

AAMRL-TR-88-060

AD-A203 849

**ROUTEMAP MODEL FOR
PREDICTING NOISE EXPOSURE
FROM AIRCRAFT OPERATIONS
ON MILITARY TRAINING ROUTES**



**Michael J. Lucas
Kenneth J. Plotkin**

**Wyle Laboratories
2001 Jefferson Davis Highway
Arlington VA 22202**

**DTIC
ELECTE
FEB 07 1989
S & D
H**

September 1988

Final Report for Period January 1988 to September 1988

Approved for public release; distribution is unlimited.

**HARRY G. ARMSTRONG AEROSPACE MEDICAL RESEARCH LABORATORY
HUMAN SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6573**

89 2 7 072

NOTICES

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Please do not request copies of this report from the Armstrong Aerospace Medical Research Laboratory. Additional copies may be purchased from:

National Technical Information Service
5285 Port Royal Road
Springfield, Virginia 22161

Federal Government agencies and their contractors registered with Defense Technical Information Center should direct requests for copies of this report to:

Defense Technical Information Center
Cameron Station
Alexandria, Virginia 22314

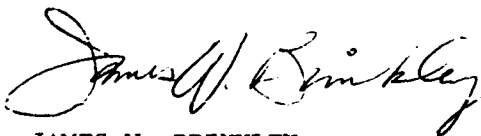
TECHNICAL REVIEW AND APPROVAL

AAMRL-TR-88-060

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER



JAMES W. BRINKLEY
Director
Biodynamics & Bioengineering Division
Harry G. Armstrong Aerospace Medical Research Laboratory

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
1a. REPORT SECURITY CLASSIFICATION Unclassified			1b. RESTRICTIVE MARKINGS		
2a. SECURITY CLASSIFICATION AUTHORITY N/A			3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution is unlimited		
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE					
4. PERFORMING ORGANIZATION REPORT NUMBER(S) WR 88-15			5. MONITORING ORGANIZATION REPORT NUMBER(S) AAMRL-TR-88-060		
6a. NAME OF PERFORMING ORGANIZATION Wyle Laboratories		6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION AAMRL/BBE		
6c. ADDRESS (City, State, and ZIP Code) 2001 Jefferson Davis Highway, Suite 701 Arlington, Virginia 22202			7b. ADDRESS (City, State, and ZIP Code) Wright-Patterson Air Force Base Ohio 45433-6573		
8a. NAME OF FUNDING / SPONSORING ORGANIZATION AAMRL		8b. OFFICE SYMBOL (If applicable) EFE	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F33615-85-C-0534		
8c. ADDRESS (City, State, and ZIP Code) Wright-Patterson Air Force Base, Ohio 45433-6573			10. SOURCE OF FUNDING NUMBERS		
			PROGRAM ELEMENT NO. 62202F	PROJECT NO. 7231	TASK NO. 34
11. TITLE (Include Security Classification) ROUTEMAP Model for Predicting Noise Exposure from Aircraft Operations on Military Training Routes					
12. PERSONAL AUTHOR(S) Michael J. Lucas, Kenneth J. Plotkin					
13a. TYPE OF REPORT Final		13b. TIME COVERED FROM Jan88 TO Sep88		14. DATE OF REPORT (Year, Month, Day) September 1988	
15. PAGE COUNT 93					
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) Noise Noise Model Military Aircraft Operations Noise Modeling Community Noise Exposure		
FIELD	GROUP	SUB-GROUP			
20	01				
19. ABSTRACT (Continue on reverse if necessary and identify by block number) Low-altitude, high-speed training operations are routinely conducted along specially designated Military Training Routes (MTRs). The location of these routes is continually changed for a variety of reasons. Each new route requires an environmental assessment to determine the community noise impact. This report describes computer program ROUTEMAP which calculates the noise level on the ground along an MTR corridor. Program ROUTEMAP is a menu-driven program that runs on any IBM PC or PC-compatible computer. ROUTEMAP requires MS DOS Version 2.0 or later, with at least one megabyte of available disk space, 640K of random access memory, and an 8087/80287 math coprocessor. (CONTINUED)					
20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a. NAME OF RESPONSIBLE INDIVIDUAL Jerry Speakman			22b. TELEPHONE (Include Area Code) 513/255-3664		22c. OFFICE SYMBOL AAMRL/BBE

Paragraph 19, Abstract, Continued

- The model requires the Air Force planner to specify the nature of the flight activity for the segment of the route in question. The information needed for each aircraft type are the number of day and night operations during a month, and nominal values for the airspeed, engine power setting, and altitude. In addition, the user must input whether the activity is usually under visual or instrument flying rules and if there are single or multiple flight tracks within the route corridor.

With this input data, the program computes the onset rate-adjusted monthly day-night average A-weighted sound level, Ldnmr, in dB for ground positions located within 13 miles of the route centerline. For comparison purposes, the program also computes the monthly average A-weighted noise exposure level without the penalty for high onset rates and without the penalty for operations during the night.

- The program also computes the probability of being highly annoyed as a function of the Ldnmr values. This information, along with the noise-compatible land-use guides normally associated with planning around airbases, can be used to interpret the noise resulting from military training route operations. (SDD)

PREFACE

This research was performed for the Harry G. Armstrong Aerospace Medical Research Laboratory at Wright-Patterson Air Force Base, Ohio, under Project/Task 723134, Exploratory Noise and Sonic Boom Research. Mr. Jerry D. Speakman of the Biodynamic Environment Branch, Biodynamics and Bio-engineering Division, was the technical monitor for this effort.



Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

TABLE OF CONTENTS

	<u>Page</u>
1.0 INTRODUCTION	1
2.0 MODEL FORMULATION	2
3.0 PROGRAM DESCRIPTION	4
3.1 Computer Program Capabilities and Features	4
3.2 General Description of Program Organization	4
4.0 USER'S GUIDE	7
4.1 Getting Started	7
4.2 Running ROUTEMAP	8
4.2.1 Main Menu	8
4.2.2 Configuration of a Route	10
4.2.2.1 Selecting a Training Mission	10
4.2.2.2 Selecting an Aircraft	14
4.2.2.3 Input Operation Parameters	14
4.2.3 Set Default Values	16
4.2.4 Calculating the Noise Levels	16
4.2.5 Analysis of Results	20
4.2.5.1 Plotting	20
4.2.5.2 Rank Order Contributors	21
4.2.5.3 Combination of Noise From Separate Routes	23
5.0 PROGRAMMER'S GUIDE	24
5.1 Convention and Annotation	24
5.2 Definition of Symbols	24
5.3 Subroutines	29
5.3.1 INITIAL	29
5.3.2 OPEN	29
5.3.3 PAUSE	29
5.3.4 MENU(IVAL,FLAG)	29

TABLE OF CONTENTS (Continued)

	<u>Page</u>
5.3.5 OPS1(ISWITCH)	31
5.3.6 OPS2(ISWITCH)	31
5.3.7 OPS3	32
5.3.8 CALC	32
5.3.9 GAUSS(XINIT,XFINL,FMEAN,VAR,PROB)	33
5.3.10 FDMI(I,J,K,P)	34
5.3.11 PLTMAIN(SWITCH,TOGGLE)	34
5.3.12 PLTMENU(SWITCH,TOGGLE)	35
5.3.13 DEVICE	35
5.3.14 PLTSPL	35
5.3.15 PLTAPP	36
5.3.16 DRAW(XARRAY,YARRAY,NPTS,FLAG1,INTEQ)	36
5.3.17 RANKODR	36
5.3.18 ADDER	37
5.3.19 FINISH	37
5.3.20 BLOCK DATA	37
5.4 COMMON Blocks	37
6.0 MAINTENANCE MANUAL	40
6.1 OMEGA10 Maintenance	40
6.2 ROUTEMAP Maintenance	40
6.3 Compiling and Linking	41
6.4 Commercial Software	42
REFERENCES	43
APPENDIX A: Computer Program Source Listing	A1

LIST OF FIGURES

<u>Fig. No.</u>		<u>Page</u>
1	Program Organization	5
2	MAIN MENU	9
3	Select Training Mission	11
4	Visual Flight Rules	13
5	NOISEFILE Data Base	15
6	Operation Parameters	15
7	Default Settings	17
8	OMEGA10	17
9	Calculating Noise Levels	18
10	Input Data Screen	18
11	Noise Distribution	19
12	Plot Menu	22
13	Device Menu	22
14	Schematic Illustrating MTR Symbol Notations	25
15	Subroutine Hierarchy Chart, ROUTEMAP	30
16	COMMON Block Location	39

LIST OF TABLE

<u>Table No.</u>		
1	Variable Definitions Used in ROUTEMAP	26

1.0 INTRODUCTION

Military training operations involving high-speed, low-level flights are routinely carried out by all the flight operation commands: SAC, TAC, MAC, ATC, ANG, and Air Force Reserve as well as by the AFSC for various research and development programs. Aircraft types involved in such operations include current technology single- or multi-engine aircraft operating at high subsonic speeds. Operations of this type are conducted on specially designated Military Training Routes (MTRs), which are continually changed for a variety of reasons. Each new route requires an environmental assessment to determine the community noise impact.

The computer program ROUTEMAP calculates the noise levels on the ground along a route corridor. Program ROUTEMAP requires the Air Force planner to specify the nature of the flight activity for each MTR in question. With this information, ROUTEMAP will:

- Compute the noise levels in decibels,
- Provide an estimate of the probability of high annoyance, and
- Rank the aircraft in terms of highest noise contributors at a designated point across the route.

ROUTEMAP is a menu-driven program that runs on any IBM PC or PC-compatible computer system. ROUTEMAP requires MS DOS Version 2.0 or later, with at least one megabyte of available disk space, at least 640K (kilobytes) of random access memory, and 8087/80287 math coprocessor.

Section 2.0 of this report contains a description of the model formulation and the underlying assumptions. Section 3.0 is a description of the model computer program. Section 4.0 is the user's guide and gives complete instructions for using the program. Section 5.0 is the programmer's guide. It contains descriptions of all subroutines, data configurations, and programming conventions. Section 6.0 is the maintenance guide that describes computer code modifications required if the format of either NOISEFILE or OMEGA10 were to be altered.

2.0 MODEL FORMULATION

Program ROUTEMAP's formulation treats operations under visual and instrument flying rules (VFR and IFR) as two separate models. Operations by the Strategic Air Command (SAC) and the Tactical Air Command (TAC) are indicative of the range of differences seen in MTRs within the United States.

SAC low-altitude, high-speed training missions are always conducted under instrument flying rules (IFR). These missions provide terrain avoidance training during flights to bomb scoring ranges. Operations follow the route centerline with a high degree of precision and have terrain-following stage lengths of between 150 and 250 nautical miles while the minimum altitude above ground level is 300 to 600 feet. SAC operates several dozen routes; these routes can range in length up to 1,000 miles.

TAC conducts missions under visual flight rules (VFR). The objective of these missions is to practice low-altitude point-to-point navigation within a designated MTR. TAC low-altitude operations have stage lengths of 200 to 300 nautical miles during which the minimum altitude above the ground varies between 100 and 500 feet. These flights tend to be within a few hundred miles of home base and are controlled by the local command. TAC has several hundred routes between 100 and 300 miles long.

Program ROUTEMAP models both VFR and IFR routes by treating an individual flight track as a point source moving along a line. Since flight tracks along an MTR seldom follow the exact same trajectory, an MTR can be viewed as a distribution of moving point sources with the resultant cumulative noise level at the ground being of interest.

Measurements on several MTRs (References 1 and 2) have established that a Gaussian distribution in the horizontal plane is the distribution that best describes the spatial activity along an MTR. The impact of flight track dispersion in the vertical plane on sound exposure level has a minimal and often negligible effect compared with dispersion in the horizontal plane. For purposes of the present model, therefore, vertical dispersion is not considered; thus the aircraft tracks are distributed laterally at a constant altitude above the ground.

TAC may have several routes in the same geographical area. These MTRs conceivably could overlap or follow some predominant terrain feature that is separated by several miles. To model the various geometrical combinations of MTRs, ROUTEMAP defines a working space called a corridor. This corridor has a maximum width of 20 miles and imposes the restriction that each MTR must be aligned to each other by an angle

not to exceed 15 degrees. ROUTEMAP will model any combination of aircraft type, flight operating parameters, and location of the MTR relative to the corridor centerline.

ROUTEMAP will only permit an MTR down the center of the corridor when modeling an IFR route. Limiting the operations in this way simulates typical SAC low-level routes.

The noise environment from MTRs is unique in several respects. Events are highly sporadic, ranging from a maximum of five to ten per day to a minimum of a few (less than ten) operations per week or two. This differs from community noise exposure scenarios where operations tend to be continuous or somewhat regular. Individual events are also different from typical community noise sources: the combination of low altitude and high air speeds results in noise signatures with high levels and short durations which can result in very rapid onset that may produce a startle response.

The primary noise metric used in this model is the onset rate-adjusted monthly day-night average A-weighted sound level, L_{dnmr} . This cumulative noise metric was developed in a study that uses the best available existing knowledge (Reference 3). It is based on an integration period equal to one calendar month with the highest number of monthly operations. Future studies may result in a change in the integration period, but this can always be normalized to equivalent monthly operations. For this reason, program ROUTEMAP requires operations on an MTR to be quantified in terms of monthly statistical averages.

Two other noise metrics computed by ROUTEMAP are equivalent continuous sound level, L_{eq} , and day-night sound level, L_{dn} . The L_{eq} is the level of a steady sound which, in a stated time period and at a stated location, has the same A-weighted sound energy as the time-varying sound. The L_{dn} is the equivalent continuous sound level, in decibels, for a 24-hour period from midnight to midnight, obtained after the addition of 10 dB to sound levels from midnight to 7:00 a.m. and from 10:00 p.m. to midnight. Note: ROUTEMAP computes L_{eq} and L_{dn} based on an integration period of a calendar month.

The metric most widely used for noise-compatible land-use planning is L_{dn} . Air Force planning policy includes interpretation of L_{dn} in terms of compatible land use.⁴ This is based on relationships between L_{dn} and the probability of highly annoying the population.⁵ In Reference 3 it is recommended that L_{dnmr} values along MTRs be applied to the same interpretive criteria as L_{dn} values in other circumstances. Based on this procedure, ROUTEMAP also computes the probability of high annoyance corresponding to the computed L_{dnmr} .

3.0 PROGRAM DESCRIPTION

3.1 Computer Program Capabilities and Features

The capabilities and features of ROUTEMAP are as follows:

- Calculates the L_{dnmr} , L_{dn} , and L_{eq} across a flight corridor. Up to 20 different aircraft operational conditions can be specified within this corridor.
- Calculations adjust for the air absorption and the ground attenuation using algorithms that approximate those employed in NOISEMAP.⁶
- Calculates the probability of highly annoyed people.⁵
- Allows for multiple MTRs across a corridor that has a width of $\pm 50,000$ feet. The centerline for each MTR can be positioned anywhere within the corridor.
- Displays the results as a table and plots the corridor noise levels on any user-specified plotting device. Graphical output utilizes PLOT88, a standard commercial subroutine library.
- Ranks the contributors to the total noise exposure at any position along the corridor.
- Uses NOISEFILE and OMEGA10 to compute the sound exposure levels. Both the data base and the program are Air Force standards.

ROUTEMAP's limitation of 20 different flight configurations is due to dimensioned size arrays. These arrays can be increased by changing appropriate dimensions as discussed in Section 5.0. The NOISEFILE data base can be increased from an existing 243 number of entries to 300 entries with the existing dimensions.

3.2 General Description of Program Organization

ROUTEMAP's organization is shown in Figure 1. The arrows indicate the order of access to the various routines and the overall flow of the program. To run ROUTEMAP requires three basic steps: define the operations across the corridor, calculate the sound pressure levels, and display the results. All three of these steps are controlled by the main program and initiated from the MAIN MENU.

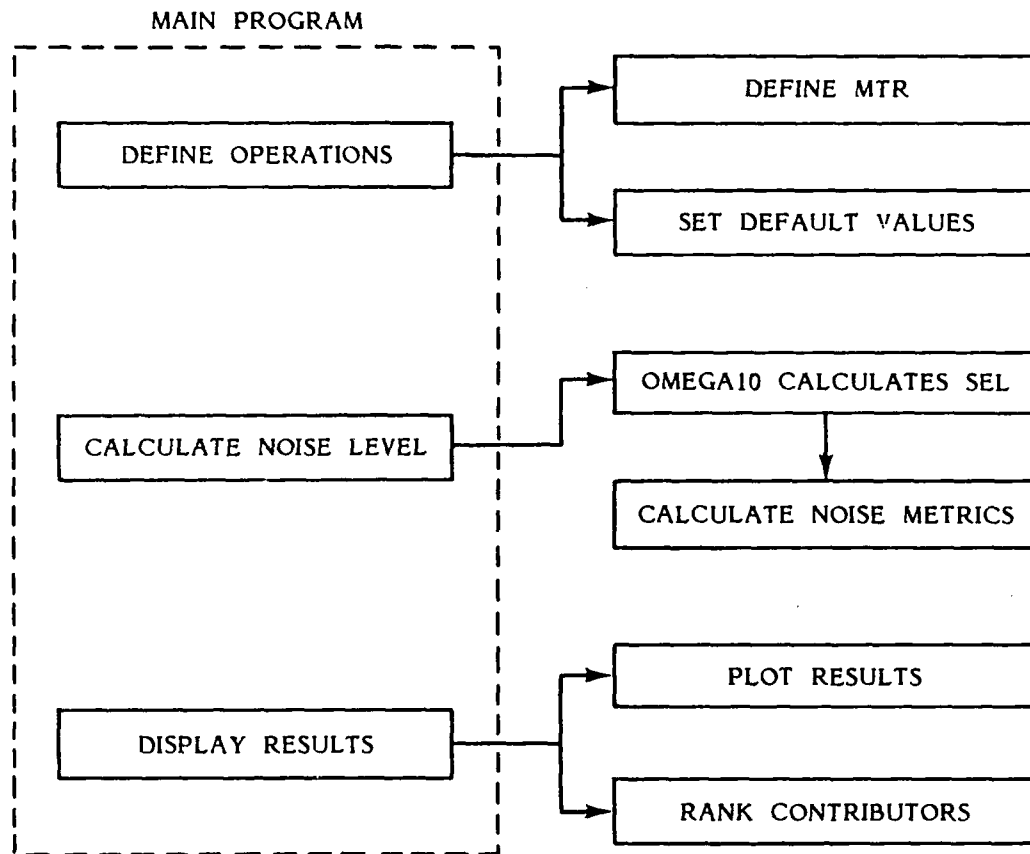


Figure 1. Program Organization.

To define the operations across the corridor, the MTRs must first be drawn on a map using the navigation points. The user must determine which Air Force command flies these routes and the kinds of operations that are performed. Next, the user must draw a corridor that has a width less than 20 miles. The flight tracks must be inside the corridor and be aligned to each other by an angle not greater than 15 degrees. See Section 4.2.5.3 for a description on adding multiple corridor noise metrics if a route is skewed by more than 15 degrees.

The user is now ready to run ROUTEMAP. Pick a point along the corridor centerline for which the noise levels at the ground perpendicular to that point are required. ROUTEMAP will step through a series of window routines that requires data input to describe the operations for the corridor cross-section. When all the operations have been entered, ROUTEMAP returns to the MAIN MENU.

Calculating the noise levels begins with ROUTEMAP writing a file to disk that will serve as an input deck to OMEGA10. ROUTEMAP executes OMEGA10 via a shell routine. OMEGA10 calculates the sound exposure levels (SEL) for the aircraft specified and writes the results to a file on disk. ROUTEMAP reads this file and uses these SEL values when calculating the noise levels across the corridor. At the completion of the calculations, a table is displayed showing the original input conditions and a second table displays the L_{dnmr} across the route.

To display the results, the user can either plot the data or rank the contributors to the total noise level at any point across the corridor. The plotting routines will plot the noise levels and the probability of high annoyance. The devices available are specified in the PLOT88 user manual and includes virtually all popular printers and plotters.

4.0 USER'S GUIDE

This section describes the procedures for properly installing and running ROUTEMAP. Topics addressed in this section include a description on the type of data an Air Force planner needs to gather prior to running ROUTEMAP, and a discussion on how to interpret and present results.

4.1 Getting Started

ROUTEMAP is delivered on two disks. One of the disks contains ROUTEMAP's source code, a batch file, and the executable version of OMEGA10. On a separate disk is a copy of NOISEFILE* renamed as NOISE.

Before installing ROUTEMAP, be sure that the host machine has 1 megabyte of available disk space, 640 kilobytes of random access memory, and a 8087/80287 math coprocessor. Also resident on the host machine must appear a copy of Microsoft FORTRAN, a SPINDRIFT library, and a PLOT88 library. These are commercially available packages; see Section 6.3 for manufacturers' addresses.

PLOT88 is a graphics subroutine library patterned after industry standard CALCOMP style calls. If a different graphics package with similar structure is available, an experienced programmer should be able to easily adapt ROUTEMAP's graphics calls to it.

To install the software it is recommended that a new directory be opened, preferably on a hard disk. This is accomplished by using the following DOS commands:

MKDIR (directory name)

CD (directory name)

Example: MKDIR ROUTE

CD ROUTE

Next, copy MAIN.FOR, OPS.FOR, CALC.FOR, PLT.FOR, RANKODR.FOR, ADDER.FOR, SPINATTR.HDR, ROUTE.BAT, OMEGA10.EXE, and NOISE from the two disks into the newly created directory. To do this, put the disk into drive A, then type the following DOS command:

COPY A:*. * C:

* These are special versions of NOISEFILE and OMEGA10, configured to operate with ROUTEMAP, and are not directly compatible with operation of NOISEMAP.

Here it is assumed that C is the destination drive. After typing this command, type ROUTE to begin the compilation and linking procedures. The new file that will appear on your directory is:

ROUTEMAP.EXE

This procedure completes the installation of ROUTEMAP.

4.2 Running ROUTEMAP

To execute the program, the user must type the program name ROUTEMAP. After pressing RETURN the screen will appear blank and the program will begin an initializing procedure. This procedure will take approximately 30 seconds on a PC-class machine and about 10 seconds on an AT-class machine. Following this procedure, a banner is displayed and the user is instructed to press ANYKEY.

4.2.1 Main Menu

ROUTEMAP is a menu-driven program that interacts with the user. The program structure is centered about a MAIN MENU (see Figure 2) that has seven options. Not all of these options are available to the user at any given time. ROUTEMAP highlights those options which can be activated. There are three main steps to running program ROUTEMAP: first, the operations are defined; second, the noise levels are calculated; and finally, the results are displayed. Each of these options can be activated from the MAIN MENU. A description of each MAIN MENU option follows:

1. Define Operations On Military Training Route – Through a series of window-driven routines, ROUTEMAP queries the user to determine for each aircraft that uses the route:
 - Whether the mission is by instrument, visual flying rules, or special operations;
 - The aircraft type;
 - The aircraft operating power;
 - The vehicle velocity;
 - The altitude above ground level; and
 - The number of daytime and nighttime monthly sorties.

With this information, program ROUTEMAP returns to the MAIN MENU. The menu display highlighted items now include Option 3.

MAIN MENU
Please Make A Selection

- (1) Define Operations On Military Training Route.
- (2) Change Default Settings.
- (3) Calculate Noise Level.
- (4) Plot Noise Level Metrics. Use Data Stored On Disk.
- (5) Rank Order Contributors To Total Noise Exposure.
- (6) Decibel Addition.
- (7) End This Program.

Your Selection :

Figure 2. MAIN MENU.

2. Change Default Settings – Only one screen appears for this option. The default settings are:

- Temperature,
- Relative Humidity, and
- Number of Days in a Month.

If a default setting is unsatisfactory, a change is made by pressing RETURN and making the appropriate update. Note changes in the default setting will only impact the noise levels if Option 2 is chosen prior to pressing Option 3.

3. Calculate Noise Level – This option performs two important functions: first, OMEGA10 is executed using the flight parameters specified in Option 1 and the results from this program are read into ROUTEMAP; second, OMEGA10 data file is then used to calculate the noise level along the ground. Results from this calculation are presented in a table. This table lists the noise level in L_{dnmr} at distances perpendicular to the route.
4. Plot Noise Level Metrics (Use Data Stored on Disk) – ROUTEMAP plots on any designated device both the noise level and the probability of high annoyance. The plot abscissa represents the corridor width and is bounded by the program calculations. Any portion of the corridor can be plotted provided the user remains within a $\pm 50,000$ -foot limit.
5. Rank Order Contributions to Total Noise Exposure – Selection of this option gives the user the capability to rank the contributors to the total noise exposure at any distance perpendicular to the route.
6. Decibel Addition – Computes the sum of two decibel levels.
7. End This Program – Terminates the program and returns the user to DOS.

4.2.2 Configuration of a Route

4.2.2.1 Selecting a Training Mission

Figure 3 shows the first menu to appear after selecting Option 1 from the MAIN MENU. The purpose for this menu is to determine the standard deviation of lateral dispersion about the route centerline. Of the four choices, the Air Force planner will typically pick either 1, 2, or 4. Option 3 entitled "Special Operations" is only used if the Air Force planner intends to use his own value for the standard deviation.

DEFINE OPERATIONS ON MILITARY TRAINING ROUTE

Select A Training Mission

- (1) Visual Flying Rules
- (2) Instrument Flying Rules
- (3) Special Operations
- (4) Retrieve Previously Saved Flight Track File

Your Selection :

Figure 3. Select Training Mission.

Option 1, Visual Flying Rules, are operations normally conducted by the Tactical Air Command. These operations tend to be during daylight hours where navigation is accomplished visually. Navigation on these routes is point to point. The points can be official segment end points, prominent landmarks which are natural to follow, or "delta points" defined by command for training purposes. Multiple MTRs can exist in a single route corridor, governed by a combination of these. Reference 2 contains a detailed discussion of this type of operation.

An Option 1 selection activates the screen shown in Figure 4. The user must decide whether the route has a single dominant MTR or is composed of multiple MTRs in the same immediate vicinity.

Consider as an example two VFR routes located in an area that has dominant terrain features separated by 2 statute miles. The Air Force planner in this situation would select a standard deviation of 1.25 miles for both routes and establish their respective centerlines 2 miles apart. If, however, the routes were located above cattle pastures with no dominant terrain features, the planner may choose to represent both routes as a single corridor with a standard deviation of 2.5 miles.

Strategic Air Command (SAC) conducts training missions under Instrument Flying Rules (Option 2). There are two general types of missions flown by SAC under IFR: missions that simulate an attack on a single site with several runs over a target via a racetrack pattern, and missions where a network of routes pass over several sites to simulate attacks on multiple targets. In either instance these missions are navigated with a high degree of precision. Typically there are no maneuvers under IFR except for navigation corrections. Operations of this kind have a lateral dispersion described by Gaussian distribution with a standard deviation of 0.5 statute mile. Reference 1 contains a detailed description of this type of operation.

Retrieve Previously Saved Flight Track File (Option 4) is used when the Air Force planner has preserved a configuration file on disk. When choosing Option 4 the user is given two opportunities to correctly enter the filename. If ROUTEMAP is unable to find the file specified the first time, a message "file not found" appears on the screen, followed by a directory listing. Pressing ANYKEY causes ROUTEMAP to ask for the desired file. The user can force the directory to be shown by deliberately entering an incorrect name. A selection of Option 4 accompanied by the successful retrieval of data terminates the configuration procedure. The user is then able to proceed from the MAIN MENU in calculating the noise levels.

VISUAL FLIGHT RULES

OPTION (1) - Specific well defined track(s): standard deviation is 1.25 statute miles.

or

OPTION (2) - Multiple scattered tracks: standard deviation is 2.5 statute miles about centerline.

Your Selection :

Figure 4. Visual Flight Rules.

4.2.2.2 Selecting an Aircraft

The "Available Aircraft From NOISEFILE" menu (see Figure 5) is activated following the selection of a training mission. NOISEFILE is a data base containing experimental noise measurements from aircraft operating at different flight configurations. NOISEFILE contains over 200 entries of which there are in excess of 50 different aircraft. Each NOISEFILE entry contains the flight parameters under which the experimental measurements were recorded. Key parameters are shown in this menu to guide the user in the appropriate selection. The user is advised to carefully examine this file using the scrolling feature and make a selection that is closest to the flight operation under investigation. OMEGA10 will use the selected datum as a reference and will scale it to the operating conditions specified in the next menu.

When selecting an aircraft the user may scroll through the file using either the UP and DOWN arrows or the PGUP and PGDN keys. When the desired aircraft and operating configuration has been identified, the user records his/her selection by pressing the ENTER key.

4.2.2.3 Input Operation Parameters

Figure 6 shows the "Operation Parameters" menu. This screen requires the monthly average operating condition for each aircraft. The training mission and aircraft type appear at the top of the screen. Depending upon the mission type, the screen will request input for five or six entries. "Centerline Track Offset", shown in the figure, will appear if the user has specified either VFR training mission having a standard deviation of 1.25 or has selected "Special Operations"; otherwise, this option is not activated.

ROUTEMAP has an allowable corridor width of $\pm 50,000$ feet. A centerline track offset must be within these limitations. If the screen does not request user input for an offset value, ROUTEMAP assumes the MTRs are down the center of the corridor.

Realistic entries for "Power" and "Speed" are essential if ROUTEMAP is to yield correct results. Values must correspond to the planned operating mode. This information can be obtained from the route designers or from experienced aircrews who fly that aircraft on similar operations. The power and speed corresponding to the datum selected in the previous menu are shown in parentheses. THESE ARE NOT DEFAULT VALUES. Their intent is to show the order of magnitude for both the power and speed, and the units in which the data are to be entered.

AVAILABLE AIRCRAFT FROM NOISEFILE				
AIRCRAFT	POWER DESCRIPTION	POWER	DRAW CONFIGURATION	AIR SPEED
E-3A	TAKEOFF POWER	1.83 EPR	GEAR DOWN. 50DEG FLAPS	250
E-3A	APPROACH POWER	1.45 EPR	GEAR DOWN. 50DEG FLAPS	250
E-3A	INTERMEDIATE POWER	1.50 EPR	GEAR DOWN. SPEED BRAKE	250
E-3A	TRAFFIC PATTERN	1.12 EPR	NO DRAG	250
KC-10A	TAKEOFF POWER	110.0 % N1	TAKEOFF POWER	230
KC-10A	APPROACH POWER	79.0 % N1	APPROACH	165

Use arrow keys to scroll desired aircraft into highlighted area.
press RETURN when done.

Figure 5. NOISEFILE Data Base.

VISUAL FLIGHT RULES OPERATION PARAMETERS	
Aircraft Type :E-3A	Configuration : 1
Power (1.83 EPR) :	
Speed (250 knots) :	
Altitude Above Ground Level (feet) :	
Centerline Track Offset (feet) :	
Number Of Daytime Monthly Sorties :	
Number Of Nighttime Monthly Sorties :	
press ESC when all done	

Figure 6. Operation Parameters.

When entering data, the user inputs a value and presses RETURN. If the entry is acceptable ROUTEMAP will then echo the number and advance to the next data entry point. As a check that all entries are properly recorded by ROUTEMAP, the user should press RETURN successive times and observe the cursor movement from one data entry point to the next. If all is well, then press ESC to advance to the next screen.

Two actions occur when the user presses ESC: first, the data are recorded by ROUTEMAP; second, ROUTEMAP flashes the screen "Enter Another Configuration (Yes/No):". A "Yes" entry causes ROUTEMAP to display the NOISEFILE data base screen. A "No" entry causes ROUTEMAP to flash the screen "Save File On Disk (Yes/No):". If a file is saved on disk it may be retrieved at a later time, thus circumventing the task of reconstructing the corridor data base. If minor changes to an existing file are desired, this file can be edited with a text editor and resubmitted to ROUTEMAP.

Following the save file on disk screen the user is returned to the MAIN MENU. This completes the configuration procedure.

4.2.3 Set Default Values

Shown in Figure 7 is the only screen to appear after pressing Option 2 from the MAIN MENU. The default setting screen contains those parameters entered into the calculation that typically do not change from one operation to the next. To change a setting, press RETURN in succession until the cursor is positioned at the entry to be revised. Now enter the value and press RETURN. If the entry is accepted by ROUTEMAP, the value will be echoed. When satisfied with the default settings, press ESC to return to the MAIN MENU.

Note: Setting the default values will ONLY impact the noise levels if Option 2 is updated prior to pressing Option 3 from the MAIN MENU.

4.2.4 Calculating the Noise Levels

Figures 8 through 11 show the succession of screens that will appear after pressing Option 3 from the MAIN MENU.

"RUNNING OMEGA10" is the first message flashed on the screen. If the operation parameters entered during the configuration process are realistic, then OMEGA10 will successfully terminate without an error message being displayed. If an error were to

DEFAULT SETTINGS

Temperature In Degrees Fahrenheit : 59

Relative Humidity : 70

Number Of Days In Month : 30

press ESC when all done

Figure 7. Default Settings.

MAIN MENU
Please Make A Selection

(1) Define Operations On Military Training Route.

RUNNING OMEGA10

(5) Rank Order Contributors To Total Noise Exposure.
(6) Decibel Addition.
(7) End This Program.

Your Selection :

Figure 8. OMEGA10.

MAIN MENU
Please Make A Selection

(1) Define Operations On Military Training Route.

CALCULATING NOISE LEVELS

(5) Rank Order Contributors To Total Noise Exposure.

(6) Decibel Addition.

(7) End This Program.

Your Selection :

Figure 9. Calculating Noise Levels.

THE INPUT DATA							
Standard Deviation = 1.25 Statute Miles							
Temp = 59		Humidity = 70		Num Days In Month = 30			
Config	Aircraft	DayOps	NiteOps	Altitude	Speed	Power	Offset
1	F-4	20.0	5.0	750	200	86.00	.528E+4
2	F-4	25.0	5.0	750	200	86.00	-.528E+4
3	F-16	30.0	.0	1000	300	85.00	.528E+4
4	F-16	25.0	.0	750	300	85.00	.528E+4
5	F-16	20.0	.0	500	300	85.00	.528E+4
6	F-5E	15.0	.0	750	325	86.00	-.528E+4
7	F-15	20.0	.0	500	280	73.50	-.528E+4
8	F-14	30.0	.0	1500	350	82.50	.000
9	C-141	30.0	.0	2000	300	1.52	.000

Figure 10. Input Data Screen.

ROUTENAP Version 1.0									
TOTAL Ldnmr AT DISTANCES PERPENDICULAR TO ROUTE									
Dist(Kft) dB	Dist(Kft) dB	Dist(Kft) dB	Dist(Kft) dB	Dist(Kft) dB	Dist(Kft) dB	Dist(Kft) dB	Dist(Kft) dB	Dist(Kft) dB	Dist(Kft) dB
-50	0	-30	0	-10	53	10	53	30	0
-49	0	-29	0	-9	54	11	53	31	0
-48	0	-28	26	-8	54	12	52	32	0
-47	0	-27	28	-7	55	13	51	33	0
-46	0	-26	32	-6	55	14	51	34	0
-45	0	-25	35	-5	55	15	50	35	0
-44	0	-24	37	-4	55	16	49	36	0
-43	0	-23	39	-3	55	17	47	37	0
-42	0	-22	41	-2	56	18	46	38	0
-41	0	-21	42	-1	56	19	45	39	0
-40	0	-20	44	0	56	20	44	40	0
-39	0	-19	45	1	56	21	42	41	0
-38	0	-18	46	2	56	22	41	42	0
-37	0	-17	47	3	55	23	39	43	0
-36	0	-16	48	4	55	24	37	44	0
-35	0	-15	49	5	55	25	35	45	0
-34	0	-14	50	6	55	26	32	46	0
-33	0	-13	51	7	55	27	28	47	0
-32	0	-12	52	8	54	28	26	48	0
-31	0	-11	53	9	54	29	0	49	0

Figure 11. Noise Distribution.

occur, ROUTEMAP will display one of several error messages. If the error is fatal, ROUTEMAP will terminate. To diagnose the nature of the error, examine OMEGA10.OUT file and check to see if a file IN and OUT were indeed created.

OMEGA10 run-time is typically 15 seconds per aircraft type. If six different kinds of aircraft operated on a given corridor, then allow about a minute and a half for OMEGA10 to calculate the sound exposure levels (SEL).

"CALCULATING NOISE LEVELS" is the second message which is flashed on the screen. At this time, ROUTEMAP is calculating the actual noise level on the ground for a corridor having a width of 100,000 feet. The run-time for this procedure ranges from several minutes to as long as an hour. The amount of time required is dependent upon the number of routes, their respective width, and the number of configurations.

At the completion of this calculation, a screen is displayed showing the original input data (see Figure 10). To make a hard copy of this screen for your records, press PRINT SCREEN. To continue to the next screen, press ANYKEY.

Next, ROUTEMAP displays the noise level in L_{dnmr} across the route (see Figure 11). As before, a hard copy of this data is made by pressing PRINT SCREEN. To preserve this data on disk for future plotting, answer "Yes" when asked to save the file. Otherwise, the data is lost at the termination of ROUTEMAP or if a new corridor is defined.

The noise distribution table, Figure 11, contains the information generally needed for preparation of an Environmental Assessment. The planner is usually interested in the maximum level and the location of the L_{dnmr} 65 dB contours. In the example shown, the maximum L_{dnmr} is 56 dB. There is no 65 dB region. If there were a need to show L_{dnmr} 50 dB contours, these are at -14,000 feet and 15,000 feet relative to the corridor centerline, and would be drawn at these distances and parallel to the centerline.

4.2.5 Analysis of Results

4.2.5.1 Plotting

Option 4, "Plot Noise Level Metrics", activates the plotting routines from the MAIN MENU. The first screen that appears is dependent on ROUTEMAP's current status. If there are no noise levels to be plotted because ROUTEMAP has not made the calculations prior to the selection of Option 4, then ROUTEMAP assumes that the file to be plotted must reside on disk. In this situation ROUTEMAP queries the user for the filename.

The second screen to appear is shown in Figure 12. The screen is divided into three sections: General, Noise Level, and Annoyance. In the General section the user inputs the plot title and the range of distances across the corridor to be plotted. This subdivision is entitled "General" because these parameters will appear in both the noise level and probability of high annoyance plots.

Next, the user must choose which noise metrics are to be plotted, whether symbols are to be used, and if all of the curves are to be collapsed onto one plot. If the user has selected more than one noise metric to be plotted, it is recommended that symbols are used to distinguish the curves.

Finally, the user indicates whether a probability of high annoyance curve is to be plotted for the corridor.

Once the user has completed data entry, press ESC to proceed to the next screen.

The third screen queries the user to select a plotting device (see Figure 13). Five devices are available to the user from this menu. If the user device does not appear on this screen, then press Option 6, "Other". The user must now enter the IOPORT and MODEL number. The user should consult the PLOT88 user manual to ascertain the appropriate IOPORT and MODEL number. The scale factor selection is done by trial and error. Note that the present devices use scale factors between 0.7 and 1.2.

At the completion of the third screen, the plots will be drawn on the appropriate device and the user will be returned to the MAIN MENU. The user has the option to display the data under different or the same conditions by pressing Option 4 from the MAIN MENU.

The plots generated by this option can be useful to a planner in developing a detailed understanding of the route noise environment. The plots are also suitable for inclusion in an Environmental Assessment if the presentation would be enhanced by that level of detail.

4.2.5.2 Rank Order Contributors

When route noise levels are high enough to cause a potential problem, it is useful to know the relative noise contribution for each type of operation. This information can form the basis of noise abatement via changes to route operation.

PLOT RESULTS

GENERAL
Enter Title :

Enter The Distances Across MTR To Be Plotted.
Minimum :-50000.
Maximum :50000.

NOISE LEVEL
Do You Want To Plot Ldnmr (Yes/No) :
Do You Want To Plot Ldn (Yes/No) :
Do You Want To Plot Leq (Yes/No) :
Do You Want Symbols Used (Yes/No) :
Do You Want To Collapse These Curves
Onto One Plot (Yes/No) :

ANNOYANCE
Do You Want To Plot The Probability Of
High Annoyance (Yes/No) :

press ESC when all done

Figure 12. Plot Menu.

AVAILABLE PLOTTING DEVICES

- (1) Epson FX-80 Or FX-85 Printer
- (2) HP 7475A Plotter
- (3) Hercules Graphics Card (HGC)
- (4) IBM Enhanced Graphics Adapter (EGA)
- (5) IBM Color Graphics Adapter (CGA)
- (6) Other

Your Selection :

Figure 13. Device Menu.

Option 5 from the MAIN MENU will permit the user to rank order the contributors to the total noise exposure at any designated position across the corridor. The user specifies the distance from the corridor centerline to the ground location point of interest to be ranked. This value must be within the $\pm 50,000$ -foot limit. Once the screen displaying rank contributions is displayed, press ANYKEY to return to the MAIN MENU.

4.2.5.3 Combination of Noise From Separate Routes

When the angle between two routes exceeds 15 degrees, or if incompatible operations (e.g., IFR versus VFR) preclude treating them as a single corridor, ROUTEMAP must be applied separately and the results combined. This section describes the method used to add two corridors together using ROUTEMAP's tabulated results. A step-by-step description follows:

1. Lay out the routes on a map. Draw the corridors and respective centerlines. Locate on the map the ground location for which the noise level is to be determined. Call this point the receiver position.
2. Draw two lines on the map which are perpendicular to the corridor centerline and intersect the receiver position.
3. Run ROUTEMAP for each corridor to obtain the noise levels at points perpendicular to each corridor's centerline.
4. PRINT SCREEN the tabulated results from ROUTEMAP.
5. Using the two tables, sum the noise contribution from both corridors. This capability is provided as an option from the MAIN MENU. The equation employed is as follows:

$$\begin{array}{c} \text{Noise Level} \\ \text{at} \\ \text{Receiver Position} \end{array} = 10 \log_{10} \sum_{i=1}^{\text{Number of Corridors}} 10^{\left[L_{dnmr}^{(\text{position})} / 10 \right]}$$

This procedure can be applied to the analysis of a particular noise-sensitive receptor. It can also be applied to a number of points around two corridors, and the results used to plot combined noise contours for use in planning or an Environmental Assessment.

5.0 PROGRAMMER'S GUIDE

5.1 Convention and Annotation

ROUTEMAP is written in standard MicroSoft FORTRAN 77. All real variables are single precision, REAL*4. All fixed point variables are INTEGER*4. ROUTEMAP makes extensive usage of SPINDRIFT and PLOT88 library routines, commercial packages which are PC and MS-DOS specific. A PLOT88 Site License is required to run ROUTEMAP. The program source and executable code are fully transportable to any IBM PC-compatible machine. Compilation and linking of source code to produce executable code (Section 6.3) requires licensed copies of Microsoft FORTRAN Version 4.01 or later, the SPINDRIFT library, and the PLOT88 library.

Extensive use is made of subroutines in order to provide reasonably well-structured code. Variable names are consistent between subroutines, although in several cases the mnemonics are similar but not identical. Executable statement labels begin with 10 in each program; format labels begin with 1000. Text strings used by SPINDRIFT are incorporated into the code and window identification numbers begin with 1 in each subroutine.

5.2 Definition of Symbols

Symbols defined herein are used throughout this report and program ROUTEMAP. A sketch of the MTR and the use of the symbol notation is shown in Figure 14. The symbols listed in Table 1 are a complete list of the variables used in ROUTEMAP. Dummy variables used for data handling and integer counters are not included in this list. Symbols that are intended to be arrays will be listed with their array dimensions. Variables I, J, K, and L will serve as running indices. Variables I and J are usually but not always used to represent:

- I – the aircraft index, and
- J – the offset distance.

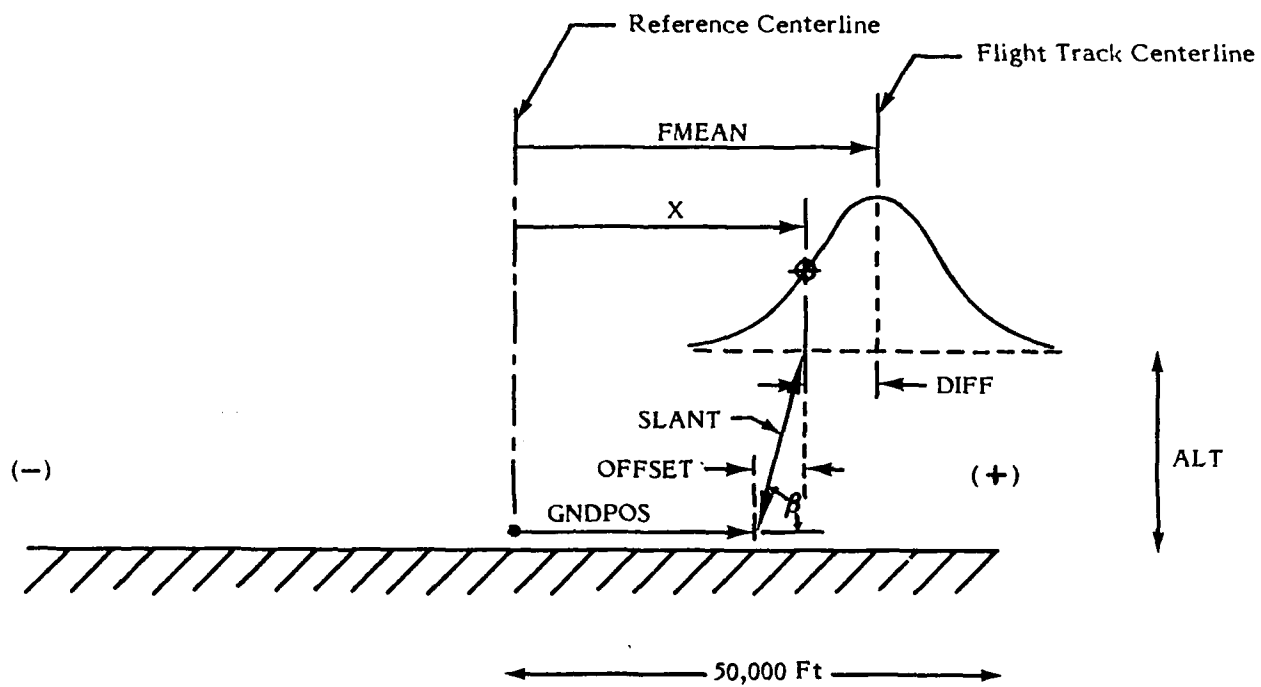


Figure 14. Schematic Illustrating MTR Symbol Notations.

Table 1
Variable Definitions Used in ROUTEMAP

<u>Symbol</u>	<u>Description</u>
ACC(300)	Aircraft code from NOISEFILE.
AIRSPD(300)	Aircraft speed from NOISEFILE.
ALT(20)	Aircraft altitude in feet (see Figure 14).
ATTEN	Lateral attenuation in dB.
BETA	Elevation angle.
DIFF	Distance as measured from center of the Gaussian distribution to point along distribution that is presently being evaluated in feet (see Figure 14).
DRAG(300)	Aircraft drag configuration description from NOISEFILE.
FACT	Scale factor for plotting routine.
FINTEN(20,147,3)	A-weighted mean square pressure.
FLDNM(201)	L_{dn} value in dB for nominal monthly integration period.
FLDNMR(201)	L_{dnmr} value in dB for nominal monthly integration period.
FLEQ(201)	L_{eq} value in dB for nominal monthly integration period.
FMAX	Maximum X axis coordinate selected by user.
FMEAN(20)	Flight track centerline measured from reference position centerline in feet (see Figure 14).
FMIN	Minimum X axis coordinate selected by user.
GNDPOS	Absolute position along flight track with respect to reference coordinate in feet (see Figure 14).
ICODE(20)	An integer array used to point to the NOISEFILE record that the user has picked for the corridor configuration.
ICOL()	Cursor column position. Closed parentheses are used to indicate the array dimensions vary between subroutines.
IOPORT	INTEGER*2 variable used by PLOT88.

Table 1 (Continued)

<u>Symbol</u>	<u>Description</u>
IROW()	Cursor row position. Closed parentheses are used to indicate the array dimensions vary between subroutines.
ISWITCH	Integer variable that points to the mission type.
MAX	The total number of aircraft records stored in NOISE-FILE. Not to be confused with the number of lines in file.
MODEL	INTEGER*2 variable used by PLOT88.
MONTH(20)	Number of days in the month.
NAME(300)	Aircraft name from NOISEFILE.
NUMAIR	Number of aircraft type.
NUMD(20)	Number of daytime monthly sorties.
NUMN(20)	Number of nighttime monthly sorties.
OFFSET	Distance from receiver location to point along Gaussian distribution that is presently being evaluated (see Figure 14).
ONSET	Onset rate in dB per second. The Onset rate is defined by slope of the A-weighted time history between points 5 dB above the ambient and 5 dB below the maximum level.
ONPEN	Onset rate adjustment in dB, which is applied only when the maximum A-weighted sound level exceeds the ambient level by at least 15 dB.
OPCR(300)	Operation power code from the NOISEFILE data base.
POWDES(300)	Aircraft power description from NOISEFILE.
POWSET(300)	Aircraft power setting from NOISEFILE.
POWUNT(300)	Aircraft power setting corresponding units from NOISEFILE.
PROB	Probability of aircraft flying at specified X position along military route.
RANK(20,201)	L_{dnmr} values for each individual aircraft at 500-foot increments across corridor.

Table 1 (Continued)

<u>Symbol</u>	<u>Description</u>
SCN(1024)	Scratch pad used to construct screens, prior to their display.
SEL(20,25)	Sound exposure level for a specific aircraft, power setting, and airspeed, in dB.
SIGMA	Standard deviation of track width in feet.
TITLE	Plot title.
TOGGLE	Logical*1 variable that tells the plotting program whether the user has called the main plotting menu before.
VEL(20)	Aircraft airspeed in feet per second.
X	Distance as measured from reference position to point along Gaussian distribution that is presently being evaluated (see Figure 14).
XARRAY(203)	Absolute grid coordinates used to plot X axis.
XFINL	Upper integral limit for normal distribution calculation.
XINIT	Lower integral limit for normal distribution calculation.
YARRAY(203)	Absolute grid coordinates used to plot Y axis.

5.3 Subroutines

ROUTEMAP subroutine hierarchy chart is shown in Figure 15. The main program is described in Section 3.2. The following subsections describe each subroutine, presented in the order they appear in the hierarchy chart. The descriptions here and in Section 3.2 are intended to be read together with the source listing in Appendix A if a detail understanding of the code is desired.

5.3.1 INITIAL

This subroutine initializes the SPINDRIFT library, the display, and the keyboard. Subroutine INITIAL captures the present screen image and stores it in array INTSCN. The keyboard numberlock is cleared. The video mode is determined and set to 80x25 color if it is not a monochrome display.

5.3.2 OPEN

This routine draws the opening banner. The zoom feature is accomplished by repeatedly drawing windows of increased size. After each window is drawn, a two-tone harmonic is sounded using a SPINDRIFT library routine. A call made to subroutine PAUSE, draws in the lower left-hand corner a window instructing the user to press ANYKEY.

5.3.3 PAUSE

This routine draws a window in the lower left-hand corner of the screen with a message to press ANYKEY.

5.3.4 MENU(IVAL,FLAG)

This routine draws the MAIN MENU. Logical array FLAG is passed to subroutine MENU through the call statement. Array FLAG indicates the present status of ROUTEMAP. If FLAG is true, then an option appearing in the MAIN MENU can be activated. Subroutine MENU uses the FLAG array to set the brightness attribute when displaying the menu. After the menu is displayed, the user is queried for a selection. If a key other than a number key is depressed the keystroke is ignored. Following a successful entry, IVAL is returned to the main program.

ROUTEMAP	MAIN	INITIAL		
		OPEN	PAUSE	
		MENU		
		OPS1		
		OPS2	PAUSE	
		OPS3		
	PROGRAM	CALC	GAUSS	
			FDMI	
	PLTMAIN		PLTMENU	PAUSE
			DEVICE	
			PLTSPL	DRAW
			PLTAPP	DRAW
		RANKODR		
		ADDER		
		FINISH		

Figure 15. Subroutine Hierarchy Chart, ROUTEMAP.

5.3.5 OPS1(ISWITCH)

This subroutine determines the type of training mission. The screen entitled "Define Operations On Military Training Route" is displayed with the four options. The computer waits until a key is pressed. If the key pressed is not a number key then the keystroke is ignored. Out-of-range entries are ignored. If the user requests the retrieval of a previously saved data base then the user is queried for the filename. A "file not found" message is displayed only once before returning the user to the previous screen. Integer variable ISWITCH is passed from subroutine OPS1 to the main program.

5.3.6 OPS2(ISWITCH)

This subroutine determines the operating parameters across the corridor. The subroutine is divided into three major sections. First, subroutine OPS2 queries the user for the standard deviation. Second, the aircraft type is recorded. Last, the operating parameters for the designated aircraft are determined. The program iterates between last two sections until all aircraft configurations are entered or twenty configurations have been specified.

Integer variable ISWITCH passed from the main program instructs subroutine OPS2 which training mission has been specified. A computed GO TO statement decides which screen is to be displayed. If ISWITCH is one, then the user must select a standard deviation (SIGMA) of 1.25 or 2.5 statute miles. An ISWITCH value of 2 causes an information banner to be displayed and the value of SIGMA is set to 0.5 statute mile. ISWITCH equal to 3 indicates that the user has chosen to enter his own value for SIGMA. If SIGMA exceeds 2.5 statute miles, then the message "Standard Deviation Out of Range" is flashed and the user is queried to enter a new value. All values of SIGMA are converted to feet before the next screen is displayed.

Integer variable NUMAIR counts the number of aircraft configurations. Its value is set to zero prior to entering loop 80. Next, NUMAIR is incremented by 1 and the screen entitled "Available Aircraft From NOISEFILE" is displayed. COMMON block NOISE passes to subroutine OPS2 character arrays that contain key descriptive parameters from NOISEFILE. These character arrays are incrementally displayed when the arrow or page keys are pressed. Briefly, the scrolling feature operates by having the keyboard continually sampled to determine if a key has been pressed and, if so, whether it was a key that constitutes a screen action. The most important aspect of this routine is

recording which aircraft file in the NOISEFILE data base has been selected. Using an integer counter variable I, the NOISEFILE recorded number is stored in integer array ICODE.

When drawing the operation parameter screen, variable ISWITCH is used to display the heading that describes the mission type. Entry labels are drawn to the screen. If integer variable INPUT is equal to 1, then the program will accept a centerline track offset. An internal read and a print window library call are used to draw on the screen: the aircraft name, the power setting, and the air speed. Data entry is achieved by continually sampling the keyboard, checking to see if a key has been pressed and, if so, whether it was a number, a RETURN, or an ESC key. If the key is a numerical key, the entry is read. The cursor is positioned to the beginning of the line; the entry is then echoed.

Subroutine OPS2 flashes a message asking if another configuration is desired. If the answer is yes, then program execution continues at loop 80. If the answer is no, subroutine OPS2 determines whether a file is to be saved on disk and the filename. Control is returned to the main program.

5.3.7 OPS3

This routine will change the default settings. The default parameters are passed to OPS3 through COMMON block AIRPAR. Entry labels are drawn and the current default settings are written to the screen. The keyboard is continually sampled to determine if a key has been pressed. A keystroke will cause a screen action only if an ESC, RETURN, or numeric key is pressed.

5.3.8 CALC

This routine calculates the noise level across the corridor. There are two parts to this program: a table is constructed of adjusted mean pressure versus offset per aircraft; in the second part, pressure squared weighted by the probability of an event along the track per aircraft are summed for each ground position. Knowing the pressure squared sum across the track, it is a simple calculation to determine the values of the L_{dnmr} , L_{dn} , and L_{eq} noise metrics.

The table construction begins with an array of 25 sound exposure levels (SEL) tabulated as a function of profile distance. Using these SEL values, a table is constructed for the slant range for offset values ranging from 0 to 73,000 feet versus SEL.

The pressure squared versus offset table accounts for both attenuations and penalties. Two attenuation mechanisms considered are air and ground absorption. OMEGA10 uses the humidity and temperature data from subroutine OPS3 to determine the air absorption coefficient and, in turn, the sound exposure level reduction. The sound exposure levels stored in array SEL are passed to CALC via COMMON block AIRPAR. Lateral attenuation is accounted in subroutine CALC only when BETA, the elevation angle, is less than 45 degrees. This equation is an approximation of the lateral attenuation methodology used in NOISEMAP 6.0 (Reference 6).

Subroutine CALC accounts for increases in noise intrusion from high-rate, low-altitude aircraft by calculating the Onset rate as described in Reference 7. The exact form of this equation was determined by curve fitting Onset Rate versus Speed, Altitude, Offset, and SEL for over 450 experimental cases. If the rate is greater than 15 dB/sec and less than 30 dB/sec, the SEL values are penalized by:

$$\text{Onset Penalty} = 16.6 \log_{10} (\text{Onset}/15.0)$$

If the onset rate is greater than 30 dB/sec, then a 5 dB penalty is built into the tabulated values.

The second part of subroutine CALC determines the pressure squared along the track, accounting for span-wise variability in aircraft trajectory and the different types of aircraft flying along the corridor. The algorithm consists of three nested loops. The innermost loop integrates a Gaussian distribution over a range of $-2*\text{SIGMA} - 10,000$ feet to $2*\text{SIGMA} + 10,000$ feet, while the second loop iterates on the different aircraft specified by the subroutine OPS2. The running index for the outer loop is ground position which ranges between $\pm 50,000$ feet.

5.3.9 GAUSS(XINIT,XFINL,FMEAN,VAR,PROB)

This subroutine computes the probability over an interval. XINIT and XFINL are the lower and upper limits in the integration. VAR is the variance for the user-specified training mission. PROB is the probability computed for indicated conditions.

5.3.10 FDMI(I,J,K,P)

This FUNCTION uses finite difference techniques to interpolate the tabulated sound intensity table stored in FINTEN. Array FINTEN is passed from CALC through COMMON block SPACE. Four integer variables are passed to FINTEN from CALC; variable I specifies the configuration number, variable J points to the present corridor position, variable K indicates which of the three noise metrics is to be used, and variable P is used as the pivotal point for the interpolation.

The difference equation used in the analysis depends on pointer J. If the calculation is at the top or bottom of the difference table, the equation used is either forward or backward technique, respectively. At any other position on the table, the technique employed is Stirling central difference.

5.3.11 PLTMAIN(SWITCH,TOGGLE)

PLTMAIN is the main plotting program for ROUTEMAP. Logical variables SWITCH and TOGGLE are status variables passed from the main program. SWITCH equal to true indicates that the plotting data resides on disk. When SWITCH is true, PLTMAIN queries the user for a filename. A "file not found" message is displayed only once. The user is given two tries to enter the correct filename before PLTMAIN returns to the main program. If SWITCH is equal to false, noise level data is passed to the subroutine through COMMON block RESULT.

Subroutine calls are made to PLTMENU and DEVICE to determine the plot parameters and attributes. These parameters in turn are passed between plotting subroutines through COMMON block PLT. The FLAG array instructs which plots the user requests. The meaning of FLAG equal to true follows:

- FLAG(1) – plot L_{dnmr}
- FLAG(2) – plot L_{dn}
- FLAG(3) – plot L_{eq}
- FLAG(4) – use symbols for any of the above three curves
- FLAG(5) – collapse curves on to one plot
- FLAG(6) – plot probability of high annoyance

Subroutine PLTSPL draws the noise level curves if any one of the first three FLAG elements is true. A true value for FLAG(6) enables subroutine PLTAPP which draws the probability of high annoyance curve.

5.3.12 PLTMENU(SWITCH,TOGGLE)

This subroutine determines the plotting parameters. The techniques employed in this subroutine are analogous to previous menus that require data entry. Data is passed from this subroutine through the PLT COMMON block. Logical variables TOGGLE and SWITCH are used to preserve and display previously entered plot menu parameters only when the sound intensity data is not acquired by reading a file from the disk.

5.3.13 DEVICE

This subroutine determines the IOPORT and MODEL number. A screen is displayed offering a selection from five possible devices. Variable ICOUNT records the option selection. If ICOUNT is less than or equal to 5, then data arrays IO and MOD assign a value to IOPORT and MODEL. If ICOUNT equals 6, then a screen is displayed requesting the user to input the model number, IOPORT number, and scale factor. COMMON block PLT passes the parameters.

5.3.14 PLTSPL

This subroutine draws the noise level plots. COMMON blocks RESULT and PLT pass the noise levels and plotting parameters to the subroutine. FMIN and FMAX are the corridor coordinate values and are truncated to an integer multiple of 500 feet. NPTS is the number of points used to plot the noise level curves.

PLOT88 calls are used extensively throughout the subroutine. These subroutine calls have been patterned after the industry standard CALCOMP style calls and are therefore interchangeable with other similar graphics packages, provided adaptations are made to ROUTEMAP's graphics calls. The program initializes PLOT88 using the IOPORT and MODEL number. The plot is scaled to correct for different size plotting areas. The program centers and draws the plot title by determining the number of characters and, knowing a character width, measures to the left of the centerline the appropriate distance. Logical variable FLAG locations 1 through 3 determine which curves are to be drawn. FLAG variables 4 and 5 are used to determine whether symbols are to be drawn and if the curves are to be collapsed onto one plot.

5.3.15 PLTAPP

This subroutine draws the probability of high annoyance. The programming logic used for plotting parameter initialization and the drawing of axes is identical to subroutine PLTSPL. DO loop 10 contains the probability calculations.

5.3.16 DRAW(XARRAY,YARRAY,NPTS,FLAG1,INTEQ)

This subroutine draws a line. Variables XARRAY and YARRAY contain the absolute grid coordinates for the line. NPTS is equal to the number of line segments minus one. A true value for FLAG1 instructs the subroutine to include symbols when drawing the line. INTEQ is the variable defining the symbol to be drawn.

Stored in XARRAY and YARRAY, position NPTS plus one is the minimum coordinate value and in position NPTS plus two is the separation between tic marks. These values are required to determine the X and Y plotting position. The program mechanics is based on a series of move and draw commands using PLOT88 calls. See PLOT88 reference manual for detail description on plotting subroutine library calls.

5.3.17 RANKODR

This subroutine ranks the contributors to the noise exposure at a specified corridor position. Logical array FLAG is assigned a true value. The first screen requests the distance to be ranked across the corridor. A message is flashed if the distance, variable DIST, exceeds the corridor limits of $\pm 50,000$ feet. Integer variable IVAL points to the vector in the RANK matrix whose values for L_{dnmr} correspond to the distance selected. COMMON block RESULT passed matrix RANK to subroutine RANKODR. The RANK matrix contains the L_{dnmr} values for each aircraft type at 500-foot incremental positions along the corridor.

The rank contributors screen headings are drawn. The distance value selected is echoed on the screen. Loop 80 contains the sorting algorithm. The logic to this algorithm is a derivative of a bubble sort, the difference being that array values are not assigned; instead, a flag is switched and the result is printed on the screen. A FLAG equal to true means the L_{dnmr} for that particular aircraft has not yet been identified as a ranked element. The routine begins at loop 50 where it looks for the first element in the list that has not been ranked. Once it is found, its value is assigned as the maximum value,

VALMAX. ITRACK points to where VALMAX is located in the RANK matrix. Loop 60 looks through the RANK vector to identify a noise value greater than VALMAX that has as of yet not been listed as a rank element. If there is a noisier aircraft that meets this condition, then both VALMAX and ITRACK are reassigned. Upon the completion of the DO 60 loop, the ITRACK value points to the FLAG which is set to false, the annoyance calculated, and the results printed on the screen. Loop 80 continues to iterate until all aircraft are ranked and in turn all FLAGs are set equal to false.

5.3.18 ADDER

This subroutine adds the noise level for two decibel numbers. Data entry is achieved by continually sampling the keyboard, checking to see if a key has been pressed, and, if so, whether the key was a number, ESC, or RETURN key. If the key is a numeric key, the entry is read only if the value is greater than or equal to zero or less than or equal to 140 dB. When both the entries are read, the sum is computed and echoed beside the equality sign.

5.3.19 FINISH

This routine sets the computer status to its original mode prior to the execution of ROUTEMAP. Displayed on the terminal is the screen image stored in INITSCN.

5.3.20 BLOCK DATA

Relative humidity, air temperature, and the number of days in a month are assigned in the data block. All other real and integer COMMON arrays are set equal to a zero value.

5.4 COMMON Blocks

The following COMMON blocks and their contents are used in ROUTEMAP:

- /AIRPAR/ ALT(20), FMEAN(20), ICODE(20), IREL, ITEMP, MONTH, NUMAIR, FNUMD(20), FNUMN(20), POWER(20), SEL(20,25), SIGMA, VEL(20)
- /FRAME/ SCN(1024)
- /NOISE/ MAX, ACC(300), AIRSPD(300), DRAG(300), NAME(300), OPCR(300), POWDES(300), POWSET(300), POWUNT(300)

- /PLT/ TITLE, FMIN, FMAX, FLAG(6), IOPORT, MODEL, FACT, ICOUNT
- /RESULTS/ FLDNMR(201), FLDNM(201), FLEZ(201), RANK(20,201)
- /SPACE/ FINTEN(20,147,3)

Figure 16 is a chart showing the location of the COMMON blocks in the programs. An open circle indicates the COMMON is in the program. A solid circle indicates that some or all of the contents of the block are defined by data statements within that routine.

	AIRPAR	FRAME	INIT	NOISE	PLT	RESULT	SPACE
MAIN	●	●		●		○	
INITIAL		●	●				
OPEN		●					
PAUSE		●					
MENU		●					
OPS1	●	●					
OPS2	●	●		○			
OPS3	●	●					
CALC	○					●	●
GAUSS							
FDMI							●
PLTMAIN		●			○	●	
PLTMENU		●			●		
DEVICE		●			●		
PLTSPL					○	○	
PLTAPP					○	○	
DRAW							
RANKODR	○	●		○		○	
ADDER		●					
FINISH		●	○				
BLOCK	●					●	

● Contents in COMMON defined.

○ COMMON in program.

Figure 16. COMMON Block Location.

6.0 MAINTENANCE MANUAL

Program maintenance is limited to modifying ROUTEMAP's input/output (I/O) format structure to reflect any format I/O change in OMEGA10. The addition of new records to NOISEFILE will not impact ROUTEMAP provided the format structure remains the same and the total number of records does not exceed 300. Presently NOISEFILE contains 243 records.

6.1 OMEGA10 Maintenance

The OMEGA10 version used with ROUTEMAP has been modified by AAMRL to accommodate low-altitude flights. This modification resulted in an output file that had the addition of two more cards per aircraft. OMEGA10 has been changed to facilitate data entry. The following items document those changes:

- WRITE(1,9000), WRITE(1,9100), and WRITE(1,9200) are now comment statements.
- READ(1,9300) SETUPFL, READ(1,9300) NOMAPFL, and READ(1,9300) NOISEFL are also comment statements.
- TAPE 5 filename is assigned as "OUT".
- TAPE 3 filename is assigned as "IN".
- TAPE 7 filename is assigned as "NOISE".
- TAPE 6 filename is assigned as "OMEGA10.OUT".

This completes the changes made to OMEGA10.FOR.

6.2 ROUTEMAP Maintenance

If the I/O of OMEGA10 were modified, then ROUTEMAP must reflect these changes. The input deck for OMEGA10 is prepared by ROUTEMAP in the MAIN program beginning at line 154. The input deck is written to TAPE 2 in ROUTEMAP having OUT as an assigned filename.

The output deck from OMEGA10 has an assigned filename of IN and is written to disk. Using an OPEN statement and making the assignment filename equal to IN for TAPE 3, ROUTEMAP reads OMEGA10's output deck. This instruction set appears in ROUTEMAP beginning at line 199 in the MAIN program.

6.3 Compiling and Linking

To compile ROUTEMAP, the user must use Microsoft FORTRAN Version 4.01 or later. To link the object code, PLOT88 and SPINDRIFT libraries must be resident on the system. The command procedure used to compile and link ROUTEMAP follows:

- FL /c /Gt MAIN.FOR
- FL /c /Gt OPS.FOR
- FL /c /Gt CALC.FOR
- FL /c /Gt PLT.FOR
- FL /c /Gt RANKODR.FOR
- FL /c /Gt ADDER.FOR
- LINK/E MAIN+OPS+CALC+PLT+RANKODR+ADDER /SEG:1024, ROUTEMAP, ,
PLOT88 SPINLAB7

This procedure is embodied in the batch file ROUTE.BAT.

ROUTEMAP can be run on a system with only 512 kilobytes of memory if it is linked into two smaller programs. The procedure is as follows:

- LINK/E MAIN+OPS+CALC+RANKODR+ADDER /SEG:1024, ROUTE1, ,PLOT88
SPINLAB7
- LINK/E MAIN+PLT /SEG:1024, ROUTE2, ,PLOT88 SPINLAB7

Two error messages will appear:

- Unresolved externals:
 - PLTMAI in file(s):
MAIN.OBJ (MAIN.FOR)
- Unresolved externals:
 - CALC in file(s):
MAIN.OBJ (MAIN.FOR)
 - PAUSE in file(s):
MAIN.OBJ (MAIN.FOR)
 - OPS1 in file(s):
MAIN.OBJ (MAIN.FOR)
 - ADDER in file(s):
MAIN.OBJ (MAIN.FOR)

- OPS2 in file(s):
 MAIN.OBJ (MAIN.FOR)
- OPS3 in file(s):
 MAIN.OBJ (MAIN.FOR)
- RANKODR in file(s):
 MAIN.OBJ (MAIN.FOR)

To execute, the user types ROUTE1 then proceeds with creating a configuration file, calculating the noise levels, and opening a file to store the results on disk. Once the file has been created, the user runs ROUTE2 and selects the plotting option. ROUTE2 will request an input file at which time the user enters the filename previously specified.

6.4 Commercial Software

ROUTEMAP requires three commercial software packages. These can be purchased from the manufacturers or from dealers. Care must be taken to observe license agreements if a ROUTEMAP load module is transferred to a different computer.

Microsoft FORTRAN is manufactured by Microsoft, 16011 N.E. 36th Way, Box 97017, Redmond, WA 98073-9717. Software support is provided by calling 206/882-8089.

The SPINDRIFT library is manufactured by Spindrift Laboratories, 116 South Harvard Avenue, Arlington Heights, IL 60005-1644. Software support is provided by calling 312/255-6909.

The PLOT88 library is manufactured by PLOTWORKS, Inc., P.O. Box 12385, La Jolla, CA 92037-0635. Software support is provided by calling 619/457-5090.

REFERENCES

1. Plotkin, K.J., and Croughwell, E.P., "Environmental Noise Assessment for Military Aircraft Training Routes, Volume I: SAC Low-Level Routes", Wyle Research Report WR 86-19, November 1986.
2. Plotkin, K.J., "Environmental Noise Assessment for Military Aircraft Training Routes, Volume 3: TAC Low-Level Routes", Wyle Research Report WR 87-9, November 1987.
3. Plotkin, K.J., "Environmental Noise Assessment Military Aircraft Training Routes, Volume 2: Recommended Noise Metric", AAMRL-TR-87-001.
4. "Environmental Protection, Planning in the Noise Environment", AFM 19-10, TM 5803-2, NAVFAC P-970.
5. Schultz, T.J., "Synthesis of Social Surveys on Noise Annoyance", J.Acoust.Soc.Am., 64, 377-405, 1978.
6. Sutherland, L.C., et al., "Development of a Model for Lateral Attenuation", Wyle Research Report WR 88-11, 1988.
7. Speakman, J.D., Letter Report, "Revised Equations for Predicting Onset Rates for Low-Altitude, High-Speed Aircraft Operations", 18 September 1987.

APPENDIX A

Computer Program Source Listing

ROUTEMAP

AUTHOR : MIKE LUCAS
SENIOR ENGINEER
WYLE LABORATORIES
2001 JEFFERSON DAVIS HWY., SUITE 701
ARLINGTON, VIRGINIA 22202 (703) 892-6700

SPONSOR: JERRY D. SPEAKMAN
BIODYNAMICS ENVIRONMENT BRANCH
BIODYNAMICS AND BIOENGINEERING DIV
HARRY G. ARMSTRONG AEROSPACE MEDICAL
RESEARCH LABORATORY (AFSC)
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6573

CONTRACT NUMBER F33615-85-3-0534 NOISEMAP Forecast Technology

PROGRAM MAIN

SINCLUDE: 'spinattr.hdr'

LOGICAL CHECK, FEXIST, FLAG(6), TOGGLE
CHARACTER*1 IDUMMY
CHARACTER*2 OPCR(300)
CHARACTER*3 ACC(300), AIRSPD(300)
CHARACTER*5 POWSET(300)
CHARACTER*6 POWUNT(300)
CHARACTER*7 NAME(300)
CHARACTER*16 FILNAM
CHARACTER*20 POWDES(300)
CHARACTER*25 DRAG(300)
CHARACTER*80 INPUT, OUTPUT, TEXT(11)

JOB CONTROL CARD FOR OMEGA10 INPUT. ALL BLANK NO PRINTED
OUTPUT AND ONLY SEL VALUES ARE COMPUTED.

COMMON /AIRPARA/ALT(20), FMEAN(20), ICODE(20), IREL, ITEMP,
1 MONTH, NUMAIR, FNUMD(20), FNUMN(20), POWER(20), SEL(20,25),
2 SIGMA, VEL(20)
COMMON /FRAME/SCN(1024)
COMMON /NOISE/MAX,ACC,AIRSPD,DRAG,NAME,OPCR,POWDES,POWSET,POWUNT
COMMON /RESULT/FLDNMR(201),FLDNM(201),FLEQ(201),RANK(20,201)
DATA TEXT(1) //'RUNNING OMEGA10' /
DATA TEXT(2) //'POWER OR VELOCITY OUT OF RANGE FOR OMEGA10. SEE FIL
1E "OMEGA10.OUT". ' /
DATA TEXT(3) //'CALCULATING NOISE LEVELS' /
DATA TEXT(4) //'THE INPUT DATA' /
DATA TEXT(5) //'Config Aircraft DayOps NiteOps Altitude Spe
led Power Offset' /
DATA TEXT(6) //'ROUTEMAP Version 1.0' /
DATA TEXT(7) //'TOTAL Ldnmr AT DISTANCES PERPENDICULAR TO ROUTE' /
DATA TEXT(8) //'Dist(Kft) dB Dist(Kft) dB Dist(Kft) dB Dist(K
1ft) dB Dist(Kft) dB' /
DATA TEXT(9) //'Save Results On Disk (yes/no) : ' /
DATA TEXT(10) //'File already exists. Try a new file name.' /

```

DATA TEXT(11)/'Enter File Name With Extension :'/
C
C INITIALIZE WINDOW ROUTINES
CALL INITIAL
C
C CHECK THAT NOISE AND OMEGA10.EXE ARE ON THE DISK
IF( .NOT. FEXIST('NOISE')) THEN
    CALL FINISH
    WRITE(*,*) 'FILE "NOISE" NOT FOUND'
    GO TO 200
ELSE
    END IF
IF( .NOT. FEXIST('OMEGA10.EXE')) THEN
    CALL FINISH
    WRITE(*,*) 'FILE "OMEGA10.EXE" NOT FOUND'
    GO TO 200
ELSE
    END IF
CALL CLS
CALL CUROFF
C
C READ NOISEFILE DATA
OPEN(UNIT=1,FILE='NOISE',STATUS='OLD',ERR=190)
DO 10 I = 1, 300
    READ(1,1000,END=20) ACC(I), OPCR(I)
1000    FORMAT(10X,A3,A2)
    READ(1,1100) NAME(I), AIRSPD(I)
1100    FORMAT(37X,A7,10X,A3)
    READ(1,1200) DRAG(I)
1200    FORMAT(33X,A25)
    READ(1,1300) POWDES(I),POWSET(I),POWUNT(I)
1300    FORMAT(17X,A20,1X,A5,1X,A6)
    READ(1,*)
    READ(1,*)
    READ(1,*)
10    CONTINUE
20    CONTINUE
CLOSE(UNIT=1)
MAX = I
C
C TRUE TURNS WINDOW OPTION -- ON --
FLAG(1) = .TRUE.
FLAG(2) = .TRUE.
FLAG(3) = .FALSE.
FLAG(4) = .TRUE.
FLAG(5) = .TRUE.
FLAG(6) = .FALSE.
TOGGLE = .FALSE.
C
C DISPLAY BANNER
CALL OPEN
C
C DRAW MAIN MENU
30    CALL MENU(IVAL,FLAG)
GO TO (40, 50, 60, 160, 170, 180, 190), IVAL
C
C CONFIGURE MILITARY TRAINING ROUTE
40    IF(.NOT. FLAG(1)) GO TO 30
    CALL OPS1(ISWITCH)
    IF(ISWITCH .NE. 4) CALL OPS2(ISWITCH)

```

```

        FLAG(1) = .FALSE.
        FLAG(2) = .TRUE.
        FLAG(3) = .TRUE.
        FLAG(4) = .TRUE.
        FLAG(5) = .TRUE.
        FLAG(6) = .FALSE.
        TOGGLE = .FALSE.
GO TO 30

C
C      SET THE TEMPERATURE, HUMIDITY, AND NUMBER OF DAYS IN MONTH
50      IF(.NOT. FLAG(2)) GO TO 30
        CALL OPS3
GO TO 30

C
C      RUN OMEGA10 AND CALC
60      IF(.NOT. FLAG(3)) GO TO 30
        CALL DEFWIND(1,8,0,14,79,SINGLBD,NORMAL)
        CALL WINDBUF(SCN)
        CALL CAPTSCN(SCN)
        CALL CLRB(1)
        CALL BLDB(1,2,32,TEXT(1),0,*500)
        CALL DISPSCN(SCN)
        CALL WLOCATE(1,2,32,*500)

C
C      PREPARE TEMPORARY FILE STRUCTURE
        IF(FEXIST('OUT')) CALL ERASE('OUT')
        IF(FEXIST('IN')) CALL ERASE('IN')
        IF(FEXIST('OMEGA10.OUT')) CALL ERASE('OMEGA10.OUT')
        OPEN(UNIT=2,FILE='OUT',STATUS='NEW',ERR=190)

C
C      PREPARE INPUT DECK FOR OMEGA10

C
C      JOB CONTROL CARD - BLANK
        WRITE(2,*)
        I = 0
70      I = I + 1
        J = 0
80      J = J + 1
        IF (ACC(ICODE(I)) .EQ. ACC(ICODE(I+J)) .AND.
1      I+J .LE. NUMAIR .AND. J .LT. 12) GO TO 80

C
C      OUTPUT PARAMETERS FOR EACH AIRCRAFT CARD #1
        WRITE(2,1400) ACC(ICODE(I)),ITEMP,IREL,J,POWUNT(ICODE(I))
1400      FORMAT(A3,I3,I3,10X,I2,A6)
C
        ASSIGN 1500 TO IVAR
        IF(POWER(I) .LT. 150) ASSIGN 1600 TO IVAR
        IF(POWER(I) .LT. 10) ASSIGN 1700 TO IVAR
        J = J - 1
        IF(J .GE. 6) THEN
            WRITE(2,IVAR)
1      (POWER(K),IFIX(VEL(K)),OPCR(ICODE(K)),K-I+1,K=I,I+5)
            WRITE(2,IVAR)
1      (POWER(K),IFIX(VEL(K)),OPCR(ICODE(K)),K-I+1,K=I+6,I+J)
        ELSE
            WRITE(2,IVAR)
1      (POWER(K),IFIX(VEL(K)),OPCR(ICODE(K)),K-I+1,K=I,I+J)
        END IF
1500      FORMAT(6(F5.0,I3,A2,I2))
1600      FORMAT(6(F5.1,I3,A2,I2))

```

```

1700      FORMAT(6(F5.2,I3,A2,I2))
          I = I + J
          IF(I .LT. NUMAIR) GO TO 70
          CLOSE(UNIT=2)
C
C      RUN OMEGA10
          CALL EXEC('OMEGA10.EXE',' ',*500)
          IF( .NOT. FEXIST('IN')) THEN
              CALL FINISH
              WRITE(*,*) 'FILE "IN" NOT FOUND.'
              WRITE(*,*) 'OMEGA10 FAILED TO WRITE THE FILE.'
              WRITE(*,*) 'CHECK CMEGA10.OUT TO DETERMINE CAUSE OF ERROR.'
              GO TO 200
          ELSE
              END IF
C
C      READ OMEGA10 OUTPUT DECK. READ INPUT FORMATTED FOR OMEGA10 VERSION
C      THAT BEGINS CALCULATIONS AT AN ALTITUDE OF 100 FEET.
          OPEN(UNIT=3,FILE='IN',STATUS='OLD',ERR=190)
          I = 0
90      READ(3,*,END=100)
          I = I + 1
          READ(3,1800,ERR=100) (SEL(I,J), J = 1, 6)
1800      FORMAT(22X,6F8.1)
          READ(3,*)
          READ(3,*)
          READ(3,*)
          READ(3,1900,ERR=100) (SEL(I,J), J = 7, 14)
          READ(3,1900,ERR=100) (SEL(I,J), J = 15, 22)
1900      FORMAT(6X,8F8.1)
          READ(3,2000,ERR=100) (SEL(I,J), J = 23, 25)
2000      FORMAT(6X,3F8.1)
          READ(3,*)
          READ(3,*)
          READ(3,*)
          READ(3,*)
          IF(I .LT. NUMAIR) GO TO 90
          CLOSE(UNIT=3)
          GO TO 110
C
C      DISPLAY SCREEN : POWER OR VELOCITY OUT OF RANGE FOR OMEGA10
100      CLOSE(UNIT=3)
          CALL WINDBUF(SCN)
          CALL CLR(1)
          CALL BLDB(1,2,5,TEXT(2),0,*500)
          CALL DISPSCN(SCN)
          CALL PAUSE
          FLAG(1) = .TRUE.
          FLAG(2) = .TRUE.
          FLAG(3) = .FALSE.
          FLAG(4) = .FALSE.
          FLAG(5) = .FALSE.
          FLAG(6) = .FALSE.
          TOGGLE = .FALSE.
          GO TO 30
C
C      DISPLAY SCREEN : RUNNING CALC
110      CALL WINDBUF(SCN)
          CALL CLR(1)
          CALL BLDB(1,2,28,TEXT(3),FLASH+NORMAL,*500)

```

```

      CALL DISPSCN(SCN)
C
C   EXECUTE SUBROUTINE CALC
      CALL CALC
C
C   DISPLAY INPUT DATA
      CALL DEFWIND(2,0,0.24,79,DBLBDR,NORMAL)
      CALL WINDBUF(SCN)
      CALL CLRB(2)
      IADJUST = 1
      IF(NUMAIR .NE. 20) THEN
        CALL BLDB(2, 0,33,TEXT( 4),0,*500)
        IADJUST = 0
      ELSE
      END IF
      CALL BLDB(2, 3-IADJUST, 3,TEXT( 5),0,*500)
      CALL DISPSCN(SCN)
      WRITE(OUTPUT,2100) SIGMA/5280.0
2100  FORMAT('Standard Deviation =',F6.2,' Statute Miles')
      CALL WLOCATE(2,1-IADJUST,19,*500)
      CALL PRINTW(2,OUTPUT(1:40),*500)
      WRITE(OUTPUT,2200) ITEMP, IREL, MONTH
2200  FORMAT('Temp =',I3,5X,'Humidity =',I3,5X,
1      'Num Days In Month = ',I2)
      CALL WLOCATE(2,2-IADJUST,13,*500)
      CALL PRINTW(2,OUTPUT(1:54),*500)
      DO 120 I = 1, NUMAIR
        CALL WLOCATE(2,I+3-IADJUST,2,*500)
        WRITE(OUTPUT,2300) I,NAME(ICODE(I)),FNUMD(I),FNUMN(I),
1      IFIX(ALT(I)),IFIX(VEL(I)),POWER(I),FMEAN(I)
2300  FORMAT(3X,I2,6X,A7,2X,F5.1,4X,F5.1,5X,I5,6X,I3,2X,F6.2,
1      3X,G8.3E1)
        CALL PRINTW(2,OUTPUT(1:73),*500)
120  CONTINUE
      CALL ANYKEY
C
C   DISPLAY RESULTS OF CALCULATIONS
      CALL WINDBUF(SCN)
      CALL CLRB(2)
      CALL BLDB(2, 0,30,TEXT( 6),0,*500)
      CALL BLDB(2, 1,16,TEXT( 7),0,*500)
      CALL BLDB(2, 2, 2,TEXT( 8),0,*500)
      CALL DISPSCN(SCN)
      DO 130 I = 1, 39, 2
        IROW = (I+1)/2 + 2
        CALL WLOCATE(2,IROW,2,*500)
        WRITE(OUTPUT,2400)
1      ((J+I-101)/2,IFIX(FLDNMR(J+I))), J = 0, 160, 40)
2400  FORMAT(2X,I3,4X,I3,4(5X,I3,4X,I3))
        CALL PRINTW(2,OUTPUT(1:74),*500)
130  CONTINUE
      CALL ANYKEY
      CALL WINDBUF(SCN)
      CALL CAPTSCN(SCN)
      CALL CLRB(1)
      CALL BLDB(1,2,10,TEXT( 9),0,*500)
      CALL DISPSCN(SCN)
      CALL WLOCATE(1,2,41,*500)
      CALL CURON
      READ(*,2500) IDUMMY

```

```

2500    FORMAT(A1)
        IF(IDUMMY.EQ. 'Y' .OR. IDUMMY.EQ. 'y') THEN
140      CALL WINDBUF(SCN)
        CALL CLRB(1)
        CALL BLDB(1,2,10,TEXT(10),0,*500)
        CALL DISPSCN(SCN)
        CALL PAUSE
        CALL CLS
        CALL SYSTEM('DIR/W',*500)
        CALL ANYKEY
150      CALL WINDBUF(SCN)
        CALL CLRB(1)
        CALL BLDB(1,2,10,TEXT(11),0,*500)
        CALL DISPSCN(SCN)
        CALL CURON
        CALL WLOCATE(1,2,42,*500)
        READ(*,2600,ERR=190) FILNAM
2600    FORMAT(A16)
        CALL CUROFF
        OPEN(UNIT=6,ERR=140,FILE=FILNAM,FORM='BINARY',RECL=1)
        WRITE(6) FLDNMR, FLDNM, FLEQ
        CLOSE(UNIT=6)
        ELSE
        END IF
        FLAG(1) = .TRUE.
        FLAG(2) = .FALSE.
        FLAG(3) = .FALSE.
        FLAG(4) = .TRUE.
        FLAG(5) = .FALSE.
        FLAG(6) = .TRUE.
        TOGGLE = .FALSE.
        GO TO 30
C
C      PLOT RESULTS
160      CONTINUE
C
C      IF OPS IS RUN WITHOUT RUNNING CALC THEN DO NOT ALLOW USER TO DRAW PLOT
C
C      IF FLAG(5) TRUE THEN FILE MUST BE ON DISK.
        IF(.NOT. FLAG(4)) GO TO 30
        CALL PLTMAIN(FLAG(5),TOGGLE)
        TOGGLE = .TRUE.
        GO TO 30
170      IF(.NOT. FLAG(6)) GO TO 30
        CALL RANKODR
        GO TO 30
180      CALL ADDER
        GO TO 30
C
C      SPINDRIFT ERROR
500      CALL FINISH
        IVAL = ISERROR()
        WRITE(*,*) 'SPINDRIFT ERROR NUMBER =',IVAL
        GO TO 200
190      CALL FINISH
200      STOP 'Bye-Bye.   Have a Nice Day !'
        END
C
C

```

C
C
C

```
SUBROUTINE INITIAL
INTEGER*2 INITMOD, VIDEOTYP, IVIDMOD
CHARACTER VMODE*2
LOGICAL TESTMONO
COMMON /FRAME/SCN(1024)
COMMON /IN/T/INITMOD,INITSCN
DATA TESTMONO/.FALSE./
DATA IVIDMOD/3/
```

C

```
CALL SPININIT
INITMOD = VIDEOTYP()
CALL CAPTSCN(SCN)
CALL ATOS(SCN,INITSCN,4096,0)
CALL NUMCLR
CALL INSCLR
CALL GETENVR('MONO',VMODE,*10)
TESTMONO = .TRUE.
10 IF(INITMOD .EQ. 7 .OR. TESTMONO) THEN
    RETURN
ELSE
    CALL VIDEOMOD(IVIDMOD)
END IF
RETURN
END
```

C
C
C
C
C

```
SUBROUTINE OPEN
```

C
C
C

```
DRAWS THE OPENING SCREEN USING A ZOOM FEATURE
```

```
$INCLUDE: 'spinattr.hdr'
```

```
CHARACTER*45 TEXT(8)
COMMON /FRAME/SCN(1024)
DATA TEXT(1)/'          *** ROUTEMAP ***'/
DATA TEXT(2)/'          Environmental Noise Assessment'/
DATA TEXT(3)/'          for'/
DATA TEXT(4)/'          Military Aircraft Training Routes'/
DATA TEXT(5)/'          by'/
DATA TEXT(6)/'          United States Air Force AAMRL/BBE'/
DATA TEXT(7)/'Wright-Patterson Air Force Base, Ohio 45433'/
DATA TEXT(8)/'          Version 1.0'/
```

C

```
CALL CUROFF
DO 10 I = 1, 12
    IULCR = 12 - I
    IULCC = IFIX(39.48 - FLOAT(I)*3.29)
    ILRCR = 12 + I
    ILRCC = IFIX(39.52 + FLOAT(I)*3.29)
    CALL DEFWIND(1,IULCR,IULCC,ILRCR,ILRCC,DBLBDR,NORMAL)
    CALL WINDBUF(SCN)
    CALL CAPTSCN(SCN)
    CALL CLR(1)
    CALL DISPSCN(SCN)
    CALL TONE2(I*300+2000,I*300+1000,2)
```

```

10  CONTINUE
    CALL WINDBUF(SCN)
    CALL CLR(1)
    CALL BLDB(1, 2, 17, TEXT(1), BRIGHT, *20)
    CALL BLDB(1, 5, 17, TEXT(2), BRIGHT, *20)
    CALL BLDB(1, 7, 17, TEXT(3), BRIGHT, *20)
    CALL BLDB(1, 9, 17, TEXT(4), BRIGHT, *20)
    CALL BLDB(1, 12, 17, TEXT(5), BRIGHT, *20)
    CALL BLDB(1, 15, 17, TEXT(6), BRIGHT, *20)
    CALL BLDB(1, 16, 17, TEXT(7), BRIGHT, *20)
    CALL BLDB(1, 18, 17, TEXT(8), BRIGHT, *20)
    CALL DISPSCN(SCN)
    CALL WAIT(1, 27, *20)
    CALL PAUSE
20  RETURN
    END

C
C
C
C
C
C
    SUBROUTINE MENU(IVAL, FLAG)
SINCLUDE: 'spinattr.hdr'
    INTEGER*2 KEYTYPE
    INTEGER IA(1)
    LOGICAL ISKEY, FLAG(6)
    CHARACTER*1 CHR, GETCHX
    CHARACTER*60 TEXT(11)
    COMMON /FRAME/SCN(1024)
    DATA TEXT(1)/'MAIN MENU'/
    DATA TEXT(2)/'Please Make A Selection'/
    DATA TEXT(3)/'(1) Define Operations On Military Training Route.'/
    DATA TEXT(4)/'(2) Change Default Settings.'/
    DATA TEXT(5)/'(3) Calculate Noise Level.'/
    DATA TEXT(6)/'(4) Plot Noise Level Metrics. Use Data Stored On Di
1sk.'/
    DATA TEXT(7)/'(4) Plot Noise Level Metrics.'/
    DATA TEXT(8)/'(5) Rank Order Contributors To Total Noise Exposure
1.'/
    DATA TEXT(9)/'(6) Decibel Addition.'/
    DATA TEXT(10)/'(7) End This Program.'/
    DATA TEXT(11)/'Your Selection :'/

C
10  CALL DEFWIND(1, 0, 5, 24, 4, DBLBDR, NORMAL)
    CALL WINDBUF(SCN)
    CALL CLS
    CALL CAPTSCN(SCN)
    CALL CLR(1)

C
C
    TITLE
    CALL BLDB(1, 2, 31, TEXT(1), 0, *30)
    CALL BLDB(1, 3, 24, TEXT(2), 0, *30)

C
C
    OPERATIONS
    IF(FLAG(1)) THEN
        CALL BLDB(1, 6, 8, TEXT(3), BRIGHT, *30)
    ELSE
        CALL BLDB(1, 6, 8, TEXT(3), 0, *30)
    END IF

C

```

```

C   DEFAULT
    IF(FLAG(2)) THEN
        CALL BLDB(1, 8, 8, TEXT(4),BRIGHT,*30)
    ELSE
        CALL BLDB(1, 8, 8, TEXT(4),0,*30)
    END IF

C   CACULATE NOISE LEVEL
C   IF(FLAG(3)) THEN
        CALL BLDB(1,10, 8, TEXT(5),BRIGHT,*30)
    ELSE
        CALL BLDB(1,10, 8, TEXT(5),0,*30)
    END IF

C   PLOT MENU ITEM
C   IF(FLAG(4)) THEN
        IF(FLAG(5)) THEN
            CALL BLDB(1,12, 8, TEXT(6),BRIGHT,*30)
        ELSE
            CALL BLDB(1,12, 8, TEXT(7),BRIGHT,*30)
        END IF
    ELSE
        CALL BLDB(1,12, 8, TEXT(7),0,*30)
    END IF

C   RANK ORDER
C   IF(FLAG(6)) THEN
        CALL BLDB(1,14, 8, TEXT(8),BRIGHT,*30)
    ELSE
        CALL BLDB(1,14, 8, TEXT(8),0,*30)
    END IF

C   END PROGRAM
C   CALL BLDB(1,16, 8, TEXT(9),BRIGHT ,*30)
    CALL BLDB(1,18, 8, TEXT(10),BRIGHT ,*30)
    CALL BLDB(1,21,40,TEXT(11),0,*30)
    CALL DISPSCN(SCN)
    CALL CURON
20  CALL WLOCATE(1,21,56,*30)
    IF(.NOT. ISKEY(CHR,KEYTYPE)) THEN
        GO TO 20
    ELSE
        CALL CTOA(CHR,IA,0)
        IF((KEYTYPE .EQ. 3) .AND. (IA(1).GE.49 .AND. IA(1).LE.55)) THEN
            IVAL = IA(1) - 48
            CHR = GETCH(KEYTYPE)
        ELSE
            CHR = GETCHX(KEYTYPE)
            CALL WLOCATE(1,21,56,*30)
            CALL PRINTW(1,' ',*30)
            GO TO 20
        END IF
    END IF
    CALL CUROFF
30  RETURN
    END

```

C

```
SUBROUTINE FINISH
INTEGER*2 INITMOD
COMMON /FRAME/SCN(1024)
COMMON /INIT/INITMOD,INITSCN
CALL VIDEOMOD(INITMOD)
CALL STOA(INITSCN,SCN.0,*10)
CALL DISPSCN(SCN)
10 CALL SPINWRAP
RETURN
END
```

10

C
C
C
C
C

```
BLOCK DATA
COMMON /AIRPARA/ALT(20), FMEAN(20), ICODE(20), IREL, ITEMP,
1 MONTH, NUMAIR, FNUMD(20), FNUMN(20), POWER(20), SEL(20,25),
2 SIGMA, VEL(20)
COMMON /RESULT/FLDNMR(201),FLDNM(201),FLEQ(201),RANK(20,201)
DATA ALT/20*0.0/, FMEAN/20*0.0/, ICODE/20*0/, IREL/70/, ITEMP/59/
DATA MONTH/30/, NUMAIR/0/, FNUMD/20*0.0/, FNUMN/20*0.0/,RANK/4020*
10.0/
DATA POWER/20*0.0/, SIGMA/0.0/, VEL/20*0.0/
DATA FLDNMR/201*0.0/, FLDNM/201*0.0/, FLEQ/201*0.0/
END
```

C
C
C
C
C

```

SUBROUTINE OPS1(ISWITCH)
$INCLUDE: 'spinattr.hdr'
INTEGER*2 KEYTYPE
INTEGER IA(1)
LOGICAL FLAG, ISKEY
CHARACTER*1 CHR, GETCHX
CHARACTER*16 OLDNAM
CHARACTER*50 TEXT(10)
COMMON /AIRPARA/ALT(20), FMEAN(20), ICODE(20), IREL, ITEMP,
1 MONTH, NUMAIR, FNUMD(20), FNUMN(20), POWER(20), SEL(20,25),
2 SIGMA, VEL(20)
COMMON /FRAME/SCN(1024)
DATA TEXT(1)/'DEFINE OPERATIONS ON MILITARY TRAINING ROUTE'/
DATA TEXT(2)/'Select A Training Mission'/
DATA TEXT(3)/'(1) Visual Flying Rules'/
DATA TEXT(4)/'(2) Instrument Flying Rules'/
DATA TEXT(5)/'(3) Special Operations'/
DATA TEXT(6)/'(4) Retrieve Previously Saved Flight Track File'/
DATA TEXT(7)/'Your Selection :'/
DATA TEXT(8)/'Enter File Name With Extension :'/
DATA TEXT(9)/'File Not Found'/
DATA TEXT(10)/'Error Reading File'/
10 CALL DEFWIND(1,0,5,15,73,DBLBDR,NORMAL)
CALL WINDBUF(SCN)
FLAG = .FALSE.
CALL CLS
CALL CAPTSCN(SCN)
CALL CLR(1)
CALL BLDB(1, 1,10,TEXT(1),0,*500)
CALL BLDB(1, 4,10,TEXT(2),0,*500)
CALL BLDB(1, 6,10,TEXT(3),0,*500)
CALL BLDB(1, 7,10,TEXT(4),0,*500)
CALL BLDB(1, 8,10,TEXT(5),0,*500)
CALL BLDB(1, 9,10,TEXT(6),0,*500)
CALL BLDB(1,12,40,TEXT(7),0,*500)
CALL DISPSCN(SCN)
CALL CURON
20 CALL WLOCATE(1,12,56,*500)
IF(.NOT. ISKEY(CHR,KEYTYPE)) THEN
  GO TO 20
ELSE
  CALL CTOA(CHR,IA,0)
  IF((KEYTYPE .EQ. 3) .AND. (IA(1).GE.49 .AND. IA(1).LE.52)) THEN
    ISWITCH = IA(1) - 48
    CHR = GETCHX(KEYTYPE)
    CALL CUROFF
  ELSE
    CHR = GETCHX(KEYTYPE)
    CALL WLOCATE(1,12,56,*500)
    CALL PRINTW(1,'',*500)
    GO TO 20
  END IF
END IF
CALL DEFWIND(2,8,0,14,79,SINGLBD,NORMAL)
IF(ISWITCH .EQ. 4) THEN
  All

```

```

30      GO TO 40
      IF(FLAG) GO TO 10
      CALL WINDBUF(SCN)
      CALL CLR(2)
      CALL BLDB(2,2,10,TEXT(9),0,*500)
      CALL DISPSCN(SCN)
      CALL PAUSE
      CALL CLS
      CALL SYSTEM('DIR/W',*500)
      CALL ANYKEY
      CALL CLRKB
      FLAG = .TRUE.
40      CALL WINDBUF(SCN)
      CALL CLR(2)
      CALL BLDB(2,2,10,TEXT(8),0,*500)
      CALL DISPSCN(SCN)
      CALL CURON
      CALL WLOCATE(2,2,42,*500)
      READ(*,1100,ERR=40) OLDNAM
      CALL CUROFF
1100     FORMAT(A16)
      OPEN(UNIT=4,FILE=OLDNAM,STATUS='OLD',ERR=30)
      READ(4,1200,ERR=60) IREL, ITEMP, MONTH, NUMAIR, SIGMA
1200     FORMAT(11X,I3.3,9X,I3.3,13X,I2.2,17X,I2.2,9X,F10.1)
      DO 50 I = 1, NUMAIR
          READ(4,1300,ERR=60) ALT(I), FMEAN(I), FNUMD(I),
1          FNUMN(I), ICODE(I), POWER(I), VEL(I)
1300     FORMAT(10X,F10.1,10X,F10.1,13X,F6.1,15X,F6.1,8X,I3.3,9X,
1          F10.2,12X,F10.2)
50      CONTINUE
      CLOSE(UNIT=4)
      ELSE
      ENDIF
      GO TO 500
60      CALL WINDBUF(SCN)
      CALL CLR(2)
      CALL BLDB(2,2,10,TEXT(10),0,*500)
      CALL DISPSCN(SCN)
      CALL PAUSE
      CLOSE(UNIT=4)
      GO TO 10
500     RETURN
      END

```

C
C
C
C
C

```

      SUBROUTINE OPS2(ISWITCH)
$INCLUDE: 'spinattr.hdr'
      DIMENSION IA(1), IROW(6), ICOL(6), VALUE(6)
      INTEGER*2 KEYTYPE
      LOGICAL ISKEY
      CHARACTER IDUMMY, CHR, GETCHX
      CHARACTER*2 OPCR(300)
      CHARACTER*3 ACC(300), AIRSPD(300)
      CHARACTER*5 POWSET(300)
      CHARACTER*6 POWUNT(300)
      CHARACTER*7 NAME(300)
      CHARACTER*16 NEWNAM

```

```

CHARACTER*20 POWDES(300)
CHARACTER*25 DRAG(300)
CHARACTER*80 OUTPUT, TEXT(50)
COMMON /AIRPARA/ALT(20), FMEAN(20), ICODE(20), IREL, ITEMP,
1 MONTH, NUMAIR, FNUMD(20), FNUMN(20), POWER(20), SEL(20,25),
2 SIGMA, VEL(20)
COMMON /FRAME/SCN(1024)
COMMON /NOISE/MAX,ACC,AIRSPD,DRAG,NAME,OPCR,POWDES,POWSET,POWUNT
DATA IROW /8,10,12,14,16,18/
DATA ICOL /32,29,46,42,45,47/
DATA TEXT(1) //'VISUAL FLIGHT RULES' /
DATA TEXT(2) //'OPTION (1) - Specific well defined track(s): standa
1rd' /
DATA TEXT(3) // 'deviation is 1.25 statute miles.' /
DATA TEXT(4) // 'or' /
DATA TEXT(5) //'OPTION (2) - Multiple scattered tracks: standard dev
1iation' /
DATA TEXT(6) // 'is 2.5 statute miles about centerline.'
1/
DATA TEXT(7) //'Your Selection :'/
DATA TEXT(8) //'INSTRUMENT FLYING RULES' /
DATA TEXT(9) //'Assume lateral distribution of aircraft relative' /
DATA TEXT(10) // 'to route centerline described by a Gaussian' /
DATA TEXT(11) // 'distribution with a standard deviation of' /
DATA TEXT(12) // '0.5 Statute Miles.' /
DATA TEXT(13) //'SPECIAL OPERATIONS' /
DATA TEXT(14) //'Enter the standard deviation in Statute Miles (x.xx
1) :'/
DATA TEXT(15) //'Standard Deviation Out of Range' /
DATA TEXT(16) //'Standard Deviation =' /
DATA TEXT(17) //'AVAILABLE AIRCRAFT FROM NOISEFILE' /
DATA TEXT(18) // 'POWER
1 DRAG AIR' /
DATA TEXT(19) // 'AIRCRAFT DESCRIPTION POWER
1 CONFIGURATION SPEED' /
DATA TEXT(20) /
1 'Use arrow keys to scroll desired aircraft into highlighted area.'
2/
DATA TEXT(21) //'press RETURN when done.' /
DATA TEXT(22) //'OPERATION PARAMETERS' /
DATA TEXT(23) //'Power ( ) :'/
DATA TEXT(24) //'Velocity ( knots) :'/
DATA TEXT(24) //'Speed ( knots) :'/
DATA TEXT(25) //'Altitude Above Ground Level (feet) :'/
DATA TEXT(26) //'Centerline Track Offset (feet) :'/
DATA TEXT(27) //'Number Of Daytime Monthly Sorties :'/
DATA TEXT(28) //'Number Of Nighttime Monthly Sorties :'/
DATA TEXT(29) //'press ESC when all done' /
DATA TEXT(30) //'Enter Another Configuration (Yes/No) :'/
DATA TEXT(31) //'Save File On Disk (Yes/No) :'/
DATA TEXT(32) //'Enter File Name With Extension :'/
DATA TEXT(33) //'File already exists. Try a new file name.' /
GO TO (10, 30, 40), ISWITCH

```

C
C
10

```

VISUAL FLIGHT RULES SCREEN
CALL DEFWIND(1,0,5,17.73,DBLBDR,NORMAL)
CALL WINDBUF(SCN)
CALL CLS
CALL CAPTSCN(SCN)
CALL CLRB(1)

```

```

CALL BLDB(1, 2,24,TEXT(1),0,*500)
CALL BLDB(1, 5, 7,TEXT(2),0,*500)
CALL BLDB(1, 6, 7,TEXT(3),0,*500)
CALL BLDB(1, 8, 9,TEXT(4),0,*500)
CALL BLDB(1,10, 7,TEXT(5),0,*500)
CALL BLDB(1,11, 7,TEXT(6),0,*500)
CALL BLDB(1,14,40,TEXT(7),0,*500)
CALL DISPSCN(SCN)
CALL CURON
20 CALL WLOCATE(1,14,56,*500)
IF(.NOT. ISKEY(CHR,KEYTYPE)) THEN
    GO TO 20
ELSE
    CALL CTOA(CHR,IA,0)
    IF((KEYTYPE .EQ. 3) .AND. (IA(1).GE.49 .AND. IA(1).LE.50)) THEN
        INPUT = IA(1) - 48
        CHR = GETCH(KEYTYPE)
        CALL CUROFF
    ELSE
        CHR = GETCHX(KEYTYPE)
        CALL WLOCATE(1,14,56,*500)
        CALL PRINTW(1,' ',*500)
        GO TO 20
    END IF
END IF
IF(INPUT .EQ. 1) THEN
    SIGMA = 6600.0
ELSE
    SIGMA = 13200.0
END IF
GO TO 70

C
C
30 INSTRUMENT FLIGHT RULES SCREEN
CALL DEFWIND(2,0.5,11,73,DBLBDR,NORMAL)
CALL WINDBUF(SCN)
CALL CLS
CALL CAPTSCN(SCN)
CALL CLR(2)
CALL BLDB(2, 2,23,TEXT( 8),0,*500)
CALL BLDB(2, 5,10,TEXT( 9),0,*500)
CALL BLDB(2, 6,10,TEXT(10),0,*500)
CALL BLDB(2, 7,10,TEXT(11),0,*500)
CALL BLDB(2, 8,10,TEXT(12),0,*500)
CALL DISPSCN(SCN)
CALL PAUSE
SIGMA = 2640.0
INPUT = 2
GO TO 70

C
C
40 SPECIAL OPERATIONS SCREEN
CONTINUE
CALL DEFWIND(3,8.0,14,79,DBLBDR,NORMAL)
CALL WINDBUF(SCN)
CALL CLS
CALL CAPTSCN(SCN)
CALL CLR(3)
CALL BLDB(3,2.5,TEXT(14),NORMAL,*500)
CALL DISPSCN(SCN)
CALL CURON
GO TO 60

```

```

50  READ(*,1000) IDUMMY
1000 FORMAT(A1)
    CALL WLOCATE(3,2,58,*500)
    CALL PRINTW(3,'          ',*500)
60  CALL WLOCATE(3,2,58,*500)
    IF(.NOT. ISKEY(CHR,KEYTYPE)) THEN
        GO TO 60
    ELSE
        CALL CTOA(CHR,IA,0)
        IF((KEYTYPE .EQ. 3) .AND.
1      ((IA(1).GE.46 .AND. IA(1).LE.57) .AND. IA(1).NE.47)) THEN
            READ(*,1100,ERR=50) SIGMA
1100    FORMAT(F6.2)
        ELSE
            CHR = GETCHX(KEYTYPE)
            CALL WLOCATE(3,2,58,*500)
            CALL PRINTW(3,'          ',*500)
            GO TO 60
        END IF
    END IF
    IF(SIGMA .GT. 2.5 .OR. SIGMA .LE. 0.0) THEN
        CALL CLRB(3)
        CALL BLDB(3,2,10,TEXT(15),NORMAL,*500)
        CALL DISPSCN(SCN)
        CALL CUROFF
        CALL PAUSE
        GO TO 40
    ELSE
        END IF
        CALL CLRB(3)
        CALL BLDB(3,2,10,TEXT(15),NORMAL,*500)
        CALL DISPSCN(SCN)
        CALL WLOCATE(3,2,31,*500)
        WRITE(*,1200,ERR=30) SIGMA
1200    FORMAT(F6.2,'      Is this correct (Yes/No) :',S)
        READ(*,1300) IDUMMY
1300    FORMAT(A1)
        IF(IDUMMY .EQ. 'N' .OR. IDUMMY .EQ. 'n') GO TO 40
        SIGMA = SIGMA*5280.0
        INPUT = 1
        CALL CUROFF

C
C  BEGIN AIRCRAFT SELECTION
70  NUMAIR = 0
    CALL DEFWIND(4,0,0,24,79,DBLBDR,NORMAL)
80  CONTINUE
    NUMAIR = NUMAIR + 1
    I = -4
90  CONTINUE
    CALL WINDBUF(SCN)
    CALL CLRB(4)
    CALL BLDB(4, 1,23,TEXT(17),0,*500)
    CALL BLDB(4, 3, 0,TEXT(18),0,*500)
    CALL BLDB(4, 4, 0,TEXT(19),0,*500)
    CALL BLDB(4,19, 8,TEXT(20),0,*500)
    CALL BLDB(4,20,29,TEXT(21),0,*500)
    IF(I .LT. -4) I = MAX
    IF(I .GT. MAX) I = -4

C
C  DISPLAY TABLE FROM NOISEFILE INCLUDE SCROLLING FEATURE

```

```

DO 100 J = 7, 17
IATTR = NORMAL
IF(J .EQ. 12) IATTR = BRIGHT
  IF(I .GE. 1 .AND. I .LE. MAX) THEN
    CALL BLDB(4,J, 3,NAME(I),IATTR,*500)
    CALL BLDB(4,J,12,POWDES(I),IATTR,*500)
    CALL BLDB(4,J,33,POWSET(I),IATTR,*500)
    CALL BLDB(4,J,39,POWUNT(I),IATTR,*500)
    CALL BLDB(4,J,47,DRAG(I),IATTR,*500)
    CALL BLDB(4,J,73,AIRSPD(I),IATTR,*500)
  ELSE
    END IF
    I = I + 1
100  CONTINUE
    I = I - 11
    CALL DISPSCN(SCN)
C
C  CHECK TO SEE IF THE KEYBOARD HAS BEEN PRESSED
110  CONTINUE
    CHR = GETCHX(KEYTYPE)
    CALL CTOA(CHR,IA,0)
    IF(KEYTYPE .EQ. 4 .AND. IA(1) .EQ. 13) GO TO 120
    IF(KEYTYPE .EQ. 8) THEN
      IF(IA(1) .EQ. 72) THEN
        I = I - 1
        GO TO 90
      ELSE
        END IF
      IF(IA(1) .EQ. 73) THEN
        I = I - 10
        GO TO 90
      ELSE
        END IF
      IF(IA(1) .EQ. 80) THEN
        I = I + 1
        GO TO 90
      ELSE
        END IF
      IF(IA(1) .EQ. 81) THEN
        I = I + 10
        GO TO 90
      ELSE
        END IF
    ELSE
      GO TO 110
    END IF
120  CONTINUE
C
C  RECORD AIRCRAFT CODE FOR OMEGA10
    ICODE(NUMAIR) = I + 5
C
C
C  SHOW SCREEN REQUESTING DATA INPUT
    CALL DEFWIND(5,0,5,23,73,DBLBDR,NORMAL)
    CALL WINDBUF(SCN)
    CALL CLS
    CALL CAPTSCN(SCN)
    CALL CLR3(5)
    IF(ISWITCH .EQ. 1) THEN
      CALL BLDB(5, 1,25,TEXT(1),0,*500)

```

```

ELSE
  IF(ISWITCH .EQ. 2) THEN
    CALL BLDB(5, 1,23, TEXT(8),0,*500)
  ELSE
    CALL BLDB(5, 1,26, TEXT(13),0,*500)
  END IF
END IF
CALL BLDB(5, 2,24,TEXT(22),0,*500)
CALL BLDB(5, 8,10,TEXT(23),0,*500)
CALL BLDB(5,10,10,TEXT(24),0,*500)
CALL BLDB(5,12,10,TEXT(25),0,*500)
IF(INPUT .EQ. 1) CALL BLDB(5,14,10,TEXT(26),0,*500)
CALL BLDB(5,16,10,TEXT(27),0,*500)
CALL BLDB(5,18,10,TEXT(28),0,*500)
CALL BLDB(5,21,22,TEXT(29),0,*500)
CALL DISPSCN(SCN)
CALL CURON

C
C   RECORD ZERO VALUES TO ACCOUNT FOR WHEN THERE IS NO INPUT
DO 130 I = 1, 5
  VALUE(I) = 0.0
130 CONTINUE
C
C   WRITE AIRCRAFT TYPE AND NUMBER OF CONFIGURATIONS ON SCREEN
CALL WLOCATE(5,5,10,*500)
WRITE(OUTPUT,1400) NAME(ICODE(NUMAIR)), NUMAIR
1400 FORMAT('Aircraft Type :',A7,10X,'Configuration :',I2,$)
CALL PRINTW(5,OUTPUT(1:49),*500)
C
C   WRITE POWER ON SCREEN
CALL WLOCATE(5,8,17,*500)
WRITE(OUTPUT(1:12),1500) POWSET(ICODE(NUMAIR)),
1 POWUNT(ICODE(NUMAIR))
1500 FORMAT(A5,1X,A6)
CALL PRINTW(5,OUTPUT(1:12),*500)
C
C   WRITE VELOCITY ON SCREEN
CALL WLOCATE(5,10,17,*500)
WRITE(OUTPUT(1:4),1600) AIRSPD(ICODE(NUMAIR))
CALL PRINTW(5,OUTPUT(1:4),*500)
1600 FORMAT(A3,1X)
C
C   INPUT RESULTS, ALLOW FOR EDITING FEATURES
I = 1
GO TO 150
140 READ(*,1700) IDUMMY
1700 FORMAT(A1)
CALL WLOCATE(5,IROW(I),ICOL(I),*500)
CALL PRINTW(5, ' ',*500)
150 CONTINUE
IF(INPUT .NE. 1 .AND. I .EQ. 4) I = 5
IF(I .EQ. 7) I = 1
CALL WLOCATE(5,IROW(I),ICOL(I),*500)
IF(.NOT. ISKEY(CHR,KEYTYPE)) THEN
  GO TO 150
ELSE
  CALL CTOA(CHR,IA,0)
  IF(KEYTYPE .EQ. 16 .AND. IA(1) .EQ. 27) GO TO 170
  IF(KEYTYPE .EQ. 4 .AND. IA(1) .EQ. 13) THEN
    I = I + 1

```

```

        CHR = GETCHX(KEYTYPE)
        GO TO 150
    ELSE
    END IF
    IF((KEYTYPE .EQ. 3) .AND.
1      ((IA(1).GE.45 .AND. IA(1).LE.57) .AND. IA(1).NE.47)) THEN
        CALL PRINTW(5,'',*500)
        CALL WLOCATE(5,IROW(I),ICOL(I),*500)
        READ(*,1800,ERR=140) VAR
1800    FORMAT(G10.0)
        CALL WLOCATE(5,IROW(I),ICOL(I),*500)
        IF(I .NE. 4 .AND. VAR .LT. 0.0) GO TO 160
        IF(I .EQ. 4 .AND.
1      (VAR .LT. -50000 .OR. VAR .GT. 50000)) GO TO 160
        VALUE(I) = VAR
160    WRITE(*,1900) VALUE(I)
1900    FORMAT(G10.3,$)
        IF(VALUE(I) .EQ. VAR) I = I + 1
        GO TO 150
    ELSE
        CHR = GETCHX(KEYTYPE)
        CALL WLOCATE(5,IROW(I),ICOL(I),*500)
        CALL PRINTW(5,'',*500)
        GO TO 150
    END IF
END IF
170  CONTINUE
    CHR = GETCHX(KEYTYPE)
C
C    COPY INPUT RESULTS INTO COMMON ARRAYS
    POWER(NUMAIR) = VALUE(1)
    VEL(NUMAIR) = VALUE(2)
    ALT(NUMAIR) = VALUE(3)
    IF(INPUT .EQ. 1) THEN
        FMEAN(NUMAIR) = VALUE(4)
    ELSE
        FMEAN(NUMAIR) = 0.0
    END IF
    FNUMD(NUMAIR) = VALUE(5)
    FNUMN(NUMAIR) = VALUE(6)
C
C    CHECK TO SEE IF USER WANTS TO ENTER ANOTHER AIRCRAFT
    CALL DEFWIND(6,8,0,14,79,SINGLBD,NORMAL)
    CALL CAPTSCN(SCN)
    CALL WINDBUF(SCN)
    CALL CLR(6)
    CALL BLDB(6,2,10,TEXT(30),0,*500)
    CALL DISPSCN(SCN)
    CALL CURON
    CALL WLOCATE(6,2,48,*500)
    READ(*,2000) IDUMMY
2000  FORMAT(A1)
    CALL CUROFF
    IF((IDUMMY .EQ. 'Y'.OR. IDUMMY .EQ. 'y').AND. NUMAIR .LT. 20) THEN
        CALL CLS
        GO TO 80
    ELSE
        CALL WINDBUF(SCN)
        CALL CLR(6)
        CALL BLDB(6,2,10,TEXT(31),0,*500)

```

```

      CALL DISPSCN(SCN)
      CALL CURON
      CALL WLOCATE(6,2,38,*500)
      READ(*,2100) IDUMMY
2100  FORMAT(A1)
      IF(IDUMMY.EQ. 'Y'.OR. IDUMMY.EQ. 'y') THEN
        GO TO 190
180    CALL WINDBUF(SCN)
        CALL CLRB(6)
        CALL BLDB(6,2,10,TEXT(33),0,*500)
        CALL DISPSCN(SCN)
        CALL PAUSE
        CALL CLS
        CALL SYSTEM('DIR/W',*500)
        CALL ANYKEY
        CALL CLRKB
190    CALL CAPTSCN(SCN)
        CALL WINDBUF(SCN)
        CALL CLRB(6)
        CALL BLDB(6,2,10,TEXT(32),0,*500)
        CALL DISPSCN(SCN)
        CALL CURON
        CALL WLOCATE(6,2,42,*500)
        READ(*,2200,ERR=180) NEWNAM
2200  FORMAT(A16)
        CALL CUROFF
        OPEN(UNIT=5,FILE=NEWNAM,STATUS='NEW',ERR=180)
        WRITE(5,2300) IREL, ITEMP, MONTH, NUMAIR, SIGMA
2300  FORMAT('Rel. Hum. =',I3.3,2X,'Temp. =',I3.3,2X,'Num. Days =',
1     I2.2,2X,'Num Airplanes =',I2.2,2X,'Sigma =',F10.1)
        DO 200 I = 1, NUMAIR
          WRITE(5,2400) ALT(I), FMEAN(I), FNUMD(I),
1          FNUMN(I), ICODE(I), POWER(I), VEL(I)
2400  FORMAT('Altitude =',F10.1,2X,'Offset =',F10.1,2X,
1      'Num. Days =',F6.1,2X,'Num. Nights =',F6.1,2X,
2      'Code =',I3.3,2X,'Power =',F10.2,2X,'Velocity =',F10.2)
200    CONTINUE
        CLOSE(UNIT=5)
      ELSE
      END IF
    END IF
500  RETURN
    END

```

C
C
C
C
C

```

      SUBROUTINE OPS3
      SINCLUDE: 'spinattr.hdr'
      DIMENSION IA(1), IROW(3), ICOL(3), IVALUE(3)
      INTEGER*2 KEYTYPE
      LOGICAL ISKEY
      CHARACTER IDUMMY, CHR, GETCHX
      CHARACTER*50 TEXT(5)
      COMMON /AIRPAR/ALT(20), FMEAN(20), ICODE(20), IREL, ITEMP,
1 MONTH, NUMAIR, FNUMD(20), FNUM(20), POWER(20), SEL(20,25),
2 SIGMA, VEL(20)
      COMMON /FRAME/SCN(1024)
      DATA TEXT(1)/'DEFAULT SETTINGS'/

```

```

DATA TEXT(2)/'Temperature In Degrees Fahrenheit :'/
DATA TEXT(3)/'Relative Humidity :'/
DATA TEXT(4)/'Number Of Days In Month :'/
DATA TEXT(5)/'press ESC when all done'/
DATA IROW/3,6,9/
DATA ICOL/45,29,35/
CALL CURON
CALL DEFWIND(1,4,9,18,69,DBLBDR,NORMAL)
CALL WINDBUF(SCN)
CALL CLS
CALL CAPTSCN(SCN)
CALL CLRB(1)
CALL BLDB(1, 1,22,TEXT(1),0,*100)
CALL BLDB(1, 3,10,TEXT(2),0,*100)
CALL BLDB(1, 6,10,TEXT(3),0,*100)
CALL BLDB(1, 9,10,TEXT(4),0,*100)
CALL BLDB(1,12,18,TEXT(5),0,*100)
CALL DISPSCN(SCN)
IVALUE(1) = ITEMP
IVALUE(2) = IREL
IVALUE(3) = MONTH
DO 10 I = 1, 3
    CALL WLOCATE(1,IROW(I),ICOL(I),*100)
    IF(I .EQ. 1) THEN
        WRITE(*,1000) IVALUE(I)
    ELSE
        WRITE(*,1100) IVALUE(I)
    END IF
    FORMAT(1X,I3,$)
    FORMAT(I3,$)
1000
1100
10    CONTINUE
C
C    INPUT RESULTS, ALLOW FOR EDITING FEATURES
I = 1
GO TO 30
20    READ(*,1200) IDUMMY
1200    FORMAT(A1)
    CALL WLOCATE(1,IROW(I),ICOL(I),*100)
    CALL PRINTW(1,'          ',*100)
30    CONTINUE
    IF(I .EQ. 4) I = 1
    CALL WLOCATE(1,IROW(I),ICOL(I),*100)
    IF(.NOT. ISKEY(CHR,KEYTYPE)) THEN
        GO TO 30
    ELSE
        CALL CTOA(CHR,IA,0)
        IF(KEYTYPE .EQ. 16 .AND. IA(1) .EQ. 27) GO TO 50
        IF(KEYTYPE .EQ. 4 .AND. IA(1) .EQ. 13) THEN
            I = I + 1
            CHR = GETCHX(KEYTYPE)
            GO TO 30
        ELSE
            END IF
            IF((KEYTYPE .EQ. 3) .AND.
1          ((IA(1).GE.45 .AND. IA(1).LE.57) .AND. IA(1).NE.47)) THEN
                CALL PRINTW(1,'          ',*100)
                CALL WLOCATE(1,IROW(I),ICOL(I),*100)
                READ(*,1300,ERR=20) IVAR
                FORMAT(I5)
                CALL WLOCATE(1,IROW(I),ICOL(I),*100)
1300

```

```

        IF(I .EQ. 1 .AND. (IVAR.LT.-99 .OR. IVAR.GT.200)) GO TO 40
        IF(I .EQ. 2 .AND. (IVAR .LT. 0 .OR. IVAR .GT. 100)) GO TO 40
        IF(I .EQ. 3 .AND. (IVAR .LT. 28 .OR. IVAR .GT. 31)) GO TO 40
        IVALUE(I) = IVAR
40      WRITE(*,1400) IVALUE(I)
1400    FORMAT(I5,$)
        IF(IVALUE(I) .EQ. IVAR) I = I + 1
        GO TO 30
    ELSE
        CHR = GETCHX(KEYTYPE)
        CALL WLOCATE(1,IROW(I),ICOL(I),*100)
        CALL PRINTW(1,'          ',*100)
        GO TO 30
    END IF
END IF
50    CONTINUE
    CHR = GETCHX(KEYTYPE)
    CALL CUROFF

C
C    COPY INPUT RESULTS INTO COMMON ARRAYS
    ITEMP = IVALUE(1)
    IREL = IVALUE(2)
    MONTH = IVALUE(3)
100   RETURN
    END

C
C
C
C
C
C
    SUBROUTINE PAUSE
$INCLUDE: 'spinattr.hdr'
    COMMON /FRAME/SCN(1024)
    CALL DEFWIND(99,20,0,24,11,DBLBDR,NORMAL)
    CALL WINDBUF(SCN)
    CALL CLRB(99)
    CALL BLDB(99,1,2,'ANYKEY',NORMAL,*100)
    CALL DISPSCN(SCN)
    CALL ANYKEY
100   RETURN
    END

```

SUBROUTINE CALC
LOGICAL FEXIST

COMMON /AIRPARA/ALT(20), FMEAN(20), ICODE(20), IREL, ITEMP,
1 MONTH, NUMAIR, FNUMD(20), FNUMN(20), POWER(20), SEL(20,25),
2 SIGMA, VEL(20)
COMMON /RESULT/FLDNMR(201),FLDNM(201),FLEQ(201),RANK(20,201)
COMMON /SPACE/FINTEN(20,147,3)
DATA PIDEG,DEGPI/0.01745329252,57.29577951/

SLANT RANGE 0 TO 73,000 FEET.

DO 20 J = 1, 147

 OFFSET = FLOAT((J-1)*500)

 DO 10 I = 1, NUMAIR

 SLANT = SQRT(ALT(I)*ALT(I) + OFFSET*OFFSET)

 INT = IFIX(10.0*LOG10(SLANT) - 19.0)

 IF(INT .GT. 24) INT = 24

 IF(INT .LT. 1) INT = 1

 S1 = 10.0**((FLOAT(INT+19)/10.))

 S2 = 10.0**((FLOAT(INT+20)/10.))

 VALUE = SEL(I,INT) +

1 (((SEL(I,INT)-SEL(I,INT+1))/LOG10(S1/S2))
2 *LOG10(SLANT/S1))

 BETA = ASIN(ALT(I)/SLANT)*DEGPI

LATERAL ATTENUATION

 IF(BETA .GT. 2.0 .AND. BETA .LT. 45.0) THEN

 ATTEN = -0.3786 + 17.04/BETA

 ELSE

 IF(BETA .LE. 2.0) THEN

 ATTEN = 8.14

 ELSE

 ATTEN = 0.0

 ENDIF

 END IF

 VALUE = VALUE - ATTEN

EQUATION FOR PREDICTION OF ONSET RATES FOR LOW-ALTITUDE,
HIGH SPEED AIRCRAFT OPERATIONS. REFERENCE LETTER FROM
JERRY SPEAKMAN, AAMRL APAFB FEBRUARY 9, 1987. THIS
EQUATION APPEARS IN ZROUTE FOR THE SURPRISE EFFECT.

 IF(OFFSET .EQ. 0.) OFFSET = 1.

 ONSET = 100.0/((1.0 + 2.7183**((10.01 - 3.62*LOG10(VEL(I))

1 + 2.48*LOG10(ALT(I)) + 0.15*LOG10(OFFSET)

2 - 0.0542*VALUE))

EQUATION FOR PREDICTION OF ONSET RATE RATE. THIS EQUATION
WAS REFERENCED BY JERRY SPEAKMAN ON SEPTEMBER 18, 1987

 ONSET = 10.0**((0.002247*VEL(I) - 0.0003228*ALT(I)

1 - 0.00014*OFFSET + 0.008987*VALUE - 0.434)

 IF(ONSET .LT. 15.) THEN

 ONPEN = 0.0

```

ELSE
  IF(ONSET .GT. 30.) THEN
    ONPEN = 5.0
  ELSE
    ONPEN = 16.6*LOG10(ONSET/15.0)
  ENDIF
ENDIF

C
C   INTENSITY USED FOR Ldnmr
FINTEN(I,J,1)=((FNUMD(I)+10.0*FNUMN(I))/(FLOAT(MONTH)*86400.))
1   *10.**((VALUE + ONPEN)/10.0)

C
C   INTENSITY USED FOR Ldnm
FINTEN(I,J,2)=((FNUMD(I)+10.0*FNUMN(I))/(FLOAT(MONTH)*86400.))
1   *10.**((VALUE)/10.0)

C
C   INTENSITY USED FOR Leq
FINTEN(I,J,3)=((FNUMD(I)+FNUMN(I))/(FLOAT(MONTH)*86400.))
1   *10.**((VALUE)/10.0)

10   CONTINUE
20   CONTINUE

C
C   TABLE ALL DONE NOW INTEGRATE ACROSS THE MTR
C
C   FIRST DEFINE SOME CONSTANTS
C
C   THE CONSTANT 41 SERVES AS A 10,000 FOOT BUFFER ZONE. THIS
C   WILL ENSURE ALL THE NECESSARY LEVELS ARE CALCULATED.
KMAX = 4*IFIX(SIGMA)/500 + 41
IF(FLOAT(KMAX)/2.0 .EQ. FLOAT(KMAX/2)) KMAX = KMAX + 1
KARGU = (KMAX/2) + 1
DO 22 J = 1, 201
  DO 23 I = 1, 20
    RANK(I,J) = 0.0
23   CONTINUE
22   CONTINUE

C
C   SIGMA      KMAX      KARGU
C   6600       93       47
C   13200      147      74
C   2640       63       32
C
C   NOW INTEGRATE ACROSS THE ROUTE
DO 50 J = 1, 201
  SUM1 = 0.0
  SUM2 = 0.0
  SUM3 = 0.0
  GNDPOS = FLOAT((J-101)*500)
  DO 40 I = 1, NUMAIR
    IF(GNDPOS.LT.FMEAN(I)-(2.0*SIGMA+10000.)) .OR.
1    GNDPOS.GT.FMEAN(I)+(2.0*SIGMA+10000.)) GO TO 40
    SUM4 = 0.0
    DO 30 K = 1, KMAX
      DIFF = FLOAT((K-KARGU)*500.0)
      X = FMEAN(I) + DIFF
      XINIT = X - 249.0
      XFINL = X + 250.0
      CALL GAUSS(XINIT,XFINL,FMEAN(I),SIGMA,PROB)
      OFFSET = ABS(GNDPOS - X)

```

```

        IOFF = IFIX(OFFSET/500.0) + 1
        P = MOD(OFFSET/500.0,1.0)
        IF(P .GT. 0.5) THEN
            P = P - 1
            IOFF = IOFF + 1
        ENDIF
        SUM1 = SUM1 + PROB*FDMI(I,IOFF,1,P)
        SUM2 = SUM2 + PROB*FDMI(I,IOFF,2,P)
        SUM3 = SUM3 + PROB*FDMI(I,IOFF,3,P)
        SUM4 = SUM4 + PROB*FDMI(I,IOFF,1,P)
30      CONTINUE
        IF(SUM4 .LE. 1.0) THEN
            RANK(I,J) = 0.0
        ELSE
            RANK(I,J) = 10.0*LOG10(SUM4)
        END IF
40      CONTINUE
        IF(SUM1 .LE. 1.0) THEN
            FLDNMR(J) = 0.0
        ELSE
            FLDNMR(J) = 10.0*LOG10(SUM1)
        END IF
        IF(SUM2 .LE. 1.0) THEN
            FLDNM(J) = 0.0
        ELSE
            FLDNM(J) = 10.0*LOG10(SUM2)
        END IF
        IF(SUM3 .LE. 1.0) THEN
            FLEQ(J) = 0.0
        ELSE
            FLEQ(J) = 10.0*LOG10(SUM3)
        END IF
50      CONTINUE
        RETURN
        END

C
C
C
C
C
C
FUNCTION FDMI(I,J,K,P)
C      FINITE DIFFERENCE METHOD INTERPOLATION FOR NUMERICAL FUNCTION
C      TABULATED AT EQUAL INTERVALS
C
COMMON /SPACE/FINTEN(20,147,3)
IF(J .EQ. 1) THEN
C      NEWTON-GREGORY FORWARD FORMULA
        FDMI = FINTEN(I,J,K) + P*(FINTEN(I,J+1,K) - FINTEN(I,J,K))
1          + (P*(P-1)/2.0)*(FINTEN(I,J+2,K) - 2.0*FINTEN(I,J+1,K)
2          + FINTEN(I,J,K))
        ELSE
            IF(J .EQ. 147) THEN
C      NEWTON-GREGORY BACKWARD FORMULA
                FDMI = FINTEN(I,J,K) + P*(FINTEN(I,J,K) - FINTEN(I,J-1,K))
1                + (P*(P-1)/2.0)*(FINTEN(I,J,K) - 2.0*FINTEN(I,J-1,K)
2                + FINTEN(I,J-2,K))
            ELSE
C      STIRLING FORMULA
                FDMI = FINTEN(I,J,K) + (P/2.0)*(FINTEN(I,J+1,K) - FINTEN(I,J-1,K))
1                + (P*P/2.0)*(FINTEN(I,J+1,K) - 2.0*FINTEN(I,J,K)

```



```

C
C
C   MAIN PLOTTING PROGRAM
C
C
C   SUBROUTINE PLTMAIN(SWITCH,TOGGLE)
SINCLUDE: 'spinattr.hdr'
      INTEGER*2 IOPORT,MODEL
      LOGICAL FLAG(6),FLAG01,SWITCH,TOGGLE
      CHARACTER CHR
      CHARACTER*16 PTNAME
      CHARACTER*40 TITLE,TEXT(3)
      COMMON /FRAME/SCN(1024)
      COMMON /RESULT/FLDNMR(201),FLDNM(201),FLEQ(201),RANK(20,201)
      COMMON /PLT/TITLE,FMIN,FMAX,FLAG,IOPORT,MODEL,FACT,ICOUNT
C
      DATA TEXT(1) /'File Not Found'/
      DATA TEXT(2) /'Enter Plot File Name With Extension :'/
      DATA TEXT(3) /'Error Reading File'/
C
      IF(SWITCH) THEN
        FLAG01 = .FALSE.
C
C   IF USER DID NOT RUN CALC THEN READ IN SPL DATA
      IF USER DID NOT RUN CALC THEN READ IN SPL DATA
        CALL DEFWIND(1,8,0,14,79,SINGLBD,NORMAL)
        GO TO 20
10      IF(FLAG01) GO TO 500
        CALL WINDBUF(SCN)
        CALL CLR(1)
        CALL BLDB(1,2,10,TEXT(1),0,*500)
        CALL DISPSCN(SCN)
        CALL PAUSE
        CALL CLS
        CALL SYSTEM('DIR/W',*500)
        CALL ANYKEY
        CALL CLRKB
        FLAG01 = .TRUE.
20      CALL WINDBUF(SCN)
        CALL CLS
        CALL CAPTSCN(SCN)
        CALL CLR(1)
        CALL BLDB(1,2,10,TEXT(2),0,*500)
        CALL DISPSCN(SCN)
        CALL CURON
        CALL WLOCATE(1,2,47,*500)
        READ(*,1000) PTNAME
1000      FORMAT(A16)
        CALL CUROFF
        OPEN(UNIT=7,ERR=10,FILE=PTNAME,FORM='BINARY',RECL=1,
1          STATUS='OLD')
        READ(7,ERR=30) FLDNMR, FLDNM, FLEQ
        CLOSE(UNIT=7)
      ELSE
        END IF
        GO TO 40
30      CALL WINDBUF(SCN)
        CALL CLR(1)
        CALL BLDB(1,2,10,TEXT(3),0,*500)
        CALL DISPSCN(SCN)
        CALL PAUSE

```

```

GO TO 500
40 CONTINUE
C
C INPUT PLOT
CALL PLTMENU(SWITCH,TOGGLE)
C
C INPUT THE DEVICE
CALL DEVICE
CALL CLS
C
C DRAW THE SOUND PRESSURE LEVELS
IF(FLAG(1) .OR. (FLAG(2) .OR. FLAG(3))) CALL PLTSPL
C
C DRAW THE PROBABILITY OF HIGHLY ANNOYED ACROSS THE MTR
IF(FLAG(6)) CALL PLTAPP
C
C NOW RETURN TO THE MAIN PROGRAM
WRITE(*,1100)
1100 FORMAT(' READY TO RETURN TO MAIN MENU.',/,
1' Please press <return> to continue.', $)
READ(*,1200) CHR
1200 FORMAT(A1)
500 RETURN
END
C
C
C DETERMINE DEVICE TYPE
C
C
C SUBROUTINE DEVICE
INCLUDE: 'spinattr.hdr'
DIMENSION IA(1), IROW(3), ICOL(3)
INTEGER*2 IO(5), IOPORT, IVALUE(2), KEYTYPE, MOD(5), MODEL
LOGICAL FLAG(6), ISKEY
CHARACTER IDUMMY, CHR, GETCHX
CHARACTER*40 TITLE
CHARACTER*50 TEXT(16)
COMMON /FRAME/SCN(1024)
COMMON /PLT/TITLE, FMIN, FMAX, FLAG, IOPORT, MODEL, FACT, ICOUNT
C
C PLOTTING DEVICE DATA
DATA IO / 0, 9600, 93, 97, 99/
DATA MOD / 5, 30, 93, 96, 99/
C
C POSITION MARKERS FOR OTHER PLOTTING DEVICE SCREEN
DATA IROW / 8, 9, 10/
DATA ICOL / 35, 34, 34/
C
DATA TEXT( 1) //'AVAILABLE PLOTTING DEVICES' /
DATA TEXT( 2) //'(1) Epson FX-80 Or FX-85 Printer' /
DATA TEXT( 3) //'(2) HP 7475A Plotter' /
DATA TEXT( 4) //'(3) Hercules Graphics Card (HGC)' /
DATA TEXT( 5) //'(4) IBM Enhanced Graphics Adapter (EGA)' /
DATA TEXT( 6) //'(5) IBM Color Graphics Adapter (CGA)' /
DATA TEXT( 7) //'(6) Other' /
DATA TEXT( 8) //'Your Selection : ' /
DATA TEXT( 9) //'OTHER PLOTTING DEVICE' /
DATA TEXT(10) //'Enter the IOPORT number and MODEL number.' /
DATA TEXT(11) //'Consult PLOT88 User Reference Manual to determine' /
DATA TEXT(12) //'appropriate settings for your device.' /

```

```

DATA TEXT(13)/'IOPORT number :'/
DATA TEXT(14)/'MODEL number :'/
DATA TEXT(15)/'Scale Factor :'/
DATA TEXT(16)/'press ESC when all done'/

```

C
C

```

DETERMINE THE DEVICE
CALL DEFWIND(1,0,5,20,73,DBLBDR,NORMAL)
CALL WINDBUF(SCN)
CALL CLS
CALL CAPTSCN(SCN)
CALL CLR(1)

```

```

CALL BLDB(1, 1,20,TEXT(1),0,*500)
CALL BLDB(1, 4,12,TEXT(2),0,*500)
CALL BLDB(1, 6,12,TEXT(3),0,*500)
CALL BLDB(1, 8,12,TEXT(4),0,*500)
CALL BLDB(1,10,12,TEXT(5),0,*500)
CALL BLDB(1,12,12,TEXT(6),0,*500)
CALL BLDB(1,14,12,TEXT(7),0,*500)
CALL BLDB(1,17,40,TEXT(8),0,*500)
CALL DISPSCN(SCN)
CALL CURON

```

10

```

CALL WLOCATE(1,17,56,*500)
IF(.NOT. ISKEY(CHR,KEYTYPE)) THEN
  GO TO 10
ELSE
  CALL CTOA(CHR,IA,0)
  IF((KEYTYPE .EQ. 3) .AND. (IA(1).GE.49 .AND. IA(1).LE.54)) THEN
    ICOUNT = IA(1) - 48
    CHR = GETCHX(KEYTYPE)
  ELSE
    CALL WLOCATE(1,17,56,*500)
    CALL PRINTW(1,' ',*500)
    CHR = GETCHX(KEYTYPE)
    GO TO 10
  END IF
END IF
IF(ICOUNT .LE. 5) THEN
  CALL CUROFF
  IOPORT = IO(ICOUNT)
  MODEL = MOD(ICOUNT)

```

```

ELSE
  CALL DEFWIND(2,0,5,15,73,DBLBDR,NORMAL)
  CALL WINDBUF(SCN)
  CALL CLS
  CALL CAPTSCN(SCN)
  CALL CLR(2)
  CALL BLDB(2, 1,23,TEXT( 9),0,*500)
  CALL BLDB(2, 4,13,TEXT(10),0,*500)
  CALL BLDB(2, 6, 9,TEXT(11),0,*500)
  CALL BLDB(2, 7,15,TEXT(12),0,*500)
  CALL BLDB(2, 8,20,TEXT(13),0,*500)
  CALL BLDB(2, 9,20,TEXT(14),0,*500)
  CALL BLDB(2,10,20,TEXT(15),0,*500)
  CALL BLDB(2,13,22,TEXT(16),0,*500)
  CALL DISPSCN(SCN)

```

C
C

```

INPUT RESULTS ALLOW FOR EDITING FEATURES
  I = 1
  GO TO 30
20  READ(*,1000) IDUMMY

```

```

1000    FORMAT(A1)
        CALL WLOCATE(2,IROW(I),ICOL(I),*500)
        CALL PRINTW(2,'      ',*500)
30      CONTINUE
        IF(I.EQ. 4) I = 1
        CALL WLOCATE(2,IROW(I),ICOL(I),*500)
        IF(.NOT. ISKEY(CHR,KEYTYPE)) THEN
            GO TO 30
        ELSE
            CALL CTOA(CHR,IA,0)
            IF(KEYTYPE.EQ. 16 .AND. IA(1).EQ. 27) GO TO 40
            IF(KEYTYPE.EQ. 4 .AND. IA(1).EQ. 13) THEN
                I = I + 1
                CHR = GETCHX(KEYTYPE)
                GO TO 30
            ELSE
                END IF
            IF((KEYTYPE.EQ. 3) .AND.
1          ((IA(1).GE.46 .AND. IA(1).LE.57) .AND. IA(1).NE.47)) THEN
                IF(I.NE. 3) THEN
                    CALL PRINTW(2,'      ',*500)
                    CALL WLOCATE(2,IROW(I),ICOL(I),*500)
                    READ(*,1100,ERR=20) IVALUE(I)
                    CALL WLOCATE(2,IROW(I),ICOL(I),*500)
                    WRITE(*,1200) IVALUE(I)
1100        FORMAT(I5)
1200        FORMAT(I5,$)
                    I = I + 1
                    GO TO 30
                ELSE
                    CALL PRINTW(2,'      ',*500)
                    CALL WLOCATE(2,IROW(I),ICOL(I),*500)
                    READ(*,1300,ERR=20) FACT
                    CALL WLOCATE(2,IROW(I),ICOL(I),*500)
                    WRITE(*,1400) FACT
1300        FORMAT(F6.2)
1400        FORMAT(F6.2,$)
                    I = I + 1
                    GO TO 30
                END IF
            ELSE
                CHR = GETCHX(KEYTYPE)
                CALL WLOCATE(2,IROW(I),ICOL(I),*500)
                CALL PRINTW(2,'      ',*500)
                GO TO 30
            END IF
        END IF
        CONTINUE
40      CHR = GETCHX(KEYTYPE)
        CALL CUROFF
        IOPORT = IVALUE(1)
        MODEL = IVALUE(2)
        END IF
        CALL WLOCATE(1,0,0,*500)
500    RETURN
        END
C
C
C    LINE DRAWING ROUTINE
C

```

C

```

SUBROUTINE DRAW(XARRAY,YARRAY,NPTS,FLAG1,INTEQ)
DIMENSION XARRAY(203), YARRAY(203)
INTEGER*2 NPTS
LOGICAL FLAG1, FLAG2
CHARACTER CTEXT
XMIN = XARRAY(NPTS+1)
YMIN = YARRAY(NPTS+1)
XDELTA = XARRAY(NPTS+2)
YDELTA = YARRAY(NPTS+2)
FLAG2 = .TRUE.
IF(FLAG1) THEN
  IF(INTEQ .EQ. 1) IVAL = 11
  IF(INTEQ .EQ. 2) IVAL = 9
  IF(INTEQ .EQ. 5) IVAL = 7
ELSE
  END IF
DO 10 I = 1, NPTS
  XPOS = (XARRAY(I) - XMIN)/XDELTA
  YPOS = (YARRAY(I) - YMIN)/YDELTA
  IF(YPOS .LE. 0.0) THEN
    IF(FLAG2) THEN
      CALL PLOT(XPOS,0.0,3)
    ELSE
      CALL PLOT(XPOS,0.0,2)
    END IF
    FLAG2 = .TRUE.
  ELSE
    FLAG2 = .FALSE.
    IF(FLAG1 .AND. FLOAT(I)/FLOAT(IVAL).EQ.FLOAT(I/IVAL)) THEN
      CTEXT = CHAR(INTEQ)
      CALL SYMBOL(XPOS,YPOS,0.12,CTEXT,0.0,-2)
    ELSE
      IF(I .EQ. 1) THEN
        CALL PLOT(XPOS,YPOS,3)
      ELSE
        CALL PLOT(XPOS,YPOS,2)
      END IF
    END IF
  END IF
END IF
10 CONTINUE
RETURN
END

```

10

C
C
C
C
C

PLOTING MENU

```

SUBROUTINE PLTMENU(SWITCH,TOGGLE)
$INCLUDE: 'spinattr.hdr'
DIMENSION IA(1), IROW(9), ICOL(9), VALUE(2)
INTEGER*2 IOPORT,KEYTYPE,MODEL
LOGICAL FLAG(6),ISKEY,SWITCH,TOGGLE
CHARACTER CHR, GETCHX, CTEXT
CHARACTER*3 IDUMMY
CHARACTER*40 TITLE
CHARACTER*65 TEXT(17)
COMMON /FRAME/SCN(1024)
COMMON /PLT/TITLE,FMIN,FMAX,FLAG,IOPORT,MODEL,FACT,ICOUNT

```

C

```

DATA IROW/ 4, 7, 8,11,12,13,14,16,20/
DATA ICOL/23,29,29,46,44,44,45,36,37/
DATA TEXT( 1) //'PLOT RESULTS'/
DATA TEXT( 2) //'GENERAL'/
DATA TEXT( 3) //'Enter Title :'/
DATA TEXT( 4) //'Enter The Distances Across MTR To Be Plotted.'/
DATA TEXT( 5) //'Minimum :'/
DATA TEXT( 6) //'Maximum :'/
DATA TEXT( 7) //'NOISE LEVEL'/
DATA TEXT( 8) //'Do You Want To Plot Ldnmr (Yes/No) :'/
DATA TEXT( 9) //'Do You Want To Plot Ldn (Yes/No) :'/
DATA TEXT(10) //'Do You Want To Plot Leq (Yes/No) :'/
DATA TEXT(11) //'Do You Want Symbols Used (Yes/No) :'/
DATA TEXT(12) //'Do You Want To Collapse These Curves'/
DATA TEXT(13) //'Onto One Plot (Yes/No) :'/
DATA TEXT(14) //'ANNOYANCE'/
DATA TEXT(15) //'Do You Want To Plot The Probability Of'/
DATA TEXT(16) //'High Annoyance (Yes/No) :'/
DATA TEXT(17) //'press ESC when all done'/

```

C
C

```

DRAW PLOTTING MENU
CALL DEFWIND(1,0.5,24,73,DBLBDR,NORMAL)
CALL WINDBUF(SCN)
CALL CLS
CALL CAPTSCN(SCN)
CALL CLRB(1)
CALL BLDB(1, 1,28,TEXT( 1),0,*500)
CALL BLDB(1, 3, 5,TEXT( 2),0,*500)
CALL BLDB(1, 4,10,TEXT( 3),0,*500)
CALL BLDB(1, 6,10,TEXT( 4),0,*500)
CALL BLDB(1, 7,20,TEXT( 5),0,*500)
CALL BLDB(1, 8,20,TEXT( 6),0,*500)
CALL BLDB(1,10, 5,TEXT( 7),0,*500)
CALL BLDB(1,11,10,TEXT( 8),0,*500)
CALL BLDB(1,12,10,TEXT( 9),0,*500)
CALL BLDB(1,13,10,TEXT(10),0,*500)
CALL BLDB(1,14,10,TEXT(11),0,*500)
CALL BLDB(1,15,10,TEXT(12),0,*500)
CALL BLDB(1,16,12,TEXT(13),0,*500)
CALL BLDB(1,18, 5,TEXT(14),0,*500)
CALL BLDB(1,19,10,TEXT(15),0,*500)
CALL BLDB(1,20,12,TEXT(16),0,*500)
CALL BLDB(1,22,22,TEXT(17),0,*500)
CALL DISPSCN(SCN)
CALL CURON
IF(SWITCH .OR. .NOT. TOGGLE) THEN

```

C
C

```

WRITE DEFAULT VALUES TO SCREEN
CALL WLOCATE(1,IROW(2),ICOL(2),*500)
CALL PRINTW(1,'-50000.',*500)
CALL WLOCATE(1,IROW(3),ICOL(3),*500)
CALL PRINTW(1,'50000.',*500)

```

C
C

```

INPUT RESULTS, ALLOW FOR EDITING FEATURES
VALUE(1) = -50000.0
VALUE(2) = 50000.0
ELSE
CALL WLOCATE(1,IROW(1),ICOL(1),*500)
WRITE(*,1200) TITLE
CALL WLOCATE(1,IROW(2),ICOL(2),*500)

```

```

WRITE(*,1400) VALUE(1)
CALL WLOCATE(1,IROW(3),ICOL(3),*500)
WRITE(*,1400) VALUE(2)
DO 10 I = 1, 6
  CALL WLOCATE(1,IROW(I+3),ICOL(I+3),*500)
  IF(FLAG(I)) THEN
    CALL PRINTW(1,'Yes',*500)
  ELSE
    CALL PRINTW(1,'No',*500)
  END IF
10  CONTINUE
END IF
CALL CLRKB
I = 1
20  CONTINUE
IF(I .EQ. 10) I = 1
CALL WLOCATE(1,IROW(I),ICOL(I),*500)
IF(.NOT. ISKEY(CHR,KEYTYPE)) THEN
  GO TO 20
ELSE
  CALL CTOA(CHR,IA,0)
  IF(KEYTYP .EQ. 16 .AND. IA(1) .EQ. 27) GO TO 30
  IF(KEYTYP .EQ. 4 .AND. IA(1) .EQ. 13) THEN
    I = I + 1
    READ(*,1000) CHR
    1000  FORMAT(A1)
    GO TO 20
  ELSE
    END IF
    IF((KEYTYPE .EQ. 2 .OR. KEYTYPE .EQ. 3) .AND. I .EQ. 1) THEN
      TITLE = '
      WRITE(*,1200) TITLE
      CALL WLOCATE(1,IROW(1),ICOL(1),*500)
      READ(*,1100,ERR=20) TITLE
      CALL WLOCATE(1,IROW(1),ICOL(1),*500)
      WRITE(*,1200) TITLE
      1100  FORMAT(A40)
      1200  FORMAT(1X,A40,S)
      I = I + 1
      GO TO 20
    ELSE
      END IF
      IF(((KEYTYPE.EQ.3) .AND. (IA(1).GE.45.AND. IA(1).LE.57)) .AND.
1      (I .EQ. 2 .OR. I .EQ. 3)) THEN
        CALL PRINTW(1,' ',*500)
        CALL WLOCATE(1,IROW(I),ICOL(I),*500)
        READ(*,1300,ERR=20) VAR
        1300  FORMAT(G10.0)
        CALL WLOCATE(1,IROW(I),ICOL(I),*500)
        IF(VAR.GE.-50000.0 .AND. VAR.LE.50000.0) VALUE(I-1) = VAR
        WRITE(*,1400) VALUE(I-1)
        1400  FORMAT(F8.0,S)
        IF(VALUE(I-1) .EQ. VAR) I = I + 1
        GO TO 20
      ELSE
        END IF
        IF(KEYTYP .EQ. 2 .AND. (I .GE. 4 .AND. I .LE. 9)) THEN
          CALL PRINTW(1,' ',*500)
          CALL WLOCATE(1,IROW(I),ICOL(I),*500)
          READ(*,1500,ERR=20) IDUMMY

```

```

1500      CALL WLOCATE(1,IROW(I),ICOL(I),*500)
        FORMAT(A3)
        IF(IDUMMY(1:1) .EQ. 'Y' .OR. IDUMMY(1:1) .EQ. 'y') THEN
            CALL PRINTW(1,'Yes',*500)
            FLAG(I-3) = .TRUE.
            I = I + 1
            GO TO 20
        ELSE
            IF(IDUMMY(1:1) .EQ. 'N' .OR. IDUMMY(1:1) .EQ. 'n') THEN
                CALL PRINTW(1,'No ',*500)
                FLAG(I-3) = .FALSE.
                I = I + 1
                GO TO 20
            ELSE
                GO TO 20
            END IF
        END IF
    ELSE
        END IF
    CHR = GETCHX(KEYTYPE)
    CALL WLOCATE(1,IROW(I),ICOL(I),*500)
    CALL PRINTW(1,'          ',*500)
    GO TO 20
END IF
CONTINUE
30      CHR = GETCHX(KEYTYPE)
    CALL CUROFF
    DIFFER = VALUE(2) - VALUE(1)
    IF((VALUE(1) .LT. -50000. .OR. VALUE(2) .GT. 50000.) .OR.
1DIFFER .LT. 1500.0) THEN
        VALUE(1) = -50000.0
        VALUE(2) = 50000.0
    ELSE
        END IF
    FMIN = VALUE(1)
    FMAX = VALUE(2)
500    RETURN
    END

```

C
C
C
C
C

SOUND PRESSURE LEVEL PLOTTING PROGRAM

C
C

```

SUBROUTINE PLTSPL
DIMENSION XARRAY(203), YARRAY(203)
INTEGER*2 IOPT,MODEL,NC,NPTS,NBLANK
LOGICAL FLAG(6)
CHARACTER CTEXT
CHARACTER*40 TITLE
COMMON /RESULT/FLDNMR(201),FLDNM(201),FLEQ(201),RANK(20,201)
COMMON /PLT/TITLE,FMIN,FMAX,FLAG,IOPT,MODEL,FACT,ICOUNT

SET PLOTTING PARAMETERS
FMIN = FLOAT(IFIX(FMIN/500.0))*500.0
FMAX = FLOAT(IFIX(FMAX/500.0))*500.0
DIFF = FMAX - FMIN
DELTA = DIFF/(FLOAT(10))
AXLEN = -10.0
IFMIN = IFIX(FMIN/500.0) + 101
IFMAX = IFIX(FMAX/500.0) + 101

```

```

NPTS = (IFMAX - IFMIN) + 1
C
C
IF(FLAG(1)) THEN
  I = 1
ELSE
  IF(FLAG(2)) THEN
    I = 2
  ELSE
    I = 3
  END IF
END IF
10 CONTINUE
CALL PLOTS(0,IOPORT,MODEL)
CALL FONT(2)
C
C SET FACTOR FOR HARD COPY DEVICE OR TERMINAL
CALL FACTOR(0.7)
C
C ONE INCH EQUALS ONE CENTIMETER ON HARD COPY DEVICE
IF(ICOUNT .LE. 2) CALL FACTOR(0.787)
C
C OTHER DEVICE SCALING FACTOR
IF(ICOUNT .EQ. 6) CALL FACTOR(FACT)
C
C DRAW THE TITLE AND CENTER IT PLEASE
NC = NBLANK(TITLE)
X = 6.0 - (FLOAT(NC)/2.0)*(7.5/40.0)
CALL SYMBOL(X,6.5,0.24,TITLE,0.0,40)
C
C DRAW THE AXIS BOARDER
CALL STAXIS(-.177, 0.177, .089, .089,-1)
CALL AXIS(11.0,1.0,' ',0.5,0.90,0.0,0.0,0.0)
CALL AXIS(1.0,6.0,' ',0,AXLEN,0.0,0.0,0.0,0.0)
C
C SET THE DEFAULTS TO DRAW ANNOTATED AXIS
CALL STAXIS( .177, .177, .089, .089, -1)
C
C DRAW THE X AXIS
CALL AXIS(1.0,1.0,'Distance in feet',-16,AXLEN,0.0,FMIN,DELTA)
C
C NOW USE THE FLAGS TO DRAW THE Y AXIS
IF(FLAG(5)) THEN
  CALL AXIS(1.0,1.0,'Noise Level',11,-5.0,90.0,40.0,10.0)
  IF(FLAG(4)) THEN
    CTEXT = CHAR(1)
    IF(FLAG(1)) CALL SYMBOL(9.0,5.43,0.15,CTEXT,0.0,-1)
    IF(FLAG(1)) CALL SYMBOL(9.5,5.33,0.15,'Ldnmr',0.0,5)
    CTEXT = CHAR(2)
    IF(FLAG(2)) CALL SYMBOL(9.0,5.10,0.15,CTEXT,0.0,-1)
    IF(FLAG(2)) CALL SYMBOL(9.5,5.00,0.15,'Ldn',0.0,3)
    CTEXT = CHAR(5)
    IF(FLAG(3)) CALL SYMBOL(9.0,4.76,0.15,CTEXT,0.0,-1)
    IF(FLAG(3)) CALL SYMBOL(9.5,4.66,0.15,'Leq',0.0,3)
  ELSE
    END IF
  ELSE
    GO TO (20, 30, 40), I
    CALL AXIS(1.0,1.0,'Ldnmr in dB',11,-5.0,90.0,40.0,10.0)
    CTEXT = CHAR(1)
20

```

```

        IF(FLAG(4)) CALL SYMBOL(9.0,5.43,0.15,CTEXT,0.0,-1)
        IF(FLAG(4)) CALL SYMBOL(9.5,5.33,0.15,'Ldnmr',0.0,5)
30      GO TO 50
        CALL AXIS(1.0,1.0,'Ldn in dB',9,-5.0,90.0,40.0,10.0)
        CTEXT = CHAR(2)
        IF(FLAG(4)) CALL SYMBOL(9.0,5.10,0.15,CTEXT,0.0,-1)
        IF(FLAG(4)) CALL SYMBOL(9.5,5.00,0.15,'Ldn',0.0,3)
        GO TO 50
40      CALL AXIS(1.0,1.0,'Leq in dB',9,-5.0,90.0,40.0,10.0)
        CTEXT = CHAR(5)
        IF(FLAG(4)) CALL SYMBOL(9.0,4.76,0.15,CTEXT,0.0,-1)
        IF(FLAG(4)) CALL SYMBOL(9.5,4.66,0.15,'Leq',0.0,3)
END IF
50      CONTINUE
C
C      COPY SOUND PRESSURE LEVEL INTO PLOTTING ARRAY
        CALL PLOT(1.0,1.0,-3)
60      K = 0
        SUM = FMIN
        DO 100 J = IFMIN, IFMAX
            K = K + 1
            XARRAY(K) = SUM
            SUM = SUM + 500
            GO TO (70, 80, 90), I
70          YARRAY(K) = FLDNMR(J)
            INTEQ = 1
            GO TO 100
80          YARRAY(K) = FLDNM(J)
            INTEQ = 2
            GO TO 100
90          YARRAY(K) = FLEQ(J)
            INTEQ = 5
100     CONTINUE
        XARRAY(NPTS+1) = FMIN
        XARRAY(NPTS+2) = DELTA
        YARRAY(NPTS+1) = 40.0
        YARRAY(NPTS+2) = 10.0
        CALL DRAW(XARRAY,YARRAY,NPTS,FLAG(4),INTEQ)
        IF(FLAG(5)) THEN
            I = I + 1
            IF(FLAG(I) .AND. I .LE. 3) GO TO 60
            I = I + 1
            IF(FLAG(I) .AND. I .LE. 3) GO TO 60
            CALL PLOT(0.0,0.0,999)
        ELSE
            END IF
        IF(.NOT. FLAG(5)) THEN
            CALL PLOT(0.0,0.0,999)
            I = I + 1
            IF(FLAG(I) .AND. I .LE. 3) GO TO 10
            I = I + 1
            IF(FLAG(I) .AND. I .LE. 3) GO TO 10
        ELSE
            END IF
        RETURN
    END

```

C
C
C
C

PROBABILITY OF HIGHLY ANNOYED PLOTTING PROGRAM

```

C
SUBROUTINE PLTAPP
DIMENSION XARRAY(203), YARRAY(203)
INTEGER*2 IOPORT,MODEL,NC,NPTS,NBLANK
LOGICAL FLAG(6)
CHARACTER CTEXT
CHARACTER*40 TITLE
COMMON /RESULT/FLDNMR(201),FLDNM(201),FLEQ(201),RANK(20,201)
COMMON /PLT/TITLE,FMIN,FMAX,FLAG,IOPORT,MODEL,FACT,ICOUNT

C
C
SET PLOTTING PARAMETERS
CALL CLS
FMIN = FLOAT(IFIX(FMIN/500.0))*500.0
FMAX = FLOAT(IFIX(FMAX/500.0))*500.0
DIFF = FMAX - FMIN
DELTA = DIFF/(FLOAT(10))
AXLEN = -10.0
IFMIN = IFIX(FMIN/500.0) + 101
IFMAX = IFIX(FMAX/500.0) + 101
NPTS = (IFMAX - IFMIN) + 1

C
C
CALL PLOTS(0,IOPORT,MODEL)
CALL FONT(2)

C
C
SET FACTOR FOR HARD COPY DEVICE OR TERMINAL
CALL FACTOR(0.7)

C
C
ONE INCH EQUALS ONE CENTIMETER ON HARD COPY DEVICE
IF(ICOUNT .LE. 2) CALL FACTOR(0.787)

C
C
OTHER DEVICE SCALING FACTOR
IF(ICOUNT .EQ. 6) CALL FACTOR(FACT)

C
C
DRAW THE TITLE AND CENTER IT PLEASE
NC = NBLANK(TITLE)
X = 6.0 - (FLOAT(NC)/2.0)*(7.5/40.0)
CALL SYMBOL(X,6.5,0.24,TITLE,0.0,40)

C
C
DRAW THE AXIS BOARDER
CALL STAXIS(-.177, 0.177, .089, .089,-1)
CALL AXIS(11.0,1.0,' ',0.5,0.90,0.0,0.0,0.0)
CALL AXIS(1.0,6.0,' ',0,AXLEN,0.0,0.0,0.0)

C
C
SET THE DEFAULTS TO DRAW ANNOTATED AXIS
CALL STAXIS( .177, .177, .089, .089, -1)

C
C
DRAW THE X AXIS
CALL AXIS(1.0,1.0,'Distance in feet',-16,AXLEN,0.0,FMIN,DELTA)

C
C
COPY SOUND PRESSURE LEVEL INTO PLOTTING ARRAY
K = 0
SUM = FMIN
VALUE = 0.0
DO 10 J = IFMIN, IFMAX
  K = K + 1
  XARRAY(K) = SUM
  SUM = SUM + 500
  YARRAY(K) = 1.0/(1.0+EXP(10.43-0.132*FLDNMR(J)))
  IF(YARRAY(K) .GT. VALUE) VALUE = YARRAY(K)

```

```

10  CONTINUE
    XARRAY(NPTS+1) = FMIN
    XARRAY(NPTS+2) = DELTA
    YARRAY(NPTS+1) = 0.0
C
C  SET THE DEFAULTS TO DRAW ANNOTATED AXIS
    CALL STAXIS( .177, .177, .089, .089, 1)
    IF(VALUE .LT. 0.5) THEN
        YARRAY(NPTS+2) = 0.1
C
C  DRAW THE Y AXIS
    CALL AXIS(1.0,1.0,'Prob. of High Annoyance',23,-5.0,90.0,0.0,0.1)
    ELSE
        YARRAY(NPTS+2) = 0.2
C
C  DRAW THE Y AXIS
    CALL AXIS(1.0,1.0,'Prob. of High Annoyance',23,-5.0,90.0,0.0,0.2)
    END IF
    CALL PLOT(1.0,1.0,-3)
    CALL DRAW(XARRAY,YARRAY,NPTS,.FALSE.,1)
    CALL PLOT(0.0,0.0,999)
    RETURN
    END

```

```

SUBROUTINE RANKODR
$INCLUDE: 'spinattr.hdr'
INTEGER*2 KEYTYPE
INTEGER IA(1)
LOGICAL ISKEY, FLAG(20)
CHARACTER*1 CHR, GETCHX, IDUMMY
CHARACTER*2 OPCR(300)
CHARACTER*3 ACC(300), AIRSPD(300)
CHARACTER*5 POWSET(300)
CHARACTER*6 POWUNT(300)
CHARACTER*7 NAME(300)
CHARACTER*20 POWDES(300)
CHARACTER*25 DRAG(300)
CHARACTER*50 OUTPUT
CHARACTER*55 TEXT(6)
COMMON /AIRPARA/ALT(20), FMEAN(20), ICODE(20), IREL, ITEMP,
1 MONTH, NUMAIR, FNUMD(20), FNUMN(20), POWER(20), SEL(20,25),
2 SIGMA, VEL(20)
COMMON /FRAME/SCN(1024)
COMMON /NOISE/MAX,ACC,AIRSPD,DRAG,NAME,OPCR,POWDES,POWSET,POWUNT
COMMON /RESULT/FLDNMR(201),FLDNM(201),FLEQ(201),RANK(20,201)
DATA TEXT(1) /'Enter the distance in feet from corridor centerline
1'/
DATA TEXT(2) /'to ground location point of interest :'/
DATA TEXT(3) /'Distance Out Of Range'/
DATA TEXT(4) /'          Rank Contributors At Distance ='/
DATA TEXT(5) /'          Config      Aircraft      Ldnmr      Prob of High
1ly'/
DATA TEXT(6) /'          Number      Name                        Annoyed '
1/

C
DO 10 I = 1, 20
  FLAG(I) = .TRUE.
10 CONTINUE
20 CALL DEFWIND(1,8,0,14,79,DBLBDR,NORMAL)
  CALL WINDBUF(SCN)
  CALL CLS
  CALL CAPTSCN(SCN)
  CALL CLRB(1)
  CALL BLDB(1,1,10,TEXT(1),0,*500)
  CALL BLDB(1,2,10,TEXT(2),0,*500)
  CALL DISPSCN(SCN)
  CALL CURON
  GO TO 40
30 READ(*,1000) IDUMMY
1000 FORMAT(A1)
  CALL WLOCATE(1,2,48,*500)
  CALL PRINTW(1,' ',*500)
40 CALL WLOCATE(1,2,48,*500)
  IF(.NOT. ISKEY(CHR,KEYTYPE)) THEN
    GO TO 40
  ELSE
    CALL CTOA(CHR,IA,0)
    IF((KEYTYPE.EQ. 3) .AND.
1      ((IA(1).GE.45 .AND. IA(1).LE.57) .AND. IA(1).NE.47)) THEN
      READ(*,1100,ERR=30) DIST
      WRITE(OUTPUT,1100)
1100      FORMAT(F10.0)
    ELSE
      CHR = GETCHX(KEYTYPE)

```

```

        CALL WLOCATE(1,2,48,*500)
        CALL PRINTW(1,'',*500)
        GO TO 40
    END IF
END IF
CALL CUROFF
IF(DIST .LT. -50000.0 .OR. DIST .GT. 50000.0) THEN
    CALL DEFWIND(1,8,0,14,79,DBLBDR,NORMAL)
    CALL WINDBUF(SCN)
    CALL CLRB(1)
    CALL BLDB(1,2,20,TEXT(3),0,*500)
    CALL DISPSCN(SCN)
    CALL PAUSE
    GO TO 20
ELSE
END IF
C
C   DETERMINE WHICH VECTOR WAS PICKED
DIST = DIST + 50000.0
IVAL = IFIX(DIST/500.0) + 1
P = MOD(DIST/500.0,1.0)
IF(P .GT. 0.5) IVAL = IVAL + 1
C
C   PREPARE SCREEN
CALL DEFWIND(2,0,11,24,68,DBLBDR,NORMAL)
CALL WINDBUF(SCN)
CALL CLS
CALL CAPTSCN(SCN)
CALL CLRB(2)
CALL BLDB(2,0,1,TEXT(4),0,*500)
CALL BLDB(2,1,1,TEXT(5),0,*500)
CALL BLDB(2,2,1,TEXT(6),0,*500)
CALL DISPSCN(SCN)
DIST = (IVAL - 101)*500.0
WRITE(OUTPUT,1200) DIST
1200 FORMAT(G8.3E1)
CALL WLOCATE(2,0,40,*500)
CALL PRINTW(2,OUTPUT(1:8),*500)
DO 80 I = 1, NUMAIR
    J = 0
50   J = J + 1
    IF(.NOT. FLAG(J) .AND. J .LT. NUMAIR) GO TO 50
    IF(J .EQ. NUMAIR) THEN
        ITRACK = NUMAIR
        GO TO 70
    ELSE
    END IF
    VALMAX = RANK(J,IVAL)
    ITRACK = J
    DO 60 K = 1, NUMAIR
        IF(RANK(K,IVAL) .GT. VALMAX .AND. FLAG(K)) THEN
            VALMAX = RANK(K,IVAL)
            ITRACK = K
        ELSE
        END IF
60   CONTINUE
70   FLAG(ITRACK) = .FALSE.
    ANNOY = 1.0/(1.0 + EXP(10.43 - 0.132*RANK(ITRACK,IVAL)))
    CALL WLOCATE(2,I+2,2,*500)
    WRITE(OUTPUT,1300) ITRACK,NAME(ICODE(ITRACK)),

```

```
      1      IFIX(RANK(ITRACK,IVAL)),ANNOY
1300      FORMAT(4X,I3,9X,A7.5X,I3,9X,F6.3)
      CALL PRINTW(2,OUTPUT(1:50),*500)
80      CONTINUE
      CALL ANYKEY
500     RETURN
      END
```

```

C
C
C      ADDS TWO NUMBERS IN DECIBELS
C
C      SUBROUTINE ADDER
$INCLUDE: 'spinattr.hdr'
      DIMENSION IA(1), ICOL(3), VALUE(3)
      INTEGER*2 KEYTYPE
      LOGICAL ISKEY
      CHARACTER IDUMMY, CHR, GETCHX
      CHARACTER*34 TEXT(5)
      COMMON /FRAME/SCN(1024)
      PARAMETER (IROW=6)
      DATA TEXT(1)/'DECIBEL ADDITION'/
      DATA TEXT(2)/'Enter Noise Levels In Decibels'/
      DATA TEXT(3)/'      dB +      dB =      dB'/
      DATA TEXT(4)/'press RETURN to advance the cursor'/
      DATA TEXT(5)/'press ESC when all done'/
      DATA ICOL/10, 21, 32/
      DATA VALUE/3*0.0/
      CALL CURON
      CALL DEFWIND(1,5,15,17,65,DBLBDR,NORMAL)
      CALL WINDBUF(SCN)
      CALL CLS
      CALL CAPTSCN(SCN)
      CALL CLR(1)
      CALL BLDB(1, 1,17,TEXT(1),0,*100)
      CALL BLDB(1, 3,10,TEXT(2),0,*100)
      CALL BLDB(1, 6,10,TEXT(3),0,*100)
      CALL BLDB(1, 9, 8,TEXT(4),0,*100)
      CALL BLDB(1,10,13,TEXT(5),0,*100)
      CALL DISPCN(SCN)

C
C      INPUT RESULTS, ALLOW FOR EDITING FEATURES
      I = 1
      GO TO 20
10      READ(*,1000) IDUMMY
1000     FORMAT(A1)
      CALL WLOCATE(1,IROW,ICOL(I),*100)
      CALL PRINTW(1,'      ',*100)
20      CONTINUE
      IF(I.EQ. 3) THEN
          CALL WLOCATE(1,IROW,ICOL(I),*100)
          I = 1
          VALUE(3) = 10.0**((VALUE(1)/10.0) + 10.0**((VALUE(2)/10.0)
          IF(VALUE(3).LE. 0.0) THEN
              CALL PRINTW(1,'*****',*100)
          ELSE
              VALUE(3) = 10.0*LOG10(VALUE(3))
              WRITE(*,1100) VALUE(3)
1100      FORMAT(F5.1,$)
          END IF
      ELSE
          END IF
      CALL WLOCATE(1,IROW,ICOL(I),*100)
      IF(.NOT. ISKEY(CHR,KEYTYPE)) THEN
          GO TO 20
      ELSE
          CALL CTOA(CHR,IA,0)

```

```

IF(KEYTYPE .EQ. 16 .AND. IA(1) .EQ. 27) GO TO 30
IF(KEYTYPE .EQ. 4 .AND. IA(1) .EQ. 13) THEN
    I = I + 1
    CHR = GETCHX(KEYTYPE)
    GO TO 20
ELSE
    END IF
IF((KEYTYPE .EQ. 3) .AND.
1    ((IA(1).GE.45 .AND. IA(1).LE.57) .AND. IA(1).NE.47)) THEN
    CALL PRINTW(1,' ',*100)
    CALL WLOCATE(1,IROW,ICOL(I),*100)
    READ(*,1200,ERR=10) VALUE(I)
    IF(VALUE(I) .GT. 140.0 .OR. VALUE(I) .LT. 0.0) GO TO 20
    CALL WLOCATE(1,IROW,ICOL(I),*100)
    WRITE(*,1300) VALUE(I)
1200    FORMAT(F5.0)
1300    FORMAT(F6.1,$)
    I = I + 1
    GO TO 20
ELSE
    CHR = GETCHX(KEYTYPE)
    CALL WLOCATE(1,IROW,ICOL(I),*100)
    CALL PRINTW(1,' ',*100)
    GO TO 20
    END IF
END IF
30    CONTINUE
    CHR = GETCHX(KEYTYPE)
    CALL CUROFF
100    RETURN
    END

```

* This is the Header file for specifying Screen Attributes
 * and Window Parameters

* The Color Attributes are of 2 types: Those determining the
 * BACKGROUND Color, and those determining the CHARACTER Color.

* The Colors are Black (BLK), Dark Blue (DBLU), Green (GREEN),
 * Red (RED), Lavender (LAV), Yellow (YELLO), and White (WHITE).

* To make a particular Color combination on the screen, merely
 * add up the two colors desired for the BACKGROUND (BKG), and
 * the CHARACTERS (CHR).

* For example, to place RED characters on a GREEN background,
 * use the sum: REDCHR + GREENBKG

* If the Characters are to be of NORMAL Intensity, Nothing else
 * need be done. If the Characters are to be BRITE also, use
 * REDCHR + GREENBKG + BRITE

* If they are to FLASH also, use:
 * REDCHR + GREENBKG + BRITE + FLASH

* The WINDOW Parameters are SCROLL, WRAP, DBLBDR for Double Border,
 * SINGBD for Single Border, LIST for no carriage control,
 * and NONE for no Parameters

* If a Window was to have Scrolling and a Double Border, use
 * SCROLL + DBLBDR

* Note that if LIST is not specified, then Fortran will assume
 * that normal formatting rules are in effect: The first character
 * will be assumed to be a Carriage Control Character.

```

1  INTEGER BLKBKG, BLKCHR, DBLUBKG, DBLUCHR, GREENBKG, GREENCHR,
2    LBLUBKG, LBLUCHR, REDBKG, REDCHR, LAVBKG, LAVCHR,
3    YELLOBKG, YELLOCHR, WHITEBKG, WHITECHR, FLASH, UNDRLINE,
4    NORMAL, BRITE, BRIGHT, INVERSE, SCROLL, WRAP, DBLBDR, SINGLBD,
    UNITZERO, DEFAULT

```

```

PARAMETER (BLKCHR = 0)
PARAMETER (BLKBKG = 0)
PARAMETER (DBLUBKG = 16)
PARAMETER (DBLUCHR = 1)
PARAMETER (GREENBKG = 32)
PARAMETER (GREENCHR = 2)
PARAMETER (REDBKG = 64)
PARAMETER (REDCHR = 4)
PARAMETER (LAVBKG = 80)
PARAMETER (LAVCHR = 5)
PARAMETER (YELLOBKG = 96)
PARAMETER (YELLOCHR = 6)
PARAMETER (WHITEBKG = 112)
PARAMETER (WHITECHR = 7)
PARAMETER (UNDRLINE = 1)
PARAMETER (DEFAULT = 0)
PARAMETER (NORMAL = 7)

```

PARAMETER (FLASH = 128)
PARAMETER (INVERSE = 112)
PARAMETER (BRIGHT = 15)
PARAMETER (BRITE = 8)
PARAMETER (LBLUBKG = 48)
PARAMETER (LBLUCHR = 3)
PARAMETER (NONE = 0)
PARAMETER (SINGLBD = 1)
PARAMETER (UNITZERO = 0)
PARAMETER (DBLBDR = 3)
PARAMETER (WRAP = 4)
PARAMETER (SCROLL = 8)
PARAMETER (LIST = 16)