REFERENCE USE ONLY

REPORT NO. DOT-TSC-RSPA-78-19

THE AIRPORT NOISE PREDICTION MODEL--MOD 7

R. H. Hinckley
W. Messcher

U.S. Department of Transportation
Research and Special Programs Administration
Transportation Systems Center
Cambridge MA 02142

JULY 1978

FINAL REPORT

DOCUMENT IS AVAILABLE TO THE U.S. PUBLIC THROUGH THE NATIONAL TECHNICAL INFORMATION SERVICE, SPRINGFIELD, VIRGINIA 22161

Prepared for

U.S. DEPARTMENT OF TRANSPORTATION
RESEARCH AND SPECIAL PROGRAMS ADMINISTRATION
OFFICE OF POLICY, PLANS & ADMINISTRATION
WASHINGTON DC 20590

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
DOT-TSC-RSPA-78-19		

4. Title and Subtitle	5. Report Date
THE AIRPORT NOISE PREDICTION MODEL--MOD 7	July 1978
	6. Performing Organization Code
7. Author(s) R.H. Hinckley and W. Messcher	8. Performing Organization Report No. DOT-TSC-RSPA-78-19
9. Performing Organization Name and Address U.S. Department of Transportation Research and Special Programs Administration Transportation Systems Center Cambridge MA 02142	10. Work Unit No. (TRAIS) FA865/R8143 11. Contract or Grant No.
12. Sponsoring Agency Name and Address U.S. Department of Transportation Research and Special Programs Administration Office of Policy, Plans & Administration Washington DC 20590	13. Type of Report and Period Covered Final Report July 1974 - July 1977 14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

The MOD 7 Airport Noise Prediction Model is fully operational. The language used is FORTRAN, and it has been run on several different computer systems. Its capabilities include prediction of noise levels for single parameter changes, for multiple changes, and for an entire airport's operations. Some of the single parameters include: type of aircraft, flight paths, speed, thrust, and noise abatement procedures.

17. Key Words	18. Distribution Statement
Noise Exposure Noise Exposure Forecast (NEF) Airport Noise	DOCUMENT IS AVAILABLE TO THE U.S. PUBLIC THROUGH THE NATIONAL TECHNICAL INFORMATION SERVICE, SPRINGFIELD, VIRGINIA 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	134	

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

PREFACE

This airport noise prediction program has been developed at the Transportation Systems Center. This work was accomplished under the OST Office of Noise Abatement as part of PPA OS-707.

The MOD 7 version updates earlier versions MOD 4, 5, and 6. Major changes include:

- Updated noise tables and elimination of obsolete tables
- Addition of L_{dn} and ASDS and elimination of WECPNL
- No upper bounds of quantities except a maximum of 2,500 grid points.
- Reduced program size
- Faster running time
- Use of noise-table equations rather than points
- Additional IMOD-7 input program.

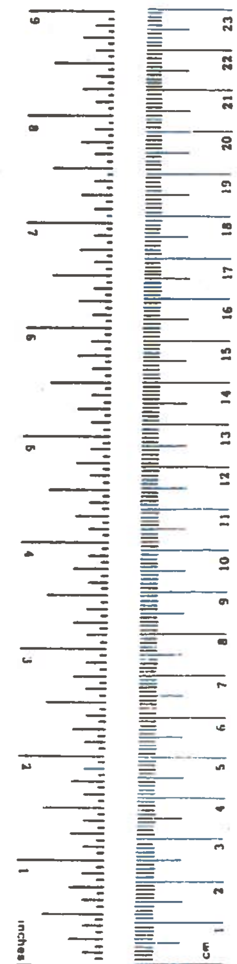
Although two authors are identified for this manual, other persons have also contributed to the content and development of the MOD-7 capabilities. Mr. John Wesler, AEQ-2, provided all of the mathematics for the accoustics as well as overall project objectives. Mr. James Steinberg and Mr. Thomas Vaughan (TSC) made contributions in the graphics area. Mr. Prescott Heald and Mr. Michael Hoffman of Kentron Int'l., Ltd., contributed significantly to the basic programming.

Mr. Leon Tritter of the Raytheon Service Company deserves credit for valuable assistance in the editing and publication of this manual.

METRIC CONVERSION FACTORS

Approximate Conversions to Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
in	inches	2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
		AREA		
in^2	square inches	6.5	square centimeters	cm^2
ft^2	square feet	0.09	square meters	m^2
yd^2	square yards	0.8	square meters	m^2
mi^2	square miles	2.6	square kilometers	km^2
	acres	0.4	hectares	ha
		MASS (weight)		
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons (2000 lb)	0.9	tonnes	t
		VOLUME		
tsp	teaspoons	5	milliliters	ml
Tbsp	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cups	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.95	liters	l
gal	gallons	3.8	liters	l
ft^3	cubic feet	0.03	cubic meters	m^3
yd^3	cubic yards	0.76	cubic meters	m^3
		TEMPERATURE (exact)		
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C



Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
m	meters	1.1	yards	yd
km	kilometers	0.6	miles	mi
		AREA		
cm^2	square centimeters	0.16	square inches	in^2
m^2	square meters	1.2	square yards	yd^2
km^2	square kilometers	0.4	square miles	mi^2
ha	hectares (10,000 m^2)	2.5	acres	
		MASS (weight)		
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1000 kg)	1.1	short tons	
		VOLUME		
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m^3	cubic meters	35	cubic feet	ft^3
m^3	cubic meters	1.3	cubic yards	yd^3
		TEMPERATURE (exact)		
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F

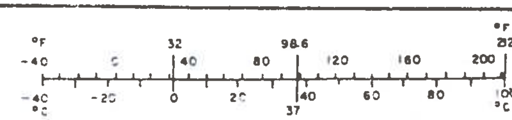


TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (cont'd)

INTRODUCTION

Purpose of Document

The purpose of this document is to provide a description of the MOD 7 Airport Noise Prediction Model and procedures for its use. Instructions are also included for preparing and running a sample case.

General Description of MOD-7 Airport Noise Prediction Model

The MOD-7 Airport Noise Prediction Model provides a user with the ability to predict noise levels for selected points around an airport for various combinations of number and types of aircraft runway usage and a wide variety of aircraft operations. The model includes four types of noise metrics as follows:

NE - Noise Exposure Forecast - equal weighting 24-hour period

NEF - Noise Exposure Forecast - day and night weightings

ASDS - Aircraft Sound Description System

L_{dn} - Day-night Average Sound Level

The computer program is written in FORTRAN IV, and is currently operational on a PRIME 300 mini-computer system. It is also operational on other systems, including an IBM 360. Inputs are disk files and outputs can be printouts of gridded numerical values and/or a computer plot of selected noise contours. The contour plotting routing is done at TSC on a PRIME 300 mini-computer with a CALCOMP 1136 drum plotter. The plotting program is readily adaptable to most computer-plotter systems.

Running time for the program is highly variable. The running time is proportional to:

Number of flight operations

Complexity of flight paths

Number of grid points.

All aircraft noise data characteristics are stored in the program. The user need enter only the airport geometry and the operational data.

BACKGROUND

Operational Use of Previous Models

The Airport Noise Prediction Model has been operational for more than three years at TSC. The applications of the model, whose capabilities are indicated in Table 1, may be categorized into the following areas.

a. Sensitivity Analyses of One Parameter. Studies have been made of many individual parameters. This type of prediction is relatively easy to prepare. CPU times on the PRIME 300 system are a few seconds. Some examples include analyses of:

Engine retrofits

Climb and descent angle changes

Thrust changes

Speed changes

Flight path parameter changes.

b. Analyses of Multiple Parameters. This category requires more input preparation and more CPU time than the single parameter analysis. Some of these applications include:

Addition of operations from a new runway

Change of runway utilization

Combinations of items in a above.

c. Noise Prediction for an Entire Airport. This includes all aircraft types, flight paths, takeoffs, landings, day and night flights, etc. These studies have been done not only for individual airports, but for multiple interacting airports. For example, a study was made of the impact of wide-body jet aircraft upon National Airport, Washington DC. One condition was that the number of passenger seating remain constant for the totals of National, Dulles, and Baltimore Airports.

TABLE 1. CAPABILITIES OF OPERATIONAL NOISE PREDICTION MODELS

INPUT PARAMETERS
- RUNWAY (S) - LOCATION (S)
- OPERATIONS DATA:
  - RUNWAY(S) USED
  - TYPE AND NUMBER OF AIRCRAFT
  - DESIRED ENGINE/FAN RETROFIT
  - FLIGHT PATH, CLIMB ANGLE, THRUST
  - TAKEOFF/LANDING
  - TIME OF DAY
  - EXCESS GROUND ATTENUATION
  - SHIELDING
- COMPUTER PROCESSING INSTRUCTIONS
  - TITLE
  - LOCATION OF GROUND OBSERVER POINTS
  - NE, NEF, ASDS,OR L_{dn} METRIC
  - OUTPUT DEVICE (S)
  - CONTOUR LEVEL
  - OBSERVER GRID ARRAY OR SINGLE POINT

PROCESSING FUNCTIONS
- SELECT AIRCRAFT DATA
- SELECT NOISE DATA
- MODIFY STORED DATA, IF INSTRUCTED
- CALCULATE NE, NEF, L_{dn}, OR ASDS
- CALCULATE CONTOURS

OUTPUTS
- CONTOUR LEVEL PLOTS ON CALCOMP PLOTTER
- LINE PRINTOUT OF NOISE LEVEL (S)

Enhancements

The noise model can produce individual grid points of calculated noise levels or contours of noise levels from the selected grid array. Often the major reason for the study will be to determine the noise impact on people. TSC has developed additional capabilities for combining on a single CALCOMP plot:

The selected contour(s) of predicted noise

A map of key locations and geography for reference

Centroids of population levels

A summation of population levels

A summation of population within the noise contour.

Figure 1 shows the above for Washington National Airport.

Of the above five capabilities listed, the contouring routine is the only one included in the MOD 7 program. The others are accomplished by post processing and are listed to show examples of additional ways to present output information.

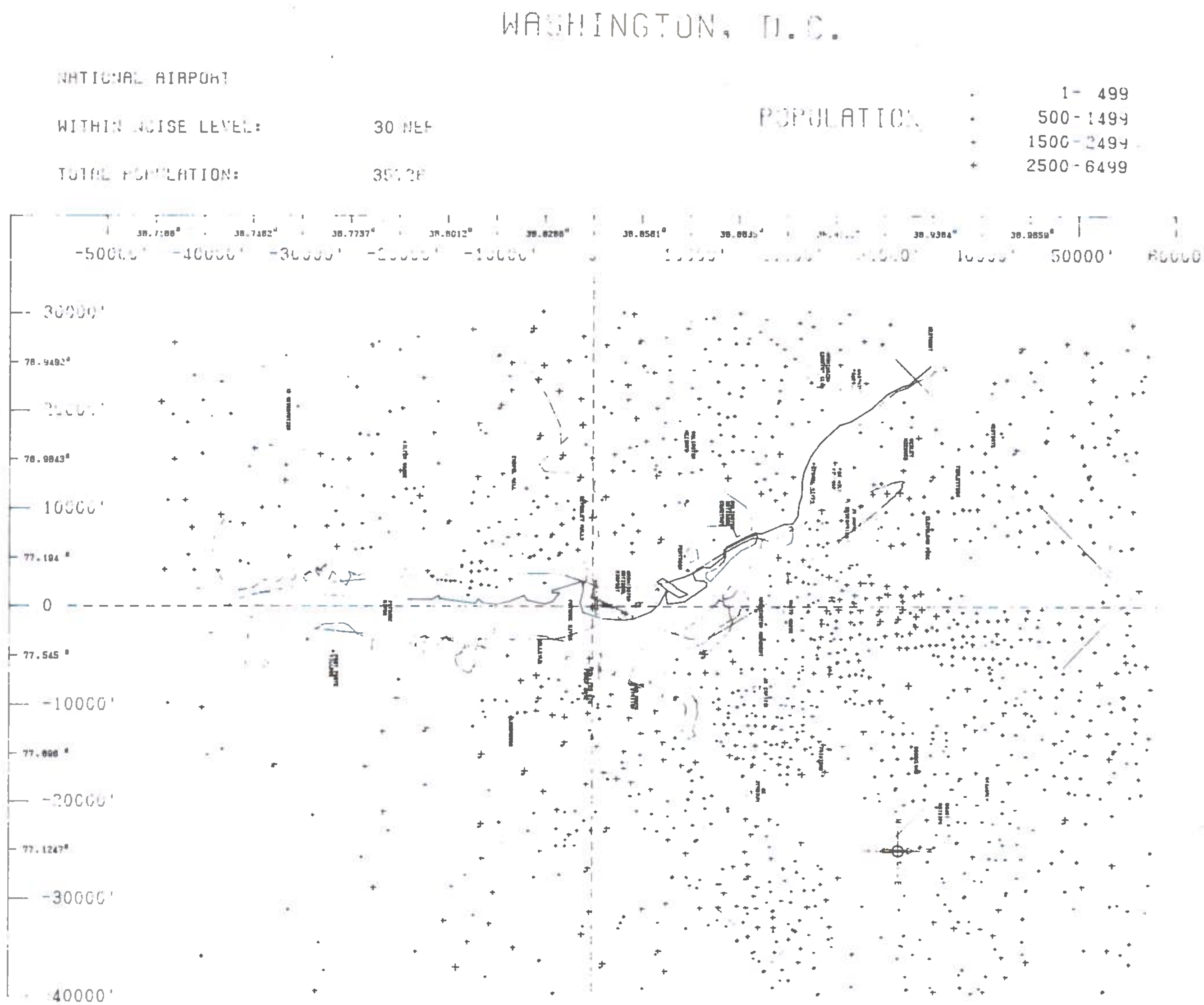


FIGURE 1. COMPUTER PLOT FOR WASHINGTON NATIONAL AIRPORT

CAPABILITIES

Options and Parameters Related to Noise Prediction Calculations

The Noise Prediction Model will calculate predicted noise values at selected grid points around an airport. The following options are available:

NE, NEF, L_{dn}, ASDS metrics

Individual observer sites on a grid of observer sites

Shielding and excess ground attenuation.

The following parameters must be supplied by the user:

Runways

Number of flight paths

Type of aircraft

Takeoff or landing

Number of day flights

Number of night flights

Flight path geometry

Engine thrusts and/or speeds.

Flight Paths

Each flight path is composed of one or more segments. A segment of a flight path is defined as one where there is no change of direction, change of turn radius, or climb or descent angle. Thus, ground roll would be one segment, and liftoff at constant climb angle and a helical turn at constant climb angle and constant radius would each be single segments.

Flights

Any number of flights may be used. One flight, as seen on a sample coding sheet, contains the following items:

One type of aircraft

Constant numbers of day and night operations

Constant flight paths for all operations

Either takeoff or landing

Same beginning speeds for all operations

Same ending speeds for all operations

Same beginning thrusts for all operations

Same ending thrusts for all operations.

When any change is made in the above parameters, an additional flight is required.

MOD-7 Noise Data Table

Table 2 provides data pertaining to aircraft and engine types associated with the flights referred to above. The appearance of an asterisk in the EPNL Data or dBA Data column denotes the existence, in the model, of noise data of the type specified for that column. For example, the Lear Jet (aircraft type 10) does have EPNL versus distance information stored in the model (indicated by an asterisk in the EPNL Data column), but does not have any dBA versus distance data (indicated by the nonexistence of an asterisk in the dBA Data column). This means that NE and NEF may be calculated for a Lear Jet, but ASDS or L_{dn} can not be calculated.

TABLE 2. MOD 7 NOISE DATA

Coding Number	EPNL Data	dBA Data	Aircraft Type	Description
1	*		747-200	(JT9D-3A)
2	*		L-1011	
3	*		707-320B	
4	*		707-320	(JT4A-3)
5	*		727-200	(JT8D-9)
6	*		DC9-32	(JT8D-7)
7	*		F27	(2 PROPJET)
8	*		2 PISTON/OVER 12000 LB.	
9	*		2 PISTON/UNDER 12000 LB.	
10	*		LEARJET	(CJ610-6)
46	*	*	747-200B	(JT9D-7/QN)
47	*	*	707-320B	(JT3D-3B)
48	*	*	727-200	(JT8D-9,-15)
49	*	*	737-200	(JT8D-9,-15)
50	*	*	747-100	(JT9D-7)
51	*	*	707-320B/QN	(JT3D-3D/QN)
52	*	*	727-200/QN	(JT8D-9/QN,-15/QN)
53	*	*	737-200/QN	(JT8D-9/QN,-15/QN)
54	*		707-320B/RFN	(JT3D/RFN)
55	*	*	DC-8-61	(JT3D-3D)
56	*	*	DC-89-63	(JT3D-7)
57	*		DC-8-63/RFN	(JT3D/RFN)
58	*		727-200/RFN	(JT8D/RFN)
59	*		737-200/RFN	(JT8D/RFN)
60	*		DC-9-30/RFN	(JT8D/RFN)
61	*		DC-8-61/RFN	(JT3D/RFN)
63	*		DC-8-61	
64	*	*	DC-10-10	(CF6-6D)
65	*	*	DC-9-30	(JT8D-9)
66	*		CONCORDE	

MOD-7 METRICS

The noise metrics available in the MOD 7 Noise Prediction Model are presented in this section. The items of interest include:

Symbols used in derivation of noise prediction equations

Elements comprising the model

Noise prediction equations.

Symbols

Terms used in the noise prediction equations are defined in Table 3, and are geometrically identified as noted in the paragraph, Noise Prediction Model Equations.

Elements Comprising the Model

The following describes the elements comprising the model:

a. Noise Exposure (NE): Calculates the cumulative noise exposure at points on the ground around an airport, assuming equal weighting throughout the period considered (normally an average 24-hour day).

b. Noise Exposure Forecast (NEF): Calculates the cumulative noise exposure in the same manner as the Noise Exposure procedure, but applies a heavier emphasis, or weighting, for nighttime flights (normally defined as those occurring between 2200 and 0700 during an average day).

In the calculation of noise exposure values, aircraft noise levels are expressed in terms of Effective Perceived Noise Levels (EPNL) in units of EPNdB. This calculated value represents the loudness sensation (called Perceived Noise Level in PNdB) created for an "average" observer, considering the sound intensity and frequency spectrum generated by an aircraft, plus corrections for single frequency (or pure tone) content in the noise spectrum (if any) and the time duration of the aircraft's flyby. In simulating

TABLE 3. DEFINITIONS OF SYMBOLS USED IN NOISE PREDICTION EQUATIONS

Symbol	Definition
N	Number of total flight paths
k	Flight path k
X_{ijk}	The subscript ijk* indicates that the value X has been evaluated at the observer point O_{ij} with respect to point P, the point along the flight path k that is the minimum distance to the observer at point O_{ij}.
d_{ijk}	Distance (feet) from point O_{ij} to P for flight k
β_{ijk}	Observer elevation angle (radians) at point O_{ij} for flight k toward P
γ	Correction factor for elevation angle β
$\tilde{v}_{ijk}$	Velocity (knots) at point P for flight k
E_{ijk}	EPNdB (or dBA) at point O_{ij} for the flight path k from noise table (Table 2) with function $g(X_{ijk})$
V_{ijk}	Speed correction term at point O_{ij} for flight path k
S_{ijk}	Shielding correction term at point O_{ij} for flight path k
Λ_{ijk}	Excess ground attenuation term at point O_{ij} for flight path k
H_{mk}	Number of operations for day (m=1), evening (m=2), and night (m=3) for specified flight k
W_m	Weighting factor on operations
E_r	Background reference noise level (dBA)
T	Number of seconds in 24-hour day (86,400)

*Figures 2 and 3 illustrate the points and subscripts referenced above.

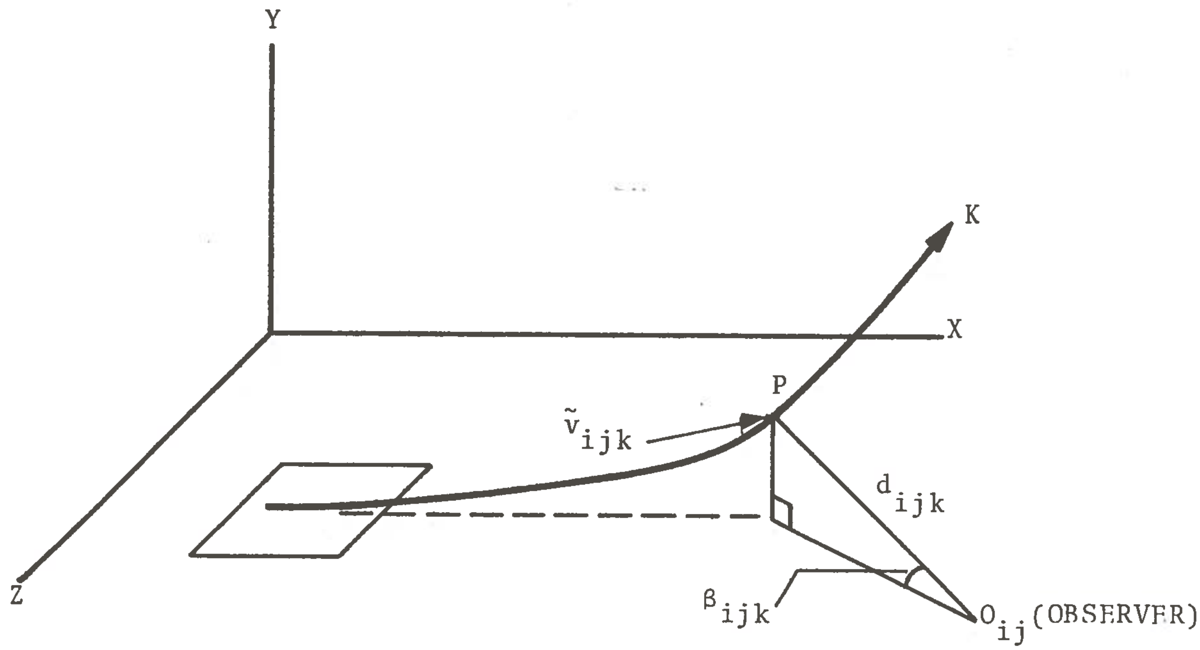


FIGURE 2. FLIGHT PROFILE

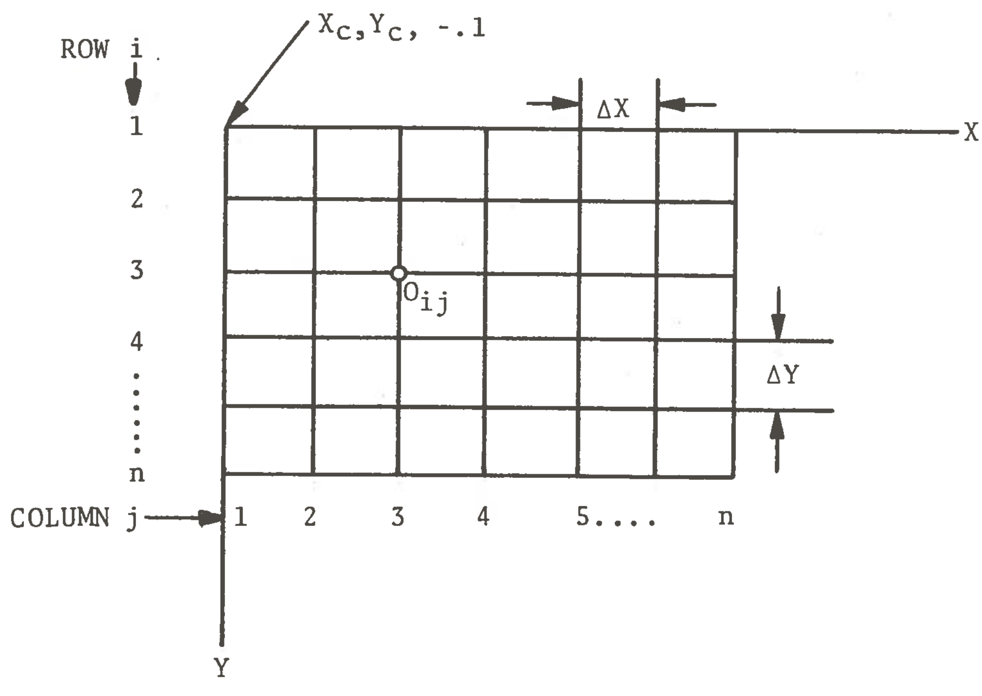


FIGURE 3. GRID MATRIX

the noise exposure near an airport, it is convenient to group aircraft into classes, based on noise characteristics and takeoff and landing profiles. Each class is assigned a set of EPNdB-versus-distance curves for various engine thrust levels, and a set of takeoff climb profiles for appropriate ranges of gross takeoff weight.

c. Aircraft Sound Description System (ASDS): This concept sums the periods of time during which the noise from aircraft exceeds a fixed A-weighted sound level during a fixed period such as a 24-hour day. The results are plotted as equal-time contours around the airport. Initially, the sound level threshold above which the time is counted has been selected as 85 dBA. The noise exposure computer model has the ability to preselect this threshold, at the user's option, using the 85 dBA value as a default value.

d. Day-night Average Sound Level (L_{dn}): This concept is identical to the NEF concept, except that A-weighted decibels are used to represent the aircraft noise levels in lieu of Effective Perceived Noise Level, and a dBA 10 penalty is added to nighttime noise levels in lieu of the 16.7 multiplier for nighttime flights used in the NEF concept. Since performance data for aircraft noise are available only as noise-thrust-altitude curves in peak A-weighted decibels, the time duration of the aircraft noise is also calculated, in order to integrate the "Effective A-weighted Noise Level."

In all procedures, the algorithms for calculating closest point of approach, noise level from closest point of approach and thrust setting, excess ground attenuation, and shielding are the same.

e. Correction Factors

(1) Velocity correction

(2) Shielding correction

(3) Excess ground attenuation correction

(4) -88: A number to ensure that the NEF is not confused numerically with other metric values.

Noise Prediction Model Equations

The noise prediction equations [Eq(1) through Eq(4)] are used for calculation of values for the noise metrics NE_{ij}, NEF_{ij}, $ASDS_{ij}$, and L_{dnij}. The subscript ijk is defined in Table 3 (see X_{ijk}), and i, j, ij, and ijk are identified at various points in Figures 2 and 3. Parameters frequently used in the equations are first defined (Table 4)* and, where required, discussed to establish the validity for their use in Eqs(1) through (4).

The equations for the four metrics are as follows:

For Metric NE

$$NE_{ij} = 10 \log_{10} \left(\sum_{k=1}^{N} 10^{B} \right) - 88 \quad \ldots\ldots\ldots\ldots \quad \text{Eq(1)}$$

where

$$B = \frac{E'_{ijk} + 10 \log_{10} \left(\sum_{m=1}^{3} W_m H_{mk} \right)}{10}$$

$$E'_{ijk} = E_{ijk} - A_{ijk} - S_{ijk} - V_{ijk}$$

$$E_{ijk} = g(d_{ijk})$$

$$W_1 = 1$$

$$W_2 = 1$$

$$W_3 = 1$$

*Table 4 follows Eq(4) and expressions for related parameters.

For Metric NEF

$$NEF_{ij} = 10 \log_{10}\left(\sum_{k=1}^{N} 10^{D} \right) -88 \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \quad \text{Eq(2)}$$

where

$$D = \frac{E'_{ijk} + 10 \log_{10}\left(\sum_{m=1}^{3} W_m H_{mk} \right)}{10}$$

$$E'_{ijk} = E_{ijk} - A_{ijk} - S_{ijk} - V_{ijk}$$

$$E_{ijk} = g(d_{ijk})$$

$$W_1 = 1$$

$$W_2 = 1$$

$$W_3 = 16.67$$

For Metric ASDS*

$$ASDS_{ij} = \begin{cases} \displaystyle\sum_{k=1}^{N} \frac{2 \sum_{m=1}^{3} H_{mk}}{1.689 \; \tilde{v}_{ijk}} \sqrt{r_{ijk}^{2} - d_{ijk}^{2}} & r_{ijk} > d_{ijk} \\ 0 & \text{Otherwise} \end{cases} \quad \text{Eq(3)}$$

where

$$r_{ijk} = g^{-1} (E'_{ijk})$$

$$E'_{ijk} = E_r + A_{ijk} + S_{ijk}$$

*For a discussion of ASDS calculations by use of TSC MOD 7, refer to Appendix A.

For Metric L_{dn}

$$L_{dn_{ij}} = E_R + 10 \log_{10} \left\{ 1 + \sum_{k=1}^{N} \frac{\tau_{ijk}}{T} \left[\left(H_{1k} + H_{2k}\right) \left(\frac{10^F - 1}{2.3} - F \right) + H_{3k} \left(\frac{10^G - 1}{2.3} - G \right) \right] \right\} \quad \ldots\ldots\ldots \quad \text{Eq (4)}$$

where

$$F = \frac{E'_{ijk} - E_R}{10}$$

$$G = \frac{E'_{ijk} - E_R + 10}{10}$$

$$\tau_{ijk} = \begin{cases} \dfrac{2}{1.689\ \tilde{v}_{ijk}} \sqrt{r^2_{ijk} - d^2_{ijk}} & r_{ijk} > d_{ijk} \\ 0 & \text{Otherwise} \end{cases}$$

$$r_{ijk} = \begin{cases} g^{-1} \ (E'_{ijk} - 10) & (E'_{ijk} - 10) > E_R \\ g^{-1} \ E_R & \text{Otherwise} \end{cases}$$

$$E'_{ijk} = E_{ijk} - A_{ijk} - S_{ijk} - V_{ijk}$$

$$E_{ijk} = g(d_{ijk})$$

TABLE 4. PARAMETERS COMMON TO NOISE PREDICTION EQUATIONS

Parameter	Nomenclature	Definition	Remarks
V_{ijk}	Velocity correction	$10 \log_{10} \left(\frac{\tilde{v}_{ijk}}{160} \right)$	Velocity measured in knots.
S_{ijk}	Shielding correction	$3 - 3 \sqrt{\lvert \sin \beta_{ijk} \rvert}$	Figure 4 shows variations in the shielding correction factor (S) with changes in the elevation angle (β).
A_{ijk}	Excess ground attenuation correction	$\begin{cases} \Delta_o & \text{for } \beta < \frac{\pi}{2Q} \\ 0 & \text{for } \beta \geq \frac{\pi}{2Q} \end{cases}$	(a) (1) β is in radians (2) $\gamma = \exp\left(-\sqrt{\lvert \tan[(Q)(\beta_{ijk})] \rvert}\right)$ Figure 5 shows γ as a function of elevation angle β. (3) Q is an empirical constant. (Commonly used values of Q are 3 and 13. The curves of Figure 5 are drawn for values of γ determined by use of these values for Q.)
		$\Delta_o = C_1 \tan^{-1} \left(\frac{\log_{10} \frac{d_{ijk}}{C_2}}{C_3} \right) + C_4$	(b) Constants $C_1 - C_4$ are thrust-dependent parameters. Their functional relationships with thrust were determined by a least squares method applied to

TABLE 4. PARAMETERS COMMON TO NOISE PREDICTION EQUATIONS (CONTINUED)

Parameter	Nomenclature	Definition	Remarks
A_{ijk} (cont.)			empirical data. The empirical data were originally supplied as a graph to Figure 6. The functions $C_1=C_1$ (thrust), $C_2=C_2$ (thrust),... etc. give rise to a family of Δ_o versus distance curves. Of that family, one curve coincides with the takeoff curve of Figure 6 when thrust equals 100% full thrust, and another curve coincides with the landing curve at thrust equal to 45% full thrust. Specifically, $C_i = \left(\frac{100\text{-Percent Thrust}}{55}\right)\left(A_{i,1}-A_{i,2}\right) + A_{i,2} \quad i\Big\vert_1^4$ where $A_{i,j} = \begin{bmatrix} 4.140 & 2.675 \\ 1200.000 & 1072.000 \\ 0.198 & 0.252 \\ 8.500 & 6.400 \end{bmatrix}$

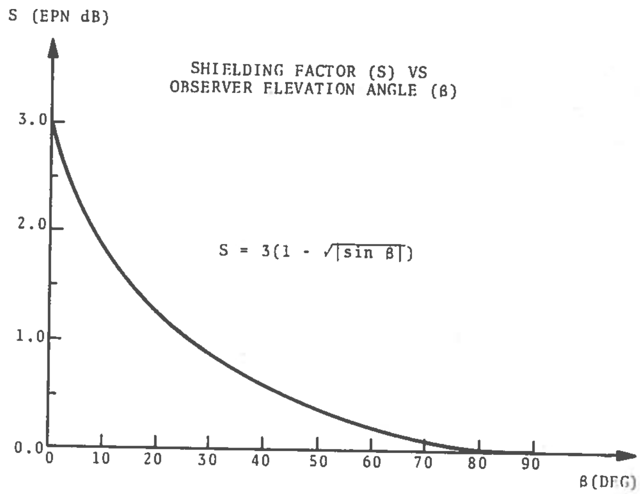


FIGURE 4. SHIELDING EFFECT GRAPH

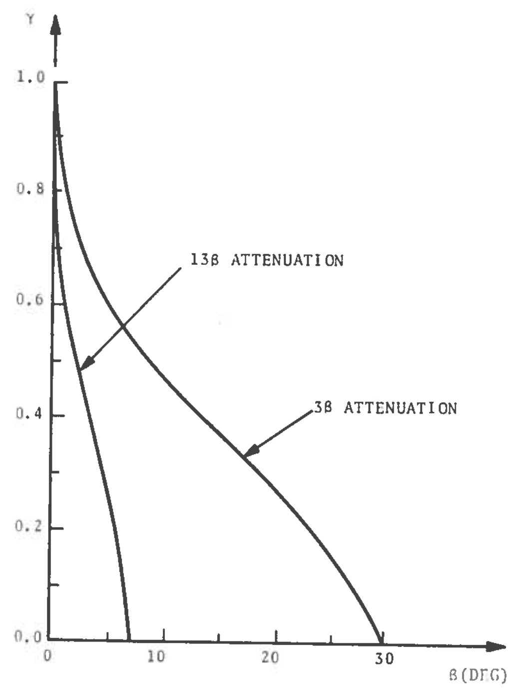


FIGURE 5. ELEVATION ANGLE CORRECTION FACTOR (GAMMA) VS ELEVATION ANGLE (BETA)

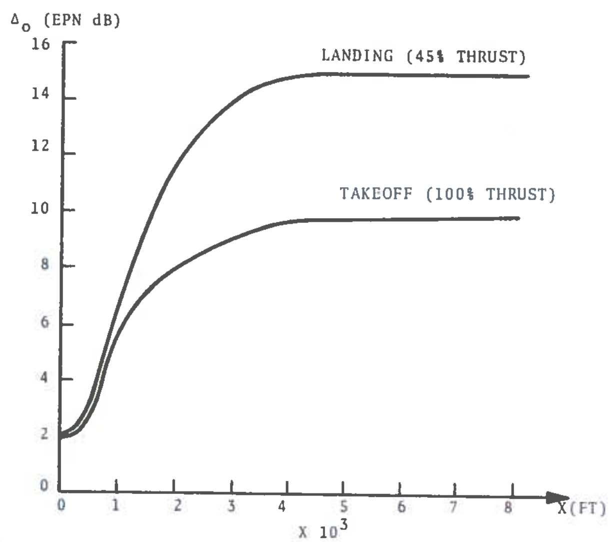


FIGURE 6. EXCESS GROUND ATTENUATION FACTOR (Δ_o) VS HORIZONTAL DISTANCE (X)

PREPARATION OF A NOISE PREDICTION SAMPLE CASE

This manual is intended for general purpose application. Certain parameters in the input can will be set at fixed values to accommodate running the MOD 7 on computer systems other than at DOT/TSC. Parameters of this type will be identified with a † symbol.

Types of Input Data

Two basic sets of input data must be provided to the computer:

(a) Command file

(b) Operation file.

The command file provides computer processing types of instructions, and the operation file translates the airport flight operations into the input data. The files, when completed, are submitted for batch processing. (Processing is discussed in the section, PROGRAMMING.)

Command File - Data Content

The command file contains the data listed in Table 5. These data are entered onto a coding sheet in the manner indicated in Figure 7.

Command File - Instructions for Preparation

Table 6 supplements information given in Table 5, and serves as instructional guide for preparation of the command file. The discussion relates to the same, and correspondingly numbered, items as in Table 5.

Observer Grid Elevation

In the sample case of Figure 7, the upper left grid corner z coordinate (elevation) is -0.1. This value for z is chosen to avoid the possibility of an observer location coinciding with a flight path. If an observer location falls on a flight path, a small discontinuity in noise values may occur. (Refer also to Figure 2 for relative observer and path locations.) This small discontinuity is due to indeterminate states in evaluating shielding and attenuation factors. Both factors depend on β, an observer elevation angle, which becomes indeterminate when the

TABLE 5. DATA CONTAINED IN COMMAND FILE

Unit Record Items	Item Classification	Suggested Value	Format
*1. XREF	For ASDS and L_{dn}	0.0	F10.2
2. NCASES †	≥ 1	1	I2
3. IOPT(1), IOPT(2), NTYP	(These items are defined in Table 6.)	020302	3I2
4. OPFILE	Name of operation file (read from device 21)		A5
5. GFILE	Name of output file (written to device 22)		A5
6. Grid †	GRID/LOCI	G	A1
7. AOFILE(1)	Name of airport geometry file		3A2
8. AOFILE(2)	(Name of) Reserved for future use. Enter DUMMY		3A2
9. NCOLS	Number of grid columns } without extra border**		I4
10. NROWS	Number of grid rows } without extra border**		I4
11. DX,DY	In feet		2F10.2
12. CORN(1),CORN(2),CORN(3)	Upper left corner/coordinates of grid (x,y,z) in feet	x,y,-0.1†	3F10.2

*This item is not in the command file in this updated version, but will appear on the terminal.

**The program will automatically add two columns and two rows (to accommodate the contour plotting routines). These added grid points must not cause the total to exceed 2500.

†See previous page.

```
01



01
2,2,2
TESTA
TESTB
G
AIRPOR
NONAME
0047
0047
     1000.     1000.
    -5000.    30000.       -0.1
```

FIGURE 7. SAMPLE CASE OF COMMAND FILE

TABLE 6. INSTRUCTIONS FOR PREPARATION OF COMMAND FILE

Item	Instruction
1. XREF	This is for ASDS and L_{dn}.
2. NCASES	This represents the number of cases being run in this batch submission. The number must equal that of separate operation files.
3. IOPT(1)	An aircraft shielding term. 1 = no shielding 2 = shielding For default, value 2 is suggested.
IOPT(2)	For excess ground attenuation 1 = no attenuation 2 = 13β 3 = 3β (the quietest) For default, value 2 is suggested.
NTYP	Selection of desired metric. 1 = NE 2 = NEF 3 = ASDS 4 = L_{dn}
4. OPFILE	Name of input (operation) file.
5. GFILE	Name of output file (normally same as OPFILE name)
6. Grid	Enter G for operation file.
7. AOFILE(1)	Name of a file containing runway geometry to be read by plotter routine.
8. AOFILE(2)	Name for output grid file, reserved for future use. (Enter DUMMY)

TABLE 6. INSTRUCTIONS FOR PREPARATION OF COMMAND FILE (CONTINUED)

Item	Instruction
*9. NCOLS	Number of columns of grid points.
10. NROWS	Number of rows for grid points.
11. DX,DY	Spacing of grid points.
12. CORN(1,2,3)	Upper left-hand corner coordinates of the grid matrix. For z coordinate use z = -0.1.

*Items 9-12

a. Items 9-12 include the description of the ground receiver grid points for which the output noise values are calculated. (The running time of the program is directly proportional to the number of grid points.)

b. The grid points are selected to enclose a constant-level noise contour and establish a "thin" density of grid points to determine the closure of the contour. When it has been determined, a readjustment of grid spacing and coverage may be made.

c. Once the first estimated shape and coverage of the contour are established, the upper left hand (x,y,z) will be established, to be followed by the x and y distances, and then by the Δx and Δy spacings, which fix the number of columns and rows.

d. When a change is made, say of DX and DY, the number of columns and the number of rows are to be changed accordingly.

e. It should be noted that a maximum of 2500 grid points may be used for each run. For a larger case a mosiac of two or more runs, each with 2500 grid points, is used.

distance between observer and flight path is zero. In such cases, the program assigns a value to β. This allows computations to proceed, but can affect the final noise computation by as much as 3 dB.

Operation File

Preparation of the operation file is in accordance with the data provided on coding sheets 1 and 2 of Figure 8. These data represent the airport runway(s), flight paths, and flight operational data. The flight paths and segments for a flight model are described geometrically, and also in legend format, in Figure 9.

In the preparation of the operation file, the elements that make up a flight and segments in the order shown on the preprinted coding sheets of Figure 8 are listed below:

1.	Number of flights	:	As defined in CAPABILITIES, paragraph titled Flights.
2.	Takeoff	:	Enter 1.
	Landing	:	Enter 2.
3.	Day operations	:	Total for 24-hour period(0700-2200)
	Evening operations	:	Enter 0 (this will be removed, since it is no longer used)
	Night operations	:	Enter total for 24-hour period (2200-0700)
4.	Segment	:	Enter number and type, where straight line = 1 and helix = 2. Note that the segment number is for editing the input case and is not read into the computer. See Figure 9, and also CAPABILITIES, paragraph titled Flight Paths.
5.	Beginning thrust	:	For this segment
6.	End thrust	:	For this segment. Note that thrust and speed changes will be reflected in the following 1,000 feet.
7.	X,Y,Z Begin	:	Coordinates for beginning of this segment. Note that Z, altitude, must be provided for the beginning of each segment, and that all X,Y,Z entries are in feet.
8.	X,Y End	:	For this segment

9.	Climb angle	:	Enter value.
10.*	Begin and End speeds	:	Enter in knots.
11.	Circulation	:	Use only for helix, where segment type=2, clockwise=-1, and counter-clockwise = +1.
12.	X,Y	:	Coordinates for center of helix

The next identical set is for the next segment of this flight. When all segments are completed for this flight, enter -1 into TYPE.

For each segment the beginning X, Y, Z coordinates must be entered. When a flight path has more than two segments, the user must calculate these coordinates. Also there is redundancy in entering thrusts, speeds etc. for both the ending of one segment and the beginning of the next segment. To save the user the task of calculating as well as entering duplicate data, a versatile input module, IMOD-7, has been written by TSC and is available for usage with MOD-7. Experience has shown that the preparation of the input operation file has been significantly simplified and the input errors have been reduced. A description of this module and a discussion on its use in the preparation of an input case are contained in Appendix B.

---

*NOTE: A speed of zero knots produces infinite noise. Consequently, a small value such as 0.5 may be substituted when zero speed is desired.

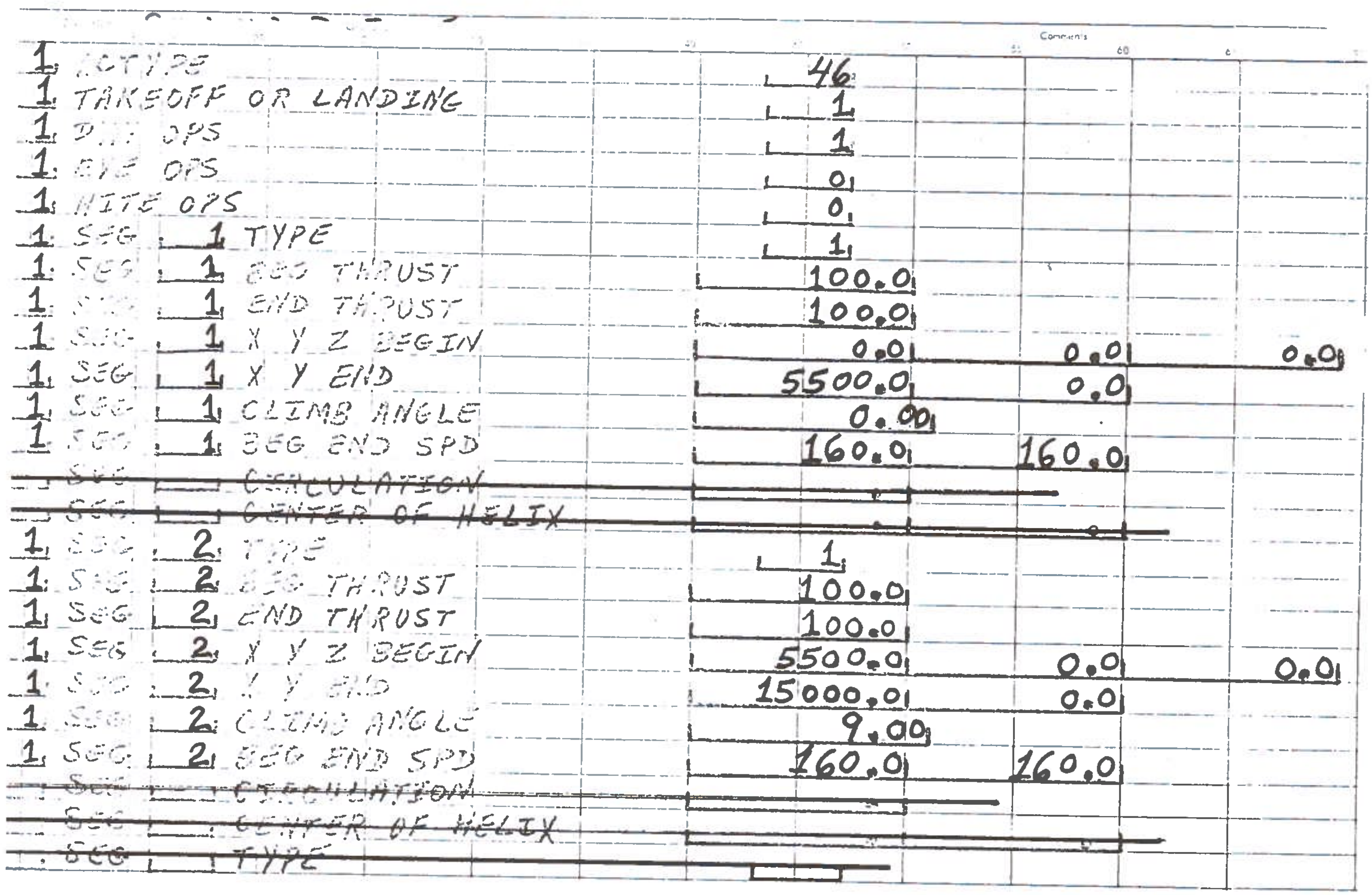
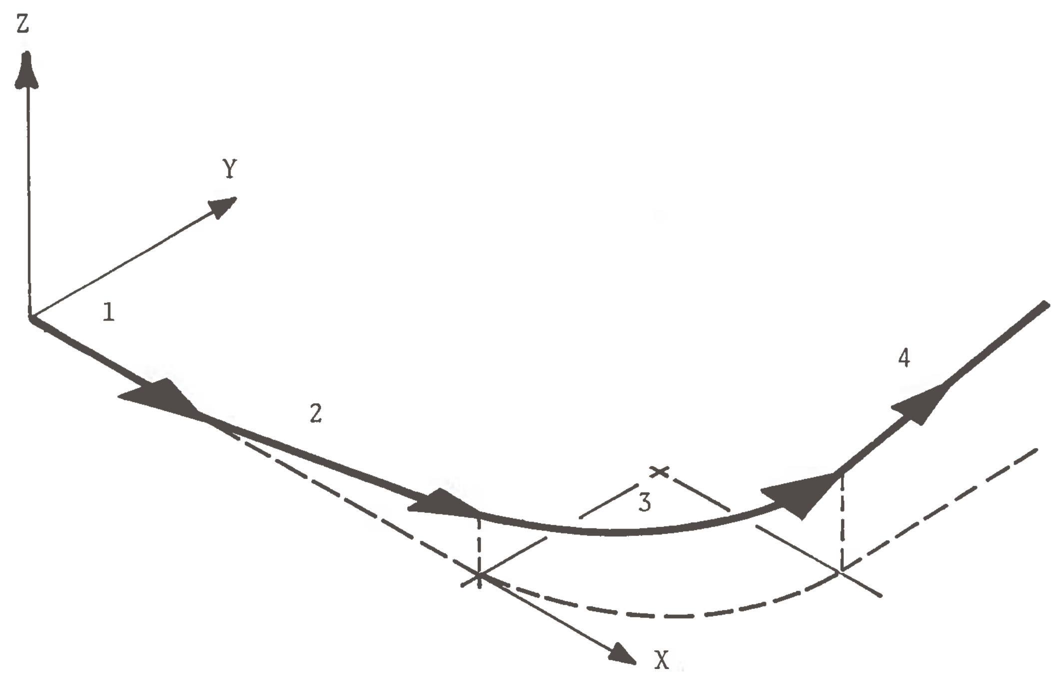


FIGURE 8A. SAMPLE CASE OF OPERATION FILE ON CODING SHEET

1	SEG	3	TYPE	2		
1	SEG	3	BEG THRUST	100.0		
1	SEG	3	END THRUST	100.0		
1	SEG	3	X Y Z BEGIN	15000.0	0.0	1500.0
1	SEG	3	X Y END	21000.0	6000.0	
1	SEG	3	CLIMB ANGLE	9.00		
1	SEG	3	BEG END SPD	160.0	160.0	
1	SEG	3	CIRCULATION	+1.0		
1	SEG	3	CENTER OF HELIX	15000.0	6000.0	
1	SEG	4	TYPE	1		
1	SEG	4	BEG THRUST	100.0		
1	SEG	4	END THRUST	100.0		
1	SEG	4	X Y Z BEGIN	21000.0	6000.0	2998.0
1	SEG	4	X Y END	21000.0	90000.0	
1	SEG	4	CLIMB ANGLE	9.00		
1	SEG	4	BEG END SPD	160.0	160.0	
	~~SEG~~		~~CIRCULATION~~			
	~~SEG~~		~~CENTER OF HELIX~~			
1	SEG	5	TYPE	-1		

FIGURE 8B. SAMPLE CASE OF OPERATION FILE ON CODING SHEET



1. TAKEOFF OR LANDING ROLL
2. TAKEOFF - INITIAL CLIMB
   LANDING - FINAL DESCENT
3. TAKEOFF - COUNTERCLOCKWISE TURN & CLIMB
   LANDING - COUNTERCLOCKWISE TURN & DESCENT
4. TAKEOFF - CLIMB
   LANDING - DESCENT

FIGURE 9. MULTIPLE SEGMENT FLIGHT PATH

SAMPLE CASE

This section demonstrates the presentation of a sample case. The data provided may be used to assist in the preparation of necessary files, and also in the validation of output grid calculations. The sample case data correspond to data already shown in previous figures and tables, and will be so noted in the following exposition.

Statement of the Sample Case

A 747-200 aircraft will take off on the X-axis runway from an origin 0,0,0. Ground roll is 5500 feet. A constant climb angle of 9 degrees will be maintained. At 15,000 feet ground track, a counterclockwise helical turn of 90° will start. The last segment is a straight line to 90,000 feet more (ground track).

Preparation of Command File for TSC PRIME 300 System

The following is the sample case command file.

Coding	Description
1	One case
020202	Shielding, 13 BETA Attenuation, NEF
TESTA	Operation File Name
TESTB	Output File Name (grid file)
G	Grid/Loci Flag
AIRPORT	Airport File Name
NONAME	Operation File Name
47	Rows of grid points
47	Columns of grid points
1000, 1000	DX, DY spacings of grids
-5000, 30,000, -0.1	X, Y, Z top left hand corner grid point

Preparation of Operation File

The operation file is prepared from the input data on the coding sheets of Figure 8. The following details are entered on the basis of these data.

```
ACTYPE            =  46 (from Table 2)
TAKEOFF           =   1
DAY OPS           =   1
EVE OPS           =   0
NITE OPS          =   0
SEG 1             =   1 (for straight line)
BEG THRUST        = 100.
END THRUST        = 100.
X, Y, Z BEGIN     = 0,0,0, (any values may be used, but these
                            simplify the calculations required)
X,Y END           = 0,5500. which is the ground roll
CLIMB ANGLE       = 0.
BEG, END SPEED    = 160. 160.
CIRCULATION       - omitted because SEG TYPE = 1
CENTER OF HELIX   - omitted because SEG TYPE = 1
```

This completes segment 1.

Segment 2 is entered in a similar fashion. X, Y, Z points are calculated by trigonometry by user.

Segment 3 is the 90° helical turn. The X, Y, Z points must be calculated. The circulation = +1 because it is counterclockwise.

Segment 4 is a straight line out to any selected value that will exceed the expected grid point coverage.

The operation file program is indicated in Figure 10.

With respect to the steps beginning with the X, Y, Z BEGIN statement contained in the data for segment 1, care must be taken in entering landings. Habit suggests entering the flight path segments in the order that they are traversed by the aircraft (Figure 11). This certainly may be done, but there are two disadvantages:

(a) A distant start point's z coordinate must be calculated.

(b) The descent angle must be entered as a negative climb angle (e.g., -3°). The omission of the minus sign is a common user error.

An easier way to structure the same case is to pretend the aircraft is "flying backwards" (Figure 12). That way the last z need not be calculated, and the climb angle can be entered as a positive value (e.g., 3°). Since speed or thrust changes are automatically averaged out over the first 1000 feet of a flight

```
NUMBER OF/FLITES                                     1


NUMBER OF/FLITES                                     1
FLITE      1 ACTYPE                                 46
FLITE      1 TAKEOFF OR LANDING                      1
FLITE      1 DAY OPS                                 1
FLITE      1 EVE OPS                                 0
FLITE      1 NITE OPS                                0
FLITE      1 SEG    1 TYPE                           1
FLITE      1 SEG    1 BEG THRUST                100.00
FLITE      1 SEG    1 END THRUST                100.00
FLITE      1 SEG    1 X Y Z BEGIN                 0.00        0.00        0.00
FLITE      1 SEG    1 X Y Z END                5500.00        0.00        0.00
FLITE      1 SEG    1 CLIMB ANGLE                 0.00
FLITE      1 SEG    1 BEG END SPD               160.00      160.00
FLITE      1 SEG    2 TYPE                           1
FLITE      1 SEG    2 BEG THRUST                100.00
FLITE      1 SEG    2 END THRUST                100.00
FLITE      1 SEG    2 X Y Z BEGIN              5500.00        0.00        0.00
FLITE      1 SEG    2 X Y Z END               15000.00        0.00     1504.65
FLITE      1 SEG    2 CLIMB ANGLE                 9.00
FLITE      1 SEG    2 BEG END SPD               160.00      160.00
FLITE      1 SEG    3 TYPE                           2
FLITE      1 SEG    3 BEG THRUST                100.00
FLITE      1 SEG    3 END THRUST                100.00
FLITE      1 SEG    3 X Y Z BEGIN             15000.00        0.00     1504.65
FLITE      1 SEG    3 X Y Z END               21000.00     5999.99     2997.99
FLITE      1 SEG    3 CLIMB ANGLE                 9.00
FLITE      1 SEG    3 BEG END SPD               160.00      160.00
FLITE      1 SEG    3 CIRCULATION                   1.
FLITE      1 SEG    3 CENTER OF HELIX         15000.00     6000.00        0.00
FLITE      1 SEG    4 TYPE                           1
FLITE      1 SEG    4 BEG THRUST                100.00
FLITE      1 SEG    4 END THRUST                100.00
FLITE      1 SEG    4 X Y Z BEGIN             21000.00     5999.99     2997.99
FLITE      1 SEG    4 X Y Z END               21000.17    89999.97    16302.26
FLITE      1 SEG    4 CLIMB ANGLE                 9.00
FLITE      1 SEG    4 BEG END SPD               160.00      160.00
FLITE      1 SEG    5 TYPE                          -1
```

FIGURE 10. OPERATION FILE PROGRAM

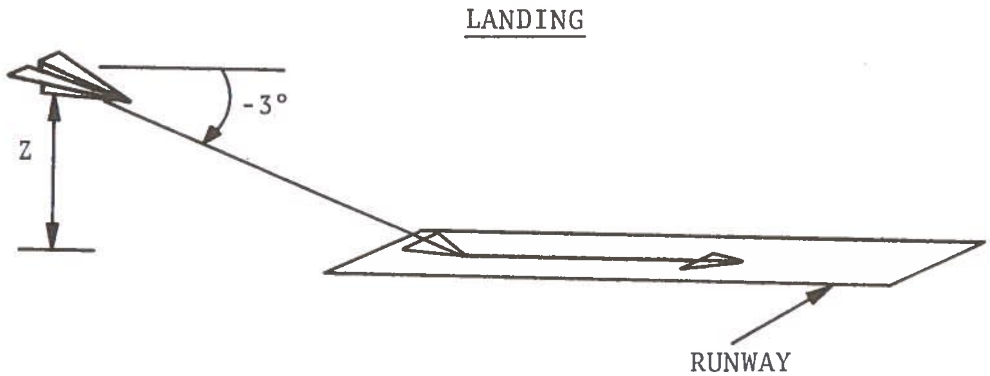


FIGURE 11. NORMAL ROUTINE OF ENTERING FLIGHT PATH SEGMENTS IN LANDING

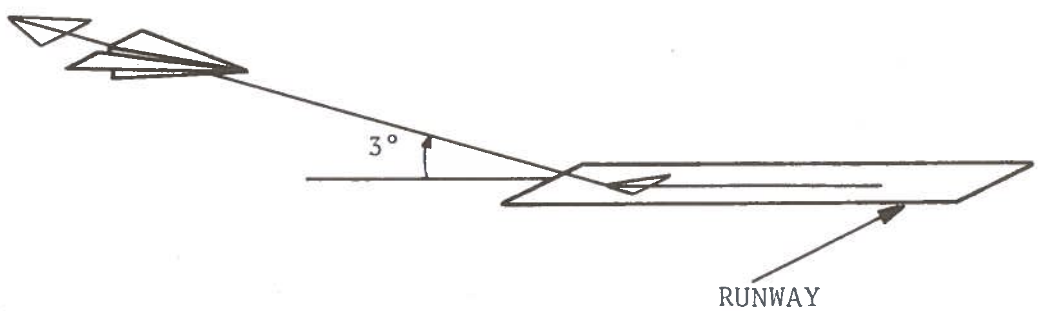


FIGURE 12. CONCEPT OF PLANE LANDING AS "BACKWARD TAKEOFF"

path segment, it is possible that "flying backwards" will affect these changes at the wrong end of the segment. If this is unacceptable, then plane direction must be reversed, or an artificial extra segment inserted to accommodate the transition.

Output Grid Calculations

The sample case output calculations of NEF levels for each observer point are indicated in the sheets of Figure 13. The order of printing is left to right and top to bottom, starting at the upper left-hand point of the specified grid matix. To draw smooth contours at the grid's edge, the program arguments the user specified grid by a one-row one-colume border around the entire grid. For example, if a user specifies a 47 x 47 grid the computer produces a 49 x 49 grid. The original grid is interior to the argumented grid. Element (2,2) of the argumented grid corresponds to element (1,1) of the original grid. Thus, the upper left corner coordinates specified in the input case belong to point (2,2) of the output grid file.

```
AIRPOR



AIRPOR
NONAME
 2
  49
  49
   1000.00    1000.00
  -5000.00   30000.00      -0.10
  0.110E-02 0.111E-02 0.111E-02 0.112E-02 0.808E-04 0.101E-03 0.126E-03
  0.157E-03 0.198E-03 0.251E-03 0.320E-03 0.409E-03 0.526E-03 0.680E-03
  0.884E-03 0.116E-02 0.152E-02 0.202E-02 0.270E-02 0.362E-02 0.490E-02
  0.665E-02 0.909E-02 0.124E-01 0.170E-01 0.232E-01 0.313E-01 0.414E-01
  0.529E-01 0.642E-01 0.728E-01 0.761E-01 0.728E-01 0.642E-01 0.529E-01
  0.414E-01 0.313E-01 0.232E-01 0.170E-01 0.124E-01 0.909E-02 0.665E-02
  0.490E-02 0.362E-02 0.270E-02 0.202E-02 0.152E-02 0.116E-02 0.884E-03
  0.137E-02 0.138E-02 0.139E-02 0.139E-02 0.140E-02 0.105E-03 0.132E-03
  0.166E-03 0.210E-03 0.267E-03 0.341E-03 0.438E-03 0.566E-03 0.737E-03
  0.965E-03 0.127E-02 0.169E-02 0.227E-02 0.306E-02 0.417E-02 0.573E-02
  0.793E-02 0.111E-01 0.156E-01 0.221E-01 0.313E-01 0.443E-01 0.617E-01
  0.833E-01 0.107E 00 0.126E 00 0.133E 00 0.126E 00 0.107E 00 0.833E-01
  0.617E-01 0.443E-01 0.313E-01 0.221E-01 0.156E-01 0.111E-01 0.793E-02
  0.573E-02 0.417E-02 0.306E-02 0.227E-02 0.169E-02 0.127E-02 0.965E-03
  0.172E-02 0.173E-02 0.174E-02 0.174E-02 0.175E-02 0.176E-02 0.137E-03
  0.173E-03 0.220E-03 0.280E-03 0.360E-03 0.464E-03 0.602E-03 0.788E-03
  0.104E-02 0.138E-02 0.185E-02 0.250E-02 0.341E-02 0.470E-02 0.655E-02
  0.925E-02 0.132E-01 0.191E-01 0.280E-01 0.414E-01 0.617E-01 0.916E-01
  0.133E 00 0.185E 00 0.232E 00 0.253E 00 0.232E 00 0.185E 00 0.133E 00
  0.916E-01 0.617E-01 0.414E-01 0.280E-01 0.191E-01 0.132E-01 0.925E-02
  0.655E-02 0.470E-02 0.341E-02 0.250E-02 0.105E-02 0.138E-02 0.104E-02
  0.216E-02 0.217E-02 0.218E-02 0.220E-02 0.221E-02 0.222E-02 0.223E-02
  0.179E-03 0.228E-03 0.292E-03 0.375E-03 0.485E-03 0.633E-03 0.831E-03
  0.110E-02 0.147E-02 0.198E-02 0.270E-02 0.371E-02 0.518E-02 0.731E-02
  0.105E-01 0.153E-01 0.227E-01 0.343E-01 0.529E-01 0.833E-01 0.133E 00
  0.214E 00 0.335E 00 0.473E 00 0.541E 00 0.473E 00 0.335E 00 0.214E 00
  0.133E 00 0.833E-01 0.529E-01 0.343E-01 0.227E-01 0.153E-01 0.105E-01
  0.731E-02 0.518E-02 0.371E-02 0.270E-02 0.198E-02 0.147E-02 0.110E-02
  0.272E-02 0.274E-02 0.276E-02 0.278E-02 0.279E-02 0.281E-02 0.282E-02
  0.284E-02 0.234E-03 0.300E-03 0.387E-03 0.502E-03 0.656E-03 0.864E-03
  0.115E-02 0.154E-02 0.208E-02 0.285E-02 0.396E-02 0.556E-02 0.793E-02
  0.115E-01 0.170E-01 0.258E-01 0.401E-01 0.642E-01 0.107E 00 0.185E 00
  0.335E 00 0.629E 00 0.113E 01 0.149E 01 0.113E 01 0.629E 00 0.335E 00
  0.185E 00 0.107E 00 0.642E-01 0.401E-01 0.250E-01 0.170E-01 0.115E-01
  0.793E-02 0.556E-02 0.396E-02 0.285E-02 0.208E-02 0.154E-02 0.115E-02
  0.346E-02 0.348E-02 0.351E-02 0.353E-02 0.355E-02 0.357E-02 0.360E-02
  0.361E-02 0.363E-02 0.305E-03 0.394E-03 0.512E-03 0.670E-03 0.884E-03
  0.118E-02 0.158E-02 0.215E-02 0.295E-02 0.411E-02 0.581E-02 0.834E-02
  0.122E-01 0.182E-01 0.280E-01 0.443E-01 0.728E-01 0.126E 00 0.232E 00
  0.473E 00 0.113E 01 0.360E 01 0.919E 01 0.369E 01 0.113E 01 0.473E 00
  0.232E 00 0.126E 00 0.728E-01 0.443E-01 0.280E-01 0.182E-01 0.122E-01
```

FIGURE 13A. OUTPUT DATA GRID MATRIX

AIRPOR

0.834E-02	0.581E-02	0.411E-02	0.295E-02	0.215E-02	0.158E-02	0.118E-02
0.440E-02	0.444E-02	0.447E-02	0.451E-02	0.454E-02	0.457E-02	0.460E-02
0.463E-02	0.466E-02	0.468E-02	0.396E-03	0.515E-03	0.675E-03	0.891E-03
0.119E-02	0.160E-02	0.217E-02	0.299E-02	0.417E-02	0.590E-02	0.848E-02
0.124E-01	0.187E-01	0.288E-01	0.458E-01	0.761E-01	0.133E 00	0.253E 00
0.541E 00	0.149E 01	0.919E 01	0.337E 04	0.919E 01	0.149E 01	0.541E 00
0.253E 00	0.133E 00	0.761E-01	0.458E-01	0.288E-01	0.187E-01	0.124E-01
0.848E-02	0.590E-02	0.417E-02	0.299E-02	0.217E-02	0.160E-02	0.119E-02
0.564E-02	0.569E-02	0.574E-02	0.579E-02	0.584E-02	0.588E-02	0.592E-02
0.596E-02	0.600E-02	0.604E-02	0.607E-02	0.515E-03	0.675E-03	0.891E-03
0.119E-02	0.160E-02	0.217E-02	0.299E-02	0.417E-02	0.590E-02	0.840E-02
0.124E-01	0.187E-01	0.288E-01	0.458E-01	0.761E-01	0.133E 00	0.253E 00
0.541E 00	0.149E 01	0.919E 01	0.337E 04	0.919E 01	0.149E 01	0.541E 00
0.253E 00	0.133E 00	0.761E-01	0.458E-01	0.288E-01	0.187E-01	0.124E-01
0.848E-02	0.590E-02	0.417E-02	0.299E-02	0.217E-02	0.160E-02	0.119E-02
0.727E-02	0.734E-02	0.741E-02	0.748E-02	0.754E-02	0.761E-02	0.767E-02
0.773E-02	0.778E-02	0.784E-02	0.789E-02	0.794E-02	0.675E-03	0.891E-03
0.119E-02	0.160E-02	0.217E-02	0.299E-02	0.417E-02	0.590E-02	0.848E-02
0.124E-01	0.187E-01	0.288E-01	0.458E-01	0.761E-01	0.133E 00	0.253E 00
0.541E 00	0.149E 01	0.919E 01	0.337E 04	0.919E 01	0.149E 01	0.541E 00
0.253E 00	0.133E 00	0.761E-01	0.458E-01	0.288E-01	0.187E-01	0.124E-01
0.848E-02	0.590E-02	0.417E-02	0.299E-02	0.217E-02	0.160E-02	0.119E-02
0.941E-02	0.952E-02	0.962E-02	0.971E-02	0.981E-02	0.990E-02	0.999E-02
0.101E-01	0.102E-01	0.102E-01	0.103E-01	0.104E-01	0.104E-01	0.891E-03
0.119E-02	0.160E-02	0.217E-02	0.299E-02	0.417E-02	0.590E-02	0.840E-02
0.124E-01	0.187E-01	0.288E-01	0.458E-01	0.761E-01	0.133E 00	0.253E 00
0.541E 00	0.149E 01	0.919E 01	0.337E 04	0.919E 01	0.149E 01	0.541E 00
0.253E 00	0.133E 00	0.761E-01	0.458E-01	0.288E-01	0.187E-01	0.124E-01
0.848E-02	0.590E-02	0.417E-02	0.299E-02	0.217E-02	0.160E-02	0.119E-02
0.123E-01	0.124E-01	0.126E-01	0.127E-01	0.128E-01	0.130E-01	0.131E-01
0.132E-01	0.133E-01	0.135E-01	0.136E-01	0.137E-01	0.130E-01	0.139E-01
0.119E-02	0.160E-02	0.217E-02	0.299E-02	0.417E-02	0.590E-02	0.848E-02
0.124E-01	0.187E-01	0.288E-01	0.458E-01	0.761E-01	0.133E 00	0.253E 00
0.541E 00	0.149E 01	0.919E 01	0.337E 04	0.919E 01	0.149E 01	0.541E 00
0.253E 00	0.133E 00	0.761E-01	0.458E-01	0.288E-01	0.187E-01	0.124E-01
0.848E-02	0.590E-02	0.417E-02	0.299E-02	0.217E-02	0.160E-02	0.119E-02
0.161E-01	0.163E-01	0.165E-01	0.167E-01	0.169E-01	0.171E-01	0.173E-01
0.175E-01	0.177E-01	0.170E-01	0.180E-01	0.182E-01	0.183E-01	0.185E-01
0.186E-01	0.160E-02	0.217E-02	0.299E-02	0.417E-02	0.590E-02	0.848E-02
0.124E-01	0.187E-01	0.288E-01	0.458E-01	0.761E-01	0.133E 00	0.253E 00
0.541E 00	0.149E 01	0.919E 01	0.337E 04	0.919E 01	0.149E 01	0.541E 00
0.253E 00	0.133E 00	0.761E-01	0.458E-01	0.288E-01	0.187E-01	0.124E-01
0.848E-02	0.590E-02	0.417E-02	0.299E-02	0.217E-02	0.160E-02	0.119E-02
0.212E-01	0.215E-01	0.218E-01	0.221E-01	0.224E-01	0.227E-01	0.230E-01
0.233E-01	0.235E-01	0.238E-01	0.240E-01	0.243E-01	0.245E-01	0.247E-01
0.250E-01	0.252E-01	0.378E-02	0.528E-02	0.750E-02	0.108E-01	0.159E-01
0.239E-01	0.370E-01	0.588E-01	0.973E-01	0.169E 00	0.313E 00	0.635E 00
0.149E 01	0.454E 01	0.285E 02	0.335E 04	0.285E 02	0.454E 01	0.149E 01

FIGURE 13B. OUTPUT DATA GRID MATRIX

AIRPOR

0.635E 00	0.313E 00	0.169E 00	0.973E-01	0.588E-01	0.370E-01	0.239E-01
0.159E-01	0.108E-01	0.750E-02	0.528E-02	0.378E-02	0.274E-02	0.201E-02
0.280E-01	0.285E-01	0.290E-01	0.294E-01	0.299E-01	0.303E-01	0.308E-01
0.312E-01	0.316E-01	0.320E-01	0.324E-01	0.328E-01	0.331E-01	0.335E-01
0.338E-01	0.341E-01	0.344E-01	0.754E-02	0.108E-01	0.157E-01	0.234E-01
0.357E-01	0.558E-01	0.903E-01	0.152E 00	0.271E 00	0.518E 00	0.110E 01
0.283E 01	0.121E 02	0.468E 02	0.707E 03	0.468E 02	0.121E 02	0.283E 01
0.110E 01	0.518E 00	0.271E 00	0.152E 00	0.903E-01	0.558E-01	0.357E-01
0.234E-01	0.157E-01	0.108E-01	0.754E-02	0.534E-02	0.384E-02	0.280E-02
0.374E-01	0.381E-01	0.388E-01	0.395E-01	0.401E-01	0.408E-01	0.415E-01
0.421E-01	0.428E-01	0.434E-01	0.440E-01	0.446E-01	0.452E-01	0.457E-01
0.462E-01	0.467E-01	0.472E-01	0.477E-01	0.135E-01	0.198E-01	0.297E-01
0.457E-01	0.723E-01	0.119E 00	0.204E 00	0.374E 00	0.751E 00	0.180E 01
0.542E 01	0.133E 02	0.474E 02	0.309E 03	0.474E 02	0.133E 02	0.542E 01
0.180E 01	0.751E 00	0.374E 00	0.204E 00	0.119E 00	0.723E-01	0.457E-01
0.297E-01	0.198E-01	0.135E-01	0.939E-02	0.662E-02	0.474E-02	0.343E-02
0.501E-01	0.512E-01	0.522E-01	0.533E-01	0.543E-01	0.553E-01	0.564E-01
0.574E-01	0.583E-01	0.593E-01	0.603E-01	0.612E-01	0.621E-01	0.630E-01
0.639E-01	0.647E-01	0.655E-01	0.662E-01	0.669E-01	0.238E-01	0.360E-01
0.558E-01	0.896E-01	0.150E 00	0.266E 00	0.517E 00	0.119E 01	0.271E 01
0.557E 01	0.135E 02	0.456E 02	0.183E 03	0.456E 02	0.135E 02	0.557E 01
0.271E 01	0.119E 01	0.517E 00	0.266E 00	0.150E 00	0.896E-01	0.558E-01
0.360E-01	0.238E-01	0.161E-01	0.111E-01	0.781E-02	0.557E-02	0.402E-02
0.675E-01	0.691E-01	0.707E-01	0.723E-01	0.739E-01	0.755E-01	0.771E-01
0.787E-01	0.803E-01	0.818E-01	0.833E-01	0.848E-01	0.863E-01	0.877E-01
0.891E-01	0.905E-01	0.918E-01	0.930E-01	0.942E-01	0.953E-01	0.427E-01
0.673E-01	0.110E 00	0.191E 00	0.363E 00	0.811E 00	0.148E 01	0.275E 01
0.564E 01	0.135E 02	0.423E 02	0.123E 03	0.423E 02	0.135E 02	0.564E 01
0.275E 01	0.148E 01	0.811E 00	0.363E 00	0.191E 00	0.110E 00	0.673E-01
0.427E-01	0.280E-01	0.188E-01	0.129E-01	0.900E-02	0.638E-02	0.459E-02
0.913E-01	0.938E-01	0.963E-01	0.988E-01	0.101E 00	0.104E 00	0.106E 00
0.109E 00	0.111E 00	0.114E 00	0.116E 00	0.119E 00	0.121E 00	0.123E 00
0.126E 00	0.128E 00	0.130E 00	0.132E 00	0.134E 00	0.136E 00	0.138E 00
0.818E-01	0.140E 00	0.260E 00	0.525E 00	0.867E 00	0.150E 01	0.278E 01
0.566E 01	0.132E 02	0.303E 02	0.894E 02	0.303E 02	0.132E 02	0.566E 01
0.278E 01	0.150E 01	0.867E 00	0.525E 00	0.260E 00	0.140E 00	0.818E-01
0.507E-01	0.327E-01	0.217E-01	0.148E-01	0.102E-01	0.722E-02	0.517E-02
0.124E 00	0.128E 00	0.131E 00	0.135E 00	0.139E 00	0.143E 00	0.147E 00
0.151E 00	0.155E 00	0.159E 00	0.163E 00	0.167E 00	0.171E 00	0.175E 00
0.179E 00	0.183E 00	0.187E 00	0.190E 00	0.194E 00	0.197E 00	0.201E 00
0.204E 00	0.189E 00	0.334E 00	0.531E 00	0.875E 00	0.151E 01	0.279E 01
0.562E 01	0.128E 02	0.343E 02	0.681E 02	0.343E 02	0.128E 02	0.562E 01
0.279E 01	0.151E 01	0.875E 00	0.531E 00	0.334E 00	0.189E 00	0.103E 00
0.612E-01	0.385E-01	0.251E-01	0.169E-01	0.116E-01	0.813E-02	0.579E-02
0.168E 00	0.174E 00	0.180E 00	0.186E 00	0.192E 00	0.199E 00	0.205E 00
0.212E 00	0.218E 00	0.225E 00	0.231E 00	0.238E 00	0.245E 00	0.251E 00
0.258E 00	0.265E 00	0.271E 00	0.278E 00	0.284E 00	0.290E 00	0.296E 00
0.302E 00	0.307E 00	0.337E 00	0.535E 00	0.880E 00	0.152E 01	0.278E 01

FIGURE 13C. OUTPUT DATA GRID MATRIX

AIRPOR

0.554E 01	0.123E 02	0.304E 02	0.536E 02	0.304E 02	0.123E 02	0.554E 01
0.278E 01	0.152E 01	0.880E 00	0.535E 00	0.337E 00	0.219E 00	0.139E 00
0.771E-01	0.463E-01	0.294E-01	0.194E-01	0.132E-01	0.914E-02	0.646E-02
0.227E 00	0.236E 00	0.246E 00	0.255E 00	0.265E 00	0.275E 00	0.286E 00
0.296E 00	0.307E 00	0.318E 00	0.329E 00	0.341E 00	0.352E 00	0.364E 00
0.375E 00	0.387E 00	0.398E 00	0.410E 00	0.421E 00	0.433E 00	0.444E 00
0.454E 00	0.465E 00	0.475E 00	0.537E 00	0.882E 00	0.152E 01	0.276E 01
0.543E 01	0.117E 02	0.268E 02	0.432E 02	0.268E 02	0.117E 02	0.543E 01
0.276E 01	0.152E 01	0.882E 00	0.537E 00	0.339E 00	0.221E 00	0.147E 00
0.100E 00	0.583E-01	0.354E-01	0.227E-01	0.151E-01	0.103E-01	0.723E-02
0.305E 00	0.319E 00	0.334E 00	0.349E 00	0.364E 00	0.380E 00	0.397E 00
0.414E 00	0.432E 00	0.450E 00	0.469E 00	0.488E 00	0.508E 00	0.528E 00
0.548E 00	0.569E 00	0.590E 00	0.611E 00	0.632E 00	0.653E 00	0.675E 00
0.695E 00	0.716E 00	0.736E 00	0.755E 00	0.882E 00	0.151E 01	0.273E 01
0.528E 01	0.110E 02	0.236E 02	0.354E 02	0.236E 02	0.110E 02	0.528E 01
0.273E 01	0.151E 01	0.882E 00	0.538E 00	0.341E 00	0.222E 00	0.148E 00
0.101E 00	0.703E-01	0.444E-01	0.273E-01	0.176E-01	0.118E-01	0.815E-02
0.403E 00	0.425E 00	0.447E 00	0.470E 00	0.494E 00	0.519E 00	0.546E 00
0.574E 00	0.603E 00	0.633E 00	0.664E 00	0.697E 00	0.731E 00	0.766E 00
0.802E 00	0.839E 00	0.877E 00	0.916E 00	0.956E 00	0.996E 00	0.104E 01
0.108E 01	0.112E 01	0.116E 01	0.120E 01	0.127E 01	0.178E 01	0.298E 01
0.551E 01	0.110E 02	0.215E 02	0.294E 02	0.200E 02	0.100E 02	0.490E 01
0.253E 01	0.143E 01	0.847E 00	0.522E 00	0.332E 00	0.217E 00	0.146E 00
0.997E-01	0.695E-01	0.462E-01	0.282E-01	0.181E-01	0.121E-01	0.831E-02
0.521E 00	0.552E 00	0.584E 00	0.619E 00	0.656E 00	0.694E 00	0.736E 00
0.779E 00	0.825E 00	0.874E 00	0.925E 00	0.979E 00	0.104E 01	0.110E 01
0.116E 01	0.123E 01	0.129E 01	0.137E 01	0.144E 01	0.152E 01	0.160E 01
0.168E 01	0.177E 01	0.185E 01	0.194E 01	0.206E 01	0.256E 01	0.396E 01
0.665E 01	0.122E 02	0.207E 02	0.242E 02	0.158E 02	0.836E 01	0.433E 01
0.237E 01	0.134E 01	0.790E 00	0.489E 00	0.306E 00	0.203E 00	0.138E 00
0.948E-01	0.665E-01	0.436E-01	0.268E-01	0.173E-01	0.117E-01	0.805E-02
0.649E 00	0.692E 00	0.738E 00	0.787E 00	0.840E 00	0.896E 00	0.957E 00
0.102E 01	0.109E 01	0.117E 01	0.125E 01	0.133E 01	0.142E 01	0.152E 01
0.163E 01	0.174E 01	0.186E 01	0.199E 01	0.213E 01	0.228E 01	0.243E 01
0.259E 01	0.277E 01	0.295E 01	0.314E 01	0.337E 01	0.411E 01	0.579E 01
0.907E 01	0.145E 02	0.199E 02	0.189E 02	0.120E 02	0.656E 01	0.356E 01
0.201E 01	0.120E 01	0.729E 00	0.459E 00	0.297E 00	0.197E 00	0.133E 00
0.917E-01	0.643E-01	0.458E-01	0.330E-01	0.221E-01	0.110E-01	0.764E-02
0.771E 00	0.826E 00	0.886E 00	0.951E 00	0.102E 01	0.110E 01	0.118E 01
0.127E 01	0.137E 01	0.147E 01	0.159E 01	0.172E 01	0.185E 01	0.200E 01
0.217E 01	0.234E 01	0.254E 01	0.275E 01	0.298E 01	0.324E 01	0.352E 01
0.382E 01	0.415E 01	0.451E 01	0.491E 01	0.537E 01	0.633E 01	0.851E 01
0.118E 02	0.158E 02	0.173E 02	0.136E 02	0.847E 01	0.475E 01	0.284E 01
0.162E 01	0.990E 00	0.627E 00	0.406E 00	0.268E 00	0.180E 00	0.123E 00
0.860E-01	0.608E-01	0.435E-01	0.316E-01	0.208E-01	0.135E-01	0.906E-02
0.860E 00	0.925E 00	0.997E 00	0.107E 01	0.116E 01	0.125E 01	0.135E 01
0.146E 01	0.159E 01	0.172E 01	0.187E 01	0.203E 01	0.221E 01	0.241E 01
0.263E 01	0.287E 01	0.314E 01	0.345E 01	0.378E 01	0.416E 01	0.450E 01

FIGURE 13D. OUTPUT DATA GRID MATRIX

AIRPOR

0.506E 01	0.560E 01	0.620E 01	0.689E 01	0.756E 01	0.902E 01	0.108E 02
0.133E 02	0.140E 02	0.127E 02	0.915E 01	0.576E 01	0.350E 01	0.215E 01
0.132E 01	0.818E 00	0.510E 00	0.342E 00	0.232E 00	0.159E 00	0.111E 00
0.783E-01	0.559E-01	0.404E-01	0.294E-01	0.190E-01	0.125E-01	0.848E-02
0.893E 00	0.962E 00	0.104E 01	0.112E 01	0.121E 01	0.131E 01	0.142E 01
0.154E 01	0.167E 01	0.182E 01	0.198E 01	0.215E 01	0.235E 01	0.257E 01
0.282E 01	0.309E 01	0.340E 01	0.374E 01	0.413E 01	0.456E 01	0.506E 01
0.562E 01	0.627E 01	0.701E 01	0.786E 01	0.861E 01	0.997E 01	0.107E 02
0.114E 02	0.103E 02	0.824E 01	0.590E 01	0.385E 01	0.248E 01	0.156E 01
0.102E 01	0.668E 00	0.440E 00	0.294E 00	0.199E 00	0.136E 00	0.968E-01
0.694E-01	0.503E-01	0.367E-01	0.262E-01	0.170E-01	0.113E-01	0.779E-02
0.860E 00	0.925E 00	0.997E 00	0.107E 01	0.116E 01	0.125E 01	0.135E 01
0.146E 01	0.159E 01	0.172E 01	0.187E 01	0.203E 01	0.221E 01	0.241E 01
0.263E 01	0.287E 01	0.314E 01	0.345E 01	0.378E 01	0.416E 01	0.458E 01
0.506E 01	0.560E 01	0.620E 01	0.689E 01	0.758E 01	0.820E 01	0.827E 01
0.778E 01	0.667E 01	0.504E 01	0.373E 01	0.253E 01	0.173E 01	0.112E 01
0.754E 00	0.519E 00	0.356E 00	0.246E 00	0.171E 00	0.120E 00	0.853E-01
0.602E-01	0.442E-01	0.327E-01	0.225E-01	0.148E-01	0.101E-01	0.702E-02
0.771E 00	0.826E 00	0.886E 00	0.951E 00	0.102E 01	0.110E 01	0.118E 01
0.127E 01	0.137E 01	0.147E 01	0.159E 01	0.172E 01	0.185E 01	0.200E 01
0.217E 01	0.234E 01	0.254E 01	0.275E 01	0.298E 01	0.324E 01	0.352E 01
0.382E 01	0.415E 01	0.451E 01	0.491E 01	0.531E 01	0.537E 01	0.534E 01
0.472E 01	0.404E 01	0.312E 01	0.236E 01	0.165E 01	0.120E 01	0.831E 00
0.569E 00	0.390E 00	0.279E 00	0.199E 00	0.142E 00	0.102E 00	0.740E-01
0.539E-01	0.396E-01	0.293E-01	0.188E-01	0.127E-01	0.879E-02	0.623E-02
0.649E 00	0.692E 00	0.738E 00	0.707E 00	0.840E 00	0.896E 00	0.957E 00
0.102E 01	0.109E 01	0.117E 01	0.125E 01	0.133E 01	0.142E 01	0.152E 01
0.163E 01	0.174E 01	0.186E 01	0.199E 01	0.213E 01	0.228E 01	0.243E 01
0.259E 01	0.277E 01	0.295E 01	0.314E 01	0.331E 01	0.337E 01	0.316E 01
0.287E 01	0.241E 01	0.193E 01	0.151E 01	0.109E 01	0.811E 00	0.597E 00
0.430E 00	0.306E 00	0.214E 00	0.158E 00	0.116E 00	0.852E-01	0.629E-01
0.466E-01	0.347E-01	0.260E-01	0.196E-01	0.144E-01	0.977E-02	0.546E-02
0.521E 00	0.552E 00	0.584E 00	0.619E 00	0.656E 00	0.694E 00	0.736E 00
0.779E 00	0.825E 00	0.874E 00	0.925E 00	0.979E 00	0.104E 01	0.110E 01
0.116E 01	0.123E 01	0.129E 01	0.137E 01	0.144E 01	0.152E 01	0.160E 01
0.168E 01	0.177E 01	0.185E 01	0.194E 01	0.202E 01	0.202E 01	0.185E 01
0.173E 01	0.147E 01	0.120E 01	0.976E 00	0.748E 00	0.547E 00	0.422E 00
0.317E 00	0.235E 00	0.172E 00	0.126E 00	0.926E-01	0.697E-01	0.525E-01
0.396E-01	0.299E-01	0.227E-01	0.173E-01	0.123E-01	0.842E-02	0.591E-02
0.403E 00	0.425E 00	0.447E 00	0.470E 00	0.494E 00	0.519E 00	0.546E 00
0.574E 00	0.603E 00	0.633E 00	0.664E 00	0.697E 00	0.731E 00	0.765E 00
0.802E 00	0.839E 00	0.877E 00	0.916E 00	0.956E 00	0.996E 00	0.104E 01
0.108E 01	0.112E 01	0.116E 01	0.120E 01	0.123E 01	0.122E 01	0.113E 01
0.107E 01	0.926E 00	0.752E 00	0.636E 00	0.510E 00	0.394E 00	0.296E 00
0.231E 00	0.177E 00	0.134E 00	0.101E 00	0.757E-01	0.563E-01	0.432E-01
0.332E-01	0.255E-01	0.196E-01	0.148E-01	0.102E-01	0.716E-02	0.512E-02
0.305E 00	0.319E 00	0.334E 00	0.349E 00	0.364E 00	0.380E 00	0.397E 00
0.414E 00	0.432E 00	0.450E 00	0.469E 00	0.488E 00	0.508E 00	0.528E 00

FIGURE 13E. OUTPUT DATA GRID MATRIX

AIRPOR

0.548E 00	0.569E 00	0.590E 00	0.611E 00	0.632E 00	0.654E 00	0.675E 00
0.695E 00	0.716E 00	0.736E 00	0.755E 00	0.769E 00	0.765E 00	0.698E 00
0.668E 00	0.596E 00	0.502E 00	0.419E 00	0.349E 00	0.280E 00	0.218E 00
0.168E 00	0.133E 00	0.103E 00	0.796E-01	0.611E-01	0.468E-01	0.358E-01
0.275E-01	0.214E-01	0.167E-01	0.120E-01	0.842E-02	0.602E-02	0.438E-02
0.227E 00	0.236E 00	0.246E 00	0.255E 00	0.265E 00	0.275E 00	0.286E 00
0.296E 00	0.307E 00	0.318E 00	0.329E 00	0.341E 00	0.352E 00	0.364E 00
0.375E 00	0.387E 00	0.398E 00	0.410E 00	0.421E 00	0.433E 00	0.444E 00
0.454E 00	0.465E 00	0.475E 00	0.484E 00	0.490E 00	0.488E 00	0.457E 00
0.428E 00	0.390E 00	0.338E 00	0.280E 00	0.240E 00	0.198E 00	0.160E 00
0.126E 00	0.989E-01	0.789E-01	0.623E-01	0.488E-01	0.381E-01	0.296E-01
0.231E-01	0.178E-01	0.134E-01	0.955E-02	0.687E-02	0.502E-02	0.372E-02
0.168E 00	0.174E 00	0.180E 00	0.186E 00	0.192E 00	0.199E 00	0.205E 00
0.212E 00	0.218E 00	0.225E 00	0.231E 00	0.238E 00	0.245E 00	0.251E 00
0.258E 00	0.265E 00	0.271E 00	0.278E 00	0.284E 00	0.290E 00	0.296E 00
0.302E 00	0.307E 00	0.312E 00	0.317E 00	0.322E 00	0.319E 00	0.302E 00
0.281E 00	0.260E 00	0.230E 00	0.196E 00	0.166E 00	0.141E 00	0.116E 00
0.944E-01	0.753E-01	0.601E-01	0.485E-01	0.387E-01	0.308E-01	0.243E-01
0.192E-01	0.152E-01	0.104E-01	0.756E-02	0.558E-02	0.416E-02	0.314E-02
0.124E 00	0.128E 00	0.132E 00	0.135E 00	0.139E 00	0.143E 00	0.147E 00
0.151E 00	0.155E 00	0.159E 00	0.163E 00	0.167E 00	0.171E 00	0.175E 00
0.179E 00	0.183E 00	0.187E 00	0.190E 00	0.194E 00	0.197E 00	0.201E 00
0.204E 00	0.207E 00	0.209E 00	0.212E 00	0.214E 00	0.212E 00	0.203E 00
0.188E 00	0.176E 00	0.159E 00	0.138E 00	0.117E 00	0.101E 00	0.853E-01
0.706E-01	0.576E-01	0.464E-01	0.376E-01	0.306E-01	0.247E-01	0.198E-01
0.159E-01	0.127E-01	0.101E-01	0.775E-02	0.451E-02	0.343E-02	0.263E-02
0.913E-01	0.938E-01	0.963E-01	0.988E-01	0.101E 00	0.104E 00	0.106E 00
0.109E 00	0.111E 00	0.114E 00	0.116E 00	0.119E 00	0.121E 00	0.123E 00
0.126E 00	0.128E 00	0.130E 00	0.132E 00	0.134E 00	0.136E 00	0.130E 00
0.140E 00	0.141E 00	0.143E 00	0.144E 00	0.146E 00	0.144E 00	0.139E 00
0.128E 00	0.121E 00	0.111E 00	0.981E-01	0.847E-01	0.729E-01	0.627E-01
0.529E-01	0.440E-01	0.361E-01	0.291E-01	0.240E-01	0.197E-01	0.161E-01
0.131E-01	0.106E-01	0.834E-02	0.617E-02	0.456E-02	0.282E-02	0.219E-02
0.675E-01	0.691E-01	0.707E-01	0.723E-01	0.739E-01	0.755E-01	0.771E-01
0.787E-01	0.803E-01	0.818E-01	0.833E-01	0.848E-01	0.863E-01	0.877E-01
0.891E-01	0.905E-01	0.918E-01	0.930E-01	0.942E-01	0.953E-01	0.964E-01
0.974E-01	0.983E-01	0.991E-01	0.998E-01	0.101E 00	0.997E-01	0.963E-01
0.892E-01	0.848E-01	0.783E-01	0.704E-01	0.618E-01	0.530E-01	0.463E-01
0.397E-01	0.336E-01	0.280E-01	0.232E-01	0.189E-01	0.157E-01	0.130E-01
0.107E-01	0.862E-02	0.651E-02	0.467E-02	0.367E-02	0.280E-02	0.182E-02
0.501E-01	0.512E-01	0.522E-01	0.533E-01	0.543E-01	0.553E-01	0.564E-01
0.574E-01	0.584E-01	0.593E-01	0.603E-01	0.612E-01	0.621E-01	0.630E-01
0.639E-01	0.647E-01	0.655E-01	0.662E-01	0.669E-01	0.676E-01	0.682E-01
0.688E-01	0.693E-01	0.698E-01	0.702E-01	0.706E-01	0.700E-01	0.679E-01
0.640E-01	0.601E-01	0.560E-01	0.510E-01	0.453E-01	0.396E-01	0.344E-01
0.300E-01	0.257E-01	0.218E-01	0.183E-01	0.152E-01	0.125E-01	0.105E-01
0.857E-02	0.660E-02	0.503E-02	0.385E-02	0.296E-02	0.229E-02	0.179E-02
0.374E-01	0.381E-01	0.388E-01	0.395E-01	0.401E-01	0.408E-01	0.415E-01

FIGURE 13F. OUTPUT DATA GRID MATRIX

AIRPOR

0.421E-01	0.428E-01	0.434E-01	0.440E-01	0.446E-01	0.452E-01	0.457E-01
0.462E-01	0.467E-01	0.472E-01	0.477E-01	0.481E-01	0.485E-01	0.489E-01
0.492E-01	0.495E-01	0.498E-01	0.500E-01	0.503E-01	0.499E-01	0.485E-01
0.461E-01	0.432E-01	0.406E-01	0.372E-01	0.335E-01	0.296E-01	0.257E-01
0.227E-01	0.197E-01	0.169E-01	0.144E-01	0.121E-01	0.102E-01	0.820E-02
0.643E-02	0.500E-02	0.389E-02	0.304E-02	0.239E-02	0.188E-02	0.149E-02
0.281E-01	0.285E-01	0.290E-01	0.294E-01	0.299E-01	0.303E-01	0.308E-01
0.312E-01	0.316E-01	0.320E-01	0.324E-01	0.328E-01	0.331E-01	0.335E-01
0.338E-01	0.341E-01	0.344E-01	0.347E-01	0.350E-01	0.352E-01	0.354E-01
0.356E-01	0.358E-01	0.360E-01	0.361E-01	0.362E-01	0.360E-01	0.351E-01
0.335E-01	0.314E-01	0.297E-01	0.275E-01	0.250E-01	0.223E-01	0.197E-01
0.173E-01	0.152E-01	0.132E-01	0.113E-01	0.967E-02	0.819E-02	0.690E-02
0.479E-02	0.381E-02	0.303E-02	0.241E-02	0.192E-02	0.154E-02	0.123E-02
0.212E-01	0.215E-01	0.218E-01	0.221E-01	0.224E-01	0.227E-01	0.230E-01
0.233E-01	0.235E-01	0.238E-01	0.240E-01	0.243E-01	0.245E-01	0.247E-01
0.250E-01	0.252E-01	0.254E-01	0.255E-01	0.257E-01	0.258E-01	0.260E-01
0.261E-01	0.262E-01	0.263E-01	0.264E-01	0.264E-01	0.262E-01	0.257E-01
0.246E-01	0.231E-01	0.219E-01	0.204E-01	0.187E-01	0.169E-01	0.151E-01
0.132E-01	0.117E-01	0.103E-01	0.895E-02	0.772E-02	0.660E-02	0.561E-02
0.456E-02	0.292E-02	0.237E-02	0.192E-02	0.155E-02	0.126E-02	0.102E-02
0.161E-01	0.163E-01	0.165E-01	0.167E-01	0.169E-01	0.171E-01	0.173E-01
0.175E-01	0.177E-01	0.178E-01	0.180E-01	0.182E-01	0.183E-01	0.185E-01
0.186E-01	0.187E-01	0.188E-01	0.190E-01	0.191E-01	0.192E-01	0.192E-01
0.193E-01	0.194E-01	0.194E-01	0.194E-01	0.195E-01	0.194E-01	0.190E-01
0.183E-01	0.171E-01	0.163E-01	0.153E-01	0.142E-01	0.129E-01	0.116E-01
0.103E-01	0.909E-02	0.806E-02	0.708E-02	0.616E-02	0.520E-02	0.432E-02
0.346E-02	0.226E-02	0.186E-02	0.153E-02	0.125E-02	0.103E-02	0.844E-03
0.123E-01	0.124E-01	0.126E-01	0.127E-01	0.128E-01	0.130E-01	0.131E-01
0.132E-01	0.133E-01	0.135E-01	0.136E-01	0.137E-01	0.138E-01	0.139E-01
0.140E-01	0.140E-01	0.141E-01	0.142E-01	0.143E-01	0.143E-01	0.144E-01
0.144E-01	0.144E-01	0.145E-01	0.145E-01	0.145E-01	0.144E-01	0.142E-01
0.137E-01	0.130E-01	0.123E-01	0.116E-01	0.108E-01	0.988E-02	0.895E-02
0.802E-02	0.708E-02	0.633E-02	0.560E-02	0.478E-02	0.395E-02	0.323E-02
0.262E-02	0.213E-02	0.146E-02	0.122E-02	0.101E-02	0.841E-03	0.697E-03
0.941E-02	0.952E-02	0.962E-02	0.971E-02	0.981E-02	0.990E-02	0.999E-02
0.101E-01	0.102E-01	0.102E-01	0.103E-01	0.104E-01	0.104E-01	0.105E-01
0.106E-01	0.106E-01	0.107E-01	0.107E-01	0.108E-01	0.108E-01	0.108E-01
0.109E-01	0.109E-01	0.109E-01	0.109E-01	0.109E-01	0.108E-01	0.107E-01
0.103E-01	0.986E-02	0.931E-02	0.882E-02	0.825E-02	0.762E-02	0.695E-02
0.627E-02	0.561E-02	0.487E-02	0.416E-02	0.351E-02	0.292E-02	0.243E-02
0.201E-02	0.166E-02	0.137E-02	0.977E-03	0.821E-03	0.688E-03	0.576E-03
0.727E-02	0.734E-02	0.741E-02	0.748E-02	0.754E-02	0.761E-02	0.767E-02
0.773E-02	0.778E-02	0.784E-02	0.789E-02	0.794E-02	0.798E-02	0.802E-02
0.806E-02	0.810E-02	0.813E-02	0.816E-02	0.819E-02	0.821E-02	0.823E-02
0.824E-02	0.825E-02	0.826E-02	0.826E-02	0.828E-02	0.822E-02	0.810E-02
0.786E-02	0.753E-02	0.711E-02	0.677E-02	0.636E-02	0.590E-02	0.542E-02
0.493E-02	0.444E-02	0.351E-02	0.302E-02	0.258E-02	0.219E-02	0.104E-02
0.155E-02	0.130E-02	0.108E-02	0.904E-03	0.666E-03	0.564E-03	0.476E-03

FIGURE 13G. OUTPUT DATA GRID MATRIX

AIRPOR

0.564E-02	0.569E-02	0.574E-02	0.579E-02	0.584E-02	0.588E-02	0.592E-02
0.596E-02	0.600E-02	0.604E-02	0.607E-02	0.611E-02	0.614E-02	0.617E-02
0.619E-02	0.622E-02	0.624E-02	0.626E-02	0.627E-02	0.629E-02	0.630E-02
0.631E-02	0.631E-02	0.632E-02	0.632E-02	0.632E-02	0.628E-02	0.619E-02
0.603E-02	0.579E-02	0.547E-02	0.522E-02	0.493E-02	0.460E-02	0.425E-02
0.389E-02	0.352E-02	0.254E-02	0.222E-02	0.192E-02	0.165E-02	0.141E-02
0.120E-02	0.102E-02	0.864E-03	0.729E-03	0.613E-03	0.462E-03	0.394E-03
0.441E-02	0.444E-02	0.448E-02	0.451E-02	0.454E-02	0.457E-02	0.460E-02
0.463E-02	0.466E-02	0.468E-02	0.471E-02	0.473E-02	0.475E-02	0.477E-02
0.479E-02	0.480E-02	0.482E-02	0.483E-02	0.484E-02	0.485E-02	0.486E-02
0.486E-02	0.486E-02	0.486E-02	0.486E-02	0.487E-02	0.483E-02	0.477E-02
0.465E-02	0.448E-02	0.423E-02	0.406E-02	0.384E-02	0.360E-02	0.334E-02
0.302E-02	0.267E-02	0.233E-02	0.166E-02	0.145E-02	0.126E-02	0.109E-02
0.943E-03	0.809E-03	0.691E-03	0.589E-03	0.500E-03	0.424E-03	0.326E-03

FIGURE 13H. OUTPUT DATA GRID MATRIX

NOISE CONTOUR PLOTTING

A major feature of the Airport Noise Prediction Model is its ability to provide noise contour plots of selected noise levels. This function is achieved by use of a CALCOMP plotter and CALCOMP software.

In the following description, the contouring process for producing output plots is performed by use of MOD-7 and PLOT-7 equipment in accordance with the block diagram arrangement shown in Figure 14.

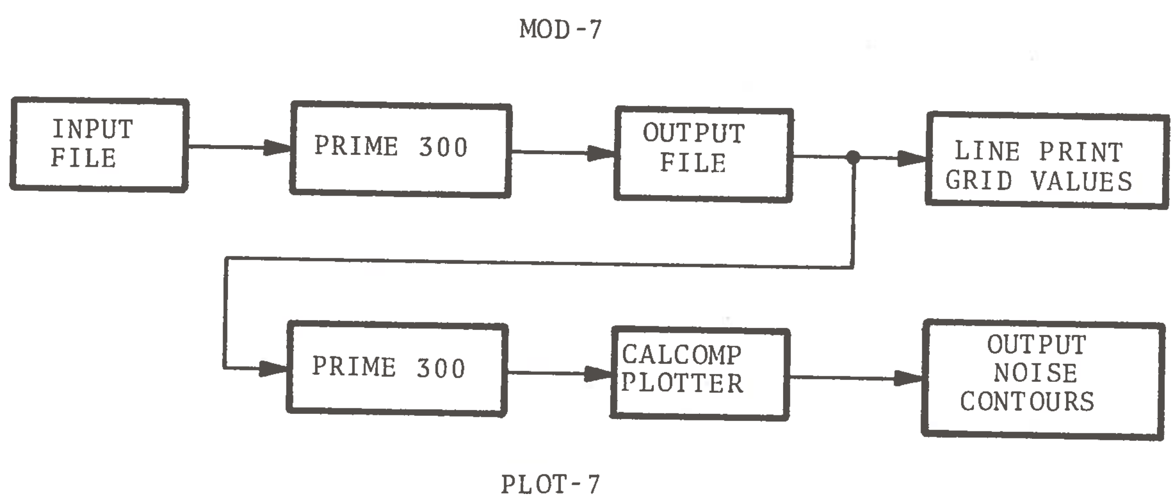


FIGURE 14. TSC PROCESSING SYSTEM FOR OBTAINING NOISE CONTOUR PLOT

The MOD-7 grid matrix output file is inserted into the PLOT-7 plotting program. The program draws any selected number of contour levels (up to 16) of equal noise exposure values, a title, runways, flight paths, and the calculated area within the contour. The following inputs are required for this program:

Number of segments (number of grid matrix output files)

Name of an airport file

Names of grid matrix output files

Desired noise levels

Title

Scaling factor of the plot.

An airport file that contains the data for the runways and flight paths is required. (If this feature is not desired, the file may consist of at least four blank lines.) An airport file of the type described is shown in Figure 15. The contents of this figure form the structure of the airport geometry file named in the command file (mnemonic AOFILE(1) of Table 5) and read by the plotting routine.

	TITLE	Airport descriptive title (unused)
	RLAT,RLONG	Latitude and longitude of airport (unused)
	NRUNS	Integer number of runways (I10)
For each runway	NAME	Runway descriptive title
For each runway	X(1),Y(1)	x-, y-coordinates (ft) of beginning of runway
For each runway	X(2),Y(2)	x-, y-coordinates (ft) of end of runway
For each runway	WDTH	Width of runway (ft)
	NPATHS	Integer number of flight paths (I10)
For each path	NAME	Flight path descriptive title
For each path	NSEG	Integer number of segments (I10)
For each path / For each segment	TYPE	Flight segment type (0 = straight line, -1 = cw*, + 1 = ccw**)
For each path / For each segment	X(1),Y(1)	x-, y-coordinates of beginning of segment
For each path / For each segment	X(2),Y(2)	x-, y-coordinates of end of segment
For each path / For each segment	X(3),Y(3)	x-, y-coordinates of center of helical path

FIGURE 15. AIRPORT FILE (.APT)

A noise contour plot is shown in Figure 16. The coordinate system shown in Figure 16 is the same as the one used in creating the operation file. The X and Y units are expressed in feet. Note that the plot would be "smoother" if a higher density of grid spacings had been selected.

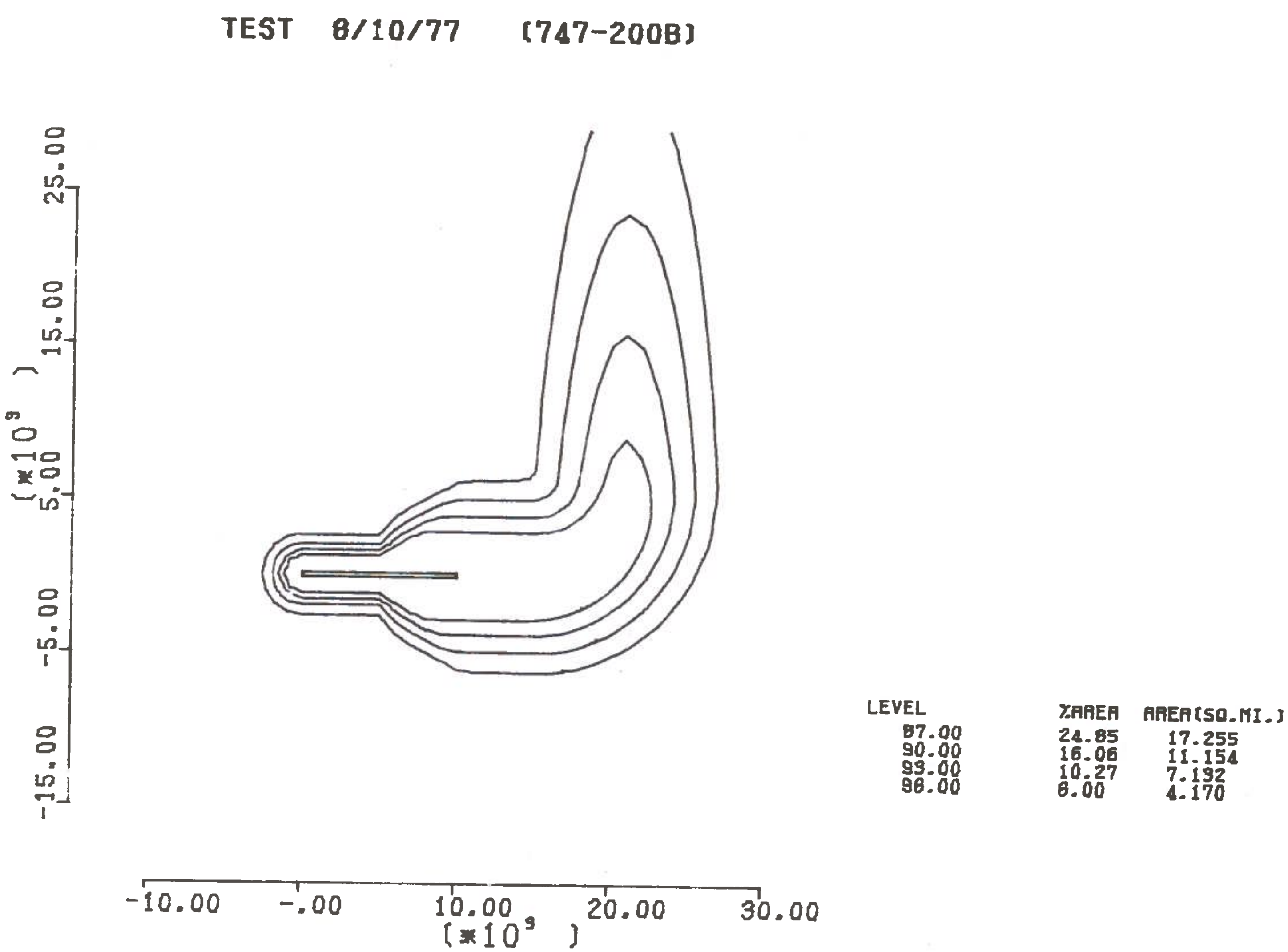


FIGURE 16. SAMPLE CASE NOISE CONTOUR PLOT

PROGRAMMING

This section describes the operations necessary to process the MOD-7 and PLOT-7 programs, using the sample case as an example.

Processing

Command and operation files are prepared in advance using local system routines. MOD-7 is then executed producing output in the form of selected metric values for each grid point specified. These values are on an output file as named by the user. On the PRIME 300 system at TSC the outputs are on a disk file. A line printout is available.

On the TSC system, plotting of noise contours is obtained from the PRIME 300 and a CALCOMP drum plotter, using the PLOT-7 routine and the CALCOMP library.

The MOD-7 Airport Noise Prediction Model currently runs on the PRIME 300 computer at TSC. The prime is a 16-bit mini computer which uses 8-bit ASCII code. All file handling is done with PRIME software, which is machine specific, and logical device numbers for FORTRAN are not necessarily transferable to other machines.

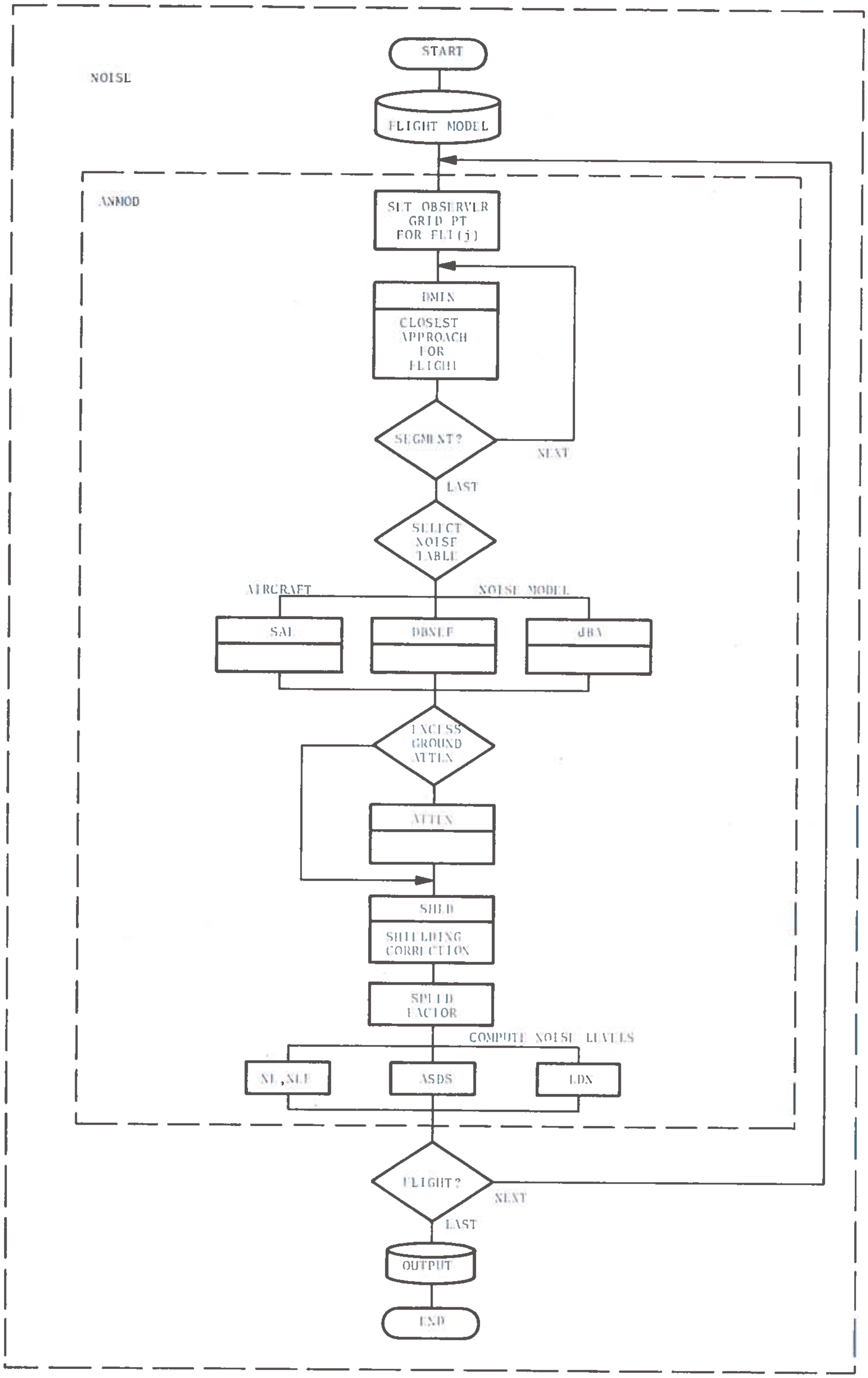


FIGURE 17. DESCRIPTIVE FLOW CHART FOR MOD-7

TABLE 7. MOD-7 PROGRAM

```
C  MOD7
C
C PRIME-300 VERSION OF AIRPORT NOISE PREDICTION MODEL
C
        INTEGER CFILE(3),OPFILE(3),GFILE(3),AOFILE
        COMMON/CNTRL/AOFILE(3,2),NCOLS,NROWS,
     X      DX,DY,CORN(3),NOBS
        COMMON/NOISE/IOPT(2),NTYP,WF(2,3),SUBC(2)
        COMMON/FLITES/FLT(145)
C
C GET COMMAND FILE NAME AND OPEN IT
        WRITE(1,1000)
1000    FORMAT(' ENTER COMMAND FILE NAME:  ',1H$)
        READ(1,1010)(CFILE(I),I=1,3)
1010    FORMAT(3A2)
        CALL SEARCH(1,CFILE,1,0)
C
C READ NO. OF CASES AND CHECK LIMITS
        READ(5,1040) NCASES
1040    FORMAT(I2)
        IF(NCASES.LE.0) GO TO 20
C
C READ ATTENUATION AND SHIELDING OPTIONS
        READ(5,1050) IOPT(1),IOPT(2),NTYP
1050    FORMAT(3I2)
        IF(IOPT(1).LT.1) IOPT(1)=1
        IF(IOPT(1).GT.2) IOPT(1)=2
        IF(IOPT(2).LT.1) IOPT(2)=1
        IF(IOPT(2).GT.3) IOPT(2)=3
C
C LOOP THROUGH ALL CASES
        DO 10 ICASE=1,NCASES
C
C READ NAME OF OPERATION(INPUT) AND GRID(OUTPUT) FILES
        READ(5,1010)(OPFILE(I),I=1,3)
        READ(5,1010)(GFILE(I),I=1,3)
C
C SET UP EACH
        CALL SEARCH(1,OPFILE,2,0)
        CALL SEARCH(2,GFILE,3,0)
C
C READ GRID/LOCI FLAG
        READ(5,1060) IGM
1060    FORMAT(A1)
        IF(IGM .NE. 1HG) GO TO 20
C
C READ AIRPORT FILE NAME
        READ(5,1070) ((AOFILE(I,J),I=1,3),J=1,2)
```

```
C  MOD7


1070     FORMAT(3A2/3A2)
         WRITE(7,1070) ((AOFILE(I,J),I=1,3),J=1,2)
         WRITE(7,1040) NTYP
C
C READ GRID INFO
         READ(5,1080) NCOLS,NROWS,DX,DY,(CORN(I),I=1,3)
1080     FORMAT(I4/I4/2F10.2/3F10.2)
       NCOLS=NCOLS+2
       NROWS=NROWS+2
         NUBS=NCOLS*NROWS
C
C NUMBER OF OBSERVERS MUST BE LESS THAN 2500 !!
         IF(NOBS.GT.2500) GO TO 20
C
C WRITE GRID INFO ONTO OUTPUT FILE
         WRITE(7,1080) NCOLS,NROWS,DX,DY,(CORN(I),I=1,3)
C
C PROCESS
         CALL MOD7
C
C CLOSE FILES AND LOOP BACK FOR MORE CASES
         CALL SEARCH(:10,GFILE,3,0)
       CALL SEARCH(4,0,3)
       CALL SEARCH(4,0,2)
10       CONTINUE
C
         GO TO 30
C
C ERROR IF HERE
20       WRITE(1,1090)
1090     FORMAT(' ERROR SOMEWHERE')
C
30       CALL SEARCH(4,0,3)
       CALL SEARCH(4,0,2)
         CALL SEARCH(4,0,1)
         CALL EXIT
         END
C
C
C
         SUBROUTINE MOD7
C
C COMPUTATIONAL MODULE FOR AIRPORT NOISE PREDICTION MODEL.
C
C INPUT:     COMMAND FILE     UNIT 1, DEVICE 5
C            OPERATION FILE   UNIT 2, DEVICE 6
C OUTPUT:    GRID FILE        UNIT 3, DEVICE 7
C
```

TABLE 7. MOD-7 PROGRAM (CONT'D)

```
C  MOD7


       INTEGER AOFILE
       COMMON/NOISE/IOPT(2),NTYP,WF(2,3),SUBC(2)
         COMMON/FLITES/FLT(145)
         COMMON/CNTRL/AOFILE(3,2),NCOLS,NROWS,
      X    DX,DY,CORN(3),NOBS
C
C RESERVE LOCATIONS FOR INTERNAL VARIABLES
       DIMENSION NFILE(3),PNOISE(3),
      X           VNOISE(2500),IONAM(3)
C
C      SET UP WEIGHTING FACTORS AND SUBTRACTION CONSTANTS
       DATA WF(1,1),WF(2,1)/1.,1./,
      *WF(1,2),WF(2,2)/1.,1./,
      *WF(1,3),WF(2,3)/1.,16.67/
       DATA SUBC(1),SUBC(2)/88.0,88.0/
C
C
       READ(6,1000) NFLTS
1000     FORMAT(40X,3X,I4)
       IF(NFLTS.LE.0) CALL EXIT
C
C DONE WITH INITIAL INPUT
C INITIALIZE ALL SUMMED NOISE VALUES TO ZERO
       DO 20 J=1,2500
       VNOISE(J)=0.0
20     CONTINUE
C
C LOOP THROUGH ALL FLIGHTS
       DO 70 ITWO=1,NFLTS
C
C READ FLIGHT SPECIFICATION FROM FILE
       READ(6,1000) IACTYP
       FLT(1)=FLOAT(IACTYP)+0.2
       READ(6,1000) ITORL
       IF(ITORL.LT.1 .OR. ITORL.GT.2) ITORL=1
       FLT(2)=FLOAT(ITORL)+0.2
       READ(6,1000) NDOPS
       IF(NDOPS.LT.0) NDOPS=1
       FLT(3)=FLOAT(NDOPS)
       READ(6,1000) NEOPS
       IF(NEOPS.LT.0) NEOPS=0
       FLT(4)=FLOAT(NEOPS)
       READ(6,1000) NNOPS
       IF(NNOPS.LT.0) NNOPS=0
       FLT(5)=FLOAT(NNOPS)
C
C READ SEGMENT DATA FOR THIS FLIGHT
       J=1
```

TABLE 7. MOD-7 PROGRAM (CONT'D)

```
C  MOD7


C
C SET SUBSCRIPT FOR FLIGHT ARRAY
30     ISUB=(J-1)*14+6
       READ(6,1000) ISGTYP
       IF(ISGTYP.LT.1 .OR. ISGTYP.GT.2) GO TO 50
       FLT(ISUB)=ISGTYP
       READ(6,1010) FLT(ISUB+1)
1010    FORMAT(40X,3F10.2)
       READ(6,1010) FLT(ISUB+2)
       READ(6,1010) FLT(ISUB+3),FLT(ISUB+4),FLT(ISUB+5)
       READ(6,1010) FLT(ISUB+6),FLT(ISUB+7)
       READ(6,1020) CLMBA
1020    FORMAT(40X,F12.2)
       FLT(ISUB+8)=CLMBA/57.29
       READ(6,1010) FLT(ISUB+12),FLT(ISUB+13)
C
C SET CIRCULATION
       FLT(ISUB+9)=ISGTYP-1
C
C TEST FOR STRAIGHT LINE
       IF(ISGTYP.EQ.1) GO TO 40
C
C NOT STRAIGHT LINE, READ HELIX DATA
       READ(6,1010) CQ
       FLT(ISUB+9)=1.
       IF(CQ.LT.0.)FLT(ISUB+9)=-1.
       READ(6,1010) FLT(ISUB+10),FLT(ISUB+11)
C
C LOOP BACK TO GET NEXT SEGMENT DATA
40     J=J+1
C
C TEST FOR NOT TOO MANY SEGMENTS ALREADY
       IF(J.LE.10) GO TO 30
       ISUB=ISUB+14
C
C DONE WITH SEGMENTS FOR THIS FLIGHT, SET FLAG
50     FLT(ISUB)=-1.0
C
C LOOP THROUGH ALL OBSERVERS
       DO 60 I=1,NOBS
C
C      RETRIEVE OBSERVER COORDINATES
       GA=CORN(1)+FLOAT((I-1)/NROWS)*DX-DX
       GB=CORN(2)-FLOAT(MOD(I-1,NROWS))*DY+DY
       GC=CORN(3)
C
C COMPUTE INCREMENTAL NOISE VALUE FOR THIS OBSERVER
       CALL ANMOD(GA,GB,GC,XNOISE,IER)
```

TABLE 7. MOD-7 PROGRAM (CONT'D)

```
C  MOD7


C
C TEST FOR ERROR RETURN FROM ANMOD
      IF(IER.NE.0) GO TO 60
C
C NO ERROR. ACCUMULATE NOISE VALUES
      VNOISE(I)=VNOISE(I)+XNOISE
C
C CONTINUE THRU OBSERVER LOOP
60    CONTINUE
C
C CONTINUE THRU FLIGHT LOOP
70    CONTINUE
C
C WRITE OUT GRID VALUES
      WRITE(7,1030) (VNOISE(J),J=1,NOBS)
1030  FORMAT(7E10.3)
C
C RETURN
      RETURN
      END
C
C
C
      SUBROUTINE ANMOD(X,Y,Z,XNOISE,IER)
C
C  THIS SUBROUTINE COMPUTES NOISE LEVELS AT A SPECIFIED
C  OBSERVATION POINT FOR A GIVEN FLIGHT.
C
C  ARGUMENTS:
C    X,Y,Z    COORDINATES OF OBSERVATION POINT (INPUT ARG.)
C    XNOISE   NOISE LEVELS COMPUTED AT OBSERVATION POINT.
C    IER      AN ERROR CODE
C     =0      NO ERRORS
C     =1      NO SEGMENTS IN FLIGHT PATH
C
C  SUBROUTINES REQUIRED
C        DMIN
C        DBEPNL
C        DBA
C        ATTEN
C        SHLD
C
C  GAIN ACCESS TO COMMON VARIABLES
      COMMON/FLITES/FLT(145)
      COMMON/NOISE/IOPT(2),NTYP,WF(2,3),SUBC(2)
C
C  SET UP ARRAYS
      DIMENSION T(21),TOPT(21)
```

TABLE 7. MOD-7 PROGRAM (CONT'D)

```
C  MOD7



C
C  SET SEGMENT COUNTER AT ZERO
      ISEG=0
C
C  ASSUME NO ERRORS
      IER=0
C
C  COMPUTE SEGMENTS BEGINNING SUBSCRIPT IN FLIGHT ARRAY
10    ISUB=ISEG*14+6
C
C  HAVE ALL SEGMENTS BEEN PROCESSED
      IF(FLT(ISUB).LE.0.0) GO TO 60
C
C  THIS SEGMENT NEEDS PROCESSING. INCREMENT SEGMENT COUNTER
      ISEG=ISEG+1
C
C  IS THIS THE FIRST SEGMENT
      IF(ISEG.EQ.1) GO TO 40
C
C  THIS IS NOT THE FIRST SEGMENT, ESTABLISH WORKING ARRAY OF
C    FLIGHT PARAMETERS.
      DO 20 I=1,9
      K=ISUB+I+2
20    T(I)=FLT(K)
      T(19)=FLT(ISUB+12)
      T(20)=FLT(ISUB+13)
      T(10)=X
      T(11)=Y
      T(12)=Z
      T(13)=FLT(ISUB+1)
      T(14)=FLT(ISUB+2)
C
C  COMPUTE CLOSET APPROACH VALUES FOR THIS SEGMENT
      CALL DMIN(T)
C
C  IS THIS SEGMENT NOT CLOSEST
      IF(T(15).GE.TOPT(15)) GO TO 10
C
C  UPDATE CLOSEST SEGMENT
      DO 30 I=1,21
30    TOPT(I)=T(I)
      GO TO 10
C
C  ASSUME OPTIMUM VALUES = 1ST SEGMENTS VALUES
40    DO 50 I=1,9
      K=ISUB+I+2
50    TOPT(I)=FLT(K)
      TOPT(10)=X
```

TABLE 7. MOD-7 PROGRAM (CONT'D)

```
      TOPT(11)=Y
      TOPT(12)=Z
      TOPT(13)=FLT(7)
      TOPT(14)=FLT(8)
      TOPT(19)=FLT(18)
      TOPT(20)=FLT(19)
      CALL DMIN(TOPT)
C
C  RETURN TO PROCESS NEXT SEGMENT
      GO TO 10
C
C  IF THERE ARE NO SEGMENTS SET ERROR CODE
60    IF(ISEG) 70,70,80
70    IER=1
      RETURN
C
C  COMPUTE EPNL VALUE FOR THIS FLIGHT.
80    KCV=ICURVE(FLT(1))
110   IF(ABS(TOPT(17))-1.)120,130,130
120   BETA=ATAN(TOPT(17)/SQRT(1.0-TOPT(17)**2))
      GO TO 140
130   BETA=1.5708
140   HDIST=TOPT(15)*COS(BETA)
C
C     ACCOUNT FOR EXCESS GROUND ATTENUATION
      KOPT=IOPT(2)
      GO TO(150,160,170),KOPT
150   VAL=0.
      GO TO 180
160   CALL ATTEN(13.,HDIST,TOPT(16),BETA,VAL)
      GO TO 180
170   CALL ATTEN(3.,HDIST,TOPT(16),BETA,VAL)

C
C     ACCOUNT FOR SHIELDING EFFECT
180   KOPT=IOPT(1)
      GO TO (200,190),KOPT
190   CALL SHLD(BETA,DVAQ)
      VAL=VAL+DVAQ
C
C  ACCOUNT FOR SPEED FACTOR
200   SPD=10.*ALOG10(TOPT(21)/160.)
C
C  COMPUTE NOISE LEVEL CONTRIBUTIONS AT THIS POINT
      GO TO (210,220), NTYP
C
C     FOR ABSOLUTE NOISE LEVEL
  210 CALL DBA(KCV,TOPT(16),TOPT(15),EPNL)
```

TABLE 7. MOD-7 PROGRAM (CONT'D)

```
      EPNL=EPNL-VAL-SPD
      RETURN
C
C
C     FOR EPNL COMPUTATION
  220 CALL DBEPNL(KCV,TOPT(16),TOPT(15),EPNL)
      EPNL=EPNL-VAL-SPD
      I=NTYP
      A=0.0
      DO 221 J=1,3
 221  A=A+WF(I,J)*FLT(J+2)
      XNOISE=10.**((EPNL+10.*ALOG10(A)-SUBC(I))/10.)
      RETURN
C
      END
C
C
C
      FUNCTION ICURVE(ACTYPE)
C
C     COMPUTES AIRCRAFT TYPE TO SELECT PROPER CURVE
C     AND CHECKS FOR INVALID AIRCRAFT TYPES.
C     CONTAINS DESCRIPTIONS OF AIRCRAFT TYPES.
C
C     *=DATA AVAILABLE FOR DBA COMPUTATIONS
C
C     TYPE DESCRIPTION
C     1     747-200 (JT9D-3A)
C     2     L-1011
C     3     707-320B
C     4     707-320 (JT4A-3)
C     5     727-200 (JT8D-9)
C     6     DC9-32 (JT8D-7)
C     7     F27 (2 PROPJET)
C     8     2 PISTON/OVER 12000 LB.
C     9     2 PISTON/UNDER 12000 LB.
C     10    LEARJET (CJ610-6)
C     46   *747-200B (JT9D-7/QN)
C     47   *707-320B (JT3D-3B)
C     48   *727-200 (JT8D-9,-15)
C     49   *737-200 (JT8D-9,-15)
C     50   *747-100 (JT9D-7)
C     51   *707-320B/QN (JT3D-3D/QN)
C     52   *727-200/QN (JT8D-9/QN,-15/QN)
C     53   *737-200/QN (JT8D-9/QN,-15/QN)
C     54    707-320B/RFN (JT3D/RFN)
C     55   *DC-8-61 (JT3D-3D)
C     56   *DC-89-63 (JT3D-7)
```

TABLE 7. MOD-7 PROGRAM (CONT'D)

```
57     DC-8-63/RFN (JT3D/RFN)
58     727-200/RFN (JT8D/RFN)
59     737-200/RFN (JT8D/RFN)
60     DC-9-30/RFN (JT8D/RFN)
61     DC-8-61/RFN (JT3D/RFN)
63     DC-8-61
64    *DC-10-10 (CF6-6D)
65    *DC-9-30 (JT8D-9)
66     CONCORDE


    COMPUTE AIRCRAFT TYPE
    J=IFIX(ACTYPE)

    CHECK FOR VALID AIRCRAFT TYPE
    IF(J.GT.10.AND.J.LT.46) GO TO 10
    IF(J.EQ.62.OR.J.GT.66) GO TO 10
    ICURVE=J
    RETURN

    HERE IF UNKNOWN AIRCRAFT TYPE
10  PRINT 15
15  FORMAT(' INVALID AIRCRAFT TYPE - PROGRAM HALTED')
    CALL SEARCH(4.0,1)
    CALL SEARCH(4.0,2)
    CALL SEARCH(4.0,3)
    CALL EXIT
    END

    SUBROUTINE DMIN(T)

    THIS SUBROUTINE COMPUTES VALUES ASSOCIATED
    WITH A POINT OF CLOSEST APPROACH ON A
    FLIGHT PATH. (EITHER STRAIGHT LINE OR HELICAL)

    ARGUMENTS:
       T = VECTOR ARRAY OF LENGTH 21

       T(1)  = X COORD OF START OF TRAJECTORY
       T(2)  = Y COORD OF START OF TRAJECTORY
       T(3)  = Z COORD OF START OF TRAJECTORY
       T(4)  = X COORD OF END OF TRAJECTORY
       T(5)  = Y COORD OF END OF TRAJECTORY
       T(6)  = CLIMB OR DESCENT ANGLE(+,-) IN RADIANS
       T(7)  = CIRCULATION(CCW,CW)(+1,-1) (0)=STRAIGHT LINE
       T(8)  = X COORD OF HELIX AXIS
       T(9)  = Y COORD OF HELIX AXIS
       T(10)= X COORD OF OBSERVER
```

TABLE 7. MOD-7 PROGRAM (CONT'D)

```
C   MOD7


C         T(11)= Y COORD OF OBSERVER
C         T(12)= Z COORD OF OBSERVER
C         T(13)= THRUST AT START OF TRAJECTORY
C         T(14)= THRUST AT END OF TRAJECTORY
C         T(15)= DISTANCE OF CLOSEST APPROACH
C         T(16)= THRUST AT CLOSEST APPROACH
C         T(17)= SINE OF OBSERVER ELEVATION ANGLE AT CLOSEST APPROACH
C         T(18)=TOTAL SEGMENT LENGTH
C         T(19)=SPEED AT SEG. BEG.
C         T(20)=SPEED AT SEG. END
C         T(21)=SPEED AT CLOSEST APPROACH
C
C      SUBPROGRAMS REQD- FUNCTION THRST(T1,T2,S,XL)
C
C      ESTABLISH LOCATIONS FOR ARRAYS
       DIMENSION T(1)
         DATA TWOPI/6.28318/
C
         TPSI=SIN(T(6))/COS(T(6))
         V1X=T(10)-T(1)
         V1Y=T(11)-T(2)
         V1Z=T(12)-T(3)
C
C        BRANCH FO HELIX OR STRAIGHT LINE
         IF(ABS(T(7))-.05) 10,10,50
C
C        STRAIGHT LINE SECTION. COMPUTE COORDS
C       OF POINT OF CLOSEST APPROACH
10       DU=T(4)-T(1)
         DV=T(5)-T(2)
         DW=TPSI*SQRT(DU**2+DV**2)
       DA=SQRT(DU*DU+DV*DV+DW*DW)
       V2M=(V1X*DU+V1Y*DV+V1Z*DW)/DA
       V2X=V2M*DU/DA
       V2Y=V2M*DV/DA
       V2Z=V2M*DW/DA
       V4X=V2X-DU
       V4Y=V2Y-DV
       V4Z=V2Z-DW
       DOT=V2X*V4X+V2Y*V4Y+V2Z*V4Z
       IF(DOT.LT.0.)GO TO 30
       DOT=V2X*DU+V2Y*DV+V2Z*DW
       IF(DOT.GT.0.)GO TO 20
C
C      AT END 1
       XN=T(1)
       YN=T(2)
       ZN=T(3)
```

TABLE 7. MOD-7 PROGRAM (CONT'D)

```
C  MOD7


      S=0.
      GO TO 40
C
C     AT END 2
20    XN=T(4)
      YN=T(5)
      ZN=T(3)+DW
      S=DA
      GO TO 40
30    XN=T(1)+V2X
      YN=T(2)+V2Y
      ZN=T(3)+V2Z
      S=V2M
40    T(18)=DA
        GO TO 90
C
C      HELICAL SECTION. COMPUTE P.O.CA. COORDS
50      DSMIN=10.
        R=SQRT((T(1)-T(8))**2+(T(2)-T(9))**2)
        DSMAX=R/4.
        IFLG=0
        IF(T(7) .GT. 0.) CIRC=1.
        IF(T(7) .LT. 0.) CIRC=-1.
        THETA1=ATAN2(T(2)-T(9),T(1)-T(8))
      IF(THETA1 .LT. 0.) THETA1=THETA1+TWOPI
        THETA2=ATAN2(T(5)-T(9),T(4)-T(8))
      IF(THETA2 .LT. 0.) THETA2=THETA2+TWOPI
        IF(CIRC.EQ.+1. .AND. THETA1.GT.THETA2) THETA2=THETA2+TWOPI
        IF(CIRC.EQ.-1. .AND. THETA2.GT.THETA1) THETA1=THETA1+TWOPI
        FM=1.
        IF(THETA1 .GT. THETA2) FM=-1.
        RH0=SQRT(V1X**2+V1Y**2+V1Z**2)
        DMIIN=RH0
        THETA=THETA1
        THMIN=THETA1
C
60      DS=RH0/4.
        IF(DS .LT. DSMIN) DS=DSMIN
        IF(DS .GT. DSMAX) DS=DSMAX
        THETA=THETA+FM*DS/R
        IF(CIRC*(THETA-THETA2) .LT. 0.) GO TO 70
        IFLG=1
        THETA=THETA2
70      XN=R*COS(THETA)+T(8)
        YN=R*SIN(THETA)+T(9)
        ZN=R*FM*(THETA-THETA1)*TPSI+T(3)
        RH0=SQRT((XN-T(10))**2+(YN-T(11))**2+(ZN-T(12))**2)
        IF(RH0 .GE. DMIIN) GO TO 80
```

TABLE 7. MOD-7 PROGRAM (CONT'D)

```
C   MOD7

        THMIN=THETA
        DMIIN=RHO
80      IF(IFLG .EQ. 0) GO TO 60
        XN=R*COS(THMIN)+T(8)
        YN=R*SIN(THMIN)+T(9)
        S=(THMIN-THETA1)*R*FM
        ZN=S*TPSI+T(3)
        S=ABS(S)/COS(T(6))
        T(18)=R*ABS(THETA2-THETA1)/COS(T(6))
C
C      COMPUTE RETURN PARAMETERS
90      T(15)=SQRT((XN-T(10))**2+(YN-T(11))**2+(ZN-T(12))**2)
      T(16)=THRST(T(13),T(14),S,T(18))
      T(21)=THRST(T(19),T(20),S,T(18))
      DZ=ZN-T(12)
      IF(DZ.EQ.0.)GO TO 100
      T(17)=DZ/T(15)
        GO TO 110
100   T(17)=0.
110    RETURN
      END
C
C
      SUBROUTINE DBEPNL(I,P,X,E)
C
C     THIS SUBROUTINE COMPUTES EFFECTIVE PERCEIVED
C     NOISE LEVEL (EPNL) IN DB
C
C     ARGUMENTS:
C       I=AIRCRAFT IDENTIFICATION NUMBER
C       P=PERCENT THRUST
C       X=DISTANCE FROM NOISE SOURCE IN FEET
C       E=COMPUTED NOISE LEVEL IN DECIBELS
C
C
      DIMENSION PL(66)
      DIMENSION AT(66),BT(66),CT(66),DT(66),ET(66)
      DIMENSION AL(66),BL(66),CL(66),DL(66),EL(66)
C
C
      DATA PL(1),PL(2),PL(3),PL(4),PL(5),PL(6)
     */58.8,43.0,45.0,27.1,28.1,26.6/,
     *PL(7),PL(8),PL(9),PL(10),PL(46),PL(47)
     */45.0,45.0,45.0,45.0,23.3,23.8/,
     *PL(48),PL(49),PL(50),PL(51),PL(52),PL(53)
     */32.9,32.6,23.3,23.3,29.7,31.0/,
     *PL(54),PL(55),PL(56),PL(57),PL(58),PL(59)
     */39.4,26.7,26.3,28.8,40.2,28.0/,
```

TABLE 7. MOD-7 PROGRAM (CONT'D)

```
C  MOD7

     *PL(60),PL(61),PL(63),PL(64),PL(65),PL(66)
     */39.8,26.9,26.7,30.0,32.0,45.0/
C
C
      DATA AT(1),AT(2),AT(3),AT(4),AT(5),AT(6)
     */-212.27,383.26,195.82,70.79,274.29,115.46/,
     *AT(7),AT(8),AT(9),AT(10),AT(46),AT(47)
     */237.73,149.83,113.67,18.43,185.67,202.73/,
     *AT(48),AT(49),AT(50),AT(51),AT(52),AT(53)
     */133.84,246.76,263.61,150.86,203.99,217.23/,
     *AT(54),AT(55),AT(56),AT(57),AT(58),AT(59)
     */163.45,127.84,105.65,285.36,162.60,275.73/,
     *AT(60),AT(61),AT(63),AT(64),AT(65),AT(66)
     */62.69,169.08,191.55,85.95,103.02,146.93/
C
C
      DATA BT(1),BT(2),BT(3),BT(4),BT(5),BT(6)
     */363.17,-306.97,-61.51,53.24,-175.41,50.21/,
     *BT(7),BT(8),BT(9),BT(10),BT(46),BT(47)
     */-114.75,-25.39,5.61,160.10,-37.38,-61.41/,
     *BT(48),BT(49),BT(50),BT(51),BT(52),BT(53)
     */34.70,-127.42,-127.48,-24.49,-52.92,-84.87/,
     *BT(54),BT(55),BT(56),BT(57),BT(58),BT(59)
     */-39.33,-2.34,24.66,-155.73,5.95,-125.29/,
     *BT(60),BT(61),BT(63),BT(64),BT(65),BT(66)
     */42.26,-18.31,-50.60,42.26,24.50,-13.67/
C
C
      DATA CT(1),CT(2),CT(3),CT(4),CT(5),CT(6)
     */-136.95,134.70,23.73,-10.36,81.27,-40.63/,
     *CT(7),CT(8),CT(9),CT(10),CT(46),CT(47)
     */33.75,2.95,-8.77,-82.61,-2.19,16.80/,
     *CT(48),CT(49),CT(50),CT(51),CT(52),CT(53)
     */-35.98,45.08,36.94,4.94,2.81,23.83/,
     *CT(54),CT(55),CT(56),CT(57),CT(58),CT(59)
     */11.09,8.56,-5.50,41.95,-33.17,24.57/,
     *CT(60),CT(61),CT(63),CT(64),CT(65),CT(66)
     */-5.32,-17.75,13.81,-12.73,-7.25,8.82/
C
C
      DATA DT(1),DT(2),DT(3),DT(4),DT(5),DT(6)
     */20.43,-26.00,-5.17,-1.78,-18.11,11.36/,
     *DT(7),DT(8),DT(9),DT(10),DT(46),DT(47)
     */-3.45,0.52,2.50,17.67,4.02,-1.60/,
     *DT(48),DT(49),DT(50),DT(51),DT(52),DT(53)
     */10.87,-6.77,-3.35,0.72,3.49,-1.98/,
     *DT(54),DT(55),DT(56),DT(57),DT(58),DT(59)
     */-1.99,-4.61,-1.33,-1.62,12.70,0.56/,
```

TABLE 7. MOD-7 PROGRAM (CONT"D)

```
     *DT(60),DT(61),DT(63),DT(64),DT(65),DT(66)
     */-2.76,9.58,-2.17,-0.79,-0.33,-2.48/
C
C
      DATA ET(1),ET(2),ET(3),ET(4),ET(5),ET(6)
     */-1.06,1.66,0.36,0.35,1.48,-1.22/,
     *ET(7),ET(8),ET(9),ET(10),ET(46),ET(47)
     */-0.07,-0.19,-0.32,-1.49,-0.69,-0.14/,
     *ET(48),ET(49),ET(50),ET(51),ET(52),ET(53)
     */-1.18,0.23,-0.18,-0.36,-0.66,-0.19/,
     *ET(54),ET(55),ET(56),ET(57),ET(58),ET(59)
     */0.09,0.52,0.23,-0.63,-1.51,-0.53/,
     *ET(60),ET(61),ET(63),ET(64),ET(65),ET(66)
     */0.41,-1.38,0.06,0.30,0.08,0.10/
C
C
      DATA AL(1),AL(2),AL(3),AL(4),AL(5),AL(6)
     */-60.27,151.60,-7.40,178.29,88.01,118.80/,
     *AL(7),AL(8),AL(9),AL(10),AL(46),AL(47)
     */120.73,124.12,140.87,87.86,137.68,88.54/,
     *AL(48),AL(49),AL(50),AL(51),AL(52),AL(53)
     */120.34,129.97,-75.68,137.49,-52.09,178.42/,
     *AL(54),AL(55),AL(56),AL(57),AL(58),AL(59)
     */186.85,467.09,51.22,264.88,167.11,227.07/,
     *AL(60),AL(61),AL(63),AL(64),AL(65),AL(66)
     */104.31,89.56,168.45,114.07,57.24,123.04/
C
C
      DATA BL(1),BL(2),BL(3),BL(4),BL(5),BL(6)
     */160.12,-16.70,69.46,-46.04,23.21,11.34/,
     *BL(7),BL(8),BL(9),BL(10),BL(46),BL(47)
     */-4.85,-8.67,-31.45,36.93,-26.71,25.18/,
     *BL(48),BL(49),BL(50),BL(51),BL(52),BL(53)
     */14.38,6.61,257.84,8.34,247.27,-26.42/,
     *BL(54),BL(55),BL(56),BL(57),BL(58),BL(59)
     */-52.24,-486.90,85.51,-113.60,-57.42,-65.52/,
     *BL(60),BL(61),BL(63),BL(64),BL(65),BL(66)
     */13.16,67.55,-16.75,6.57,145.95,16.98/
C
C
      DATA CL(1),CL(2),CL(3),CL(4),CL(5),CL(6)
     */-44.50,-7.09,21.62,8.12,-2.28,-7.53/,
     *CL(7),CL(8),CL(9),CL(10),CL(46),CL(47)
     */3.88,-3.59,4.38,-20.52,14.32,-0.49/,
     *CL(48),CL(49),CL(50),CL(51),CL(52),CL(53)
     */-13.84,-10.04,-125.69,-23.40,-135.63,-7.71/,
     *CL(54),CL(55),CL(56),CL(57),CL(58),CL(59)
     */2.96,258.56,-32.60,8.85,24.60,0.10/,
```

TABLE 7. MOD-7 PROGRAM (CONT'D)

```
C   MOD7


     *CL(60),CL(61),CL(63),CL(64),CL(65),CL(66)
     */-3.16,-55.02,-7.38,-1.19,-97.79,-13.13/
C
C
      DATA DL(1),DL(2),DL(3),DL(4),DL(5),DL(6)
     */2.14,4.04,-17.56,0.71,-2.48,0.57/,
     *DL(7),DL(8),DL(9),DL(10),DL(46),DL(47)
     */-2.27,1.68,0.66,4.35,-4.56,-2.71/,
     *DL(48),DL(49),DL(50),DL(51),DL(52),DL(53)
     */3.44,2.08,25.69,7.95,31.15,3.88/,
     *DL(54),DL(55),DL(56),DL(57),DL(58),DL(59)
     */3.40,-60.85,4.18,7.60,-6.29,4.20/,
     *DL(60),DL(61),DL(63),DL(64),DL(65),DL(66)
     */-1.87,15.83,3.38,-1.97,24.78,2.89/
C
C
      DATA EL(1),EL(2),EL(3),EL(4),EL(5),EL(6)
     */0.26,-0.66,2.28,-0.41,0.31,-0.04/,
     *EL(7),EL(8),EL(9),EL(10),EL(46),EL(47)
     */0.19,-0.28,-0.26,-0.45,0.38,0.23/,
     *EL(48),EL(49),EL(50),EL(51),EL(52),EL(53)
     */-0.42,-0.25,-2.05,-0.95,-2.71,-0.53/,
     *EL(54),EL(55),EL(56),EL(57),EL(58),EL(59)
     */-0.68,5.14,-0.29,-1.53,0.57,-0.71/,
     *EL(60),EL(61),EL(63),EL(64),EL(65),EL(66)
     */0.29,-1.72,-0.55,0.24,-2.35,-0.34/
C
C
C     COMPUTE LOG OF DISTANCE
      S=X
      IF(S.LT.100.) S=100.
      S=ALOG10(S)
C
C
C     COMPUTE THRUST PROPORTION FACTOR
      Q=(P-PL(I))/(100.-PL(I))
C
C
C     COMPUTE NOISE LEVELS AND
C     INTERPOLATE FOR THIS THRUST
      EPNLT=AT(I)+BT(I)*S+CT(I)*(S**2)+DT(I)*(S**3)+ET(I)*(S**4)
      EPNLL=AL(I)+BL(I)*S+CL(I)*(S**2)+DL(I)*(S**3)+EL(I)*(S**4)
      E=EPNLL+(EPNLT-EPNLL)*Q
      RETURN
      END
C
C
C
```

TABLE 7. MOD-7 PROGRAM (CONT'D)

```
C   MOD7



C
C
      FUNCTION THRST(T1,T2,X,XL)
C
C  THIS FUNCTION COMPUTES A THRUST VALUE FOR A SPECIFIED
C  POINT ON A FLIGHT PATH SEGMENT.
C
C  ARGUMENTS:
C    T1=THRUST AT START OF SEGMENT
C    T2=THRUST AT END OF SEGMENT
C    X=DISTANCE IN FEET ALONG SEGMENT
C    XL=TOTAL SEGMENT LENGTH
C
      DATA KEY/0/
      IF(ABS(XL) .LT. 1.) GO TO 10
      A=ABS(X)/AMIN1(1000.,ABS(XL))
      IF(ABS(X).GT.1000.0) A=1.0
      THRST=T1+A*(T2-T1)
      KEY=KEY+1
C
      RETURN
10      KEY=KEY/2
      WRITE(1,1010)XL,KEY
1010  FORMAT(13H THRST: PATH=,F10.1,5H SEG ,I5)
      CALL EXIT
      END
C
C
C
      SUBROUTINE ATTEN(CUTOFF,DIST,THRST,BETA,VAL)
C
C THIS ROUTINE COMPUTES THE ATTENUATION FACTOR TO BE
C USED IN CALCULATION THE EFFECTIVE PERCEIVED NOISE
C LEVEL (EPNL).
C
C ARGUMENTS:
C    CUTOFF = THE CUT-OFF POINT OF BETA
C    DIST = HORIZONTAL DISTANCE FROM OBSERVER IN FEET
C    THRST = THE AIRCRAFT'S THRUST
C    BETA = THE OBSERVATION ANGLE
C    VAL = THE VALUE OF ATTENUATION
C
      DIMENSION A(4,2),C(4)
C
C SET COEFFICIENTS FOR LANDING(THRUST=45.) CURVE
      DATA A(1,1),A(2,1),A(3,1),A(4,1)/4.14,1200.,0.198,8.5/
C
C SET COEFFICIENTS FOR TAKE-OFF(THRUST=100.) CURVE
```

TABLE 7. MOD-7 PROGRAM (CONT'D)

```
C  MOD7


      DATA A(1,2),A(2,2),A(3,2),A(4,2)/2.675,1072.,0.252,6.4/
C
C INITIALIZE VALUE TO ZERO
      VAL=0.0
      IF(DIST.LE.1.E-10)RETURN
C
C TEST FOR CUT-OFF POINT OF BETA
      CUT=90./CUTOFF*3.14159/180.
      IF(BETA.GE.CUT) RETURN
C
C BETA IS LESS THAN 90/CUTOFF INTERPOLATE BETWEEN LANDING
C AND TAKE-OFF CURVES TO GET COEFFICIENTS APPROPRIATE
C TO THIS THRUST
      DO 10 I=1,4
      C(I)=((100.0-THRST)/55.0)*(A(I,1)-A(I,2))+A(I,2)
10    CONTINUE
C
C COMPUTE DELTA-ZERO
      DELZRO=C(1)*ATAN(ALOG10(DIST/C(2))/C(3))+C(4)
C
C COMPUTE VALUE
      THTA=CUTOFF*BETA
      VAL=DELZRO*EXP(-SQRT(ABS(SIN(THTA)/COS(THTA))))
C
      RETURN
      END
C
C
C
      SUBROUTINE SHLD(BETA,S)
C
C     THIS SUBROUTINE COMPUTES A SHIELDING
C     CORRECTION FACTOR
C
C     BETA=OBSERVER ELEVATION ANGLE IN RADIANS
C
C     S=EPNL CORRECTION FACTOR(DB)
C
      S=3.*(1.-SQRT(ABS(SIN(BETA))))
      RETURN
      END
C
C
C
C
C
      SUBROUTINE DBA(I,P,X,E)
      DIMENSION PT(60),PL(60),AT(60),BT(60),CT(60),AL(60),BL(60),CL(60)
```

TABLE 7. MOD-7 PROGRAM (CONT'D)

```
C  MOD7

       DATA PL(1),PL(2),PL(3),PL(4),PL(5),PL(6),PL(7),PL(8),PL(9),PL(10),
      *     PL(11),PL(12),PL(13),PL(14),PL(15),PL(16),PL(17),PL(18),
      *     PL(19),PL(20),PL(21),PL(22)/
      * 23.3,23.8,26.8,32.6,23.3,25.1,29.7,31.,000.,26.7,26.7,000.,000.,
      * 000.,000.,000.,000.,000.,30.,16.,000.,35.6/
       DATA AL(1),AL(2),AL(3),AL(4),AL(5),AL(6),AL(7),AL(8),AL(9),AL(10),
      *     AL(11),AL(12),AL(13),AL(14),AL(15),AL(16),AL(17),AL(18),
      *     AL(19),AL(20),AL(21),AL(22)/
      * 121.,160.,115.,118.,128.,109.,114.,103.,000.,144.,149.,000.,
      * 000.,000.,000.,000.,000.,000.,115.,104.,000.,126./
       DATA BL(1),BL(2),BL(3),BL(4),BL(5),BL(6),BL(7),BL(8),BL(9),BL(10),
      *     BL(11),BL(12),BL(13),BL(14),BL(15),BL(16),BL(17),BL(18),
      *     BL(19),BL(20),BL(21),BL(22)/
      * 93.9,-389.,327.,244.,-11.5,244.,114.,363.,000.,-183.,
      * -315.,000.,000.,000.,000.,000.,000.,000.,292.,441.,000.,50.7/
       DATA CL(1),CL(2),CL(3),CL(4),CL(5),CL(6),CL(7),CL(8),CL(9),CL(10),
      *     CL(11),CL(12),CL(13),CL(14),CL(15),CL(16),CL(17),CL(18),
      *     CL(19),CL(20),CL(21),CL(22)/
      * .822,1.27,.768,.826,.879,.764,.736,.652,000.,.991,1.08,000.,
      * 000.,000.,000.,000.,000.,000.,.872,.667,000.,.841/
       DATA AT(1),AT(2),AT(3),AT(4),AT(5),AT(6),AT(7),AT(8),AT(9),AT(10),
      *     AT(11),AT(12),AT(13),AT(14),AT(15),AT(16),AT(17),AT(18),
      *     AT(19),AT(20),AT(21),AT(22)/
      * 132.,139.,136.,137.,136.,133.,134.,142.,000.,140.,129.,000.,
      * 000.,000.,000.,000.,000.,000.,132.,130.,000.,132./
       DATA BT(1),BT(2),BT(3),BT(4),BT(5),BT(6),BT(7),BT(8),BT(9),BT(10),
      *     BT(11),BT(12),BT(13),BT(14),BT(15),BT(16),BT(17),BT(18),
      *     BT(19),BT(20),BT(21),BT(22)/
      * 89.5,121.,78.5,17.8,27.4,-5.68,74.2,-36.2,000.,-64.8,
      * 85.4,000.,000.,000.,000.,000.,000.,000.,-49.8,4.81,000.,119./
       DATA CT(1),CT(2),CT(3),CT(4),CT(5),CT(6),CT(7),CT(8),CT(9),CT(10),
      *     CT(11),CT(12),CT(13),CT(14),CT(15),CT(16),CT(17),CT(18),
      *     CT(19),CT(20),CT(21),CT(22)/
      * .784,.869,.721,.778,.814,.790,.692,.817,000.,.726,.620,000.,
      * 000.,000.,000.,000.,000.,000.,.848,.647,000.,.771/
       DO 10 J=1,60
 10    PT(J)=100.
       S=X
       IF(S .LT. 100.) S=100.
       S=ALOG(S)**2
       K=I-45
 C
 C     CHECK FOR INVALID AIRCRAFT TYPE
       IF(K.LT.1.OR.K.GT.21) GO TO 20
       IF(PL(K).EQ.0.) GO TO 30
 C
 C     COMPUTE THRUST PROPORTION FACTOR
       Q=(P-PL(K))/(PT(K)-PL(K))
```

TABLE 7. MOD-7 PROGRAM (CONT'D)

```
C  MOD7


C
C     COMPUTE TAKEOFF AND LANDING NOISE LEVELS
      ET=AT(K)+BT(K)/(1.+S)-CT(K)*S
      EL=AL(K)+BL(K)/(1.+S)-CL(K)*S
      E=EL+(ET-EL)*Q
      IF(X .LT. 100.) E=E+20.*(100.-X)/100.
      RETURN
C
C     HERE FOR INVALID AIRCRAFT TYPE
   20 WRITE(1,1010) K
 1010 FORMAT(' INVALID AIRCRAFT TYPE',I2/' SUPPLIED TO SUBROUTINE DBA')
      CALL EXIT
C
C     HERE FOR NO DATA
   30 WRITE(1,1020) K
 1020 FORMAT(' NO DATA FOR AIRCRAFT TYPE' ,I2,/,' IN SUBROUTINE DBA')
      CALL SEARCH(4,0,1)
      CALL SEARCH(4,0,2)
      CALL SEARCH(4,0,3)
      CALL EXIT
      END
```

TABLE 7.MOD-7 PROGRAM (CONCLUDED)

TABLE 8. PLOT-7 PROGRAM.

```
C       PROGRAM PLOT7


C       PROGRAM PLOT7
C
C MODIFIED 7/29/77 BY PRESCOTT C. HEALD/KHL
C
C PRIME 300 VERSION OF THE OUTPUT MODULE FOR
C THE TSC AIRPORT NOISE PREDICTION MODEL
C
C COMMON FOR OUTPUT MODULE OF MOD7
C
      INTEGER GFILE,AFILE,TITLE
      COMMON/ZCOM/Z(2500),ZZ(2500),NUM
      COMMON/XFMCOM/X0,Y0,SF(3),U0(3),V0(3)
      COMMON/GRDCOM/NR,NC,DX,DY,X1,Y1,TAREA
      COMMON/MISCEL/RLAT,RLONG,NTYPE
      COMMON/FILES/TITLE(40),AFILE(3),GFILE(3)
      COMMON//NSEG,XDB(33),IUSE,KTITLE(40),NVALU,XXAREA(16),
     *          YYAREA(16)
      DIMENSION XDB1(33)
C
C     SET NUMBER OF NOISE VALUES TO BE PLOTTED TO ZERO
      NVALU=0
C
C     GET NUMBER OF SEGMENTS
      WRITE(1,101)
  101 FORMAT(1X,15HSEGMENTS (I2): ,1H$)
      READ(1,102) NSEG
  102 FORMAT(I2)
C
C     GET GRID FILE NAMES
      WRITE(1,100)
  100 FORMAT(1X,12HFILE NAMES: ,1H$)
      READ(1,107) (GFILE(I),I=1,3)
  107 FORMAT(3A2)
C
C     GET DESIRED NOISE LEVELS TO PLOT
      WRITE(1,103)
  103 FORMAT(1X,19HNOISE LEVELS (F5): ,1H$)
      READ(1,104) (XDB1(I),I=1,32,2)
  104 FORMAT(16F5.1)
C
C     COMPUTE NUMBER OF NOISE VALUES TO CONTOUR
      DO 2 I=1,32,2
      IF(XDB1(I) .LT. .5) GO TO 3
    2 NVALU=NVALU+1
    3 DO 1 I=1,NVALU
      II=I*2
    1 XDB1(II)=XDB1(II-1)
      XDB1(II+1)=XDB1(II)
```

```
C       PROGRAM PLOT7


      KVAL=2*NVALU+1
      DO 330 I=1,KVAL
      XDB(I)=XDB1(I)
 330  CONTINUE
C
C      GET PLOT TITLE
      WRITE(1,105)
  105 FORMAT(1X,7HTITLE: ,1H$)
      READ(1,106) (KTITLE(I),I=1,40)
  106 FORMAT(40A2)
C
C      SET UP PLOTTING FRAME
10     CALL CONTRL
C
C      DO THE CONTOURING
       CALL WORK
       GO TO 10
C
       END
C
C
       SUBROUTINE CONTRL
C
C THIS ROUTINE SETS UP A FRAME FOR CONTROL OF CERTAIN PARAMETERS
C AND PLOTTING OF AXES, RUNWAYS, ETC.
C
C COMMON FOR OUTPUT MODULE OF MOD7
C
       REAL J1
       INTEGER AFILE,GFILE,TITLE,DUMFIL(3)
       COMMON PSYMB
       COMMON/ZCOM/Z(2500),ZZ(2500),NUM
       COMMON/XFMCOM/XO,YO,SF(3),UO(3),VO(3)
       COMMON/GRDCOM/NR,NC,DX,DY,X1,Y1,TAREA
       COMMON/MISCEL/RLAT,RLONG,NTYPE
       COMMON/FILES/TITLE(40),AFILE(3),GFILE(3)
       COMMON//NSEG,XDB(33),IUSE,KTITLE(40),NVALU,XXAREA(16),
      *            YYAREA(16)
       INTEGER PLTYP,PCKPTR,DEV(3),FLG1
       LOGICAL INTFLG,PRCFLG
       DATA INTFLG,PRCFLG/.FALSE.,.FALSE./
       DATA IBLNK/2H  /,ISW1/0/
C
       PLTYP=0
       DO 40 I=1,3
       DEV(I)=PLTYP
 40    CONTINUE
       GO TO 100
```

TABLE 8. PLOT-7 PROGRAM (CONT'D) .

```
C       PROGRAM PLOT7


80     CONTINUE
C
C GET AN ITEM
100    CALL PRESET(ITEM)
       IF(ITEM.EQ.0) GO TO 100
       GO TO (110,120,120,130,140,140,150,160,170,180,190,
     X        200,210),ITEM
C
C -QUIT-
110    PRINT 111
111    FORMAT(' STOPPED AT LINE 110')
       STOP
C
C -PLOT,RUNWAYS-
120    PLTYP=1
       GO TO 80
C
C -RUN + PATHS (RUNWAY EXTENSIONS)
130    PLTYP=2
       GO TO 80
C
C -DEVICE,CAL-COMP-
140    DEV(1)=PLTYP
       GO TO 80
C
C -ARDS-    (NOT USED)
  150 GO TO 80
C
C -BOTH-
160    DEV(1)=PLTYP
       DEV(3)=PLTYP
       GO TO 80
C
C -NEW GRID-
170      ISW1=ISW1+1
C
C      SET UP FOR PLOTTER ONLY
       DEV(1)=1
       DEV(2)=0
       DEV(2)=0
C
C      DO NOT OPEN AND READ IF ALREADY DONE
       IF(INTFLG) GO TO 80
       INTFLG=.TRUE.
C
C      OPENTHE GRID FILE AND READ IN THE INFORMATION
       CALL SEARCH(1,GFILE,1,0)
         READ(5,9006) (AFILE(I), I=1,3)
```

TABLE 8. PLOT-7 PROGRAM (CONT'D).

```
C       PROGRAM PLOT7


9006      FORMAT(3A2)
          READ(5,9006) (DUMFIL(I), I=1,3)
          READ(5,9002) NTYPE
9002      FORMAT(I2)
          READ(5,9007) NC,NR,DX,DY,X1,Y1,J1
9007      FORMAT(I4/I4/2F10.2/3F10.2)
          NPNTS=NC*NR
          READ(5,9903) (Z(J),J=1,NPNTS)
9903      FORMAT(7E10.3)
        CALL SEARCH(4,0,1)
C
C       RESET COLUMN AND ROW COUNTS FOR CONTUR
        NC=NC-2
        NR=NR-2
C
C       GET SCALE FACTOR FOR PLOT
          WRITE(1,9000)
9000        FORMAT(1X,28HAIRPORT UNITS/PLOTTER INCH: ,1H$)
        READ(1,9001) SF(1)
9001    FORMAT(F10.0)
        IF(SF(1).NE.0.0) SF(1)=1.0/SF(1)
C
C       SET UP SYMBOL SCALING FACTOR
        PSYMB=1200.*SF(1)
        FLG1=0
        IF(SF(1).NE.0.0) FLG1=1
C
C       INITIALIZE PLOTTING SYSTEM
        CALL PLOTS(0,0,1)
        CALL PLOT(0.,-40.,-3)
        CALL XFMPRM(FLG1)
C
C       COMPUTE TOTAL AREA IN GRID
        TAREA=(FLOAT(NC-1)*DX)*(FLOAT(NR-1)*DY)
        GO TO 80
C
C--AXES-
180     IF(.NOT.INTFLG) GO TO 80
        RLNTH=IFIX((FLOAT(NC-1)*DX)*SF(1)+0.5)
C
C       GET LENGTH OF SYMBOL STRING FOR TITLE OF PLOT
        JJ=0
        DO 777 II=1,39
        IF(KTITLE(II) .EQ. IBLNK.AND. KTITLE(II+1).EQ.IBLNK) GO TO 778
        JJ=II
  777   CONTINUE
C
C       COMPUTE START POINT OF SYMBOL STRING
```

TABLE 8. PLOT-7 PROGRAM (CONT'D).

```
C       PROGRAM PLOT7


  778 XXL=FLOAT(JJ)*PSYMB
      FINDNT=(RLNTH/2.)+.5-XXL
C
C     RESET ORIGIN AND DRAW AXES AND TITLE
        CALL PLOT(-(UO(1)-2.5),-(VO(1)-2.5),-3)
      CALL AXIS(UO(1),VO(1)-0.5,2H  ,-2,RLNTH,0.0,XO,1.0/SF(1))
      RLNTH=IFIX((FLOAT(NR-1)*DY)*SF(1)+0.5)
      CALL AXIS(UO(1)-0.5,VO(1),2H  ,2,RLNTH,90.0,YO,1.0/SF(1))
      DO 779 II=1,2
      CALL SYMBOL(UO(1)-.5+FINDNT,VO(1)+RLNTH+1.,PSYMB,KTITLE,0.,JJ*2)
      FINDNT=FINDNT+.01
  779 CONTINUE
      GO TO 80
C
C -PRINT-
190   GO TO 80
C
C -CONTOUR-
200   IF(.NOT.INTFLG) GO TO 80
      RETURN
C
C -PROCESS-
210   IF(.NOT.INTFLG) GO TO 80
      IF(.NOT.PRCFLG) DEV(2)=PLTYP
      PRCFLG=.TRUE.
      CALL AIRDRW(DEV)
      DO 220 I=1,3
      DEV(I)=0
220   CONTINUE
      GO TO 80
C
      END
C
C
      SUBROUTINE WORK
C
C THIS ROUTINE DOES THE CONTOURING
C
C COMMON FOR OUTPUT MODULE OF MOD7
C
      INTEGER AFILE,GFILE,TITLE
      COMMON PSYMB
      COMMON/ZCOM/Z(2500),ZZ(2500),NUM
      COMMON/XFMCOM/XO,YO,SF(3),UO(3),VO(3)
      COMMON/GRDCOM/NR,NC,DX,DY,X1,Y1,TAREA
      COMMON/MISCEL/RLAT,RLONG,NTYPE
      COMMON/FILES/TITLE(40),AFILE(3),GFILE(3)
      COMMON//NSEG,XDB(33),IUSE,KTITLE(40),NVALU,XXAREA(16),
```

TABLE 8. PLOT -7 PROGRAM (CONT'D).

```
C       PROGRAM PLOT7


     *            YYAREA(16)
C
       INTEGER CNTRL(3),NOISE(2,2)
       DIMENSION PTS(4),PTX(2),PTY(2)
       LOGICAL ARFLG
       DATA ARFLG/.FALSE./,ISW2/0/
       DATA NOISE(1,1),NOISE(1,2),NOISE(2,1),NOISE(2,2)
      X     /2HAB,2HS ,2HEP,2HNL/
C
C SET-UP DEFAULT VALUES
          ISW1=1
       CNTRL(1)=1
       CNTRL(3)=0
140    CNTRL(2)=2
C
C GET SCALE VALUE
  150 IF(ISW1 .NE. 1) GO TO 200
       ISW2=ISW2+1
       IF(MOD(ISW2,NVALU*2+2) .EQ. 0) ISW2=1
       CLVL=XDB(ISW2)
C
C CHECK FOR AN ITEM
200    CALL PRESET(ITEM)
       IF(ITEM.EQ.0) GO TO 150
       ISW1=0
       GO TO (220,240,260,260,280,300,300,320,340,360),ITEM
C
C -QUIT-
C      HERE FOR NORMAL EXIT
220        CALL PLOT(XAREA+6.,0.,-3)
       PRINT 221
 221   FORMAT(' END OF EXECUTION')
         CALL EXIT
C
C -RETURN-  TO CONTROL SUBR.
240    RETURN
C
C -PLOT,CONTOUR-
260    CNTRL(1)=1
       GO TO 150
C
C -CONTOUR + AREA-
280    CNTRL(1)=2
       GO TO 150
C
C -DEVICE,PLOTTER-
300    CNTRL(2)=1
       GO TO 150
```

TABLE 8. PLOT-7 PROGRAM (CONT'D).

```
C        PROGRAM PLOT7


C
C -ARDS-
  320 GO TO 150
C
C -BOTH-
340    CNTRL(2)=4
       GO TO 150
C
C -PROCESS-
  360 ISW1=1
       IF(CNTRL(2).NE.2) GO TO 500
       CALL CUNTUR(CLVL,AREA)
       IF(NUM .LE. 0) GO TO 150
       CNTRL(3)=1
       GO TO 140
C
C CONTOUR ON PLOTTER
 500   IF(CNTRL(3).EQ.0) GO TO 140
         IF(NUM .LE. 0) GO TO 776
         IF(NUM .GT. 625) NUM=625
         CALL CONCAT(ZZ,NUM,10.)
C
C      PLOT THE CONCATENATED STRING
       DO 520 I=1,NUM
         M=4*I-3
         PTX(1)=ZZ(M)
         PTY(1)=ZZ(M+1)
         PTX(2)=ZZ(M+2)
         PTY(2)=ZZ(M+3)
         IF(I .NE. 1) GO TO 510
         XOLD=PTX(1)-20.
         YOLD=PTY(1)-20.
510      DISQ=(PTX(1)-XOLD)**2+(PTY(1)-YOLD)**2
         XOLD=PTX(2)
         YOLD=PTY(2)
         IF(DISQ .LT. 100.) GO TO 515
          CALL DRAW(1,2,PTX,PTY)
         GO TO 520
515      CALL DRAW(1,1,PTX,PTY)
 521   FORMAT(5F10.1)
520      CONTINUE
       AREA1=(AREA/TAREA)*100.0
       AREA2=AREA*3.587E-8
       IF(CNTRL(1).EQ.1) GO TO 140
       IF(ARFLG) GO TO 530
       ARFLG=.TRUE.
C
C      HERE TO PLOT SUMMARY TABLE HEADER
```

TABLE 8. PLOT-7 PROGRAM (CONT'D).

```
C       PROGRAM PLOT7


      PSYMB=PSYMB*0.65
      XAREA=(X1+FLOAT(NC-1)*DX-X0)*SF(1)+U0(1)+1.5
      YAREA=(Y1-Y0)*SF(1)+V0(1)-0.2
        TITLE(1)=2HLE
        TITLE(2)=2HVE
      TITLE(3)=2HL(
      CALL SYMBOL(XAREA,YAREA,PSYMB,TITLE,0.0,6)
      CALL SYMBOL(XAREA+6.*PSYMB,YAREA,PSYMB,NOISE(NTYPE,1),0.0,2)
      CALL SYMBOL(XAREA+PSYMB*8.,YAREA,PSYMB,NOISE(NTYPE,2),0.0,2)
      TITLE(1)=2H)
        TITLE(4)=2H%A
        TITLE(5)=2HRE
      TITLE(6)=2HA
        TITLE(2)=2H
      TITLE(3)=2H
      CALL SYMBOL(XAREA+10.*PSYMB,YAREA,PSYMB,TITLE,0.0,12)
        TITLE(1)=2H
      TITLE(2)=2HAR
      TITLE(3)=2HEA
      TITLE(4)=2H(S
      TITLE(5)=2HQ.
      TITLE(6)=2HMI
      TITLE(7)=2H.)
      CALL SYMBOL(XAREA+21.*PSYMB,YAREA,PSYMB,TITLE,0.0,14)
      NSEG1=((NSEG-1)*((NVALU*4)+3))+3
      DO 777 II=1,NVALU
      XXAREA(II)=0.
  777 YYAREA(II)=0.
      YAREA=YAREA-2.*PSYMB
      XAREA=XAREA+PSYMB*3.
C
C       PLOT SUMMARY NUMBERS
  530 III=ISW2/2
      XXAREA(III)=XXAREA(III)+AREA1
      YYAREA(III)=YYAREA(III)+AREA2
  776 IF(IUSE .LT. NSEG1) GO TO 140
      III=ISW2/2
      CALL NUMBER(XAREA,YAREA,PSYMB,CLVL,0.0,2)
        CALL NUMBER(XAREA+PSYMB*13.,YAREA,PSYMB,XXAREA(III),0.0,2)
        CALL NUMBER(XAREA+22.*PSYMB,YAREA,PSYMB,YYAREA(III),0.0,3)
      YAREA=YAREA-1.5*PSYMB
      GO TO 140
C
      END
C
C
      SUBROUTINE XFMPRM(FLG)
C
```

TABLE 8. PLOT-7 PROGRAM (CONT'D)

```
C       PROGRAM PLOT7


C COMPUTES TRANSFORMATION PARAMS.
C  IF FLG=1 DON'T DO IT FOR PLUTTER
C
C COMMON FOR OUTPUT MODULE OF MOD7
C
      INTEGER TITLE,AFILE,GFILE
      COMMON/ZCOM/Z(2500),ZZ(2500),NUM
      COMMON/XFMCOM/XO,YO,SF(3),UO(3),VO(3)
      COMMON/GRDCOM/NR,NC,DX,DY,X1,Y1,TAREA
      COMMON/MISCEL/RLAT,RLONG,NTYPE
      COMMON/FILES/TITLE(40),AFILE(3),GFILE(3)
      INTEGER FLG
      REAL DEVSZE(3),ULL(3),VLL(3)
      DATA DEVSZE(1),DEVSZE(2),DEVSZE(3)/10.,700.,1000./
      DATA ULL(1),ULL(2),ULL(3)/0.,10.,-500./,
     X     VLL(1),VLL(2),VLL(3)/0.,300.,-500./
C
C COMPUTE LARGEST SIDE OF TOTAL GRID
      XO=X1
      YO=Y1-DY*FLOAT(NR-1)
      SZE=AMAX1(DX*FLOAT(NC-1),DY*FLOAT(NR-1))
C
C TAKE CARE OF PLOTTER (PLTINT AT AIRPORT ORIGIN)
      IF(FLG.EQ.0) SF(1)=DEVSZE(1)/SZE
      UO(1)=XO*SF(1)
      VO(1)=YO*SF(1)
C
      DO 10 I=2,3
      SF(I)=DEVSZE(I)/SZE
      UO(I)=ULL(I)
      VO(I)=VLL(I)
10    CONTINUE
C
      RETURN
      END
C
C
      SUBROUTINE AIRDRW(DEV)
C
C DRAWS RUNWAYS (AND EXTENSIONS) ON DEVICES
C
      INTEGER DEV(3)
C COMMON FOR OUTPUT MODULE OF MOD7
C
      INTEGER AFILE,GFILE,TITLE
      COMMON/ZCOM/Z(2500),ZZ(2500),NUM
      COMMON/XFMCOM/XO,YO,SF(3),UO(3),VO(3)
      COMMON/GRDCOM/NR,NC,DX,DY,X1,Y1,TAREA
```

TABLE 8. PLOT-7 PROGRAM (CONT'D).

```
C       PROGRAM PLOT7


      COMMON/MISCEL/RLAT,RLONG,NTYPE
      COMMON/FILES/TITLE(40),AFILE(3),GFILE(3)
      INTEGER NAME(2)
      REAL X(3),Y(3)
C
C     OPEN AIRPORT FILE AND READ IN THE INFO
      CALL SEARCH(1,AFILE,1,1000)
        READ(5,9000) (TITLE(I),I=1,40)
9000  FORMAT(40A2)
      READ(5,9001) RLAT,RLONG
9001  FORMAT(2F15.0)
      READ(5,9002) NRUNS
9002  FORMAT(I10)
      IF(NRUNS.LE.0) GO TO 30
C
C     HERE TO DRAW RUNWAYS
      DO 20 I=1,NRUNS
      READ(5,9009) (NAME(J),J=1,3)
 9009 FORMAT(3A2)
      READ(5,9001) (X(J),Y(J),J=1,2)
      READ(5,9001) WDTH
      IF(DEV(1).NE.0) CALL RWYDRW(1,X(1),Y(1),X(2),Y(2),WDTH)
20    CONTINUE
C
C     CHECK FOR RUNWAY EXTENSIONS TO BE DRAWN
30    DO 35 I=2,3
      DEV(I)=0
      IF(DEV(I).EQ.2) GO TO 45
35    CONTINUE
  40  CALL SEARCH(4,0,1)
      RETURN
C
C     HERE FOR RUNWAY EXTENSIONS
45    READ(5,9002) NPATHS
      IF(NPATHS.LE.0) GO TO 40
      DO 60 I=1,NPATHS
      READ(5,9000) (NAME(J),J=1,2)
      READ(5,9002) NSEGS
      IF(NSEGS.LE.0) GO TO 60
      DO 70 I1=1,NSEGS
      READ(5,9001) TYPE
      READ(5,9001) (X(J),Y(J),J=1,3)
      IF(DEV(1).NE.2) GO TO 50
      CALL EXTDRW(1,TYPE,X,Y,SF(1))
50    CONTINUE
70    CONTINUE
60    CONTINUE
      GO TO 40
```

TABLE 8. PLOT-7 PROGRAM (CONT'D).

```
C       PROGRAM PLOT7


C
1000  WRITE(1,9003) AFILE
9003  FORMAT(1X,10HNO FILE = ,3A2)
      STOP
        END
C
C
      SUBROUTINE EXTDRW(DEV,C,X,Y,SF)
C
C DRAWS AN EXTENSION  C=(0=ST. LINE,-1=CW,+1=CCW)
C
      INTEGER DEV
      REAL X(3),Y(3),U(2),V(2)
      U(1)=X(1)
      V(1)=Y(1)
      IF(C.LT.-0.5 .OR. C.GT.0.5) GO TO 100
C
C STRAIGHT LINE
      U(2)=X(2)
      V(2)=Y(2)
      CALL DRAW(DEV,2,U,V)
      GO TO 300
C
C HELIX
100   ARCLTH=0.1/SF
      DTHTA=ARCLTH/SQRT((X(1)-X(3))**2+(Y(1)-Y(3))**2)
      CDT=COS(DTHTA)
      CSDT=C*SIN(DTHTA)
150   DU=U(1)-X(3)
      DV=V(1)-Y(3)
      U(2)=CDT*DU-CSDT*DV+X(3)
      V(2)=CSDT*DU+CDT*DV+Y(3)
      CALL DRAW(DEV,2,U,V)
      U(1)=U(2)
      V(1)=V(2)
C
C TEST FOR DONE
      DNOW=(U(1)-X(2))**2+(V(1)-Y(2))**2
      IF(DNOW.GT.ARCLTH**2) GO TO 150
C
C DO LAST SEGMENT
      U(2)=X(2)
      V(2)=Y(2)
      CALL DRAW(DEV,2,U,V)
C
300      RETURN
      END
C
```

TABLE 8. PLOT-7 PROGRAM (CONT'D).

```
C       PROGRAM PLOT7


C
      SUBROUTINE RWYDRW(DEV,X1,Y1,X2,Y2,W)
C
C DRAWS A RUNWAY
C
      INTEGER DEV
      REAL U(2),V(2)
C
      DX=X2-X1
      DY=Y2-Y1
      XL=DX*DX+DY*DY
      IF(XL.LE.0.0) RETURN
      XL=SQRT(XL)
      DX=DX/XL
      DY=DY/XL
      DO 10 I=1,4
      DO 5 J=1,2
      X=FLOAT(MOD((I+J-1)/2,2))*XL
      Y=(FLOAT(MOD((I+3+J-1)/2,2))-0.5)*W
      U(J)=DX*X-DY*Y+X1
      V(J)=DY*X+DX*Y+Y1
5     CONTINUE
      CALL DRAW(DEV,2,U,V)
10    CONTINUE
      RETURN
      END
C
C
      SUBROUTINE DRAW(DEV,TYPE,X,Y)
C
C DRAWS A LINE(TYPE=2) OR CONCATENATED LINE(TYPE=1)
C DEVICE (1=PLOTTER)
C
      DIMENSION X(2),Y(2)
      INTEGER DEV,TYPE
      COMMON/XFMCOM/XO,YO,SF(3),UO(3),VO(3)
      INTEGER IU(2),IV(2),IDOT(2)
      DIMENSION U(2),V(2)
C
C DO INITIAL TRANSFORMATION
      DO 5 I=1,2
      U(I)=(X(I)-XO)*SF(DEV)+UO(DEV)
      V(I)=(Y(I)-YO)*SF(DEV)+VO(DEV)
5     CONTINUE
C
C
C -PLOTTER-
      GO TO (110,120),TYPE
```

TABLE 8. PLOT-7 PROGRAM (CONT'D).

```
C       PROGRAM PLOT7


120    CALL PLOT(U(1),V(1),3)
110    CALL PLOT(U(2),V(2),2)
       RETURN
       END
C
C
       SUBROUTINE CONTUR(H,AREA)
C
C THIS ROUTINE 'DRAWS' A CONTOUR
C
C ARGUMENTS:
C      H        THE VAL OF THE CONTOUR
C      Z        THE TWO-DIMENSIONAL ARRAY (GRID)
C      NR,NC    THE NUMBER OF ROWS,COLUMNS IN THE GRID
C      DX,DY    THE GRID BOX SIZE(USER UNITS)
C      XS,YS    THE COORDINATES OF UPPER-LEFT CORNER OF GRID
C
       INTEGER AFILE,GFILE,TITLE
       COMMON/ZCOM/Z(2500),ZZ(2500),NUM
       COMMON/GRDCOM/NR,NC,DX,DY,X1,Y1,TAREA
         COMMON/FILES/TITLE(40),AFILE(3),GFILE(3)
       REAL C(4,3)
       INTEGER IO(2),JO(2)
C
C WRITE NUMBER OF LINE SEGMENTS ON FILE
       NUM=0
C
C INITIALIZE THE AREA UNDER CONTOUR
       AREA=0.0
C
C LOOP THROUGH ROW STRIPS OF GRID
       DO 1000 ING=2,NR
       I=ING-1
C
C LOOP THROUGH THE RECTANGLES IN THE I-TH ROW
       DO 1000 JJ=2,NC
C
C ALTERNATE LEFT/RIGHT SEQUENCING
       J=JJ
       IF(MOD(I,2).NE.0) J=NC-JJ+2
       J=J+1
C
C COMPUTE COORDINATES OF THE CORNERS OF THE RECTANGLES
       DO 50 L=1,4
       J2=J+MOD(L/2,2)-1
       I2=I+MOD((L+3)/2,2)-1
       C(L,1)=X1+DX*FLOAT(J2-1)-DX
       C(L,2)=Y1-DY*FLOAT(I2-1)+DY
```

TABLE 8. PLOT-7 PROGRAM (CONT'D).

```
C       PROGRAM PLOT7


      C(L,3)=Z2(Z,NR+2,I2,J2)
50    CONTINUE
C
C COMPUTE CONTOUR SEGMENTS IN THIS RECTANGLE
      CALL CRECT(C,H,AREA,I,J)
C
C CONTINUE LOOPING THROUGH RECTANGLES
1000  CONTINUE
      RETURN
      END
C
C
      REAL FUNCTION Z2(Z,NR,I,J)
C
C THIS FUNCTION RETURNS THE RIGHT VALUE FROM A ONE-DIMENSIONAL
C ARRAY AS IF IT WERE A  TWO-DIMENSIONAL ARRAY WITH NR ROWS.
      REAL Z(1)
      K=I+NR*(J-1)
      Z2=Z(K)
      RETURN
      END
C
C
      SUBROUTINE CRECT(C,CLVL,AREA,IRW,JCL)
        COMMON/ZCOM/Z(2500),ZZ(2500),NUM
      COMMON/GRDCOM/NR,NC,DX,DY,X1,Y1,TAREA
C
C THIS ROUTINE COMPUTES AND SAVES THE CONTOUR SEGMENTS WITHIN
C A SINGLE GRID RECTANGLE, USING THE AVERAGE CENTER METHOD.
C
C ARGUMENTS
C     C(I,J)      GRID COORDINATES.
C                      I=CORNER NO. (1=UPPER LEFT, REST GO CW)
C                      J=X,Y,Z OF I-TH CORNER
C     CLVL         CONTOUR LEVEL SOUGHT
C     NUM         RUNNING TALLY OF LINE SEGMENTS
C     AREA        RUNNING TALLY OF AREA UNDER CONTOUR
C
      REAL C(4,3),V(3,3),P1(3),P2(3),PT1(2,4),PT2(2,4)
      LOGICAL FLG
C
C GET COORDINATES OF CENTER POINT OF GRID
      V(1,2)=(C(1,1)+C(2,1))/2.0
      V(2,2)=(C(2,2)+C(3,2))/2.0
      V(3,2)=(C(1,3)+C(2,3)+C(3,3)+C(4,3))/4.0
C
C SET FLAG FOR FIRST TIME THROUGH RECTANGLE
      FLG=.FALSE.
```

TABLE 8. PLOT-7 PROGRAM (CONT'D).

```
C       PROGRAM PLOT7


C
C SET AREA INCREMENT TO ZERO FOR THIS RECTANGLE
20      TMP1=0.0
        TMP2=0.0
C
C INITIALIZE INTERSECTION COUNTER
        NINTR=0
C
C LOOP THROUGH THE FOUR TRIANGLES IN THIS RECTANGLE
        DO 100 K=5,8
C
C GET LEADING CORNER SUBSCRIPT
        IL=MOD((K-1),4)+1
C
C GET TRAILING CORNER SUBSCRIPT
        IT=MOD((K-2),4)+1
C
C GET CORNER COORDINATES FOR STRIKE
        DO 30 I=1,3
        V(I,1)=C(IL,I)
        V(I,3)=C(IT,I)
30      CONTINUE
C
C COMPUTE INTERSECTION
        CALL STRIKE(V,CLVL,P1,P2,TMP1,TMP2,IER)
          AREA=AREA+TMP1
C
C CHECK FOR NO INTERSECTION
        IF(IER.NE.2) GO TO 100
C
C THERE IS GOOD INTERSECTION, INCREMENT COUNTER,SAVE ENDS
50          NINTR=NINTR+1
        DO 55 I=1,2
        PT1(I,NINTR)=P1(I)
        PT2(I,NINTR)=P2(I)
55      CONTINUE
C
100     CONTINUE
C
C CHECK FOR SPECIAL CASE.
        IF(FLG.OR.(NINTR.NE.4)) GO TO 200
C
C SPECIAL CASE
        FLG=.TRUE.
        DELY=C(2,3)-C(4,3)
        DIFF1=ABS(Z2(Z,NR+2,IRW+1,JCL-2)-(C(4,3)-DELY))
       X     +ABS(Z2(Z,NR+2,IRW-2,JCL+1)-(C(2,3)+DELY))
        DELY=C(3,3)-C(1,3)
```

TABLE 8. PLOT-7 PROGRAM (CONT'D).

```
C       PROGRAM PLOT7


      DIFF2=ABS(Z2(Z,NR+2,IRW-2,JCL-2)-(C(1,3)-DELY))
     X     +ABS(Z2(Z,NR+2,IRW+1,JCL+1)-(C(3,3)+DELY))
      IF(DIFF1.LT.DIFF2) GO TO 120
      V(3,2)=(C(1,3)+C(3,3))/2.0
      GO TO 20
120   V(3,2)=(C(4,3)+C(2,3))/2.0
      GO TO 20
C
C DONE
200     IF(NINTR.LE.0) RETURN
        M=4*NUM+1
        NUM=NUM+NINTR
        IF(NUM .GT. 625) RETURN
        DO 210 I=1,NINTR
        ZZ(M)=PT1(1,I)
        ZZ(M+1)=PT1(2,I)
        ZZ(M+2)=PT2(1,I)
        ZZ(M+3)=PT2(2,I)
        M=M+4
210     CONTINUE
      RETURN
      END
C
C
      SUBROUTINE STRIKE(V,C,P1,P2,AA,AB,N)
C
C THIS ROUTINE COMPUTES THE INTERSECTION OF A 3-DIMENSIONAL
C TRIANGLE AND AN INFINITE HORIZONTAL PLANE.
C
C ARGUMENTS:
C     V(I,J)      AN ARRAY CONTAINING THE COORDINATES OF THE
C                 TRIANGLE. (I=X,Y,Z     J=CORNER NO.)
C     C           THE HEIGHT OF THE HORIZONTAL PLANE
C     P1,P2       THREE-VECTORS CONTAINING'THE ENDPOINTS OF THE
C                 LINE OF INTERSECTION (IF ANY)
C     AA          AREA ON OR ABOVE CONTOUR.
C     AB          AREA ON OR BELOW CONTOUR.
C     N           RETURN CODE
C                    N=0   NO INTERSECTION
C                    N=1   INTERSECT AT ONE NODE
C                    N=2   INTERSECT AT 2 POINTS, RETURNING LINE SEGMENT
C                    N=3   INTERSECT AT 3 POINTS, TRIANGLE + PLANE COINCI
C
       DIMEOSION V(3.3),P1(3),P2(3),X(3),Y(3),Z(3),XC(2),YC(2),A(2)
 333  FORMAT(' C=',F10.1)
C
C       TRANSFER COORDS OF CORNERS OF TRIANGLE TO TEMP. ARRAYS
        DO 10 J=1,3
```

TABLE 8. PLOT-7 PROGRAM (CONT'D).

```
C       PROGRAM PLOT7


         X(J)=V(1,J)
         Y(J)=V(2,J)
10       Z(J)=V(3,J)
C
C        SET COUNTER TO ZERO
         N = 0
C
C        SET INTERSECTING SIDE NO SUM AT ZERO
         ISS = 0
C
C        ASSUME TRIANGLE IN PLANE
         AA=ABS((X(2)-X(1))*(Y(3)-Y(1))-(X(3)-X(1))*(Y(2)-Y(1))) /2.
         AB = AA
C
C        LOOP FOR 3 SIDES
         DO 3 I=1,3
C
C        COMPUTE ENDPOINT SUBSCRIPTS
         J = I + 1. - (I/3)*3
C
C        COMPUTE SIGNED DISTANCES
         D1 = Z(I) - C
         D2 = Z(J) - C
C
C        CHECK FOR INTERSECTION WITH EDGE
         IF (D1 * D2) 2,1,3
C
C        INTERSECT AT ONE END, IS IT LEADING EDGE
1        IF (ABS(D2) .GT. .0001) GO TO 3
C
C        INCREMENT COUNTER
2        N = N + 1
C
C        CHECK IF OUTPUT ARRAY FULL
         IF(N .GE. 3) RETURN
C
C        COMPUTE INTERSECTION PROPORTIONALITY
         F = 1.0
         IF(ABS(D1-D2) .GT. .0001) F = D1/(D1-D2)
C
C        COMPUTE INTERSECTION COORDS
         XC(N) = X(I) + (X(J)-X(I)) * F
         YC(N) = Y(I) + (Y(J)-Y(I)) * F
C
C        ACCUMULATE SIDE SUM
         ISS = ISS + 1
C
3        CONTINUE
```

TABLE 8, PLOT-7 PROGRAM (CONT'D).

```
C       PROGRAM PLOT7


C
C         SET SMALL TRIANGLE AREA EQUAL TO TOTAL AREA
          AT = AA
C
C         BRANCH ON INTERSECTION MODE
          IF (N-1) 4,5,6
C
C         NO INTERSECTIONS - CHOOSE FIRST VERTEX FOR ABOVE/BELOW TEST
4         K = 1
          GO TO 7
C
C         OSCUMATION - CHOOSE ANY NON-OSCULATING VERTEX FOR ABOVE/BELOW TE
5         K = ISS
          GO TO 7
C
C         TWO INTERSECTION POINTS - CHOOSE THE INCLUDED VERTEX FOR
C         ABOVE/BELOW TEST
6         K = 8 - ISS - ((7-ISS) / 3) * 3
C
C         COMPUTE SMALL TRIANGLE AREA
          AT=ABS((XC(1)-X(K))*(YC(2)-Y(K))-(XC(2)-X(K))*(YC(1)-Y(K)))/2.
C
C         DETERMINE ABOVE/BELOW TEST
7         I = 1
          IF (Z(K) .LT. C) I = 2
C
C         COMPUTE TEMPORARY AREA ABOVE CONTOUR
          A(I) = AT
C
C         COMPUTE TEMPORARY AREA BELOW CONTOUR
          K = 3 - I
          A(K) = AA - AT
C
C         TRANSFER RESULTS TO OUTPUT ARGUMENTS
          P1(1) = XC(1)
          P1(2) = YC(1)
          P2(1) = XC(2)
          P2(2) = YC(2)
          AA = A(1)
          AB = A(2)
C
C         BRANCH TO CALLING ROUTINE
          RETURN
          END
C
C
        SUBROUTINE PRESET(ITEM)
        COMMON//NSEG,XDB(33),IUSE,KTITLE(40),NVALU,XXAREA(16),
```

TABLE 8. PLOT-7 PROGRAM (CONT'D).

```
C       PROGRAM PLOT7


     *             YYAREA(16)
      DATA ITEM1/14/,KOUNT/0/,ITEM2/0/
      IUSE=IUSE+1
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14),ITEM1
   14 ITEM=9
      IUSE=1
    1 GO TO 16
    2 GO TO 17
    3 ITEM=6
    4 GO TO 16
    5 ITEM=7
      GO TO 16
    6 ITEM=13
      GO TO 16
    7 ITEM=10
    8 GO TO 16
    9 ITEM=12
      IF(IUSE .EQ. 2) GO TO 7
      IF((IUSE .EQ. 3) .AND. (ITEM2 .NE. 13)) ITEM=3
      GO TO 16
   10 IF(IUSE .EQ. 3) GO TO 9
      ITEM=5
      IF((ITEM2 .EQ. 7) .AND. (KOUNT .NE. NVALU)) GO TO 7
      KOUNT=KOUNT+1
      IF(MOD(KOUNT,NVALU+1) .EQ. 0) GO TO 15
   11 GO TO 16
   12 GO TO 7
   13 IUSE=IUSE-4
   17 ITEM=9
      GO TO 16
   15 KOUNT=0
      ITEM=2
   16 ITEM2=ITEM1
      ITEM1=ITEM
      IF(IUSE .EQ. (NSEG*((NVALU*4)+3)+1)) ITEM=1
      RETURN
      END
C
C
      SUBROUTINE CONCAT (A,N,EPSI)
C
C     THIS ROUTINE CONCATENATES ADJACENT LINE SEGMENTS TAKEN FROM A
C     DISJOINT ARRAY.
C
C
C     A = VECTOR ARRAY OF SEGMENT END POINT COORDINATES.
C         EACH SEGMENT IS DEFINED BY 4 CONSECUTIVE ELEMENTS IN A.
C         THE FOUR ELEMENTS ARE:
```

TABLE 8. PLOT-7 PROGRAM (CONT'D).

```
C       PROGRAM PLOT7


C           1-X COORD OF END A
C           2-Y COORD OF END A
C           3-X COORD OF END B
C           4-Y COORD OF END B
C       N = NUMBER OF SEGMENTS IN A.  ACTUAL LENGTH OF A = 4 * N
C       EPSI = GAP CRITERION.  TWO ENDPOINTS WITHIN EPSI OF EACH OTHER ARE
C              CONSIDERED TO BE COINCIDENT.
C       NOTE: CONCATENATION IS DONE IN PLACE. ARRAY A ON OUTPUT
C             CONTAINS THE CONCATENATED STRING.
C
C
C       SUBPROGRAM REQ'D: FUNCTION ICIRC(I,N): A FUNCTION THAT RETURNS
C                         EQUIV LINEAR SUBSCRIPT FOR CIRCULAR SUBSCRIPT I
C                         IN CIRCULAR ARRAY OF LENGTH N.
C
        DIMENSION A(1)
C
C       COMPUTE GAP SQUARED CRITERION
        EPSQ=EPSI**2
C
C       NO SEGMENTS STORED IN TAIL YET
        LX = 0
C
C*****************************************************************************
C**** BEGIN PROCESSING A NEW STRING ***********************************************
C*****************************************************************************
C
C       ASSUME CCW ATTACHMENT TO FIRST END OF ATTACHMENT BLOCK SEGMENT.
  2     ISIDE = -1
        K2 = 1
C
C       CURRENT STRING STARTS WITH ONE ELEMENT
        LOCX = 1
C
C**** BEGIN PROCESSING A NEW ARM IN THIS STRING *************************
C
C       POINT TO ATTACHMENT BLOCK
  3     IBASE = N-LX
C
C**** LOOK FOR NEXT SEGMENT ATTACHED TO THIS ARM *************************
C
C       DETERMINE NUMBER OF UNSTRUNG ELEMENTS
  4     L = N-LX-LOCX
C
C       COMPUTE X-COORD SUBSCRIPT OF THE ATTACHMENT-END OF THE ATTACHMENT
C       BLOCK
        K3 = 4 * IBASE + K2 - 4
C
```

TABLE 8. PLOT-7 PROGRAM (CONT'D).

```
C      PROGRAM PLOT7


C      ARE ALL ELEMENTS STRUNG (NO GAP)
       IF (L.EQ.0) GO TO 52
C
C      LOOP THRU UNSTRUNG ELEMENTS
       DO 7 I=1,L
C
C      COMQUTE SUBSCRIPT OF UNSTRUNG BLOCK (X-COORD OF END A) UNDER TEST.
C
       K =  IBASE+I*ISIDE
       K = ICIRC(K,N-LX)*4-3
C
C      ASSUME THIS SEGMENT IS NOT CONTIGUOUS.
       NABR=0
C
C      COMQUTE DIST SQD FROM ATTACHMENT SEGMENT TO UNSTRUNG SEGMENT'S END
       DISQ=(A(K)-A(K3))**2+(A(K+1)-A(K3+1))**2
C
C      IS THIS SEGMENT END CONTIGUOUS IF SO FLAG
       IF (DISQ.LE.EPSQ) NABR=1
C
C      COMPUTE DIST SQD FOR END B
       DISQ=(A(K+2)-A(K3))**2+(A(K+3)-A(K3+1))**2
C
C      IS THIS SEGMENT END CONTIGUOUS IF SO FLAG
       IF (DISQ.LE.EPSQ) NABR=-1
C
C      IF CONTIGUOUS SEGMENT FOUND JUMP TO 20 OR 22
       IF (ISIDE*NABR)20,7,22
C
C      REPEAT FOR NEXT UNSTRUNG ELEMENT.
   7   CONTINUE
C
C**** LOOP COMQLETE - NO CONTIGUOUS SEGMENT ****************************
C**** END OF STRING IN THIS DIRECTION *********************************
C
C      CHECK IF THIS STRING COMPLETE
       IF (ISIDE.EQ.1) GO TO 8
C
C      REPEAT FOR CW MOTION
       ISIDE=1
       K2=3
       GO TO 3
C
C**** CONTIGUOUS SEGMENT FOUND - ATTACH TO ARM *************************
C
C      FLIP SEGMENT TO BE ATTACHED
  20        DO 21 I=1,2
            M=K+I-1
```

TABLE 8. PLOT-7 PROGRAM (CONT'D).

```
C       PROGRAM PLOT7


            DUM=A(M)
            A(M)=A(M+2)
  21        A(M+2)=DUM
C
C      DETERMINE WHICH UNSTRUNG SEGMENT IS ADJACENT TO THE ATTACHMENT BLO
  22   IBASE= ICIRC(IBASE+ISIDE,N-LX)
C
C      INCREASE CURRENT STRING SIZE BY 1
       LOCX=LOCX+1
C
C      EXCHANGE ADJACENT SEGMENT WITH CONTIGUOUS SEGMENT
            DO 23 I=1,4
            M=K+I-1
            J=IBASE*4+I-4
            DUM=A(M)
            A(M)=A(J)
  23        A(J)=DUM
C
C      LOOK FOR NEXT SEGMENT IN THIS STRING
       GO TO 4
C
C*********************************************************************
C**** STRING COMPLETE - PUT INTO TAIL ********************************
C*********************************************************************
C
C      IS THIS STRING ENTIRELY ON CCW SIDE
 8     IF (IBASE.EQ.N-LX) GO TO 52
C
C**** MOVE CW ARM TO FAR END OF CCW ARM ******************************
C
C      FLIP AND SHIFT CW VERTICES TO CLOSE GAP
C
C           GET NUMBER OF COORDS (ARRAY ELEMENTS) IN ARM.
            NN=4*IBASE
C           LOOP FOR ALL POINTS IN ARM.
            DO 50 I=1,NN,2
C           GET SUBSCRIPT OF ARRAY ELEMENT IMMEDIATELY PRECEEDING NEW
C           LOCATION OF POINT.
            JD=(N-LX-LOCX)*4+I-1
C           LOOP FOR BOTH X AND Y COORDS OF POINT.
            DO 50 M=1,2
C           COMPUTE OLD (J) AND NEW (JJ) SUBSCRIPTS.
            J = NN-I+M-1
            JJ=JD+M
C           INTERCHANGE ARRAY ELEMENTS.
            TEMP=A(J)
            A(J)=A(JJ)
  50        A(JJ)=TEMP
```

TABLE 8. PLOT-7 PROGRAM (CONT'D).

```
C       PROGRAM PLOT7


C
C      REVERSE CCW ELEMENTS
C           GET SUBSCRIPT OF ARRAY ELEMENT IMMEDIATELY PRECEEDING FIRST
C           SEGMENT OF CCW ARM.
            ND=ICIRC(IBASE-LOCX,N-LX)*4
C           COMPUTE NUMBER OF SEGMENTS IN CCW ARM
            NN=N-LX-ND/4
C           COMPUTE NUMBER OF COORDS (ARRAY ELEMENTS) IN HALF OF ARM.
            NN = 2 * NN
C           LOOP FOR ALL POINTS IN HALF OF ARM
            DO 51 I=1,NN,2
C           GET SUBSCRIPT OF ARRAY ELEMENT IMMEDIATELY PRECEEDING
C           CURRENT POINT IN BACKWARD LOOP.
            JD=4*(N-LX)-I-1
C           LOOP FOR BOTH X AND Y COORDS OF POINT.
            DO 51 M=1,2
C           COMPUTE FORWARD (J) AND BACKWARD (JJ) LOOP SUBSCRIPTS.
            J=ND+M+I-1
            JJ=JD+M
C           INTERCHANGE ARRAY ELEMENTS.
            TEMP=A(J)
            A(J)=A(JJ)
  51        A(JJ)=TEMP
C
C**** ARRANGE LATEST STRING TO MINIMIZE DISTANCE BETWEEN ***************
C**** ITS BUTTOM AND TOP OF PREVIOUS STRING *****************************
C
C      IS TAIL EMPTY
  52   IF (LX.EQ.0) GO TO 53
C
C      COMQUTE DISTS SQD TO BOTH ENDS OF LATEST STRING FROM END OF
C      PREVIOUS STRING
       JJ=4*(N-LX)+1
       J=4*(N-LX-LOCX)+1
       DISQ1=(A(JJ-2)-A(JJ))**2+(A(JJ-1)-A(JJ+1))**2
       DISQ2 = (A(J)-A(JJ))**2+(A(J+1)-A(JJ+1))**2
C
C      IS PRESENT ORIENTATION CORRECT
       IF (DISQ1.LE.DISQ2) GO TO 53
C
C      REVERSE CURRENT STRING
            LLX = 2 * LOCX
            DO 54 I=1, LLX, 2
            J1=J+I-2
            J2 = JJ-I-2
            DO 54 M=1,2
            J1=J1+1
            J2=J2+1
```

TABLE 8. PLOT-7 PROGRAM (CONT'D).

```
C      PROGRAM PLOT7


            TEMP=A(J1)
            A(J1)=A(J2)
  54        A(J2)=TEMP
C
C**** ADD CURRENT STRING TO TAIL ***************************************
C
  53  LX=LX+LOCX
      IF (N-LX.EQ.0) RETURN
C
C**********************************************************************
C**** START NEXT STRING BY PUTTING NEXT CLOSEST UNSTRUNG ELEMENT *******
C**** INTO POSIUION NEXT TO TAIL ***************************************
C**********************************************************************
C
      NN=4*(N-LX)
C
C**** FIND UNSTRUNG POINT CLOSEST TO TOP OF LATEST STRING **************
C
            IMIN=1
            DMIN=(A(1)-A(NN+1))**2+(A(2)-A(NN+2))**2
            DO 61 I=3,NN,2
            DSQ=(A(I)-A(NN+1))**2+(A(I+1)-A(NN+2))**2
            IF (DMIN.LE.DSQ) GO TO 61
            IMIN=I
            DMIN=DSQ
  61        CONTINUE
C
C**** DETERMINE WHICH SEGMENT THIS IS **********************************
      IMIN=((IMIN-1)/4)*4
C
C**** INTERCHANGE THIS WITH SEGMENT NEXT TO TAIL ***********************
C
            DO 62 I=1,4
            J=NN-4+I
            IMIN=IMIN+1
            TEMP=A(IMIN)
            A(IMIN)=A(J)
  62        A(J)=TEMP
C
C     BEGIN PROCESSING NEW STRING.
C
      GO TO 2
C
      END
C
C
      FUNCTION ICIRC (I,N)
C     I = CIRCULAR SUBSCRIPT.


C     N = LENGTH OF ARRAYS.
      ICIRC = MOD(I-1,N) + 1
      IF (ICIRC.LE.0) ICIRC = ICIRC + N
      RETURN
      END
```

TABLE 8. PLOT-7 PROGRAM (CONCLUDED)

APPENDIX A
BASIC DERIVATIONS FOR ASDS

ASDS (Aircraft Sound Description System) measures the time span that an observer is exposed to noise above a specified threshold, and is measured in seconds. The threshold is given in dBA, with 85 as a typical value.

TSC's MOD-7 program calculates ASDS in the following manner:

Given a description of an aircraft's flight path and an observer's location, a point of closest approach is found. Shown as point 1 in Figure A-1, it is separated from the observer

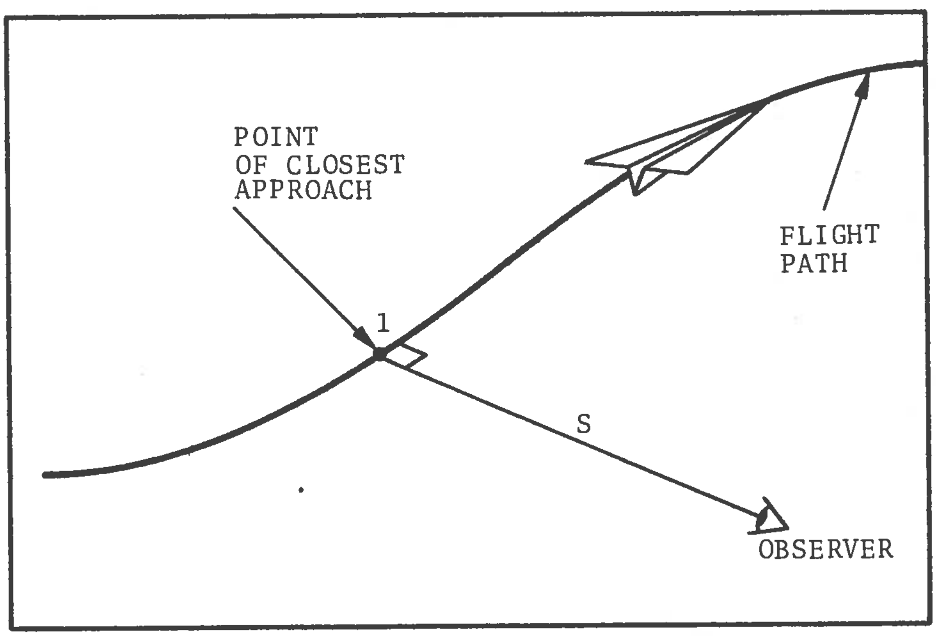


Figure A-1. Aircraft Flight Path

by distance S. S is usually orthogonal to the flight path at the point of closest approach.

To simplify computation, the flight path is idealized as a straight line, using a tangential approximation at point 1. This is shown in Figure A-2.

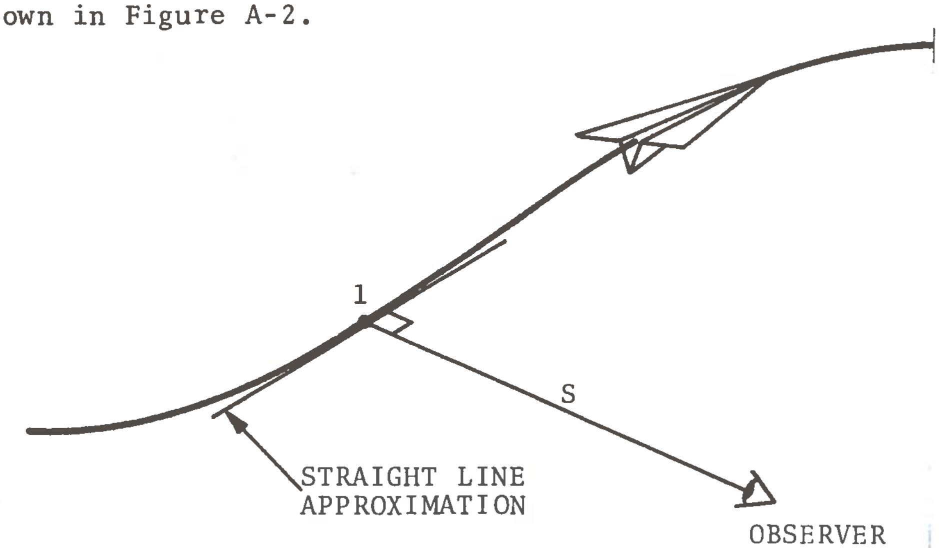


Figure A-2. Approximation of Aircraft Flight Path

Somewhere downstream of point 1 there is a point where the aircraft, approaching point 1, just exceeds the observer's audible threshold. This point is designated as point 2 in Figure A-3. Point 3 is a companion point upstream of 1 where the aircraft's noise recedes beneath the audible threshold.

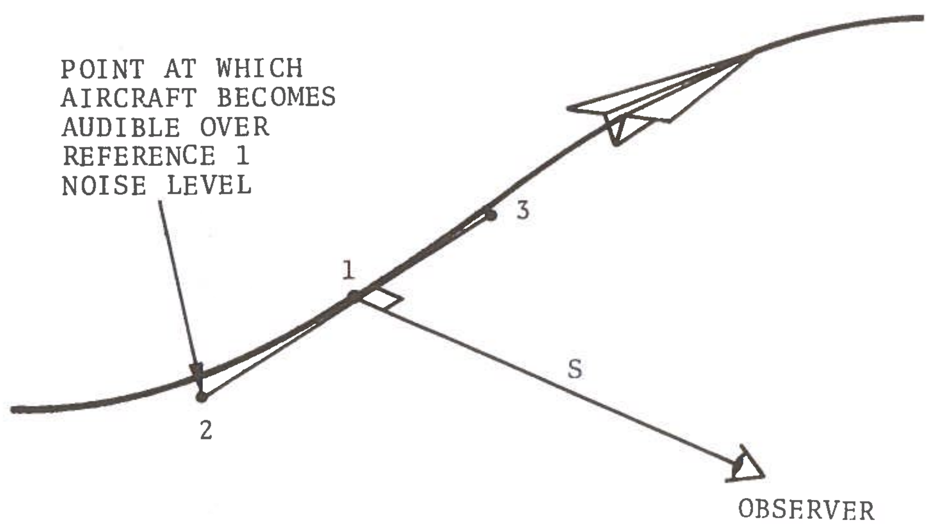


Figure A-3. Threshold Points with Respect to Reference Noise Level

Figure A-4 shows the distances from points 2 and 3 to the observer as R. The length from 2 to 3 is important in the computation of ASDS, and will be calculated from R.

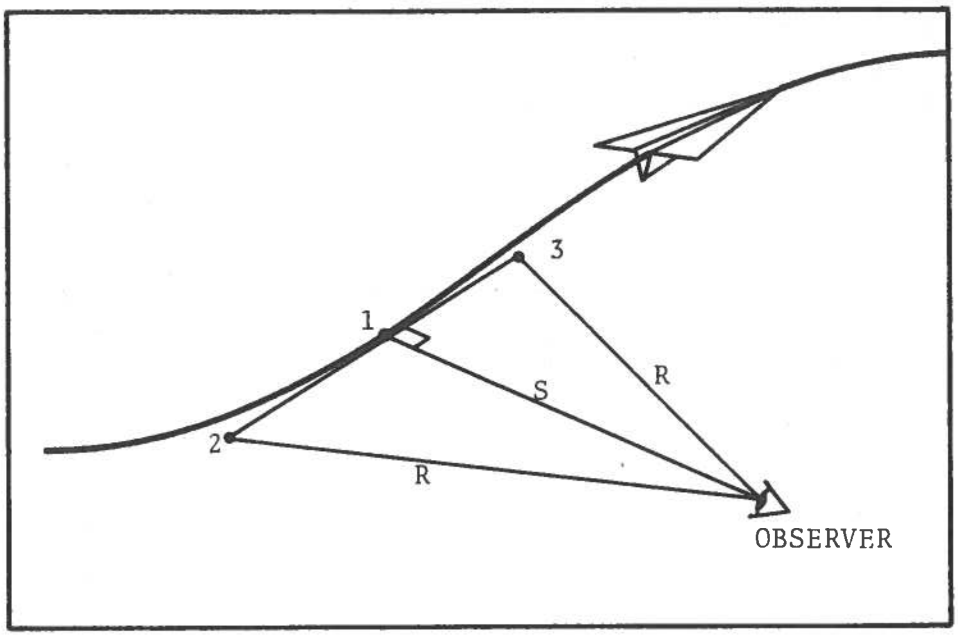


Figure A-4. Distances of Threshold Points to Observer

By using the aircraft's noise curve it is possible to calculate R. R is obtained by solving for the point at which the noise curve crosses the line of threshold noise level, correcting for attenuation and shielding effects. Figure A-5 depicts the process graphically. In the program, subroutine DGET is used for the same purpose. DGET uses a Newton-Raphson iterative method with a central difference approximation for the curve's derivative.

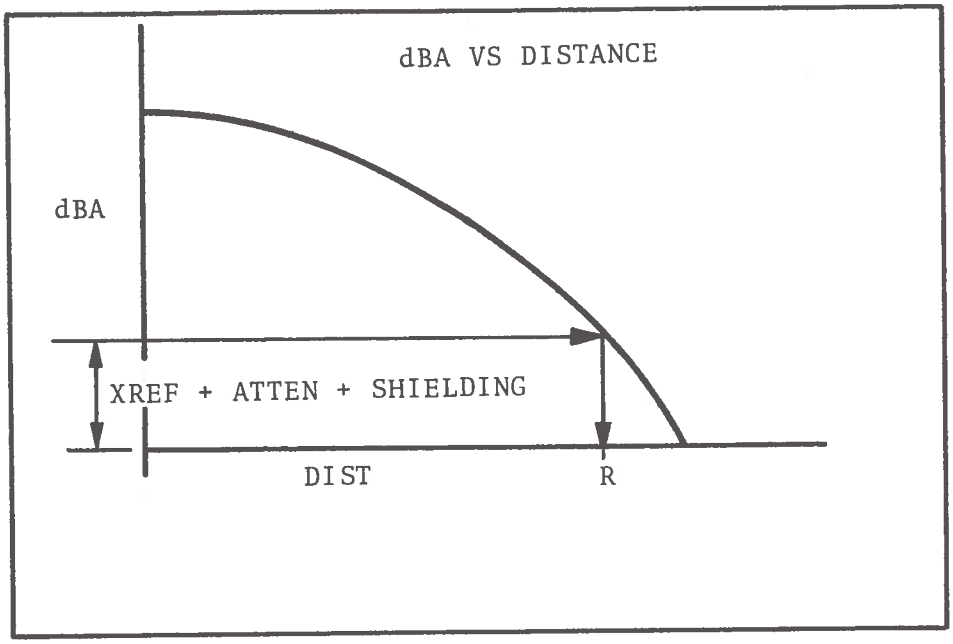


Figure A-5. Aircraft Noise Curve (dBA vs Distance)

Once R is known, the Pythagorean Theorem yields the distance the aircraft has traversed while above the threshold noise level. This distance, shown in Figure A-6, is expressed as

$$2\sqrt{R^2 - S^2} .$$

The time duration follows by dividing the distance by the plane's speed. Hence,

$$t = 2\sqrt{R^2 - S^2}/V .$$

When V is given in knots, a conversion factor of 1.689 knots/(ft/sec) is used. Thus, the formula becomes

$$t = 2\sqrt{R^2 - S^2}/(1.689V) .$$

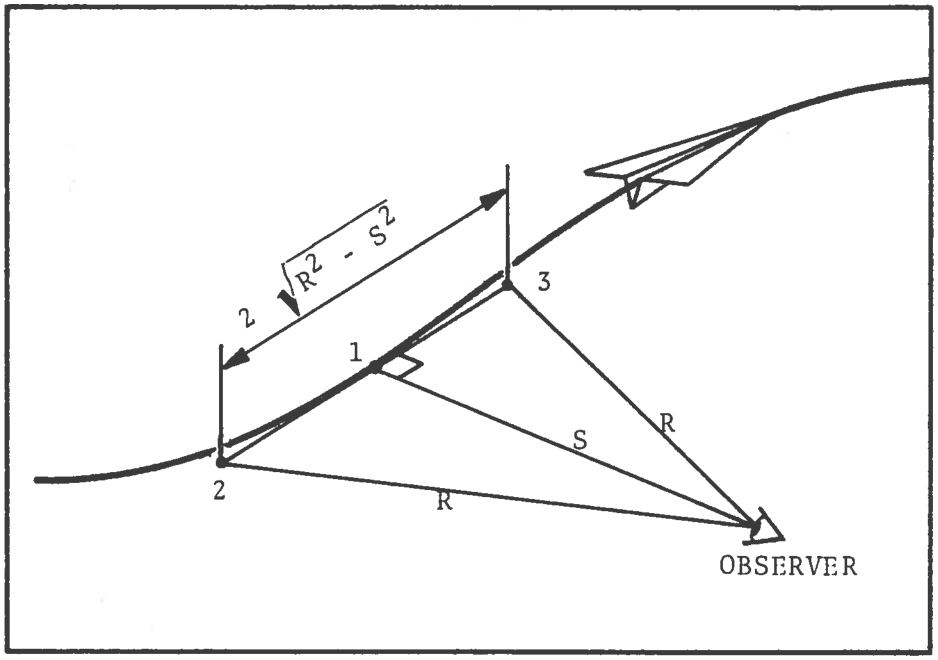


Figure A-6. Distance Traveled Between Threshold Points

The final ASDS index for an observer represents a summation over t for all flights and all operations.

APPENDIX B
PREPARATION AND EXECUTION SPECIFICATIONS FOR IMOD-7

The purpose of this input module (IMOD-7) is to reduce the amount of effort required by the planner in the preparation of an input case. A listing of the IMOD-7 program appears at the end of this appendix. Inherently, the reliability of the input case will increase. This method essentially eliminates redundant entries and nearly all user calculations of the flight path geometry. A set of initial default values may be used or overridden with new entries.

The output of IMOD-7 is a file identical to the input file currently used on MOD-7. The user has the option of using either the current method or the IMOD-7 method. The basic MOD-7 program is not changed by use of IMOD-7. IMOD-7 requires an input file called SMARY.FLT and writes the output to a file called TESTA.OPR, which is immediately usable as input to MOD-7. Program IMOD-7 is distinct from MOD-7, and is utilized sequentially with MOD-7.

The following section describes the use of IMOD-7 to prepare and execute a noise prediction case for MOD-7. This example differs from the illustrated case used earlier. This was done to illuminate special features of IMOD-7.

Preparation of SMARY.FLT

Introduction

The basic methodology used affects two areas:

(1) Redundant entries - The present method requires many redundant entries. These all can be eliminated by the use of default values, i.e., by not making an entry and automatically using the value for that parameter from the previous flight path segment. In the case of the first segment (ground roll), a set of default values may be used or overridden. The default value will always be the last entry for that parameter. For example, if the aircraft speed at the end of the first segment is 160 knots and does not change for any following segments, then no further entries for speed are required, and 160 knots will be automatically used. In practice an aircraft speed of 0 yields spurious results in MOD-7. However, a beginning speed of 0 knots is used in the following case for illustrative purposes only.

(2) Flight path geometry - The present method requires X,Y, and Z for the beginning of each segment. IMOD-7 eliminates all X,Y,Z hand calculations, and requires only the length of ground track data (and climb angle). The helical turns require the ground track X,Y, starting and ending points, the radius, and the X,Y of the center. Another associated feature is the ability to align the runways for easiest input calculations. For example, if the main runway is East-West, where East is zero degrees, then the ground tracks are more easily defined. Should the main runway not be at 0°, the ground tracks may be prepared as if it were 0°, and one entry for the rotation of the angle of reference from DIRECTION (DIR) added to the flight parameter section. Counterclockwise is the positive direction.

Details of Preparation

The coding sheets (Figure B-1) included in the following example are used for IMOD-7. Punched cards with three-character acronym and data entry are prepared for input to IMOD-7. The first coding sheet entry must contain the number of "flights" punched in columns 1-5, right justified. Following that, each flight will consist of parameter cards containing the parameter keyword punched in columns 1-3 and the parameter values punched left justified in columns 11-20 (and, in the case of the XYZ parameter, in columns 21-30 and 31-40), each value incorporating a decimal point.

There are two groups of parameters: those pertinent to the entire flight, and those pertinent to individual segments of a flight. They are as follows:

FLIGHT PARAMETERS	ACT	Aircraft type code number
	NDO	Number of daytime operations
	NNO	Number of nighttime operations
	DIR	Rotation angle of flight pattern
	XYZ	Starting coordinates
	EOF	End of flight flag
SEGMENT PARAMETERS	NSG	Segment number
	ISG	Segment type code
	TH1	Starting thrust
	TH2	Ending thrust
	SP1	Starting speed
	SP2	Ending speed

MOD7 INPUT CODING FORM

Left justify and include a decimal point for all numbers

1–10	11–20	21–30	31–40	DESCRIPTION
1.				Number of "flights"(enter once per case)-
ACT	96.			Aircraft type (enter once per flight)
NDO	1.			Day operations (enter once per flight)
~~NNO~~				Night operations (enter once per flight)
DIR	134.1			Angle of rotation (enter once per flight)
XYZ	23862.	11753.	0.	Starting coordinates (enter once per flight)
NSG	1.			Segment number
ISG	1.			Segment type
~~TH1~~				Start thrust
~~TH2~~				End thrust
SP1	0.5			Start speed
SP2	160.			End speed
~~ANG~~				Climb angle
GND	5000.			Length of linear segment ground track
~~CIR~~				Circulation angle of helical segment
~~RAD~~				Radius of circulation of helical segment
~~EOF~~				(Circle EOF if no more segments in flight)
NSG	2.			Next segment number
ISG				Segment type
~~TH1~~				(Omit if same as preceding TH2)**
~~TH2~~				(Omit if same as current TH1)
SP2	250.			(Omit if same as preceding SP2)**
ANG	10.			(Omit if same as preceding ANG)
GND	7000.			(Omit if same as preceding GND)
~~CIR~~				(Omit if same as preceding CIR)
~~RAD~~				(Omit if same as preceding RAD)
~~EOF~~				(Circle EOF if no more segments in flight)

**TH1 and SP1 are automatically set equal to the TH2 and SP2 of the preceding segment

Figure B-1. IMOD-7 Input Coding Form (Sheet 1 of 3)

Left justify and include a decimal point for all numbers

MOD7 INPUT CODING FORM

DESCRIPTION

Number of "flights"(enter once per case)

Aircraft type
Day operations
Night operations
Angle of rotation
Starting coordinates
(enter once per flight)

Segment number
Segment type
Start thrust
End thrust
Start speed
End speed
Climb angle
Length of linear segment ground track
Circulation angle of helical segment
Radius of circulation of helical segment
(Circle EOF if no more segments in flight)
Next segment number
Segment type
(Omit if same as preceding TH2)**
(Omit if same as current TH1)
(Omit if same as preceding SP2)**
(Omit if same as preceding ANG)
(Omit if same as preceding GND)
(Omit if same as preceding CIR)
(Omit if same as preceding RAD)
(Circle EOF if no more segments in flight)

**TH1 and SP1 are automatically set equal to the TH2 and SP2 of the preceding segment

Figure B-1. IMOD-7 Input Coding Form (Sheet 2 of 3)

Left justify and include a decimal point for all numbers

MOD7 INPUT CODING FORM

1–3	4–40	DESCRIPTION
		Number of "flights"(enter once per case)
ACT		Aircraft type (enter once per flight)
NDO		Day operations (enter once per flight)
[illegible]		Night operations (enter once per flight)
[illegible]		Angle of rotation (enter once per flight)
XYZ		Starting coordinates (enter once per flight)
NSG	5.	Segment number
[illegible]		Segment type
[illegible]	73.	Start thrust
[illegible]		End thrust
[illegible]		Start speed
[illegible]		End speed
[illegible]	0.	Climb angle
[illegible]	20000.	Length of linear segment ground track
[illegible]		Circulation angle of helical segment
[illegible]		Radius of circulation of helical segment
[illegible]		(Circle EOF if no more segments in flight)
[illegible]		Next segment number
ISG		Segment type
[illegible]		(Omit if same as preceding TH2)**
[illegible]		(Omit if same as current TH1)
[illegible]		(Omit if same as preceding SP2)**
[illegible]		(Omit if same as preceding ANG)
[illegible]		(Omit if same as preceding GND)
[illegible]		(Omit if same as preceding CIR)
[illegible]		(Omit if same as preceding RAD)
[illegible]		(Circle EOF if no more segments in flight)

**TH1 and SP1 are automatically set equal to the TH2 and SP2 of the preceding segment

Figure B-1. IMOD-7 Input Coding Form (Sheet 3 of 3)

GND Length of ground track (linear segments)
ANG Climb angle
CIR Circulation angle (helical segments)
RAD Radius of circulation (helical segments)

In general, SMARY.FLT would be arranged as follows:

Number of flights
ACT for first flight
NDO and/or NNO, DIR, and XYZ of first flight

NSG (with a value of 1) for first segment
ISG for first segment
TH1, TH2, SP1, SP2, ANG for first segment
GND if linear segment, CIR and RAD if helical

NSG for next segment
The other segment parameters if different from their default values

NSG for 3rd segment
Parameters
.
.
.
EOF if no more segments in flight
ACT for next flight
.
.
.

etc.

The coding sheet must be designed for all entries for every segment. In practice, many entries for a segment will be the same as the previous segment. Where a previous parameter value is to be used, no entry of acronym or data is required. (A line should be drawn through this item on the coding sheet to insure that a card with an acronym and no data is not prepared.)

Range	Mnemonic	Definition
1. to 999.	ACT	Aircraft code (See Table 2) must be specified for first flight and any subsequent flight where it changes from flight immediately preceding.
positive positive	NDO NNO	Day-night-operations flags are initialized to 1 and zero, respectively, for each flight.

Range	Mnemonic	Definition
-180° to 180°	DIR	Provides for rotating the frame of reference. If a runway is at an angle to the x-axis, it is convenient to provide all geometrical parameters as though it were parallel to the x-axis, and set an appropriate value to DIR to restore the true orientation in the actual plot. DIR is initialized to zero (rotating through the top quadrants is positive angle) for each flight.
	EOF	EOF is the only parameter with no value. It flags the end of the flight (and the end of the last segment).
1. to 99.	NSG	The segment number must be specified for every segment. Its presence signals the end of the previous segment.
1. to 2.	ISG	Segment type is initialized to type 1 (linear) for the first segment of each flight. For subsequent segments it retains whatever value it had for the preceding segment unless specified otherwise.
0. to 100. 0. to 100 positive positive	TH1 TH2 SP1 SP2	Beginning and ending thrusts and beginning and ending speeds are initialized for the first segment of each flight to the typical values of 100% and 160 knots, respectively. In subsequent segments, if TH1 and SP1 are not specified, they are set equal to the TH2 and SP2, respectively, of the preceding segment. If TH2 and SP2 are not specified, they are set equal to the TH1 and SP1 of the <u>same</u> segment.

Range	Mnemonic	Definition
	XYZ	The starting coordinates are initialized to 0., 0., 0. for the first segment of each flight.
positive	GND	The length of the ground track for linear segments is initialized for the first segment of each flight to a typical value of 5500 feet. For subsequent segments it retains the value of the preceding segment unless specified. (Note that the ground track length is for each individual segment, not the cumulative length.)
-90° to 90°	ANG	Climb angle is initialized to zero for first segment of each flight. For subsequent segments it retains the value of the preceding segment unless specified.
positive	CIR RAD	Angle of circulation and radius of curvature for helical segments. They are initialized for the first segment of each flight to the typical values of +90 degrees and 6000 feet, respectively. For subsequent segments they retain the values of the preceding segment unless specified. (Counterclockwise is positive angle.)

NOTE: A warning is typed to the user if, by the second segment, the following parameters are not encountered: ACT, TH1, TH2, SP1, SP2, XYZ, GND, ANG, and one or the other of NDO and NNO. Also, if CIR and RAD are not encountered during the first segment to type 2 (helical), the user is warned.

Execution of the Program on the TSC DEC System-10

Log into area [4147,333] and perform the following steps:

```
.AS DSK 21
.AS DSK 22
.AS DSK 23
.RUN PREPAR
```

If SMARY.FLT exists on disk, the program will execute, typing out warnings if necessary, and writing the output in standard MOD 7 format to TESTA.OPR. (Note that a scratch file, LAST.FLT, is written and can be deleted after the run.)

Creation of a New Load Module (DEC System-10)

```
.COMPILE IMOD 7.F4
FORTRAN:  IMOD 7.F4
EXIT

.LOAD IMOD 7.REL,SYS:USRLIB/LIB
LOADING
IMOD 7 7K CORE
EXIT
.SAVE PREPAR
Job saved
.
```

To incorporate IMOD-7 into MOD-7 as a subroutine, change the blank common in IMOD-7 to a dimension statement, and rewrite the section that sets up the input and output files so that IMOD-7 will read from, and write to, files whose names change with each call. This would allow the use of MOD-7's ability to process several "operation" files during a single execution. (The number of operation files to be read is the first parameter on the "command" file.) The flow chart for the subroutine is given in Figure B-2.

Examples

The DIRECTION (DIR) Parameter

The DIR parameter must be used whenever the runway is not located in an East-West direction, and/or when the direction of the first segment is not due east. With a single entry, the planner can re-orient the runway to an East-West direction which permits the IMOD-7 to perform all sign and coordinate calculations.

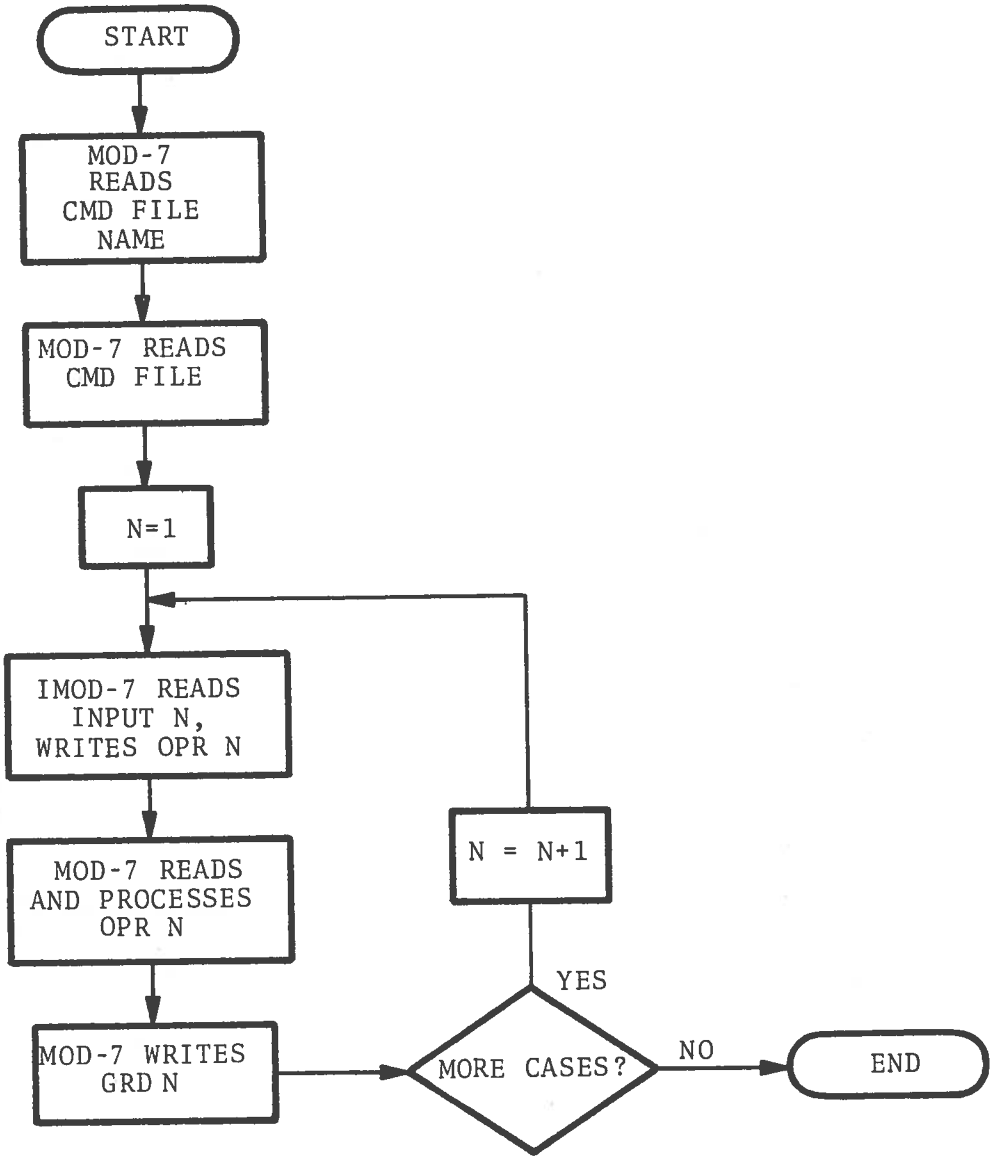


Figure B-2. Flow Chart for Subroutine Incorporating IMOD-7 into MOD-7

Two examples of the DIR parameter will be described. The first is a simple example where the runway is East-West and the aircraft takes off in a westerly direction. The second example includes a fairly complex landing onto a runway with a 30° angle from East-West. This example will illustrate both the DIR function and the best method of setting up a landing.

Example 1. Takeoff, East-West Direction

The takeoff, shown in Figure B-3 at the eastern end of the runway at XYZ coordinates (10000,0,0), rolls 7500 feet down the runway, climbs at 5° for a ground distance of another 15,000 feet, and on out at a 3° climb. Since IMOD-7 assumes all takeoffs to be exactly due east, the flight pattern shown in Figure B-4 must be rotated 180° with respect to the pattern shown in Figure B-3. In order to translate the coordinates computed by IMOD 7 back to the original westerly takeoff, the DIR function is set equal to the amount of rotation, i.e., 180°.

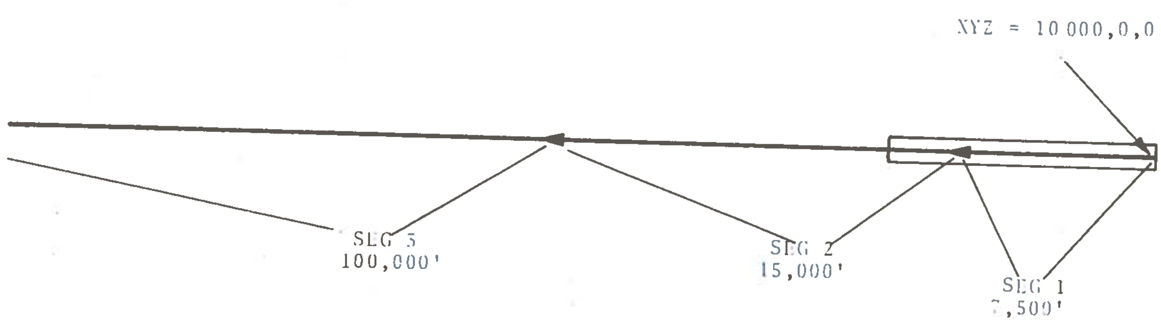


Figure B-3. Eastern End of Runway Pointing East-West

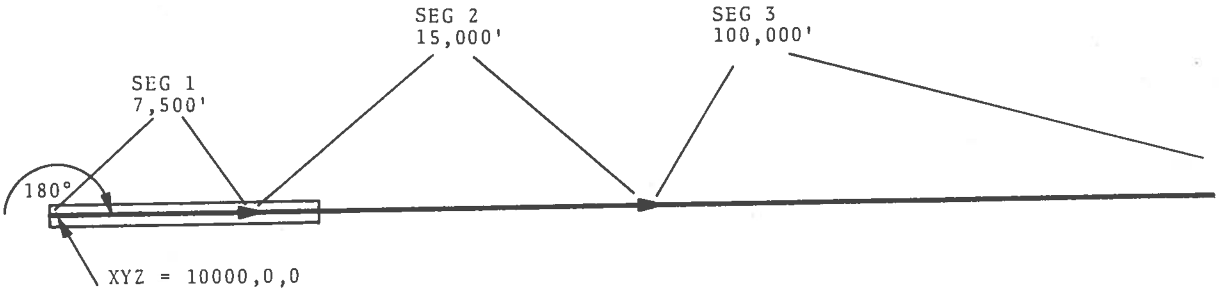


Figure B-4. Runway Direction Reversed 180°, Corresponding to Takeoff Due East

The input file to IMOD-7 might be as follows:

```
1.
ACT    1.
XYZ    10000.    0.    0.
DIR    180.
NSG    1.
ISG    1.
TH1    100.
TH2
SP1
SP2
ANG    0.0
GND    7500.
NSG    2.
ISG    1.
TH1    100.
TH2    100.
SP1    90.
SP2    200.
ANG    5.0
GND    15000.
NSG    3.
ISG    1.
TH1    100.
TH2    100.
SP1    200.
SP2    250.
ANG    3.0
GND    100000.
EOF
```

The resultant output would be as follows:

```
NUMBER OF/FLITES                                    1
FLITE     1 ACTYPE                                   1
FLITE     1 TAKEOFF OR LANDING                       1
FLITE     1 DAY OPS                                  1
FLITE     1 EVE OPS                                  Ø
FLITE     1 NITE OPS                                 Ø
FLITE     1 SEG   1 TYPE                             1
FLITE     1 SEG   1 BEG THRUST                     1ØØ.ØØ
FLITE     1 SEG   1 END THRUST                     1ØØ.ØØ
FLITE     1 SEG   1 X Y Z BEGIN                  1ØØØØ.ØØ        Ø.ØØ        Ø.ØØ
FLITE     1 SEG   1 X Y Z END                     25ØØ.ØØ        Ø.ØØ        Ø.ØØ
FLITE     1 SEG   1 CLIMB ANGLE                      Ø.ØØ
FLITE     1 SEG   1 BEG END SPD                      0.50       9Ø.ØØ
FLITE     1 SEG   2 TYPE                             1
FLITE     1 SEG   2 BEG THRUST                     1ØØ.ØØ
FLITE     1 SEG   2 END THRUST                     1ØØ.ØØ
FLITE     1 SEG   2 X Y Z BEGIN                   25ØØ.ØØ        Ø.ØØ        Ø.ØØ
FLITE     1 SEG   2 X Y Z END                   -125ØØ.ØØ        Ø.ØØ     1312.33
FLITE     1 SEG   2 CLIMB ANGLE                      5.ØØ
FLITE     1 SEG   2 BEG END SPD                     9Ø.ØØ      2ØØ.ØØ
FLITE     1 SEG   3 TYPE                             1
FLITE     1 SEG   3 BEG THRUST                     1ØØ.ØØ
FLITE     1 SEG   3 END THRUST                      7Ø.ØØ
FLITE     1 SEG   3 X Y Z BEGIN                 -125ØØ.ØØ        Ø.ØØ     1312.33
FLITE     1 SEG   3 X Y Z END                  -1125ØØ.ØØ        Ø.ØØ     6553.11
FLITE     1 SEG   3 CLIMB ANGLE                      3.ØØ
FLITE     1 SEG   3 BEG END SPD                    2ØØ.ØØ      25Ø.ØØ
FLITE     1 SEG   4 TYPE                            -1
```

Example 2. IMOD-7 Processing for Input to MOD-7

The next example will show the landing and DIR concepts, input coding for IMOD-7, and the output generated, which becomes the input to MOD-7, the Noise Prediction Model.

There are two ways to construct the landing input coding:

(1) As a conventional landing - In this case the climb angles must be negative, and the planner must calculate the X,Y, and Z coordinates for the start of each segment. The need for hand calculation of these coordinates negates a major feature of the IMOD-7 routine.

(2) As a "backwards takeoff" - In this case the landing is treated as a backwards takeoff where the IMOD-7 will calculate all the coordinates, thus relieving the planner of considerable effort. All climb angles in this case are positive.

An example of the use of the "backwards takeoff" method follows.

Figure B-5 shows a landing pattern consisting of four segments (numbered backwards, starting from 4). The aircraft comes in from the northwest on a heading of 150°, makes a 90° left turn which aligns it with a 10,000-foot runway, continues to descend, touching down at a point 2500 feet down the runway, and coasts to the far end. The XYZ parameter is the end of the last segment of the landing, i.e., the right end of the runway, which now becomes the first segment of the takeoff (numbered 1 in Figure B-5). In the example, the origin of the coordinate system is taken to be the left end of the runway (as might be the case if there were several runways at this airport). It is assumed that, for some reason, the coordinate system must be used as is and cannot be changed to make the origin be the starting point. Knowing the runway to be 10,000 feet long and making an angle of 30° with the x-axis, the coordinates of its right end are found to be (8660.25, 5000,0), and this

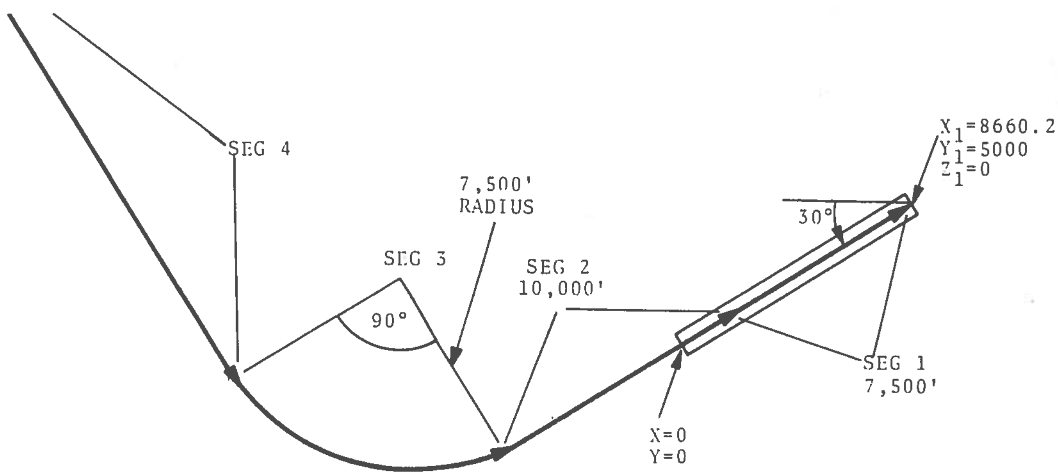


Figure B-5. Landing Pattern (Segments 1-4) is the value assigned to XYZ for IMOD-7. With XYZ set, the flight pattern is rotated so that the aircraft (looking at it now as taking off) takes off parallel to the x-axis of the coordinate system and in the eastward (positive x) direction, as shown in Figure B-6. The amount of rotation necessary is the value of the DIR parameter, taking into consideration the following convention: The range of DIR is -180° to +180° inclusive; the sign is positive if the minimum rotation required is clockwise, and negative if counterclockwise, so DIR is set to -150.

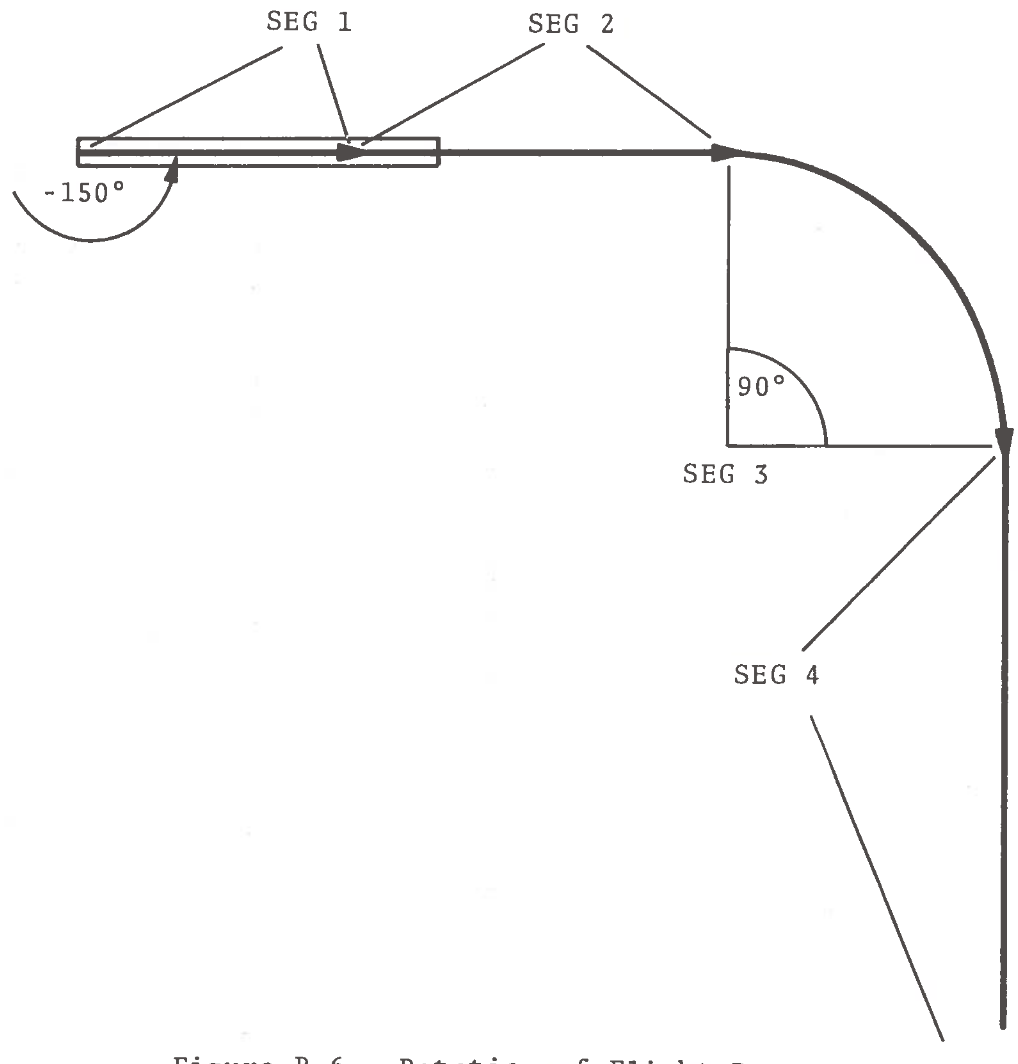


Figure B-6. Rotation of Flight Pattern

The rest of the IMOD-7 parameters are set as though the flight pattern in Figure B-6 were an ordinary takeoff, except for thrusts and speeds. There is a 7500-foot ground roll followed by a climb giving a ground track length of 10,000 feet, a right 90° turn (CIR = -90°) with a radius of curvature of 7500 feet, and, finally, a linear track parallel to the y-axis. Using the DIR parameter, IMOD-7 automatically translates the geometry into the original pattern of Figure B-5 for input to MOD-7.

The input to IMOD-7 file might be as follows:

```
1.
ACT      1.
XYZ      8660.25   5000.   0.
DIR      -150.
NSG      1.
ISG      1.
TH1      0.5
TH2      34.
SP1      0.5
SP2      70.
ANG      [illegible]
GND      7500.
NSG      2.
ISG      1.
TH1      32.
TH2      54.
SP1      70.
SP2      150.
ANG      3.0
GND      10000.
NSG      3.
ISG      2.
CIR      -90.
RAD      7500.
NSG      4.
ISG      1.
TH1      70.
TH2      70.
SP1      150.
SP2      200.
GND      20000.
EOF
```

The resultant output file from IMOD-7 would be as follows:

```
NUMBER OF/FLITES                                  1
FLITE    1 ACTYPE                                 1
FLITE    1 TAKEOFF OR LANDING                     1
FLITE    1 DAY OPS                                1
FLITE    1 EVE OPS                                0
FLITE    1 NITE OPS                               0
FLITE    1 SEG    1 TYPE                          1
FLITE    1 SEG    1 BEG THRUST                 0.50
FLITE    1 SEG    1 END THRUST                30.00
FLITE    1 SEG    1 X Y Z BEGIN             8661.25     5000.00       0.00
FLITE    1 SEG    1 X Y Z END               2165.06     1250.00       0.00
FLITE    1 SEG    1 CLIMB ANGLE                0.00
FLITE    1 SEG    1 BEG END SPD                0.50       70.00
FLITE    1 SEG    2 TYPE                          1
FLITE    1 SEG    2 BEG THRUST                30.00
FLITE    1 SEG    2 END THRUST                50.00
FLITE    1 SEG    2 X Y Z BEGIN             2165.06     1250.00       0.00
FLITE    1 SEG    2 X Y Z END              -6495.19    -3750.00     524.08
FLITE    1 SEG    2 CLIMB ANGLE                3.00
FLITE    1 SEG    2 BEG END SPD               70.00      150.00
FLITE    1 SEG    3 TYPE                          2
FLITE    1 SEG    3 BEG THRUST                50.00
FLITE    1 SEG    3 END THRUST                50.00
FLITE    1 SEG    3 X Y Z BEGIN            -6495.19    -3750.00     524.08
FLITE    1 SEG    3 X Y Z END             -16736.43    -1011.65    1141.74
FLITE    1 SEG    3 CLIMB ANGLE                3.00
FLITE    1 SEG    3 BEG END SPD              150.00      150.00
FLITE    1 SEG    3 CIRCULATION                 -1.
FLITE    1 SEG    3 CENTER OF HELIX       -10245.19     2745.19       0.00
FLITE    1 SEG    4 TYPE                          1
FLITE    1 SEG    4 BEG THRUST                70.00
FLITE    1 SEG    4 END THRUST                70.00
FLITE    1 SEG    4 X Y Z BEGIN           -16736.43    -1011.65    1141.74
FLITE    1 SEG    4 X Y Z END             -26751.67    16290.32    2189.90
FLITE    1 SEG    4 CLIMB ANGLE                3.00
FLITE    1 SEG    4 BEG END SPD              150.00      200.00
FLITE    1 SEG    5 TYPE                         -1
```

Figure B-7 shows DIR values for the same flight pattern at four different orientations. The following IMOD-7 input file, shown with its resultant output, will produce the landing of Figure B-5.

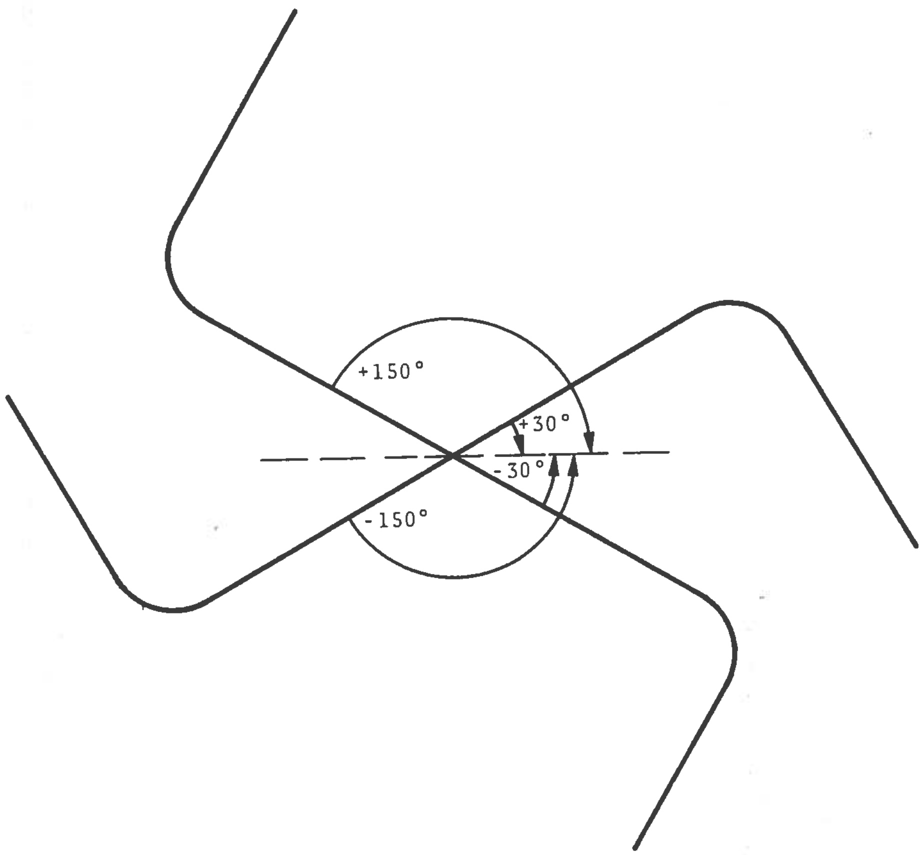


Figure B-7. DIR Values for Flight Pattern of Example 2

Examples of IMOD-7 input files, along with their resultant output, are shown in the next four pages of this appendix. The IMOD-7 listing appears at the end of Appendix B.

```
1.
[illegible]   46.
[illegible]   1.
[illegible]   134.1
XYZ           23[illegible]61.    11753.    [illegible]
NSG           1.
ISG           1.
[illegible]   [illegible]
[illegible]   16[illegible].
GAP           [illegible]
[illegible]   [illegible]
[illegible]   25[illegible].
ANG           1[illegible].
GAP           7[illegible].
NSG           [illegible].
ISG           [illegible].
[illegible]   16.15
GAP           7[illegible].
NSG           4.
ISG           1.
[illegible]   [illegible]41.
[illegible]   [illegible]
TH1           93.
ANG           [illegible].
GAP           2[illegible]03.
[illegible]
```

In this example of one runway in a multiple runway airport the XYZ happens not to be 0., 0., 0. Note also that the runway makes an angle of 134.1 degrees with the positive x-axis.

Segment one is a 5000-foot ground roll, from zero to 160 knots at 100% thrust.

In segment two, the aircraft climbs at a 10 degree angle, speeds up to 250 knots covering 7000 feet on the ground.

The aircraft turns left through 16.15 degrees on a helix of radius 7000 feet climbing at 4 degrees.

The last segment is level flight and reduced thrust.

The output generated from the above input is shown on the next page. Teletype communication is shown below:

```
RUN INMOD

TH1 TH2 ANG  OMITTED FROM  1ST SEG. FLT  1


CPU TIME: 5.79  ELAPSED TIME: 8:22.17
NO EXECUTION ERRORS DETECTED

EXIT
```

```
NUMBER OF/FLITES                                1
FLITE    1 ACTYPE                              46
FLITE    1 TAKEOFF OR LANDING                   1
FLITE    1 DAY OPS                              1
FLITE    1 EVE OPS                              0
FLITE    1 NITE OPS                             0
FLITE    1 SEG    1 TYPE                        1
FLITE    1 SEG    1 BEG THRUST             100.00
FLITE    1 SEG    1 END THRUST             100.00
FLITE    1 SEG    1 X Y Z BEGIN          23661.00   11753.00      -0.10
FLITE    1 SEG    1 X Y Z END            20381.44   15343.63      -0.10
FLITE    1 SEG    1 CLIMB ANGLE              0.00
FLITE    1 SEG    1 BEG END SPD              0.50     160.00
FLITE    1 SEG    2 TYPE                        1
FLITE    1 SEG    2 BEG THRUST             100.00
FLITE    1 SEG    2 END THRUST             100.00
FLITE    1 SEG    2 X Y Z BEGIN          20381.44   15343.63      -0.10
FLITE    1 SEG    2 X Y Z END            15510.05   20370.52    1234.19
FLITE    1 SEG    2 CLIMB ANGLE             10.00
FLITE    1 SEG    2 BEG END SPD            160.00     250.00
FLITE    1 SEG    3 TYPE                        2
FLITE    1 SEG    3 BEG THRUST             100.00
FLITE    1 SEG    3 END THRUST             100.00
FLITE    1 SEG    3 X Y Z BEGIN          15510.05   20370.52    1234.19
FLITE    1 SEG    3 X Y Z END            13962.40   21573.24    1582.24
FLITE    1 SEG    3 CLIMB ANGLE             10.00
FLITE    1 SEG    3 BEG END SPD            250.00     250.00
FLITE    1 SEG    3 CIRCULATION                1.
FLITE    1 SEG    3 CENTER OF HELIX      10483.16   15499.13       0.00
FLITE    1 SEG    4 TYPE                        1
FLITE    1 SEG    4 BEG THRUST             100.00
FLITE    1 SEG    4 END THRUST             100.00
FLITE    1 SEG    4 X Y Z BEGIN          13962.40   21573.24    1582.24
FLITE    1 SEG    4 X Y Z END             6984.97   25569.89    3000.08
FLITE    1 SEG    4 CLIMB ANGLE             10.00
FLITE    1 SEG    4 BEG END SPD            250.00     250.00
FLITE    1 SEG    5 TYPE                        1
FLITE    1 SEG    5 BEG THRUST              93.00
FLITE    1 SEG    5 END THRUST              93.00
FLITE    1 SEG    5 X Y Z BEGIN           6984.97   25569.89    3000.08
FLITE    1 SEG    5 X Y Z END           -10369.65   35510.57    3000.08
FLITE    1 SEG    5 CLIMB ANGLE              0.00
FLITE    1 SEG    5 BEG END SPD            250.00     250.00
FLITE    1 SEG    6 TYPE                       -1
```

Output generated from the input shown on preceding page.

(1)			(2)		
1.			1.		
ACT	52.		ACT	52.	
NDO	5.		NDO	5.	
NNO	0.				NNO,DIR, and XYZ are all initialized to zero
DIR	0.				
XYZ	0.	0.		0.	
NSG	1.		NSG	1.	
ISG	1.		ISG	1.	
TH1	100.				TH1 and TH2 are initialized to 100%
TH2	100.				
SP1	0.		SP1	0.5	
SP2	150.		SP2	150.	
ANG	0.				ANG is initialized to zero
GND	5000.		GND	5000.	
NSG	2.		NSG	2.	
ISG	1.				ISG retains the value to which it was previously set
TH1	100.				
TH2	100.				
SP2	200.		SP2	200.	(Note that SP1 is automatically set to the previous SP2)
ANG	5.		ANG	5.	
GND	5000.				GND retains its value
NSG	3.		NSG	3.	
ISG	2.		ISG	2.	
ANG	5.				ANG retains its value
CIR	45.		CIR	45.	
RAD	4500.		RAD	4500.	
NSG	4.		NSG	4.	
ISG	1.		ISG	1.	
TH1	80.		TH1	80.	
TH2	80.				TH2 is automatically set to the current TH1
SP2	350.		SP2	350.	
ANG	3.		ANG	3.	
GND	20000.		GND	20000.	
EOF			EOF		

These two input files produce identical output (shown on next page).

```
NUMBER OF FLITES                               1
FLITE     1 ACTYPE                            52
FLITE     1 TAKEOFF OR LANDING                 1
FLITE     1 DAY OPS                            5
FLITE     1 EVE OPS                            0
FLITE     1 NITE OPS                           0
FLITE     1 SEG   1 TYPE                       1
FLITE     1 SEG   1 BEG THRUST               100.00
FLITE     1 SEG   1 END THRUST               100.00
FLITE     1 SEG   1 X Y Z BEGIN                0.00       0.00       -0.10
FLITE     1 SEG   1 X Y Z END               5000.00       0.00       -0.10
FLITE     1 SEG   1 CLIMB ANGLE                0.00
FLITE     1 SEG   1 BEG END SPD                0.50     160.00
FLITE     1 SEG   2 TYPE                       1
FLITE     1 SEG   2 BEG THRUST               100.00
FLITE     1 SEG   2 END THRUST               100.00
FLITE     1 SEG   2 X Y Z BEGIN             5000.00       0.00       -0.10
FLITE     1 SEG   2 X Y Z END              10000.00       0.00      437.34
FLITE     1 SEG   2 CLIMB ANGLE                5.00
FLITE     1 SEG   2 BEG END SPD              160.00     200.00
FLITE     1 SEG   3 TYPE                       2
FLITE     1 SEG   3 BEG THRUST               100.00
FLITE     1 SEG   3 END THRUST               100.00
FLITE     1 SEG   3 X Y Z BEGIN            10000.00       0.00      437.31
FLITE     1 SEG   3 X Y Z END              13181.98    1318.02      746.63
FLITE     1 SEG   3 CLIMB ANGLE                5.00
FLITE     1 SEG   3 BEG END SPD              200.00     200.00
FLITE     1 SEG   3 CIRCULATION                1.
FLITE     1 SEG   3 CENTER OF HELIX        10000.00    4500.00        0.00
FLITE     1 SEG   4 TYPE                       1
FLITE     1 SEG   4 BEG THRUST                80.00
FLITE     1 SEG   4 END THRUST                80.00
FLITE     1 SEG   4 X Y Z BEGIN            13181.96    1318.02      746.58
FLITE     1 SEG   4 X Y Z END              27324.12   15460.16     1794.83
FLITE     1 SEG   4 CLIMB ANGLE                3.00
FLITE     1 SEG   4 BEG END SPD              200.00     350.00
FLITE     1 SEG   5 TYPE                      -1
```

The output file generated by either of the two input files on the preceding page.

TABLE B-1. IMOD-7 PROGRAM

```
C     IMOD7.F4 TRANSLATES SHORT MOD7 INPUT FILE INTO
C              STANDARD FORM
C
C
C
      COMMON DUM(80),LIST(12),ITABS(12),NUMS(4)
      DATA LIST/3HACT,3HTH1,3HTH2,3HXYZ,3HGND,3HANG,
     *          3HSP1,3HSP2,3HNDO,3HNNO,3HCIR,3HRAD/
      DATA NUMS/2HST,2HND,2HRD,2HTH/
      MONE=-1
C
      CALL IFILE(1,'SMARY','FLT')
      CALL OFILE(21,'TESTA','OPR')
      READ(1,1) FLTS
1     FORMAT(F5.0)
      NOF=FLTS
      WRITE(21,2) NOF
2     FORMAT('NUMBER OF/FLITES',27X,I4)
      DO 100 IF=1,NOF
      CALL OFILE(22,'LAST','FLT')
      NSEG=0
      NTEM=0
      IDO=1
      INO=0
      TH1=100.
      TH2=100.
      SP1=160.
      SP2=160.
      ANG=0.
      XX1=0.
      YY1=0.
      ZZ1=-.1
      GND=5500.
      XX2=XX1
      YY2=YY1
      IAC=0
      NSG=0
      ISG=1
      INSEG=.FALSE.
      DIRFLG=.FALSE.
      ZZ2=ZZ1
      THETA=0.
      DXX=COSD(THETA)
      DYY=SIND(THETA)
      CIR=0.
      RAD=0.
      DO 222 J=1,12
222   ITABS(J) = LIST(J)
9999  READ(1,3) ITEM,VALUE,VAL2,VAL3
3     FORMAT(A5,5X,3F10.2)
      IF(NSG.GT.1)GO TO 226
      DO 225 J=1,10
```

```
      IF(ITEM.NE.ITABS(J))GO TO 225
      ITABS(J)=0
      IF(J.LT.9)GO TO 226
      JJ = 11-(J/5)
      ITABS(JJ) = 0
      GO TO 226
225   CONTINUE
226   CONTINUE
      IF(ITEM.EQ.'ACT') GO TO 200



      IF(ITEM.EQ.'NDO') GO TO 201
      IF(ITEM.EQ.'NNO') GO TO 202
      IF(ITEM.EQ.'EOF') GO TO 215
      IF(ITEM.EQ.'ISG') GO TO 204
      IF(ITEM.EQ.'NSG') GO TO 215
      IF(ITEM.EQ.'TH1') GO TO 206
      IF(ITEM.EQ.'TH2') GO TO 207
      IF(ITEM.EQ.'XYZ') GO TO 208
      IF(ITEM.EQ.'GND') GO TO 209
      IF(ITEM.EQ.'ANG') GO TO 210
      IF(ITEM.EQ.'SP1') GO TO 211
      IF(ITEM.EQ.'SP2') GO TO 212
      IF(ITEM.EQ.'CIR') GO TO 213
      IF(ITEM.EQ.'RAD') GO TO 214
      IF(ITEM.EQ.'DIR') GO TO 216
      TYPE 4, ITEM
4     FORMAT(1X,A5,' UNRECOGNIZABLE ENTRY,IGNORED')
      GO TO 9999
5     FORMAT('FLITE',I5,1X,'SEG',I4,1X,'TYPE',20X,I4)
C   INCOMPLETE DATA FOR FIRST FLITE
301   TYPE 6
6     FORMAT(/' INCOMPLETE FIRST FLITE DATA')
      GO TO 1000
C   SAME FLITE
302   CALL IFILE(23,'LAST','FLT')
1302  READ(23,7) (DUM(ID),ID=1,80)
7     FORMAT(80A1)
      ENDS=EOF1(23)
      IF (ENDS) GO TO 1002
      WRITE(21,7) (DUM(ID),ID=1,80)
      GO TO 1302
C   SEGMENT TYPE
204   ISG=VALUE
      GO TO 99
C   A/C TYPE
200   IAC=VALUE
      GO TO 99
C   DAY OPS
```

TABLE B-1. IMOD-7 PROGRAM (CONT'D)

```
201     IDO=VALUE
        GO TO 99
C    NITE OPS
202     INO=VALUE
        GO TO 99
C
303     TYPE 8
8       FORMAT(/' TRYING TO ENTER NEW SEG WHILE STILL IN OLD SEG!')
        GO TO 1000
C
304     TYPE 9
9       FORMAT(/' JUMPING SEGMENT')
        GO TO 1000
C
C    BEGIN THRUST
206     TH1=VALUE
        TH2=TH1
        GO TO 99
C    END THRUST
207     TH2=VALUE
        GO TO 99
C    BEG POINT
208     XX1=VALUE
        YY1=VAL2
        ZZ1=VAL3
        GO TO 99
C    GROUND TRACK
209     GND=VALUE
        GO TO 99
C    CLIMB ANGLE
210     ANG=VALUE
        GO TO, 99
C    BEGIN SPEED
211     SP1=VALUE
        GO TO 99
C    END SPEED
212     SP2=VALUE
        GO TO 99
C    CIRCULATION
213     CIR=VALUE
        GO TO 99
C  RADIUS OF CIRCULATION
214     RAD=VALUE
        GO TO 99
216     THETA=THETA+VALUE
        DXX=COSD(THETA)
        DYY=SIND(THETA)
```

TABLE B-1. IMOD-7 PROGRAM (CONT'D)

```
      IF(DIPFLG)WRITE(5,235)
235   FORMAT(' WARNING: INCORRECT USE OF DIR')
      DIRFLG=.TRUE.
      GO TO 99
C  END OF SEGMENT
215   IF(NSEG.LT.1) GO TO 205
      IF(NSG.NE.1)GO TO 220
      IFLG=0
      DO 221 J=1,10
      IF(ITABS(J).EQ.0)GO TO 221
      IFLG=1
      WRITE(5,223) ITABS(J)
223   FORMAT(1H+,A4,S)
221   CONTINUE
      IORD=MINO(NSG,4)
      IF(IFLG.EQ.1) WRITE(5,224) NSG,NUMS(IORD),IF
224   FORMAT(15H+ OMITTED FROM ,I2,A2,' SEG, FLT ',I2//)
220   IF(ISG.LE.1) GO TO 305
      IFLG=0
      IF(CIR.NE.0.)GO TO 230
      IFLG=1
      CIR=90.
      WRITE(5,223) LIST(11)
230   IF(RAD.NE.0.)GO TO 231
      IFLG=1
      RAD=6000.
      WRITE(5,223) LIST(12)
231   IORD=MINO(NSG,4)
      IF(IFLG.EQ.1)WRITE(5,224) NSG,NUMS(IORD),IF
      PLMS=SQRT(DXX**2+DYY**2)*SIGN(1.,CIR)
      XX0=XX1-RAD*DYY*PLMS
      YY0=YY1+RAD*DXX*PLMS
      IF(DXX.NE.0.) GO TO 400
      THETA=90.
      GO TO 401
400   THETA=ATAN2(DYY,DXX)
      THETA=THETA*180.*7./22.




401   THETA=THETA+CIR
      DXX=COSD(THETA)
      DYY=SIND(THETA)
      YY2=YY0-RAD*DXX*SIGN(1.,CIR)
      XX2=XX0+RAD*DYY*SIGN(1.,CIR)
      ZZ2=ZZ1+SIND(ANG)/COSD(ANG)*RAD*ABS(CIR)/180.*22./7.
      GO TO 306
C   LINEAR SEGMENT
305   XX2=XX1+GND*DXX
      YY2=YY1+GND*DYY
```

TABLE B-1. IMOD-7 PROGRAM (CONT'D)

```
      ZZ2=ZZ1+SIND(ANG)/COSD(ANG)*GND
      ABSL=SQRT((XX2-XX1)**2+(YY2-YY1)**2)
      DXX=(XX2-XX1)/ABSL
      DYY=(YY2-YY1)/ABSL
C   WRITE OUT SEGMENT
306   IF(NSG.GT.1) GO TO 307
C   WRITE OUT A/C TYPE
      WRITE(21,10) IF,IAC,IF
10    FORMAT('FLITE',I5,1X,'ACTYPE',26X,I4,/
     *        'FLITE',I5,1X,'TAKEOFF OR LANDING',14X,4H   1)
      WRITE(22,10) IF,IAC
C   WRITE OUT DAY, EVENING, AND NITE OPS (EVE, NEVER USED, IS DUMMIED)
      WRITE(21,11) IF,IDO,IF
11    FORMAT('FLITE',I5,1X,'DAY OPS',25X,I4/
     1        'FLITE',I5,1X,'EVE OPS',25X,4H   0)
      WRITE(22,11) IF,IDO,IF
      WRITE(21,12) IF,INO
12    FORMAT('FLITE',I5,1X,'NITE OPS',24X,I4)
      WRITE(22,12) IF,INO
C   WRITE OUT SEGMENT TYPE
307   WRITE(21,13) IF,NSG,ISG
13    FORMAT('FLITE',I5,' SEG',I4,1X,'TYPE',20X,I4)
      WRITE(22,13) IF,NSG,ISG
C   WRITE OUT BEGIN THRUST
      WRITE(21,14) IF,NSG,TH1
14    FORMAT('FLITE',I5,' SEG',I4,1X,'BEG THRUST',11X,F10.2)
      WRITE(22,14) IF,NSG,TH1
C   WRITE OUT END THRUST
      WRITE(21,15) IF,NSG,TH2
15    FORMAT('FLITE',I5,' SEG',I4,1X,'END THRUST',11X,F10.2)
      WRITE(22,15) IF,NSG,TH2
      TH1=TH2
C   WRITE OUT BEGIN COORDINATES
      WRITE(21,16) IF,NSG,XX1,YY1,ZZ1
16    FORMAT('FLITE',I5,' SEG',I4,1X,'X Y Z BEGIN',10X,3F10.2)
      WRITE(22,16) IF,NSG,XX1,YY1,ZZ1
C   WRITE OUT END  COORDINATES
      WRITE(21,17) IF,NSG,XX2,YY2,ZZ2
17    FORMAT('FLITE',I5,' SEG',I4,1X,'X Y Z END',12X,3F10.2)
      WRITE(22,17) IF,NSG,XX2,YY2,ZZ2
C   WRITE OUT CLIMB ANGLE
      WRITE(21,18) IF,NSG,ANG
18    FORMAT('FLITE',I5,' SEG',I4,1X,'CLIMB ANGLE',10X,F10.2)
      WRITE(22,18) IF,NSG,ANG
C   WRITE OUT BEGIN AND END SPEEDS
      WRITE(21,19) IF,NSG,SP1,SP2
19    FORMAT('FLITE',I5,' SEG',I4,1X,'BEG END SPD',10X,2F10.2)
      WRITE(22,19) IF,NSG,SP1,SP2
      SP1=SP2
      IF(ISG.LT.2) GO TO 205
C   WRITE OUT CIRCULATION AND CENTER OF HELIX
```

TABLE B-1. IMOD-7 PROGRAM (CONT'D)

```
      CIR=SIGN(1.,CIR)
      WRITE(21,20) IF,NSG,CIR
20    FORMAT('FLITE',I5,' SEG',I4,1X,'CIRCULATION',10X,E10.0)
      WRITE(22,20) IF,NSG,CIR
      WRITE(21,21) IF,NSG,XX0,YY0,ZZ0
21    FORMAT('FLITE',I5,' SEG',I4,1X,'CENTER OF HELIX',6X,3F10.2)
      WRITE(22,21) IF,NSG,X0,Y0,Z0
205   IF(ITEM.EQ.'EOF') GO TO 203
C     END OF LSEG, BEGIN OF NEW SEG
      LL=NSG
      NSG=VALUE
      IF((NSG-LL).GT.1.OR.(NSG-LL).LT.0) GO TO 304
      GO TO 308
C     END OF FLITE
203   IF(IF.GT.1.AND.NTEM.EQ.0) GO TO 302
      IF(IF.LE.1.AND.NTEM.EQ.0) GO TO 301
      NSG=NSG+1
      WRITE(21,5) IF,NSG,NONE
      WRITE(22,5) IF,NSG,NONE
      GO TO 1001
C   SET START COORDINATE OF NEXT SEG
308   IF(NSEG.EQ.0)GO TO 309
      XX1=XX2
      YY1=YY2
      ZZ1=ZZ2
      GO TO 99
309   NSEG=1
99    NTEM=NTEM+1
      GO TO 9999
1001  CALL RELEAS(22)
      CALL OFILE(23,'LAST','FLT')
      CALL IFILE(22,'LAST','FLT')
9901  READ(22,7) (DUM(ID),ID=1,80)
      BOTM=EOF1(22)
      IF(BOTM) GO TO 1902
      WRITE(23,7) (DUM(ID),ID=1,80)
      GO TO 9901
1002  CALL RELEAS(23)
100   CONTINUE
1000  CALL RELEAS(1)
      CALL RELEAS(21)
      CALL RELEAS(22)
      STOP
      END
```

TABLE B-1. IMOD-7 PROGRAM (CONCLUDED)

110 copies

U. S. DEPARTMENT OF TRANSPORTATION
TRANSPORTATION SYSTEMS CENTER
KENDALL SQUARE, CAMBRIDGE, MA. 02142