190)

LENGTH = 8 * NFREQ + 3 * NTS + NRT

WHERE

NFREQ = NUMBER OF FREQUENCIES
NTS = NUMBER OF SOURCE TIMES
NRT = NUMBER OF RECEPTION TIMES

GDS REQUIREMENTS

(Maximum Allocation of GDS - 2000)

ALLOCATION REQUIRED FOR THE FOLLOWING TABLES:

1. ATM(TMOD)
2. ATM(AAC) (IF USER PARAMETER "ABSORP" IS .TRUE.)
3. YYYYYY(XXXNNN) NOISE TABLE INPUT

Propagation Module (PRO)

PURPOSE - PRO TAKES NOISE DATA WHICH IS GENERATED BY THE NOISE MODULES IN THE SOURCE FRAME OF REFERENCE AND APPLIES ALL OF THE APPROPRIATE COMPUTATIONS TO TRANSFER IT TO THE OBSERVER FRAME OF REFERENCE.

AUTHOR - WKB(L03/02/00)

INPUT

USER PARAMETERS

		DEFAULT
IPRINT	OUTPUT PRINT OPTION CODE (INTEGER) =0 NO PRINTED OUTPUT =1 PRINT INPUT DATA ONLY =2 PRINT OUTPUT DATA ONLY =3 PRINT BOTH INPUT AND OUTPUT DATA	3
IOUT	=1 PRINT OUTPUT AS SOUND PRESSURE LEVEL, SPL, IN DECIBELS (INTEGER) =2 PRINT OUTPUT AS DIMENSIONLESS MEAN-SQUARE PRESSURE =3 PRINT OUTPUT IN BOTH FORMS	2
SIGMA	SPECIFIC FLOW RESISTANCE OF THE GROUND, KG/(S M**3) ((SLUG/(S FT**3))) (REAL)	2.5E+5
IUNITS	INPUT UNITS FLAG (CHARACTER) =SI , INPUTS ARE IN SI UNITS =ENGLISH, INPUTS ARE IN ENGLISH UNITS	SI
NBAND	NUMBER OF SUBBANDS PER ONE-THIRD OCTAVE BAND (INTEGER) (NOTE, NBAND MUST BE ODD-E.G., 1,3,5,...)	5
RS	SOURCE RADIUS, M (FT) (REAL) USER SHOULD NOTE THAT RS VALUE MUST BE THE SAME USED BY THE NOISE MODULES IN GENERATING THE NOISE TABLES.	1.0
SURFACE	TYPE OF SURFACE TO BE USED IN CALCULATING GROUND EFFECTS (CHARACTER) =SOFT, NON HARD GROUND SURFACE =HARD, HARD GROUND SURFACE	SOFT
COH	INCOHERENCE COEFFICIENT (REAL)	0.01
ABSORP	=.TRUE. INCLUDE THE EFFECTS OF ATMOSPHERIC ABSORPTION =.FALSE. DO NOT INCLUDE THE EFFECTS OF ATMOSPHERIC ABSORPTION	F
GROUND	=.TRUE. INCLUDE GROUND EFFECTS =.FALSE. DO NOT INCLUDE GROUND EFFECTS	F

PROTIME	THREE LETTER ID (XXX) TO APPEND TO FLI TO DEFINE THE UNIT MEMBER FLI(FLIXXX) THAT WAS GENERATED BY THE FLIGHT MODULE CONTAINING THE SOURCE TIMES TO BE USED BY THE NOISE MODULES. THIS ID ALSO BECOMES THE FIRST THREE CHARACTERS OF EACH TABLE (ONE TABLE FOR EACH SOURCE TIME) MEMBER THAT IS GENERATED BY A NOISE MODULE. FOR EXAMPLE, IF THE VALUE OF PARAMETER PROTIME IS F01, THEN THE SOURCE TIMES ARE WRITTEN ON UNIT MEMBER FLI(FLIF01) BY THE FLIGHT MODULE. IF THERE ARE 20 SOURCE TIMES ON UNIT MEMBER FLI(FLIF01), THEN THERE 20 TABLES GENERATED FOR EACH NOISE MODULE CALLED AND THE MEMBER NAMES FOR THESE TABLES ARE THE SAME FOR ALL NOISE SOURCES WITH ONLY THE UNIT NAME BEING DIFFERENT. FOR THIS EXAMPLE, IF NOISE MODULES FAN AND CORE ARE CALLED, THE TABLES ARE ON UNIT MEMBERS FAN(F01001), FAN(F01002), ... FAN(F01020), CORE(F01001), CORE(F01002), .. CORE(F01020).	XXX
PROSUM	CONTAINS THE NAMES OF SOURCE UNITS THAT ARE TO BE SUMMED BEFORE PROPAGATION. IF THE NOISE DATA FROM THE CORE AND FAN MODULES ARE TO BE SUMMED, THEN PROSUM WILL CONTAIN TWO ELEMENTS - 4HCORE AND 4HFAN .	

DATA BASE UNITS AND MEMBERS

ATM(TMOD) TYPE 1 TABLE CONTAINING ATMOSPHERIC PROPERTIES (SEE MODULE ATM)

- INDEPENDENT VARIABLES
  1. ALTITUDE
  2. ORDERED POSITION
- DEPENDENT VARIABLES IN THE FOLLOWING ORDERED POSITION
  1. PRESSURE
  2. DENSITY
  3. TEMPERATURE
  4. SPEED OF SOUND
  5. AVERAGE SPEED OF SOUND
  6. HUMIDITY
  7. COEFFICIENT OF VISCOSITY
  8. COEFFICIENT OF THERMAL CONDUCTIVITY
  9. CHARACTERISTIC IMPEDANCE (RHO C)

ATM(AAC) TYPE 1 TABLE CONTAINING ATMOSPHERIC ABSORPTION COEFFICIENTS (SEE MODULE ABS)

- INDEPENDENT VARIABLES
  1. ALTITUDE
  2. FREQUENCY

GEO(GEOM) GEOMETRY DATA FOR ALL OBSERVERS RELATIVE TO ONE NOISE SOURCE COORDINATE SYSTEM SEE DESCRIPTION IN DATA BASE STRUCTURES. (SEE MODULE GEO)

FLI(FLIXXX) FLIGHT DATA CONTAINING SOURCE TIMES USED BY THE NOISE MODULES. THE XXX IS REPLACED BY THE ID FOUND IN PARAMETER PROTIME. (SEE DESCRIPTION IN DATA BASE STRUCTURES)

YYYYYY(XXXNNN) TYPE 1 TABLE CONTAINING DIMENSIONLESS MEAN SQUARE PRESSURE. THE YYYYYY IS THE UNIT NAME ASSOCIATED WITH A NOISE MODULE (E.G., FAN OR CORE). THE XXX IS THE ID FOUND IN PARAMETER PROTIME. THE NNN IS A COUNTER STARTING AT 001 AND CONTINUING FOR THE NUMBER OF SOURCE TIMES. THERE IS ONE TABLE PER SOURCE TIME. IF THERE ARE TWENTY SOURCE TIMES, THEN THERE ARE TWENTY TABLES FOR EACH NOISE MODULE.

INDEPENDENT VARIABLES

1. FREQUENCY, HZ
2. POLAR DIRECTIVITY ANGLE, DEG
3. AZIMUTHAL DIRECTIVITY ANGLE, DEG

OUTPUT

USER PARAMETERS

NERR =.TRUE. , ERROR ENCOUNTERED, PRO TERMINATED ABNORMALLY
=.FALSE., NO ERRORS ENCOUNTERED, PRO TERMINATED SUCCESSFULLY

DATA BASE UNITS AND MEMBERS

PRO(PRES) DIMENSIONLESS MEAN SQUARE PRESSURE AT THE OBSERVER AS A FUNCTION OF FREQUENCY AND TIME. (SEE DESCRIPTION IN DATA BASE STRUCTURES.)

SCRATCH(XXXNNN) UNIT SCRATCH CONTAINS THE RESULT OF SUMMING NOISE TABLES

DATA BASE STRUCTURES

THE FORMAT OF GEO(GEOM) IS AS FOLLOWS:

RECORD	WORD	DESCRIPTION
1		RECORD FORMAT IS I,3RS,I,RS
	1	OBSERVER INDEX FOR FIRST OBSERVER
	2	X COORDINATE OF OBSERVER
	3	Y COORDINATE OF OBSERVER
	4	Z COORDINATE OF OBSERVER
	5	NUMBER OF RECEPTION TIMES ASSOCIATED WITH THIS OBSERVER (ASSUME VALUE IS N)
	6	OBSERVER'S HEIGHT
2		RECORD FORMAT IS RS
	1	
	.	RECEPTION TIMES FOR CURRENT OBSERVER
	.	INDEX
	N	

RECORDS 3 THROUGH N+2 CONTAIN GEOMETRY DATA FOR EACH RECEPTION TIME. RECORD 3 CONTAINS GEOMETRY DATA FOR THE FIRST RECEPTION TIME, RECORD 4 FOR THE SECOND RECEPTION TIME,... RECORD N+2 FOR THE N TH RECEPTION TIME.

3		RECORD FORMAT IS RS
	1	DISTANCE OF SOURCE FROM OBSERVER
	2	EMISSION TIME, SEC
	3	DIRECTIVITY ANGLE, DEG
	4	ELEVATION ANGLE, DEG
	5	AZIMUTH ANGLE, DEG
4		REPEAT OF RECORD 3 FOR SECOND RECEPTION TIME
.	.	
.	.	
N+3		RECORD FORMAT IS I,3RS,I,RS
	1	OBSERVER INDEX FOR SECOND OBSERVER
	2	X COORDINATE OF OBSERVER
	3	Y COORDINATE OF OBSERVER
	4	Z COORDINATE OF OBSERVER
	5	NUMBER OF RECEPTION TIMES ASSOCIATED WITH THIS OBSERVER (ASSUME VALUE IS M)
N+4		RECORD FORMAT IS RS
	1	
	.	RECEPTION TIMES FOR CURRENT OBSERVER
	.	INDEX
	M	

RECORD N+5 THROUGH RECORD N+M+4 CONTAIN GEOMETRY DATA FOR EACH RECEPTION TIME STORED IN THE SAME MANNER AS DESCRIBED ABOVE IN RECORDS 3 THROUGH N+2.

THE PATTERN AS SEEN IN RECORDS 1 THROUGH N+2 AND RECORDS N+3 THROUGH N+M+4 CONTINUES FOR ALL OBSERVERS

THE FORMAT OF FLI(FLIXXX) IS AS FOLLOWS:

RECORD	WORD	DESCRIPTION
1		RECORD FORMAT IS RS
	1	VALUE OF FIRST SOURCE TIME
	2	MACH NUMBER
	3	POWER SETTING
	4	AMBIENT SPEED OF SOUND
	5	DENSITY
	6	VISCOSITY
	7	LANDING GEAR INDICATOR
	8	FLAP SETTING
	9	HUMIDITY
2		RECORD FORMAT IS RS
	.	SAME INFORMATION AS IN RECORD 1 EXCEPT FOR THE SECOND SOURCE TIME
3	.	
.	.	
.	.	
.	.	

THE FORMAT OF PRO(PRES) IS AS FOLLOWS:

RECORD	WORD	DESCRIPTION
1		RECORD FORMAT IS I, A8
	1	NUMBER OF NOISE SOURCES PROPAGATED TO THE OBSERVERS, NS.
	2-(NS+1)	MODULE NAMES OF NOISE SOURCES PROPAGATED TO THE OBSERVERS
2		RECORD FORMAT IS 2I,2RS
	1	OBSERVER INDEX FOR THE FIRST OBSERVER
	2	NUMBER OF RECEPTION TIMES ASSOCIATED WITH THIS OBSERVER (ASSUME VALUE IS N)
	3	AIR DENSITY AT THE OBSERVER (RE RHO)
	4	SPEED OF SOUND AT THE OBSERVER (RE C)
3		RECORD FORMAT IS RS
	1	
	.	RECEPTION TIMES FOR CURRENT OBSERVER
	.	INDEX
	N	
4		RECORD FORMAT IS RS
	1	DIMENSIONLESS MEAN SQUARE PRESSURE FOR THE FIRST FREQUENCY AND THE FIRST RECEPTION TIME
	2	DIMENSIONLESS MEAN SQUARE PRESSURE FOR THE SECOND FREQUENCY AND THE FIRST RECEPTION TIME
	.	
	.	
	NF	DIMENSIONLESS MEAN SQUARE PRESSURE FOR THE LAST FREQUENCY AND THE FIRST RECEPTION TIME
5		RECORD FORMAT IS RS
	1	DIMENSIONLESS MEAN SQUARE PRESSURE FOR
	.	ALL FREQUENCIES FOR THE SECOND
	.	RECEPTION TIME
	NF	
6		RECORD FORMAT IS RS
	1	DIMENSIONLESS MEAN SQUARE PRESSURE FOR
	.	ALL FREQUENCIES FOR THE THIRD
	.	RECEPTION TIME
	NF	
.		
.		
N+4		SAME AS RECORD 2 BUT DATA IS FOR SECOND OBSERVER
N+5		SAME AS RECORD 3 BUT DATA IS FOR SECOND OBSERVER

RECORDS 2 THROUGH N+3 REPEAT FOR ALL OBSERVERS. THE VALUE OF N DIFFERS FOR EACH OBSERVER.

ERRORS

NON-FATAL

FUNCTIONAL MODULE ERRORS

1. REQUIRED UNIT MEMBER NOT AVAILABLE
2. INSUFFICIENT LDS DYNAMIC STORAGE
3. UNIT MEMBER NOT OF CORRECT FORMAT
4. MEMBER MANAGER ERROR OCCURRED ON READING OR OPENING A UNIT MEMBER
5. REQUIRED USER PARAMETER PROSUM IS TYPE --- WITH ---
   - EXPECTED TYPE ALPHA AND THE NUMBER OF ELEMENTS .LE. 10.
6. UNABLE TO INTERPOLATE TABLE ---- FOR VALUES -, -, -, -
7. ERROR ENCOUNTERED IN BUILDING TABLE

PRO MODULE ERRORS

1. TABLES OF NOISE SOURCES TO BE SUMMED ARE NOT COMPATIBLE

FATAL - NONE

LDS REQUIREMENTS

(Maximum Allocation of LDS - 6190)

LENGTH = 3*NFREQ + NB + 6*NB*NFREQ

WHERE NFREQ = NUMBER OF FREQUENCIES
NB = VALUE OF USER PARAMETER "NBAND"

GDS REQUIREMENTS

(Maximum Allocation of GDS - 2000)

ALLOCATION REQUIRED FOR THE FOLLOWING TABLES:

1. ATM(TMOD)
2. ATM(AAC) (IF USER PARAMETER "ABSORP" IS .TRUE.)
3. NUMBER OF YYYYYY(XXXNNN) NOISE TABLES THAT ARE TO BE SUMMED PLUS ONE

Noise Levels Module (LEV)

PURPOSE - TO SUM THE NOISE UNIT MEMBERS PROPAGATED FROM DIFFERENT SOURCE COORDINATE SYSTEMS AND COMPUTE THE REQUESTED NOISE LEVELS FOR EITHER 1/3 OCTAVE OR NARROW BAND DATA.

AUTHOR - DSW(L03/02/00)

INPUT

USER PARAMETERS		DEFAULT
IAWT	OPTION FOR A-WEIGHTED SOUND PRESSURE LEVEL BASED ON 1/3 OCTAVE BAND DATA (L)	F
IDWT	OPTION FOR D-WEIGHTED SOUND PRESSURE LEVEL BASED ON 1/3 OCTAVE BAND DATA (L)	F
IOSPL	OPTION FOR OVERALL SOUND PRESSURE LEVEL BASED ON 1/3 OCTAVE BAND DATA (L)	F
IOUT	=1 PRINT OUTPUT AS SOUND PRESSURE LEVEL, SPL, IN DECIBELS (INTEGER) =2 PRINT OUTPUT AS DIMENSIONLESS MEAN-SQUARE PRESSURE (DEFAULT) =3 PRINT OUTPUT IN BOTH FORMS	2
IPNL	OPTION FOR PERCEIVED NOISE LEVEL BASED ON 1/3 OCTAVE BAND DATA (L)	F
IPNLT	OPTION FOR TONE-CORRECTED PERCEIVED NOISE LEVEL BASED ON 1/3 OCTAVE BAND DATA (L)	T
NAWT	OPTION FOR A-WEIGHTED SOUND PRESSURE LEVEL BASED ON NARROW BAND DATA (L)	F
NDWT	OPTION FOR D-WEIGHTED SOUND PRESSURE LEVEL BASED ON NARROW BAND DATA (L)	F
NOSPL	OPTION FOR OVERALL SOUND PRESSURE LEVEL BASED ON NARROW BAND DATA (L) (THE ABOVE OPTION CODES HAVE THE FOLLOWING) (VALUES:) (.TRUE. - COMPUTE) (.FALSE. - DO NOT COMPUTE)	F
IPRINT	PRINTED OUTPUT OPTION (I) 0 - NO PRINT DESIRED 1 - INPUT PRINT ONLY 2 - OUTPUT PRINT ONLY 3 - BOTH INPUT AND OUTPUT PRINT	3
MEMSUM	CONTAINS THE UNIT NAME AND MEMBER NAME OF THE 1/3 OCTAVE BAND NOISE UNIT MEMBERS TO BE SUMMED (*A8)	NONE
MEMSUMN	CONTAINS THE UNIT NAME AND MEMBER NAME OF THE NARROW BAND NOISE UNIT MEMBERS TO BE SUMMED (*A8)	NONE

UNIT MEMBERS (DESCRIBED UNDER DATA BASE STRUCTURES)

SFIELD(FREQ)	UNIT MEMBER OF 1/3 OCTAVE BAND FREQUENCIES
OBSERV(COORD)	UNIT MEMBER OF OBSERVER POSITIONS
XXXXXX(YYYYYY)	UNIT MEMBERS OF 1/3 OCTAVE BAND NOISE DATA WITH NAMES DEFINED BY USER PARAMETER MEMSUM
ZZZZZZ(WWWWWW)	UNIT MEMBERS OF NARROW BAND NOISE DATA WITH NAMES DEFINED BY USER PARAMETER MEMSUMN

OUTPUT

USER PARAMETERS - NONE

SYSTEM PARAMETERS

NERR =.FALSE. - IF NO ERROR
=.TRUE. - IF AN ERROR OCCURS

UNIT MEMBERS (DESCRIBED UNDER DATA BASE STRUCTURES)

LEV	(AWGT)	UNIT MEMBER OF A-WEIGHTED SOUND PRESSURE LEVEL BASED ON NARROW BAND DATA
LEV	(PNLT)	UNIT MEMBER OF TONE-CORRECTED PERCEIVED NOISE LEVEL
LEV	(PRES)	UNIT MEMBER OF SUMMED DIMENSIONLESS MEAN SQUARE PRESSURE AT THE OBSERVER AS A FUNCTION OF FREQUENCY AND TIME
SCRATCH	(FILE)	SCRATCH UNIT MEMBER TO STORE NARROW BAND DATA CONVERTED TO 1/3 OCTAVE BANDS

DATA BASE STRUCTURES

OBSERV(COORD)	MULTI-RECORD UNIT MEMBER IN 3RS FORMAT CONTAINING ONE RECORD FOR EACH OBSERVER WITH VALUES OF THE X, Y, AND Z COORDINATES
SFIELD(FREQ)	ONE RECORD UNIT MEMBER IN *RS FORMAT CONTAINING VALUES OF THE 1/3 OCTAVE BAND CENTER FREQUENCIES IN HERTZ
XXXXXX(YYYYYY)	UNFORMATTED MULTI-RECORD UNIT MEMBER CONTAINING 1/3 OCTAVE BAND NOISE DATA AS GENERATED BY THE PRO MODULE. THE UNIT MEMBER NAMES ARE DEFINED BY USER PARAMETER MEMSUM. ZERO, ONE, OR MORE UNIT MEMBERS MAY BE PROVIDED, AND IF TWO OR MORE ARE PROVIDED, THEY ARE SUMMED. THE UNIT MEMBERS HAVE THE FOLLOWING FORMAT:

RECORD	FORMAT	DESCRIPTION
1	I,*A8	NO. OF NOISE SOURCES, NAMES OF SOURCES
2	2I,2RS	OBSERVER INDEX, NUMBER OF OBSERVER TIMES, AMBIENT DENSITY, AMBIENT SPEED OF SOUND FOR FIRST OBSERVER
3	*RS	ARRAY OF OBSERVER TIMES
4	*RS	1/3 OCTAVE BAND MEAN-SQUARE PRESSURES FOR FIRST OBSERVER TIME
.		.
.		.
1+M*(2+N)	*RS	1/3 OCTAVE BAND MEAN SQUARE PRESSURES FOR NTH OBSERVER TIME AND MTH OBSERVER (M IS THE NUMBER OF OBSERVERS) (AND N IS THE NUMBER OF) (OBSERVER TIMES. N MAY BE) (DIFFERENT FOR EACH OBSERVER)

ZZZZZZ(WWWWWW) UNFORMATTED MULTI-RECORD UNIT MEMBER CONTAINING NARROW BAND NOISE DATA AS GENERATED BY THE PRT MODULE. THE UNIT MEMBER NAMES ARE DEFINED BY USER PARAMETER MEMSUMN. ZERO, ONE, OR MORE UNIT MEMBERS MAY BE PROVIDED, AND IF TWO OR MORE ARE PROVIDED, THEY ARE SUMMED. THE UNIT MEMBERS HAVE THE FOLLOWING FORMAT:

RECORD	FORMAT	DESCRIPTION
1	I,*A8	NO. OF NOISE SOURCES, NAMES OF NOISE SOURCES
2	2I,2RS	OBSERVER INDEX, NUMBER OF OBSERVER TIMES, AMBIENT DENSITY, AMBIENT SPEED OF SOUND
3	*RS	ARRAY OF OBSERVER TIMES
4	*RS	ARRAY OF NARROW BAND FREQUENCIES FOR FIRST TIME
5	*RS	ARRAY OF MEAN-SQUARE PRESSURES FOR FIRST TIME
.		.
.		.
M*(2+2N)	*RS	ARRAY OF FREQUENCIES FOR NTH OBSERVER TIME AND MTH OBSERVER
1+M*(2+2N)	*RS	ARRAY OF PRESSURES FOR NTH OBSERVER TIME AND MTH OBSERVER

LEV (AWGT) SAME AS LEV(PNLT), SUBSTITUTING AWGT FOR PNLT

LEV (PNLT) UNFORMATTED MULTI-RECORD OUTPUT UNIT MEMBER IN THE FOLLOWING FORMAT OF A HEADER RECORD AND THREE RECORDS PER OBSERVER:

RECORD	FORMAT	DESCRIPTION
1	I,*A8	NO. OF NOISE SOURCES, NAMES OF NOISE SOURCES
2	2I	OBSERVER INDEX, NUMBER OF TIMES
3	*RS	ARRAY OF TIMES FOR FIRST OBSERVER
4	*RS	ARRAY OF PNLT FOR ALL TIMES FOR FIRST OBSERVER
.		.
.		.
3*M-1	2I	OBSERVER INDEX, NUMBER OF TIMES
3*M	*RS	ARRAY OF TIMES FOR MTH OBSERVER
1+3*M	*RS	ARRAY OF PNLT FOR ALL TIMES FOR MTH OBSERVER

LEV (PRES) UNFORMATTED MULTI-RECORD UNIT MEMBER
FOLLOWING FORMAT

RECORD	FORMAT	DESCRIPTION
1	I,*A8	NO. OF NOISE SOURCES, NAMES OF NOISE SOURCES
2	2I,2RS	OBSERVER INDEX, NUMBER OF RECEPTION TIMES, AIR DENSITY, SPEED OF SOUND
3	*RS	RECEPTION TIMES FOR CURRENT OBSERVER INDEX
4	*RS	SUMMED DIMENSIONLESS MEAN SQUARE PRESSURE FOR ALL FREQUENCIES FOR THE FIRST RECEPTION TIME
5	*RS	SUMMED DIMENSIONLESS MEAN SQUARE PRESSURE FOR ALL FREQUENCIES FOR THE SECOND RECEPTION TIME
6	*RS	SUMMED DIMENSIONLESS MEAN SQUARE PRESSURE FOR ALL FREQUENCIES FOR THE THIRD RECEPTION TIME
.		
.		
N+4	*RS	SAME AS RECORD 2 BUT DATA IS FOR SECOND OBSERVER
N+5	*RS	SAME AS RECORD 3 BUT DATA IS FOR SECOND OBSERVER

RECORDS 2 THRU N+3 REPEAT FOR ALL OBSERVERS.
THE VALUE OF N DIFFERS FOR EACH OBSERVER.

SCRATCH(FILE) UNFORMATTED MULTI-RECORD UNIT MEMBER
CONTAINING THE NARROW BAND DATA CONVERTED TO
1/3 OCTAVE BANDS AND SUMMED FOR EACH OBSERVER
TIME AND OBSERVER IN THE FOLLOWING FORMAT:

RECORD	FORMAT	DESCRIPTION
1	2I,2RS	OBSERVER INDEX, NUMBER OF OBSERVER TIMES, AMBIENT DENSITY, AMBIENT SPEED OF SOUND FOR FIRST OBSERVER
2	*RS	ARRAY OF OBSERVER TIMES
3	*RS	1/3 OCTAVE BAND MEAN-SQUARE PRESSURES FOR FIRST OBSERVER TIME
.		.
.		.
M*(2+N)	*RS	1/3 OCTAVE BAND MEAN SQUARE PRESSURES FOR NTH OBSERVER TIME AND MTH OBSERVER (M IS THE NUMBER OF OBSERVERS) (AND N IS THE NUMBER OF) (OBSERVER TIMES. N MAY BE) (DIFFERENT FOR EACH OBSERVER)

ERRORS

NON-FATAL

1. MEMBER MANAGER ERROR OCCURRED ON SPECIFIED UNIT MEMBER.
2. INSUFFICIENT DYNAMIC STORAGE.
3. REQUIRED UNIT MEMBER NOT AVAILABLE.
4. REQUIRED USER PARAMETER TYPE OR NUMBER OF ELEMENTS IS INVALID.
5. INVALID INPUT DATA.

FATAL - NONE

INPUT DATA SIZE RESTRICTIONS	MAXIMUM NUMBER OF ENTRIES
NUMBER OF FREQUENCIES	24

LDS REQUIREMENTS

(Maximum Allocation of LDS - 6190)

$$LENGTH = NFREI * (5+NMEMI) + 3*LRECN + 2 * (NMEMI+NMEMN+1) + 3 * (1+NMEMI+NMEMN) + MAX(LRECI,LRECN)$$

WHERE:

NFREI	NUMBER OF 1/3 OCTAVE BAND FREQUENCIES
NMEMI	NUMBER OF 1/3 OCTAVE BAND NOISE MEMBERS
NMEMN	NUMBER OF NARROW BAND NOISE MEMBERS
LRECI	LENGTH OF LONGEST RECORD ON 1/3 OCTAVE BAND NOISE MEMBERS
LRECN	LENGTH OF LONGEST RECORD ON NARROW BAND NOISE MEMBERS

GDS REQUIREMENTS

(Maximum Allocation of GDS - 2000)

Effective Noise Module (EFF)

PURPOSE - TO COMPUTE EFFECTIVE PERCEIVED NOISE LEVEL (EPNL)
AS A FUNCTION OF OBSERVER POSITION

AUTHOR - CBF(L03/02/00)

INPUT

USER PARAMETERS	DEFAULT
DTIME - RECEPTION TIME INCREMENT (S)	.5
IPRINT - PRINTED OUTPUT OPTION	3

0 - NO PRINT DESIRED
1 - INPUT PRINT ONLY
2 - OUTPUT PRINT ONLY
3 - BOTH INPUT AND OUTPUT PRINT

MEMBERS - DESCRIBED UNDER DATA BASE STRUCTURES
NOTE : MEMBERS ARE SPECIFIED BY UNIT(MEMBER) NAME
OBSERV(COORD)
LEV(PNLT)

OUTPUT

SYSTEM PARAMETER
NERR - EXECUTIVE SYSTEM PARAMETER FOR ERROR ENCOUNTERED DURING EXECUTION OF A FUNCTIONAL MODULE. NERR SET TO .TRUE. IF ERROR ENCOUNTERED

MEMBERS
EFF(EPNL)

DATA BASE STRUCTURES

OBSERV(COORD) MULTI-RECORD MEMBER WITH FORMAT = 3RS, CONTAINING ONE RECORD FOR EACH OBSERVER WITH VALUES OF THE X,Y,Z COORDINATES

LEV(PLNT) UNFORMATTED MULTI-RECORD MEMBER IN THE FOLLOWING FORMAT OF A HEADER RECORD FOLLOWED BY THREE RECORDS PER OBSERVER

RECORD	FORMAT	DESCRIPTION
1	I,*A8	NO. OF NOISE SOURCES, ARRAY OF NOISE SOURCES
2	2I	OBSERVER INDEX, NO. OF TIMES
3	*RS	ARRAY OF TIMES
4	*RS	ARRAY OF TONE-CORRECTED PERCEIVED NOISE LEVEL FOR ALL TIMES
.	.	.
.	.	.
.	.	.
1+3*M		(WHERE M IS THE NUMBER OF OBSERVERS)

EFF(EPNL) UNFORMATTED MULTI-RECORD OUTPUT MEMBER IN THE FOLLOWING FORMAT OF A HEADER RECORD FOLLOWED BY A RECORD FOR EACH OBSERVER CONTAINING THE OBSERVER INDEX AND EPNL VALUE

RECORD	FORMAT	DESCRIPTION
1	I,*A8	NO. OF SOURCES, ARRAY OF NOISE SOURCES
2	I,RS	OBSERVER INDEX, EFFECTIVE PERCEIVED NOISE LEVEL (EPNL)
.	.	.
.	.	.
.	.	.
N+1		(WHERE N IS THE NUMBER OF OBSERVERS)

ERRORS

NON-FATAL

1. INSUFFICIENT DYNAMIC STORAGE (FMNMSG ERROR 2)
2. MEMBER MANAGER ERROR OCCURRED ON SPECIFIED UNIT MEMBER. (FMNMSG ERROR 4)

LDS REQUIREMENTS

(Maximum Allocation of LDS - 6190)

LENGTH = 12 * NOBS + NS

WHERE NOBS = NUMBER OF OBSERVERS
NS = NUMBER OF NOISE SOURCES

GDS REQUIREMENTS

(Maximum Allocation of GDS - 2000)

REFERENCES

1. Zorumski, William E.; and Weir, Donald S.: Aircraft Noise Prediction Program Theoretical Manual, Part 3, Propeller Aerodynamics and Noise. NASA TM-83199, 1986.

2. IBM Personal Computer DOS Reference Manual. International Business Machines Corporation, 1984.

3. Gillian, Ronnie E.: Aircraft Noise Prediction Program Users' Manual, NASA TM-84436, 1983.

4. Zorumski, William E.: Aircraft Noise Prediction Program Theoretical Manual, Part 1, NASA TM-83199, 1982.

APPENDIX A

GLOSSARY OF ANOPP TERMS AND ACRONYMS

A.1 Terms

Alternate names - The set of names, established in an Alternate Names Table File, which corresponds to a set of reference names. The set of alternate names is available for retrieval by a functional or an executive system module during the execution of that module.

Alternate names - External file which contains an alternate names table. The name of an alternate names file always includes a '.ANT' extension.

ANOPP run - One execution of ANOPP, the sequence of functional module execution.

BATCH file - A DOS file which contains a list of commands and/or executable file names.

Data base - A collection of data units residing on random-access mass-storage files.

Data element - One or more words residing on a formatted data record. Its data type and number of words are determined by the corresponding element code in the member format.

Data member - Any member other than a procedure member or a table member. (See member.)

Data record - An ordered set of data elements or words residing on a member. The record may be unformatted or it may be formatted as fixed, variable, or card image according to the member format.

Data table - A table of data available to the functional module for processing. It resides on a one-record member having an internal format corresponding to a defined data table type.

Data unit - The highest level of the ANOPP data base structure that can be referenced directly (by name) during an ANOPP run. It is a set of named numbers and it resides alone on an external random-access file.

Data unit file - External file which contains a data unit. The name of a data unit file always includes a '.UNT' extension.

DOS - IBM-PC Disk Operating System.

DOS file - An external file that can be accessed by DOS.

APPENDIX A

Element	-	A word or group of words on a record. (See data element.)
Element code	-	The descriptor within a data member format specification used to identify the data type of an element within the data member's records.
Executive System	-	The ANOPP modules that handle data management.
External file	-	A file residing on a disk under the management of the external system.
External system	-	The operating system that controls all job processing on the host computer.
Functional module-(FM)	-	A named set of one or more modules recognized by the ANOPP executive system that performs a specific utility or noise prediction task.
Global dynamic storage (GDS)	-	A section of free core storage defined and maintained by the executive system for storage of directories and tables.
Local dynamic storage (LDS)	-	A section of free core storage maintained by the executive system for internal data storage.
Member	-	The lowest level of the ANOPP data base structure that can be referenced directly (by name) during an ANOPP run. It is a logically contiguous set of records and it resides on a data unit.
Member format	-	A specification which describes the composition of data records residing on a member.
Module	-	A FORTRAN program that is part of the ANOPP system.
Parameter	-	See user parameter.
Permanent file	-	An external file established permanently at the host computer installation.
Preprocessor	-	A program which allows the user to create, change or list named data items required by functional modules.
Procedure	-	A batch file which controls the execution of one or more functional modules.
Record	-	An ordered set of elements or words residing on a member. (See data record.)
Scratch file	-	A temporary file on which a data unit is established.
Table member	-	A member containing a one-record member corresponding to a defined data table type.

User parameter - Array of one or more values may be established.

User parameter table (UPT) - A table used by the executive system which links the user parameter name with its current value, type, and number of elements.

User parameter file (UPF) - External file that contains a user parameter table. The name of a UPF always includes a '.PAR' extension.

A.2 Acronyms

ANOPP	Aircraft Noise Prediction Program
ANTEDT	Alternate Names Table Editor
IBGPREP	Blade Geometry Preprocessor
EPNL	Effective perceived noise level
FLPPREP	Flight Path Preprocessor
FM	Functional module
GDS	Global dynamic storage
LDS	Local dynamic storage
MMEDT	Member Manager Preprocessor
OASPL	Overall sound pressure level
PAS	Propeller Analysis System
PLEPREP	Propagation (and Noise Level) Preprocessor
PNL	Perceived noise level
PNLT	Tone-corrected perceived noise level
PRFPREP	Performance Preprocessor
SNSPREP	(Subsonic) Noise Preprocessor
SPL	Sound pressure level
TMEDT	Table Manager Editor
UPF	User parameter file
UPFEDT	User Parameter File Editor

APPENDIX A

A.3 IBM-PC Version of ANOPP-PAS Commands

ALT	Executes ANTEDT
ANOPP	Starts ANOPP-PAS
BKUP	Executes backup for input/output data
CHANGE	Changes the current ANOPP subdirectory
FPATH	Executes ATM, ABS, SFO, GEO
GEOM	Executes RBS, RBA, BLM or IBS, IBA, IBL
INPUT	Executes prediction preprocessors: PRFPREP, SNSPREP, FLPPREP, PLEPREP
LEVELS	Executes PRT, PRO, LEV, EFF
MAKE	Creates an ANOPP subdirectory and executes IBGPREP
NOISE	Executes SPN, PTE
PARAM	Executes UPFEDT
PAS	Executes all ANOPP-PAS functional modules in sequence
PERF	Executes PRP, PLD
PLOT	Executes graphics plot postprocessor
REDUCE	Reduces data unit file size
RUN	Executes a functional module
SMALL	Reduces the size of all data units listed in Section 4
TABLE	Executes TMEDT
UNIT	Executes MMEDT

APPENDIX B

SUMMARY OF FUNCTIONAL MODULES FOR IBM PC VERSION OF ANOPP-PAS

Module name	Module title	Brief description (a)
ABS	Atmospheric Absorption Module	Computes atmospheric absorption coefficients as function of altitude and frequency using either ANSI or SAE method
ATM	Atmospheric Module	Computes atmospheric properties as function of altitude using hydrostatic model
BLM, IBL	Blade Section Boundary Layer Module	Computes the blade skin friction and section drag distribution (IBL is the improved version of the module)
EFF	Effective Noise Module	Computes EPNL
GEO	Geometry Module	Calculates source-to-observer geometry
LEV	Noise Levels Module	Computes OASPL, A-weighted SPNL, D-weighted SPL, PNL, and PNLT
PLD	Propeller Loads Module	Computes final blade pressure and skin friction distribution
PRO	Propagation Module	Transfers 1/3 octave noise data to the observer frame of reference
PRP	Propeller Performance Module	Computes the induced velocity field thrust, torque, and efficiency for a given propeller under specified operating conditions
PRT	Tone Propagation Module	Transfers noise data to the observer frame of reference
PTE	Propeller Trailing Edge Module	Predicts the broadband noise due to the interaction of the blade turbulent boundary layer with the trailing edge.
RBA, IBA	Blade Section Aerodynamics Module	Computes the blade pressure and blade section lift distributions (IBA is the improved version of the module)
RBS, IBS	Blade Shape Module	Formulates functional representation of blade surface geometry (IBS is the improved version of the module)

SUMMARY OF FUNCTIONAL MODULES FOR IBM PC VERSION OF ANOPP-PAS - Continued

Module name	Module title	Brief description (a)
SFO	Steady Flyover Module	Computes the aircraft flight path
SPN	Subsonic Propeller Noise Module	Predicts the propeller noise signature

[a]For more complete descriptions of the modules and the methods used, see Section VI.

APPENDIX C

ERROR DIAGNOSTICS AND RECOVERY TECHNIQUES

Error messages generated during an ANOPP job can be any one of the following:

1. An external system diagnostic
2. An ANOPP executive system diagnostic
3. An ANOPP functional module error message

The diagnostics produced by the external operating system (DOS) are usually fatal, resulting in no program execution or job termination. The following describes some of the error messages that are printed to the screen monitor:

Error Message	Type of Error	Recovery Techniques
Bad command or file name	DOS cannot find required executable (.EXE) or BATCH file	Check the following: • Current directory must be C:\ANOPP • System Diskette must be installed on C:\ANOPP • Executable files (.EXE) must be installed on C:\ANOPP\EXE
Error Number 1000-1999	Argument passed to intrinsic function out of range	• Check input parameters and Unit members entries
Error Number 2000-2499	Input file (.PAR or .UNT) has incorrect format	• Check or recreate appropriate files
Error Number 2500-2550	Wrong data type entered (from terminal) (or) file name not found	• Rerun program with correct entry responses • Create required file on C:\ANOPP
Error Number 3000-3999	Required file not found or file has incorrect format	• Create required file or change to appropriate format
Error Number 4001	Your system is not configured with required math coprocessor	• Correct System

APPENDIX C

Error Message	Type of Error	Recovery Techniques
Error Number 4002	Incorrect DOS Version	• Replace DOS with Version 2.1 (or higher)
**FILE ACCESS ERROR	Required .PAR or .ANT file not installed on C:\ANOPP	• Create required file

All ANOPP executive system diagnostics have the following general form:

prefix (ERROR NUMBER v) *** (CALLER sysrtn) error message

where prefix identifies the executive subsystem that produced the diagnostic, v is the number of the specified system error, sysrtn is the system subroutine that called the error processor, and error message is the error condition that occurred. These messages will appear in the FM results files. The diagnostics prefixes and the executive subsystems that produce them are:

*** EXEC	ERROR - the executive management system
*** DBM	ERROR - the data base management (DBM) member manager
*** DTM	ERROR - the table manager system
*** XTB	ERROR - the table manager system utilities
*** DSM	ERROR - the dynamic storage manager
*** UTILITY	ERROR - the internal system utilities

Most error messages the normal ANOPP user encounters relate to a specific condition that the user can correct through changes to his ANOPP input. However, some error messages are applicable to internal execution problems. These are meaningful to and require the attention of a programmer maintaining the ANOPP system or a specific functional module. The following table lists the executive system diagnostics and suggests recovery techniques to be used:

Error Message	Type of Error	Recovery Techniques
DBM ERROR	Missing data base member	• Verify data member via MMEDT
	Erroneous data found on specified unit or member	• Reexamine data requirements of FM; and verify name and contents of member via MMEDT
DTM ERROR XTB ERROR	Insufficient GDS (Table too large)	• Decrease input size of data used to create table

Error Message	Type of Error	Recovery Techniques
	Erroneous table identification or content	• Verify table requirements of FM; and verify name and contents of table via TMEDT
DSM ERROR	Insufficient core or GDS/LDS overlap	• Decrease size of input data • Check Size Restriction in Section VI.
Utility ERROR	Invalid or erroneous user parameter or member data type or values	• Recheck data requirements of FM.

Error messages produced by specific functional modules are documented in Section VI.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-181689		

4. Title and Subtitle	5. Report Date
AIRCRAFT NOISE PREDICTION PROGRAM PROPELLER ANALYSIS SYSTEM IBM-PC VERSION USER'S MANUAL, VERSION 2.0	August 1988
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Sandra K. Nolan	
	10. Work Unit No. 505-63-51-07

9. Performing Organization Name and Address	11. Contract or Grant No.
Planning Research Corporation 303 Butler Farm Road Hampton, VA 23666	NAS1-18000
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Contractor Report
National Aeronautics and Space Administration Langley Research Center Hampton, VA 23665-5225	14. Sponsoring Agency Code

15. Supplementary Notes

Langley Technical Monitor: Robert A. Golub

16. Abstract

The IBM-PC version of the Aircraft Noise Prediction Program (ANOPP) Propeller Analysis System (PAS) is a set of computational programs for predicting the aerodynamics, performance, and noise of propellers. The ANOPP-PAS is a subset of a larger version of ANOPP which can be executed on CDC or VAX computers. This manual provides a description of the IBM-PC version of the ANOPP-PAS and its prediction capabilities, and instructions on how to use the system on an IBM-XT or IBM-AT personal computer. Sections within the manual document installation, system design, ANOPP-PAS usage, data entry preprocessors, and ANOPP-PAS functional modules and procedures. Appendices to the manual include a glossary of ANOPP terms and information on error diagnostics and recovery techniques.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Aircraft noise Noise prediction Propeller performance Propeller noise	Unclassified - Unlimited Subject Category - 71

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	235	A11

NASA FORM 1626 OCT 86