FAA -NO-70-6

FINAL REPORT

CONTRACT FA68WA-1900

A DIGITAL COMPUTER PROGRAM FOR COMPUTATION OF NOISE EXPOSURE FORECAST CONTOURS

JUNE 1970

Prepared For

DEPARTMENT OF TRANSPORTATION
FEDERAL AVIATION ADMINISTRATION
OFFICE OF NOISE ABATEMENT

By

Richard D. Horonjeff
Allan Paul

BOLT BERANEK AND NEWMAN INC.
15808 Wyandotte Street
Van Nuys, California 91406

FINAL REPORT

FAA-NO-70-6

A DIGITAL COMPUTER PROGRAM FOR COMPUTATION OF NOISE EXPOSURE FORECAST CONTOURS

Prepared By

Richard D. Horonjeff
Allan Paul

This report has been prepared by Bolt,Beranek and Newman,Inc., for the Office of Noise Abatement, Federal Aviation Administration, under Contract No. FA68WA-1900. The contents of this report reflect the views of the contractor, who is responsible for the facts and the accuracy of the data presented herein, and do not necessarily reflect the official views of policy of the FAA. This report does not constitute a standard, specification or regulation.

BOLT, BERANEK AND NEWMAN, INC.
15808 Wyandotte Street
Van Nuys, California 91406

ABSTRACT

A detailed description of the digital computer program developed for the computation and plotting of Noise Exposure Forecast contours is presented. All routines including flow charts, program listings, and internal data organizations are provided. A brief summary of the basic Noise Exposure Forecast concepts and equations are included. Additional information and examples of the results of the computation program can be found in the following reports prepared in performance of Contract FA68WA-1900: FAA-NO-69-2, FAA-NO-70-7, FAA-NO-70-8, and FAA-NO-70-9.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (Con't)

LIST OF FIGURES (Con't)

I. INTRODUCTION

This report presents a detailed description of a digital computer program which was originally developed by Bolt, Beranek and Newman, Inc. under Federal Aviation Administration Contract FA67WA-1705, and refined under Contract FA68WA-1900, as an aid in producing contours of equal noise exposure using procedures set forth in Reference 1. The program documentation and the preparation of this report was conducted in performance of Task I under Phase II of Contract FA68WA-1900. The material in this report is arranged to provide both the user and the programmer with the information necessary to use, or modify the routines. It is helpful if the reader has some knowledge of the Noise Exposure Forecast (NEF) concept; however, this is not absolutely necessary.

Section II presents a users guide to the program and includes a description of data input and output formats and the organization of the data elements for program execution. An alphabetic listing of error messages and their probable causes is also presented.

Section III outlines the general computational procedures for determining the contour locations.

Appendix A provides samples of the input data and associated program output for the NEF contours calculated for a single flight path.

Appendix B provides a listing of the mathmetical relationships pertinent to the program.

Appendix C provides a set of flow diagrams and the program listings for all routines.

II. PROGRAM USAGE

A. Discussion of Data Required

This section of the report outlines the types of data required in order to compute the NEF contours for each flight path. The general flow of control within the program and the format required for data input is explained. (A complete set of flow diagrams and program listings may be found in Appendix C.) To compute the NEF contour distances, data concerning aircraft noise, performance, and volume of activity for each flight path must be read by the program.

The aircraft noise data is described in terms of the Effective Perceived Noise Level (EPNL). An EPNL vs Aircraft-to-Observer Slant Distance function is specified in terms of a list of EPNL values at various slant distances. This function varies only with slant distance and is not dependent upon the position on the flight path.

The aircraft performance data is described in terms of an Altitude Profile and a Delta-EPNL Profile. The Altitude Profile consists of a list of X, Y values (assumed to be connected by straight line segments) which specify the altitude of the aircraft (Y-coordinate) as a function of the aircraft's distance along the flight track (X-coordinate) from start of takeoff roll.

The Delta-EPNL Profile also consists of a list of X, Y values. This profile allows the user to modify the overall noise level characteristics of the aircraft as a function of distance along the flight track. This capability is useful for introducing power cutbacks at selected points along the flight path. The program assumes that distances are expressed in feet and noise levels in EPNdB.

Aircraft classes, based on physical attributes of the aircraft, are defined in terms of noise and performance data. Classes such as "four-engine turbojet" and "two-engine turbofan" are used. Each aircraft class is subdivided into trip length categories. These categories allow for varying performance characteristics within a single aircraft class. The trip length category for each aircraft class must be defined in terms of an altitude profile, Delta-EPNL profile, and EPNL profile. These profiles need not be unique to a particular aircraft trip length. Usually a modest number of profiles are sufficient to define several different classes of aircraft.

The level of flight activity is the volume of operations occurring during each of two daily time periods. These time periods include the daytime hours (0700-2200) and the nighttime hours (2200-0700). Each time period must have a level of flight activity specified.

A generalized flow diagram of the input, output, and computation stages of the program is shown in Fig. 1. The data is read and checks are made to determine that all necessary data has been entered. If no input errors have been encountered the program will proceed to compute the desired contours. Processing of the flight paths is sequential until all data has been processed.

B. Input Data Specifications and Formats

Data is prepared on punched cards for program input. A unique format exists for each type of data. These formats are detailed in the paragraphs below. Examples are shown in Fig. A-1.

1.) AIRCRAFT DESCRIPTOR

An aircraft class is defined in terms of a numeric name, an alphanumeric descriptor and the numeric names of the three profiles (altitude, delta-EPNL, and EPNL) to be used for each of eight trip length categories. Category 1 has been arbitrarily assigned to landings and the remaining categories to takeoffs. Category #1 ordinarily uses altitude profile #1 because this is the landing profile generated by the program as a function of the glide slope and runway length. Table I shows the card format for entering this data.

2.) ALTITUDE PROFILES

An Altitude Profile is defined in terms of a numeric name, the number of X, Y coordinate values describing the profile, and the coordinate values. The X coordinate is the distance along the flight track and the Y coordinate, the altitude. The origin of the coordinate system is arbitrary. However, it is typically on the end of the runway at which the aircraft commences its takeoff. The landing profile is generated by the program, making this assumption. For flight paths which involve operations on more than one runway, a common profile origin point must be chosen. This origin is usually related to one of the runways and the profiles involving operations from the other runways

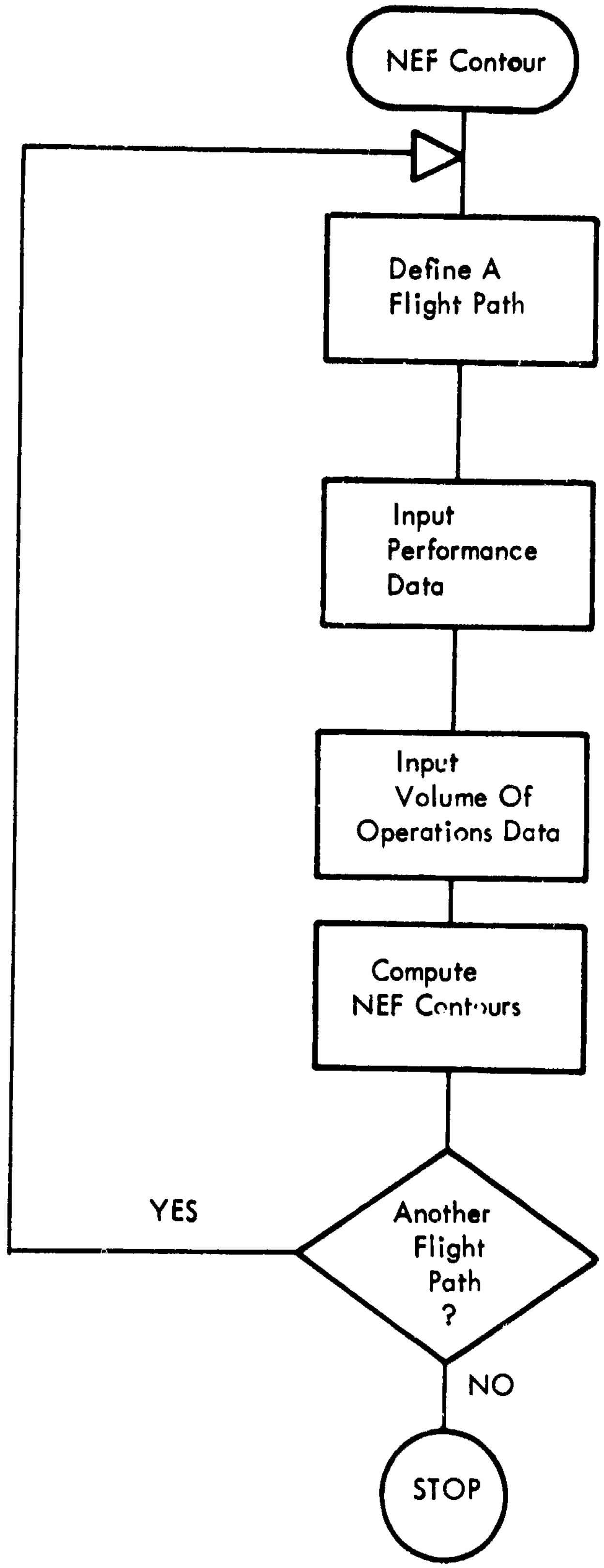


FIGURE 1. BLOCK DIAGRAM OF PROGRAM FLOW

TABLE I
AIRCRAFT DESCRIPTOR CARD FORMAT

Columns	Contents		Mode	Range
1-2	Numeric Name		Integer	1-99
3-14	Alphanumeric Descriptor		Alpha	
17-18	Altitude Profile Name	Trip Category #1 (Landings)	Integer	1-99
19-20	Delta-EPNL Profile Name		"	1-99
20-21	EPNL Profile Name		"	1-99
25-26	Altitude Profile Name	Trip Category #2 (Takeoffs 0-500 Mi)	"	1-99
27-28	Delta-EPNL Profile Name		"	1-99
29-30	EPNL Profile Name		"	1-99
33-34	Altitude Profile Name	Trip Category #3 (Takeoffs 500-1000 mi)	"	1-99
35-36	Delta-EPNL Profile Name		"	1-99
37-38	EPNL Profile Name		"	1-99
41-42	Altitude Profile Name	Trip Category #4 (Takeoffs 1000-1500 mi)	"	1-99
43-44	Delta-EPNL Profile Name		"	1-99
45-46	EPNL Profile Name		"	1-99
49-50	Altitude Profile Name	Trip Category #5 (Takeoffs 1500-2500 mi)	"	1-99
51-52	Delta EPNL Profile Name		"	1-99
53-54	EPNL Profile Name		"	1-99
57-58	Altitude Profile Name	Trip Category #6 (Takeoffs 2500-3500 mi)	"	1-99
59-60	Delta-EPNL Profile Name		"	1-99
61-62	EPNL Profile Name		"	1-99
65-66	Altitude Profile Name	Trip Category #7 (Takeoffs 3500-4500 mi)	"	1-99
67-68	Delta-EPNL Profile Name		"	1-99
69-70	EPNL Profile Name		"	1-99
73-74	Altitude Profile Name	Trip Category #8 (Takeoffs 4500 + mi)	"	1-99
75-76	Delta-EPNL Profile Name		"	1-99
77-78	EPNL Profile Name		"	1-99

TABLE II

ALTITUDE PROFILE AND DELTA-EPNL PROFILE

CARD FORMAT

Column	Contents	Mode	Range
1-3	Numeric Name	Integer	1-99
4-6	Number of points to describe profile	Integer	1-10
7-8	Blank		
9-14	X_1	Real	
15-20	Y_1	"	
21-26	X_2	"	
27-32	Y_2	"	
33-38	X_3	"	
39-44	Y_3	"	
45-50	X_4	"	
51-56	Y_4	"	
57-62	X_5	"	
63-68	Y_5	"	
69-74	X_6	"	
75-80	Y_6	"	

To enter more than six coordinates, use an additional card leaving columns 1-8 blank. X_7 must appear in columns 9-14, etc.

must be adjusted for the difference in origins. Although the program generates the coordinates of the landing profile, the user must define a dummy profile whose numeric name is "1". It must be the first altitude profile defined in the program. The restrictions on the coordinate values are that no number may be negative and successive X values must always be increasing. Twelve coordinate values are allowed. Table II shows the card format for entering this data.

3.) DELTA-EPNL PROFILES

A Delta-EPNL Profile is defined in terms of a numeric name, the number of X, Y values describing the profile, and the coordinate values. The X coordinate is the position on the flight track and the Y coordinate is the EPNL correction to be <u>added</u> to the EPNL computed from the EPNL vs Slant Distance function. As with the altitude profile, the origin of the coordinate system is arbitrary, however, the origin for all altitude and Delta-EPNL profiles must be the same. The only restriction on the coordinate values is that successive X values must always be increasing. There is no restriction on the Y value. The card format for this profile is identical to the altitude profile format and is shown in Table II.

4.) EPNL PROFILE

An EPNL vs Slant Distance Profile is defined in terms of a numeric name, and two lists of 35 EPNL values each. These EPNL values correspond to assumed values of Slant Distance starting at 12.6 feet ($10^{1.1}$) and increasing by a factor of $10^{0.1}$ up to 31622 feet ($10^{4.5}$). One list is to be used when the aircraft is on or near the ground and the sound propagation path is from ground location to ground location. The second list is to be used when the aircraft is airborne and the propagation path is from air-to-ground. An integer code is used to specify the propagation path associated with the data (1 = ground-to-ground, 2 = air-to-ground). The program expects the data to be on eight successive cards; however, the cards may be in any order since the rank of the data is also coded on each card. Table III shows the card format for entering the EPNL data.

TABLE III

EPNL PROFILE CARD FORMAT

Column	Contents	Mode	Range
1-2	Numeric Name	Integer	1-99
3-4	Propagation Path Code	"	1-2
5-7	Lowest EPNL rank (L1)	"	1,11,21,31
8-10	Highest EPNL rank	"	10,20,30,35
11-17	$EPNL_{L1}$	Real	
18-24	$EPNL_{L1 + 1}$	"	
25-31	$EPNL_{L1 + 2}$	"	
32-38	$EPNL_{L1 + 3}$	"	
39-45	$EPNL_{L1 + 4}$	"	
46-52	$EPNL_{L1 + 5}$	"	
53-59	$EPNL_{L1 + 6}$	"	
60-66	$EPNL_{L1 + 7}$	"	
67-73	$EPNL_{L1 + 8}$	"	
74-80	$EPNL_{L1 + 9}$	"	

TABLE IV
VOLUME OF OPERATIONS CARD FORMAT

Columns	Contents	Mode	Range
1-3	Aircraft Class Numeric Name	Alpha	1-99
4-6	Time Period Code	Integer	1-2
7-14	Volume For Trip Length Category #1 (landings)	Real	≥0
15-22	#2 (Takeoffs)	"	≥0
23-30	#3 (")	"	≥0
31-38	#4 (")	"	≥0
39-46	#5 (")	"	≥0
47-54	#6 (")	"	≥0
55-62	#7 (")	"	≥0
63-70	#8 (")	"	≥0

TABLE V
FLIGHT PATH IDENTIFICATION CARD FORMAT

Columns	Contents	Mode	Range
1-51	Alphanumeric Descriptor	Alpha	≥0
52-58	Path Position To Commence Computation	Real	≥0
59-65	Path Position To Terminate Computation	Real	≥0
66-70	Interval On Path Between Computations	Real	≠0
71-76	Runway Length	Real	≥0
71-80	Glide Slope (in degrees)	Real	≥0

5.) VOLUME OF OPERATIONS

For each flight path the average volume of operations on the path must be specified by trip length category for each aircraft class to be considered. The numeric name of the aircraft class, the time period code (1 = daytime, 2 = nighttime) and the volume of operations for each trip length category must be specified. The specified volume of operations may never be negative. Zero values are, of course, valid. Two cards per aircraft class are expected by the program. The first must be for the daytime period and the second for the nighttime period. Table IV shows the card format for this data.

6.) FLIGHT PATH IDENTIFICATION

In order to compute the contours for a given flight path, an alphanumeric descriptor and five parameters describing the flight path are required.* The parameters specify (1) the point on the track (distance from start of take-off roll) to commence computing the contour, (2) the point on the track to terminate computing the contour, (3) the interval along the track at which to compute the contour, (4) the length of the runway on which the aircraft arrived or departed and (5) the glide slope used by the aircraft approaching said runway.

The computations may be made by moving in either direction along the flight track (i.e. the specified interval may be either positive or negative. However, care should be taken to insure that the stopping position is of lesser value than the starting position if the interval is negative and conversely, if positive.

The runway length and glide slope need not be specified if no landing operations are to be considered on the current path segment. The card format for the flight path identification data is shown in Table V.

---

* For convenience and ease of interpretation all input and output data are with reference to the flight track (i.e. the vertical projection of the flight path on the ground).

C. Organizing the Input Data for Processing

Data to be read is organized into blocks corresponding to each flight path. Each block is initiated by a Flight Path Identification Card and terminated by a Blank Card. The required data elements (aircraft descriptor, noise, and performance data along with the volume of operations data) are contained between these two cards. The only limitation on the order in which the elements appear is that aircraft decriptor, noise, and performance data must precede the volume of operations data. Only those aircraft whose volume of operations has been specified in this block will be considered in computing the contours for this path.

The program assumes all data cards to specify volume of operations. If other information is to be read in a control card specifying the type of data and/or the action to be taken must preceed the information. The control card format and list of commands is given in Table VI. An example of the card deck setup to compute contours for a flight path is shown in Figs. A-2.

In most cases the aircraft descriptor, noise and performance data will be the same for all flight paths. Therefore, this data need be entered only once (for example, with the data for the first flight path).

Since data storage space is limited, however, the capability to delete information no longer needed is desirable. To accomplish this certain cards (listed in Table VI) will allow particular areas in storage to be erased or fixed. A large amount of data will be used over and over in the computations for a multi-flight-path job. These data can then be followed by an appropriate "fix" command. When at a later time, it is desired to clear part of storage to make room for more special profiles, only the information read in after the last fix command will be erased. This way only the "transient" data (i.e. data used for one or two paths) will be erased, without the need of re-entering all "standard" profiles. In this manner many flight paths with many special profiles can be handled even though only limited storage space is available. The volume of operation storage area is automatically cleared upon entering the data for a new flight path.

If an error is detected during the data input or computation phase, an error message is printed. These messages are tabulated in Table VII. Unless an input error is irrecoverable data input will continue; however, no contour will be computed for the flight path.

TABLE VI

CONTROL CARD FORMAT AND CODES FOR ENTERING AIRCRAFT DESCRIPTOR, NOISE, AND PERFORMANCE DATA

Column	Contents	Range
1-3	Operation To Be Performed	See Below
4-6	Number of Times Operation Is To Be Repeated	

Operation Code	Operation To Be Performed
101	Read an Aircraft Descriptor
102	Read an Altitude Profile
103	Read a Delta-EPNL Profile
104	Read an EPNL Profile
111	Fix currently defined aircraft descriptors in permanent table.
112	Fix currently defined altitude profiles in permanent table.
113	Fix currently defined Delta-EPNL profiles in permanent table.
114	Fix currently defined EPNL functions in permanent table.
115	Fix all current aircraft descriptor, noise and performance data.
121	Delete all aircraft descriptors not in permanent table.
122	Delete all altitude profiles not in permanent table.
123	Delete all Delta-EPNL profiles not in permanent table.
124	Delete all EPNL functions not in permanent table
125	Delete all aircraft descriptors, noise, and performance data not in permanent tables.
999	Stop all processing.

TABLE VII

ERROR MESSAGES

Number	Error Message	Cause
1	ALTITUDE PROFILE TABLE FULL FOLLOWED BY #4	The Altitude Profile just entered cannot be stored.
2	ALTITUDE PROFILE UNDEFINED FOLLOWED BY #4	An Altitude Profile named on the Aircraft Descriptor Card has not been entered.
3	AIRCRAFT DESCRIPTOR TABLE FULL FOLLOWED BY #4	The Aircraft Descriptor just entered cannot be stored.
4	CANNOT ENTER NAME =	The Numeric Name of the erroneous profile is printed.
5	CARD MISSING OR OUT OF ORDER	The data input routine has detected an error in the card sequence.
6	CONDENSED A/C TABLE FULL	The number of unique aircraft is greater than 50.
7	DELTA-EPNL PROFILE UNDEFINED FOLLOWED BY #4	A Delta-EPNL Profile, named on the Aircraft Descriptor Card, has not been entered.
8	DELTA-EPNL PROFILE TABLE FULL FOLLOWED BY #4	The Delta-EPNL Profile just entered cannot be stored.
9	DUPLICATE ALTITUDE PROFILE NAME FOLLOWED BY #4	An Altitude Profile of the same numeric name has been previously defined.
10	DUPLICATE AIRCRAFT DESCRIPTOR NAME FOLLOWED BY #4	An Aircraft class of the same numeric name has been previously defined.
11	DUPLICATE DELTA-EPNL PROFILE NAME FOLLOWED BY #4	A Delta-EPNL Profile of the same numeric name has been previously defined.
12	DUPLICATE PNLEFF PROFILE NAME FOLLOWED BY #4	An EPNL Profile of the same numeric name has been previously defined.
13	EPNL FUNCTION UNDEFINED FOLLOWED BY #4	The EPNL Profile named on the Aircraft Descriptor card, has not been entered.
14	FUNCTION EPNL-INVALID INPUT PARAMETER	The EPNL calculation subrouting has seen an error in the calling parameters.

TABLE VII (Con't)

Number	Error Message	Cause
15	GLIDE SLOPE CANNOT BE NEGATIVE	The Glide Slope, on the Flight Path Identification Card, is in error.
16	INTERVAL IS OR WRONG SIGN	The start, stop, and interval parameters, on the Flight Path Identification Card, are inconsistent.
17	INVALID NUMBER OF ALTITUDE POINTS FOLLOWED BY #4	The Altitude Profile has too many or too few points.
18	INVALID NUMBER OF DELTA-EPNL POINTS FOLLOWED BY #4	The Delta-EPNL Profile has too many or too few points.
19	INVALID OPERATION CODE FOLLOWED BY CODE	The Operation Code on the Control Card cannot be recognized by the program.
20	INVALID PNLEFF PROPAGATION PATH FOLLOWED BY #4	The propagation path code on the EPNL Profile Card is not 1 or 2.
21	INVALID PNLEFF RANK LIMITS FOLLOWED BY #4	The rank parameter on the EPNL Profile Card is invalid.
22	PNLEFF PROFILE NAMES DO NOT MATCH FOLLOWED BY #4	The EPNL Profile Cards are inconsistenly identified.
23	PNLEFF PROFILE TABLE FULL FOLLOWED BY #4	The EPNL vs. Slant Distance function just entered cannot be stored.
24	PROGRAM CANNOT COMPUTE THIS CONTOUR	An error was detected during the input of this Flight Path Block.
25	RUNWAY LENGTH MUST BE GREATER THAN ZERO	The runway parameter, on the Flight Path Identification Card, is in error.
26	RUNWAY TOO SHORT FOR GLIDE SLOPE	The glide slope intersection is off the runway.
27	STARTING POINT CANNOT BE NEGATIVE	The start parameter, on the Flight Path Identification Card, is invalid.
28	STOPPING POINT CANNOT BE NEGATIVE	The stop parameter on the Flight Path Identification Card, is invalid.
29	UNDEFINED FOLLOWED BY THE AIRCRAFT CLASS NUMERIC NAME AND THE VOLUME OF OPERATIONS DATA	The aircraft class, on the Volume of Operations Card, has not been defined.

III. PROCEDURES USED FOR COMPUTING CONTOURS

As discussed in Section I, the purpose of the computer program is to generate the distance relationships between the flight track and contours of equal NEF value. This section of the report discusses the techniques used to determine these distance relationships. Specifically, the program generates the perpendicular (sideline) distance from the flight track to each contour at regular intervals along the track. The method chosen to compute these distances is a heuristic one based on knowledge of the NEF and distance behavioral relationships.

The dashed curve in Fig. 2 shows a typical relationship between the NEF value and sideline distance at a particular point on the flight track. An important attribute of this curve is its smooth decrease in NEF value with increasing distance. The iterative technique shown graphically in Fig. 2 is used to determine the sideline distance to the desired NEF contour. Two trial distances, Y_{t1} and Y_{t2}, are chosen. The corresponding NEF values are then computed for these distances and a straight line is drawn through these points. The point of intersection of this line with the desired NEF value yields a new trial distance, Y_{t3}. The NEF value at this new distance is then computed. The new (t3) and previous (t2) trials are used, as above, to compute another trial distance. The iterative procedure continues until the difference between two successive trial distances becomes sufficiently small. The perpendicular distance reported is the last trial distance calculated. The computations proceed from point to point along the flight track. The change in shape of the NEF vs sideline distance curve is gradual for successive points on the track. Thus, the last sideline distance calculated for the contour is used to determine the first two trial distances for the next point along the flight track.

The NEF contour computation is terminated at the end of the flight track or when the contour closes. The contour is assumed to have closed if, during the iterative procedure, a trial value is less than 12.6 feet. The point of closure on the flight track is determined by calculating the NEF at two nearby points on the flight track. A logarithmic interpolation is performed to obtain the point of closure.

In the above procedure, it was required to calculate the NEF value for a given sideline distance. This NEF value is computed as the sum, on an energy basis, of the NEF

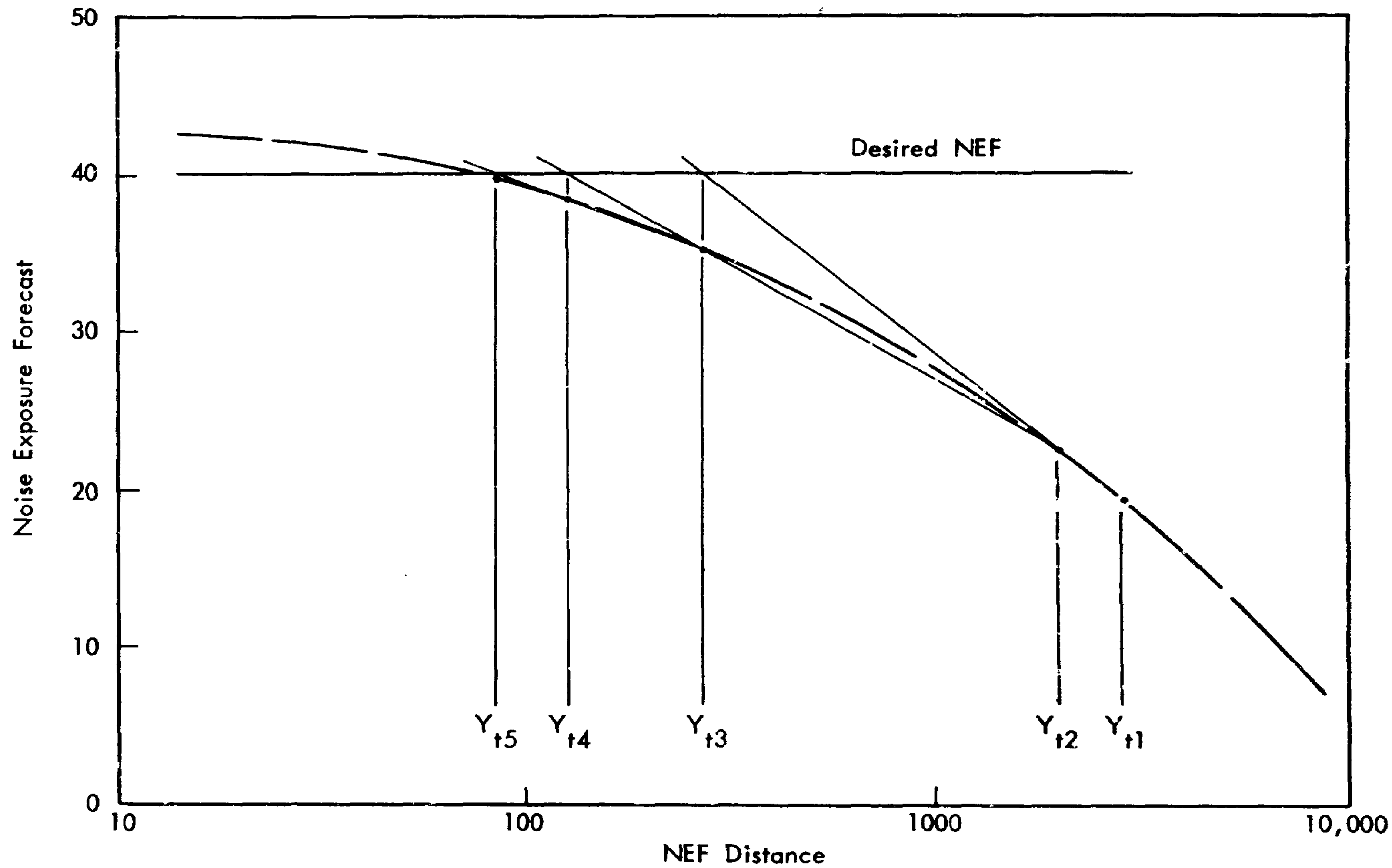


FIGURE 2. GRAPHIC ILLUSTRATION OF ITERATIVE PROCESS TO DETERMINE NEF DISTANCE

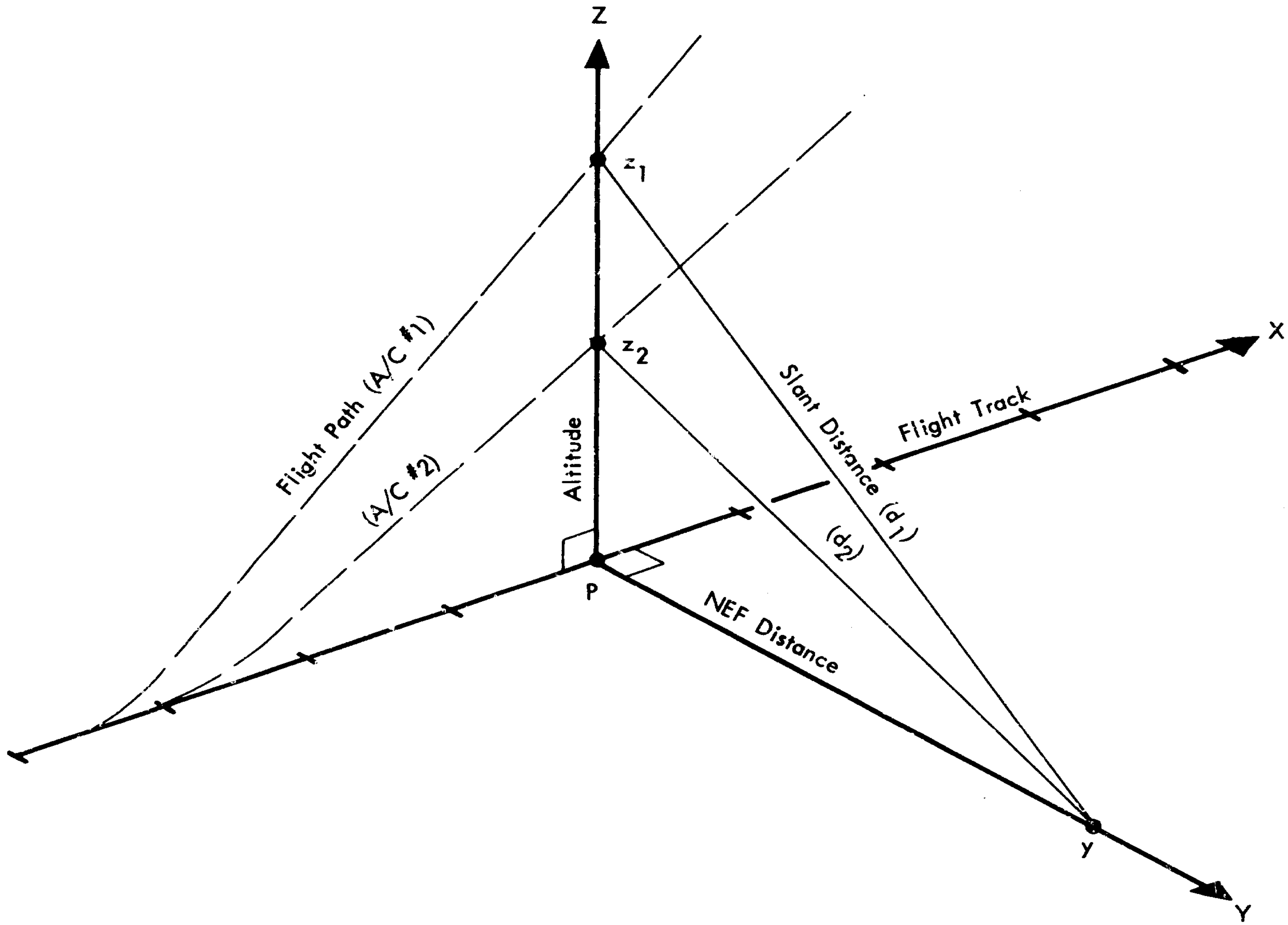


FIGURE 3. GEOMETRIC RELATIONSHIPS USED IN CALCULATING THE NEF DISTANCE

contributions from the operations of each aircraft trip length. For purposes of computation, each aircraft trip length category is treated as an individual aircraft type with known noise and performance characteristics. Figure 3 shows the geometrical relationship between the aircraft and ground positions for two aircraft types. The EPNL at the ground location Y is determined from the slant distance to the aircraft. The NEF contribution of the aircraft is then evaluated according to Equation 1, Appendix B from the EPNL and volume of operations for this aircraft. The NEF contributions for all aircraft are then summed according to Equation 3, Appendix B. The summed NEF is the actual Noise Exposure Forecast Value at the point.

If it is desired to change the numeric value of the NEF contour to be computed the array NEFCON (Fig. C-14C) can be set equal to the desired values. This cannot be done without limit, however. The EPNL information is supplied for slant distances up to 31622 feet. A contour which has a larger distance than 31622 feet to the flight path will therefore be in error. (The value for 31622 feet will be computed). Simarily a contour with a value such that the slant distance is less than 12.6 feet will also be in error.

REFERENCES

1. K.N. Stevens, A. C. Pietrasanta, et al "Procedure for Estimating Noise Exposure and Resulting Community Reactions from Air Base Operations"; WADC TN 57-10, 1967.

2. "Land Use Planning Relating to Aircraft Noise", Bolt Beranek and Newman Inc. Technical Report for the Federal Aviation Administration, October 1964. Also published by the Department of Defense as AFM 86-5, TM 5-365, NAVDOCKS p-98, "Land Use Planning with Respect to Aircraft Noise".

3. "Procedures for Developing Noise Exposure Forecast Areas for Aircraft Operations", Federal Aviation Administration Technical Report DS-67-10, August 1967.

4. "Techniques for Developing Noise Exposure Forecasts", Federal Aviation Administration Technical Report, DS-67-14, August 1967.

5. W. C. Sperry: "Aircraft Noise Evaluation", Federal Aviation Administration Technical Report, FAA-NO-68-34, September 1968.

APPENDIX A

SAMPLE NOISE EXPOSURE FORECAST CONTOUR CALCULATIONS

APPENDIX A

SAMPLE NOISE EXPOSURE FORECAST CONTOUR CALCULATIONS

The NEF program needs for its execution several types of data. All data input is on 80 column punched cards. The various formats for the different types of data are shown in Figs. A-1A through A-1G.

Some examples of input data are shown in Fig. A-2. The results of this simple computation are shown in the Figs. A-3. The program can handle considerably more complex problems, such as the example given in Fig. A-4. The complexity of the volume of operations, the total number of operations and the length of flight track for which the contours are computed affect only the execution time of the program. The first example in Appendix A (Figs. A-3) has 7 entries in two classes for a total of 65 aircraft. The contours are computed up to 30,000 feet. Execution time for this example is 65 seconds using an IBM System/360 Model 30. The second example (Figs. A-4) has 69 entries in eight classes for a total of 342 aircraft. In addition the contours are computed to 100,000 feet. The execution time in this case, on the same machine, is 368 seconds.

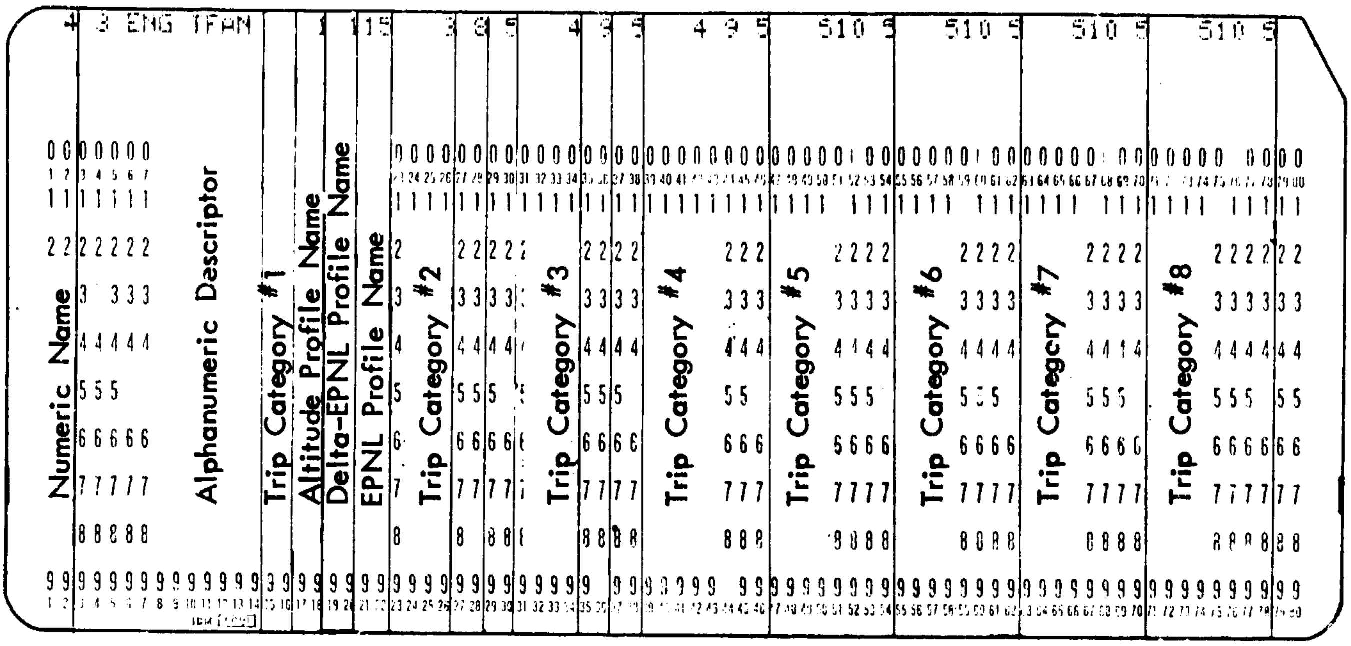


FIGURE A - 1A. AIRCRAFT DESCRIPTOR DATA CARD

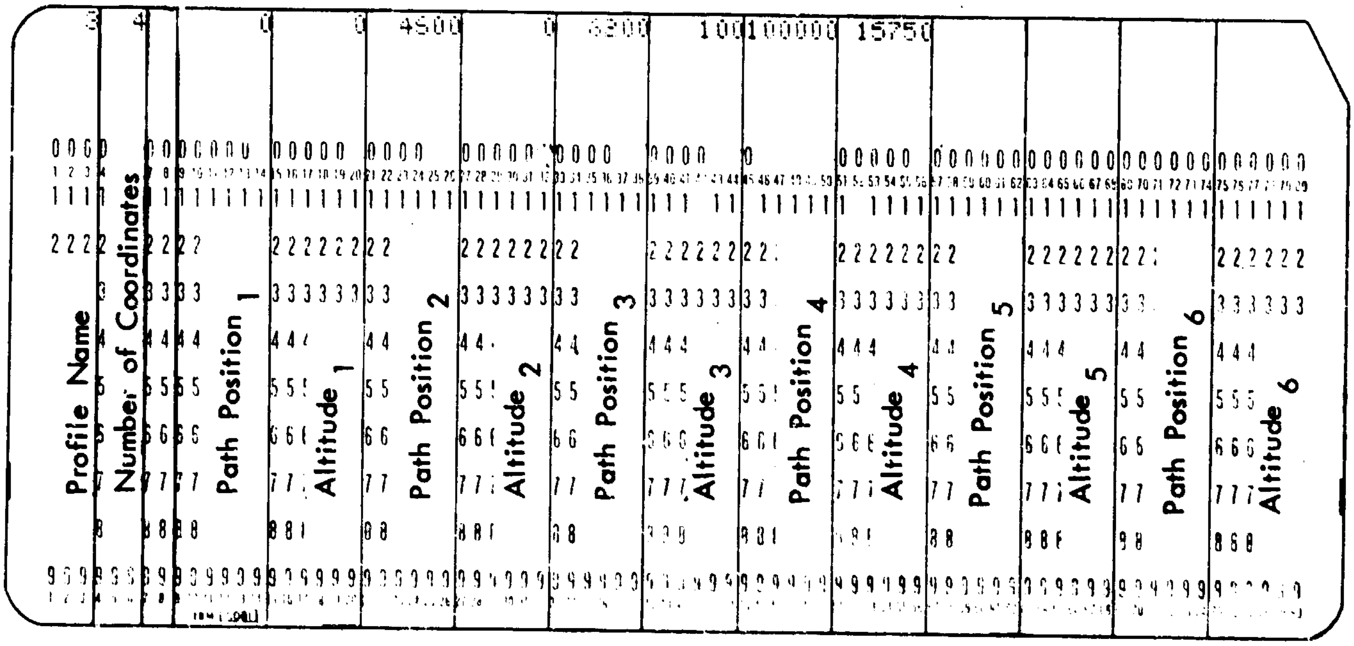


FIGURE A - 1B. ALTITUDE PROFILE DATA CARD

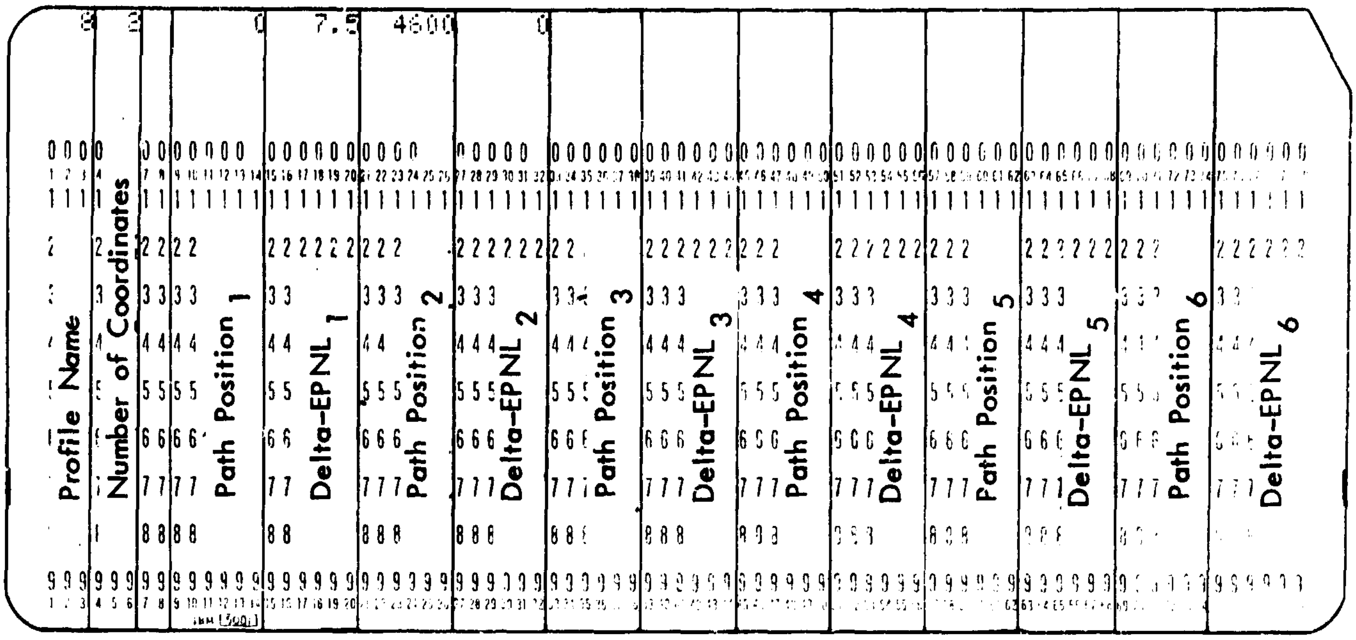


FIGURE A - 1C. DELTA-EPNL PROFILE DATA CARD

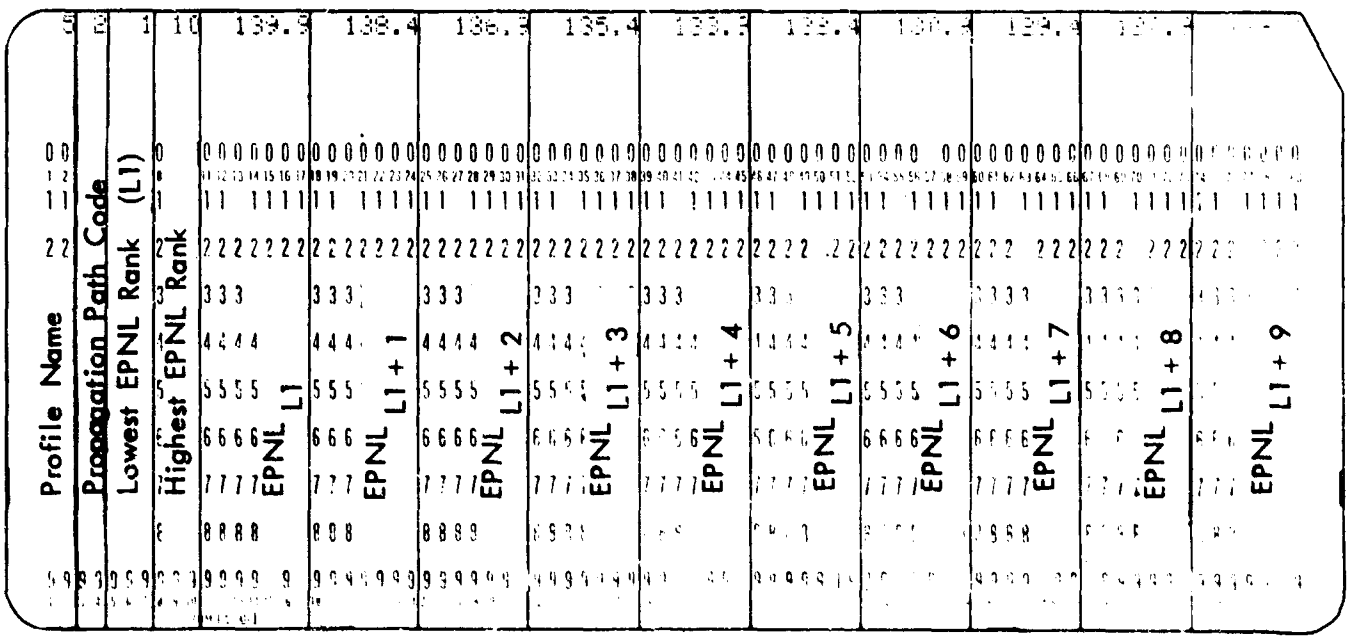


FIGURE A - 1D. EPNL PROFILE DATA CARD

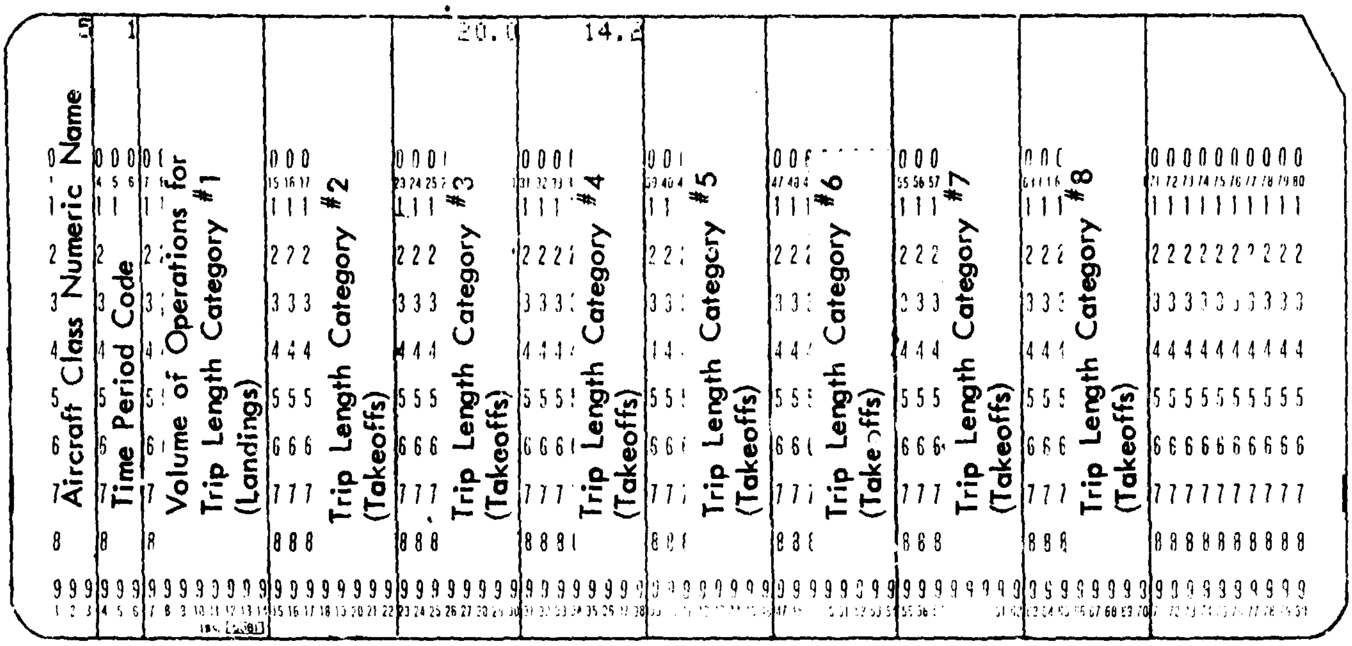


FIGURE A - 1E. VOLUME OF OPERATIONS DATA CARD

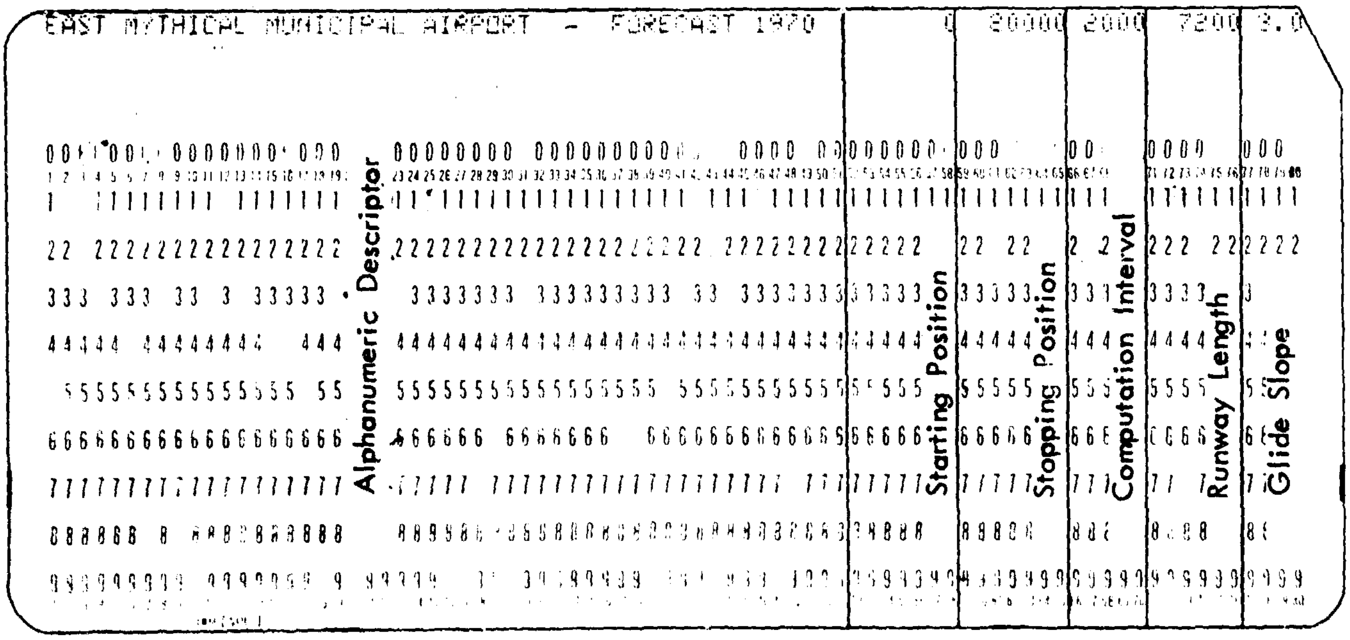


FIGURE A - 1F. FLIGHT PATH IDENTIFICATION DATA CARD

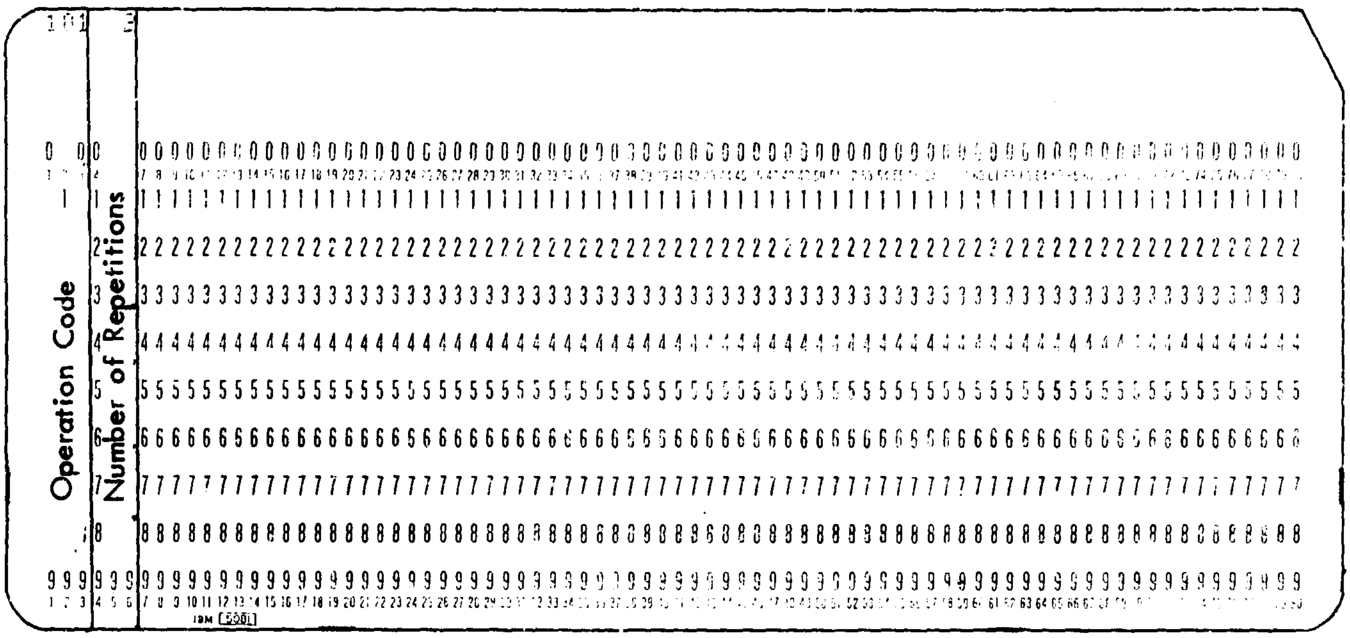


FIGURE A - 1G. PERFORMANCE DATA CONTROL CARD

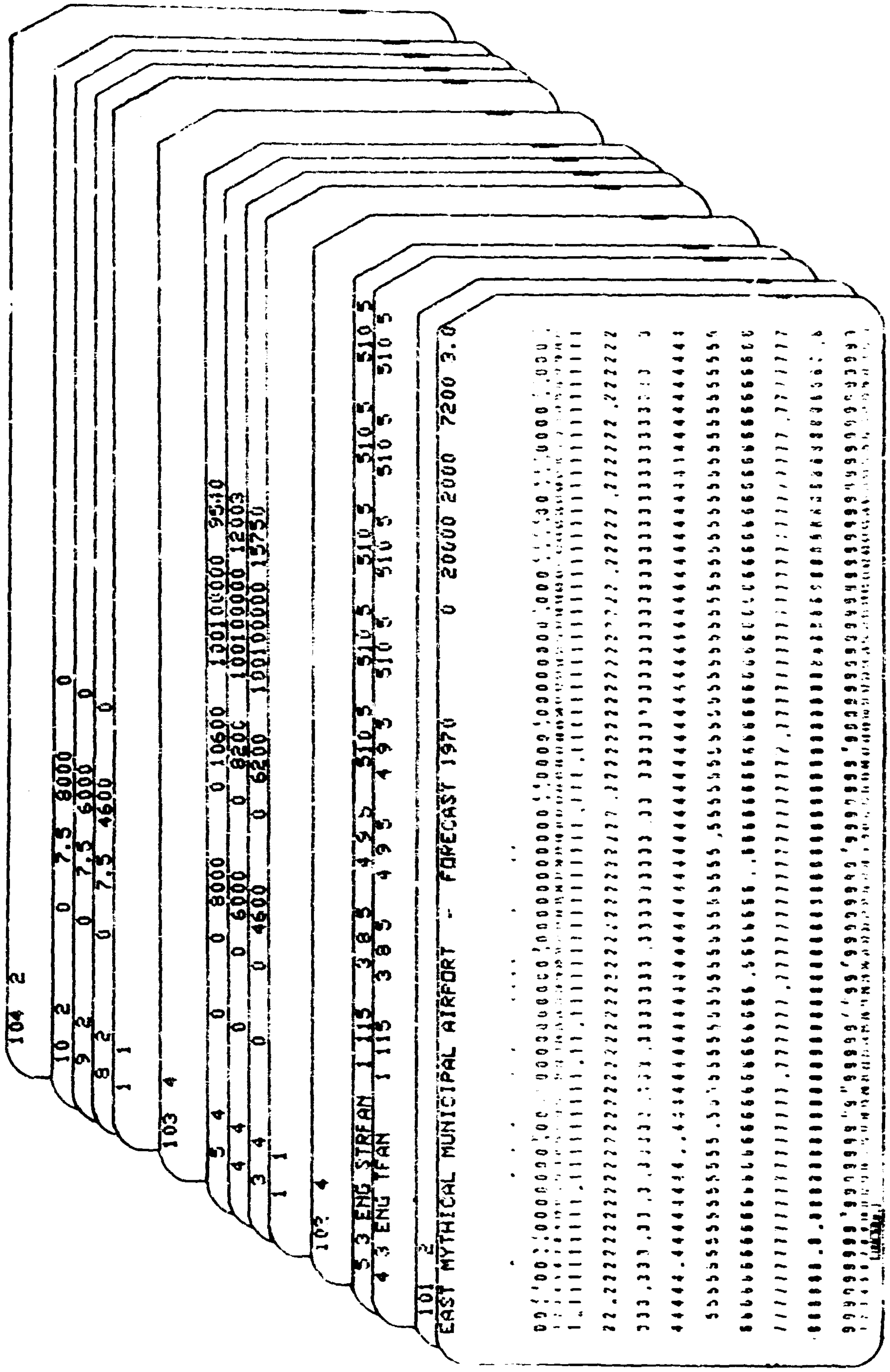


FIGURE A - 2 A. EXAMPLE OF INPUT DATA

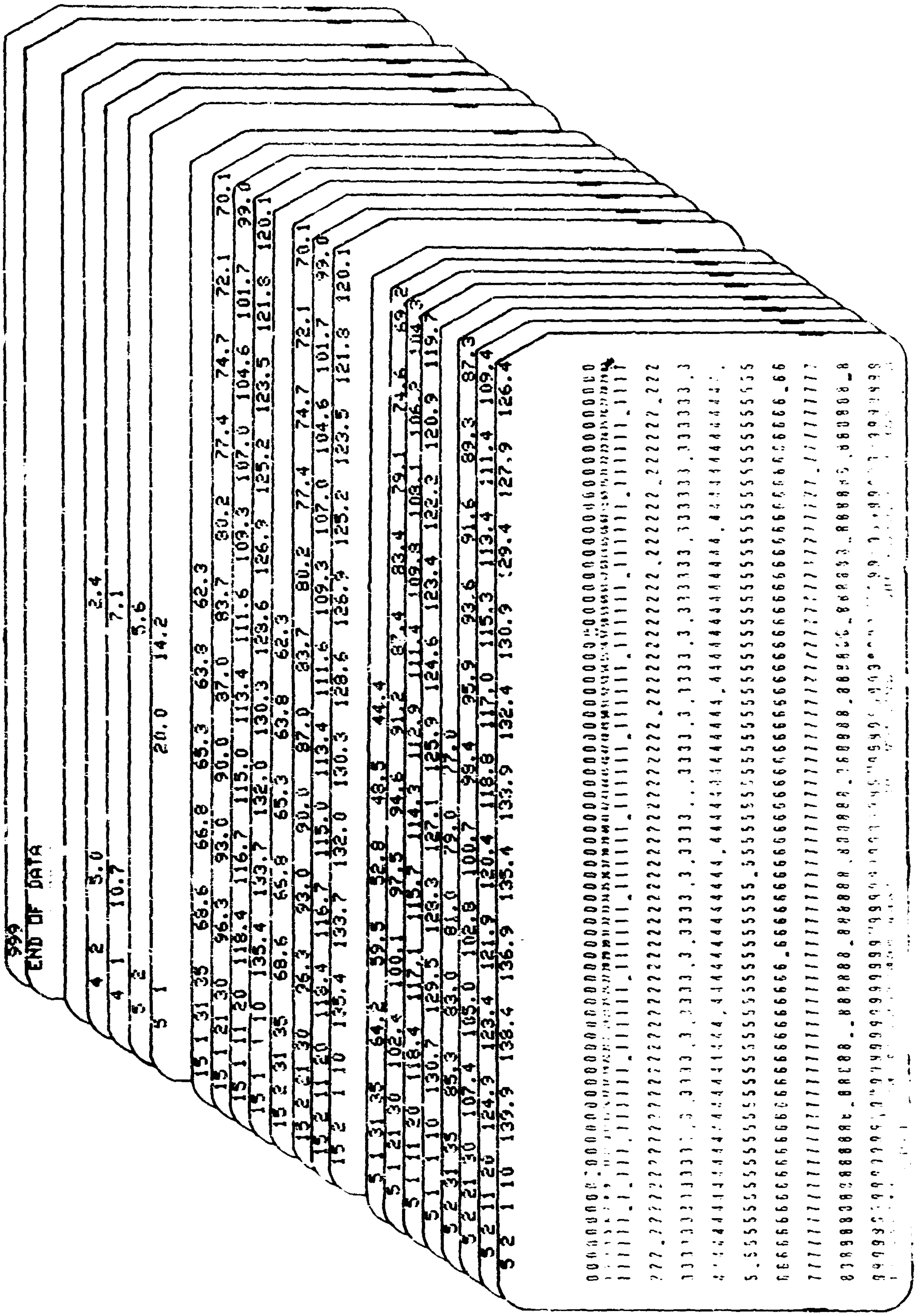


FIGURE A - 2B. EXAMPLE OF INPUT DATA

EAST MYTHICAL MUNICIPAL AIRPORT - FORECAST 1970

START AT 0.
STOP AT 30000.
INTERVAL 2000.

RUNWAY LENGTH 7200.
GLIDE SLOPE 3.0

AIRCRAFT CLASSIFICATION		TRIP LENGTH	VOLUME OF OPERATIONS	
NAME	DESCRIPTION	CATEGORY	DAYTIME	NIGHTTIME
5	3 ENG STRFAN	1	0.0	0.0
		2	0.0	0.0
		3	20.000	0.0
		4	14.200	5.600
		5	0.0	0.0
		6	0.0	0.0
		7	0.0	0.0
		8	0.0	0.0
4	3 ENG TFAN	1	10.700	5.000
		2	0.0	0.0
		3	0.0	0.0
		4	7.100	2.400
		5	0.0	0.0
		6	0.0	0.0
		7	0.0	0.0
		8	0.0	0.0

FIGURE A - 3A EXAMPLE OF COMPUTER OUTPUT - FLIGHT PATH VOLUME OF OPERATIONS LISTING (EXAMPLE 1)

EAST MYTHICAL MUNICIPAL AIRPORT - FORECAST 1970

DISTANCE TO CONTOUR IN FEET

POINT ON FLIGHT TRACK	25	30	35	40	45	50	55
0	5096	3836	2796	1906	1173	646	321
2000	4521	3386	2418	1590	920	497	236
4000	4115	2977	2061	1307	732	384	178
6000	3607	2608	1743	1053	590	302	137
8000	3362	2394	1575	994	689	365	164
10000	3601	2886	1968	1192	746	463	201
12000	4720	3146	2260	1466	809	362	0
14000	5005	3841	2487	1394	686	0	
16000	5885	4077	2422	1286	472		
18000	6896	4042	2333	1134	0		
20000	6856	3921	2220	920			
22000	6808	3840	2081	599			
24000	6751	3744	1911	0			
26000	6687	3633	1703				
28000	6615	3505	1445				
30000	6534	3360	1115				
CONTOUR CLOSING POINT ON FLIGHT TRACK				23463	17559	13807	11634

FIGURE A-3B. EXAMPLE OF COMPUTER OUTPUT - CONTOUR DISTANCE LISTING (EXAMPLE 1)

EAST MYTHICAL MUNICIPAL AIRPORT - FORECAST 1990

START AT 0.
STOP AT 10000.
INTERVAL 2000.

RUNWAY LENGTH 14000.
GLIDE SLOPE 3.0

AIRCRAFT CLASSIFICATION NAME	DESCRIPTION	TRIP LENGTH CATEGORY	VOLUME OF OPERATIONS DAYTIME	NIGHTTIME
1	4 ENG TJET	1	0.575	0.133
		2	2.844	0.356
		3	2.134	0.711
		4	2.844	0.711
		5	4.378	1.422
		6	1.995	0.0
		7	0.0	0.0
		8	0.0	0.0
2	4 ENG TFAN	1	3.407	0.537
		2	22.367	7.111
		3	12.300	1.422
		4	18.433	2.844
		5	17.352	2.485
		6	2.560	0.356
		7	0.711	0.356
		8	0.0	0.0
3	4 ENG STFFAN	1	0.459	0.115
		2	1.422	0.569
		3	0.0	0.0
		4	0.0	0.0
		5	4.115	2.276
		6	2.134	0.0
		7	0.0	0.0
		8	1.351	0.0
4	3 ENG TFAN	1	1.744	0.212
		2	37.761	4.267
		3	2.134	0.675
		4	1.356	0.142
		5	0.0	0.0
		6	0.0	0.0
		7	0.0	0.0
		8	0.0	0.0

EAST MYTHICAL MUNICIPAL AIRPORT - FORECAST 1990

AIRCRAFT CLASSIFICATION NAME	DESCRIPTION	TRIP LENGTH CATEGORY	VOLUME OF OPERATIONS DAYTIME	NIGHTTIME
5	2 ENG STFFAN	1	1.224	0.193
		2	26.981	4.267
		3	2.487	0.356
		4	0.0	0.0
		5	0.0	0.0
		6	0.0	0.0
		7	0.0	0.0
		8	0.0	0.0
6	2 ENG TFAN	1	1.744	0.212
		2	37.761	4.267
		3	2.134	0.675
		4	1.956	0.142
		5	0.0	0.0
		6	0.0	0.0
		7	0.0	0.0
		8	0.0	0.0
7	4 ENG JUMFAN	1	1.315	0.326
		2	2.062	0.356
		3	0.0	0.0
		4	0.0	0.0
		5	23.922	6.756
		6	5.689	0.711
		7	0.0	0.0
		8	0.0	0.0
8	AIR PLS	1	1.209	0.207
		2	11.236	1.422
		3	0.0	0.0
		4	0.0	0.0
		5	17.779	3.555
		6	0.0	0.0
		7	0.0	0.0
		8	0.0	0.0

FIGURE A-4A. EXAMPLE OF COMPUTER OUTPUT - FLIGHT PATH VOLUME OF OPERATIONS LISTING (EXAMPLE 2)

EAST MYTHICAL MUNICIPAL AIRPORT - FORECAST 1990

POINT ON FLIGHT TRACK	DISTANCE TO CONTOUR IN FEET 25	30	35	40	45	50	55
0	7740	6165	4769	3697	2903	2021	1333
2000	6646	5219	4048	3112	2277	1550	951
4000	5665	4435	3422	2567	1787	1144	654
6000	5302	4156	3204	2362	1619	1065	803
8000	5267	4596	3951	3367	2322	1442	880
10000	7369	6622	5903	3685	2321	1363	711
12000	9647	8809	6161	3729	2220	1185	394
14000	11972	10867	6348	3679	2069	889	99
16000	14298	10865	6307	3555	1840	472	0
18000	16637	11197	6298	3391	1537	0	
20000	18649	11309	6194	3192	1142		
22000	18604	11268	6066	2953	637		
24000	18647	11353	5919	2665	0		
26000	19309	11311	5751	2316			
28000	19493	11222	5560	1992			
30000	19431	11114	5345	1371			
32000	19431	10996	5103	744			
34000	19428	10868	4830	0			
36000	19349	10728	4523				
38000	19277	10577	4175				
40000	19194	10414	3781				
42000	19099	10239	3327				
44000	18996	10052	2797				
46000	18886	9851	2163				
48000	18770	9636	1366				
50000	18648	9406	0				
52000	18520	9161					
54000	18385	8859					
56000	18243	8616					
58000	18095	8315					
60000	17940	7992					
62000	17777	7644					
64000	17608	7269					
66000	17431	6862					
68000	17246	6419					
70000	17053	5934					
72000	16853	5393					
74000	16644	4812					
76000	16427	4113					
78000	16202	3271					
80000	15967	2178					
82000	15724	0					
84000	15472						
86000	15211						
88000	14942						
90000	14664						
92000	14379						
94000	14088						
96000	13791						
98000	13492						
100000	13190						
CONTOUR CLOSING POINT ON FLIGHT TRACK		81726	49709	33175	23584	17763	14480

FIGURE A-48. EXAMPLE OF COMPUTER OUTPUT - CONTOUR DISTANCE LISTING (EXAMPLE 2)

APPENDIX B

SUMMARY OF BASIC NOISE EXPOSURE FORECAST CONCEPTS AND EQUATIONS

APPENDIX B

SUMMARY OF BASIC NOISE EXPOSURE FORECAST CONCEPTS AND EQUATIONS

The NEF procedures have evolved from predecessor procedures described in references 1 and 2. A basic assumption in these predecessor procedures which also underlies several of the procedures used abroad for estimating noise exposures is that estimates of the total noise environment resulting from aircraft operations must be based upon a measure of noise that can be closely related to subjective reactions of people to the noise plus corrections to take into account the number of occurrences of noise intrusions per daytime and per nighttime period.

The NEF procedures outlined in this document are based on the parallel studies of references 3 and 4. The current procedure combines elements of the procedures from the predecessor studies hence certain calculation steps or constants given in this document differ from these given in either reference 3 or 4.

In calculation of NEF, aircraft noise levels are expressed in terms of the effective perceived noise level (EPNL) as defined in reference 5. The EPNL represents an integration of the tone-corrected perceived noise level (determined at half second intervals) over the upper 10 PNdB of a flyover time history. The EPNL value, therefore includes adjustments for the effective duration of the flyover signal and for the presence of discrete frequency components.

In estimating the total noise exposure near an airport or flight path one is faced with predicting or calculating the noise levels resulting from a wide range of a variety of aircraft. To simplify the calculations and provide a systematic method for calculation, it is convenient to group the aircraft in classes based upon consideration of the aircraft noise characteristics and takeoff and landing performance. Each class is assigned a description of the noise in terms of a set of EPNL vs distance curves and a set of takeoff and landing profiles.

In estimating the effective perceived noise levels, it is assumed that the noise level at any point under or to one side of the aircraft flight path (or to either side of the path during takeoff roll) depends upon the following:

a) Type of aircraft (particularly the type of engine),

b) Type of operation (takeoff, landing, or takeoff roll),

c) Engine power setting,

d) Distance to aircraft (dependent upon the aircraft takeoff, and landing profiles as well as the location of the flight track with respect to the ground observer).

For a given class of aircraft at a particular power setting (i.e. takeoff power) it is assumed that the aircraft noise characteristics may be described by a single EPNL vs distance curve.

The total noise exposure produced by aircraft operations at a given point is viewed as being composed of the effective perceived noise levels produced by different aircraft classes flying along different flight paths. For aircraft class i on flight path j, the NEF (ij) can be expressed as

$$\text{NEF (ij)} = \text{EPNL (ij)} + 10 \log \left[\frac{\text{N (day) (ij)}}{\text{K (day)}} + \frac{\text{N (night) (ij)}}{\text{K (night)}}\right] - C \qquad \text{(Eq. 1)}$$

where

NEF (ij) = Noise Exposure Forecast value produced by aircraft class (i) along flight path segment (j).

EPNL (ij) = Effective perceived noise level produced at the given point by aircraft class (i) flying along flight path segment (j)

K = Constant normalizing the adjustment in NEF values due to volume of operations. Different values of K are used for daytime and nighttime movements.

C = Arbitrary normalization constant.

K (day) is chosen so that for 20 movements of a given aircraft class per daytime period, the adjustment for number of operations is zero. Hence,

$$10 \log \frac{20}{\text{K (day)}} = 0 \qquad \text{K (day)} = 20$$

K (night) is chosen such that for the same average number of operations per hour during daytime or nighttime periods the NEF value for nighttime operations would be 10 units higher than for daytime operation. Hence,

$$10 = 10 \log \frac{K\ (day)}{K\ (night)} \quad \frac{9}{15}$$

where 9 and 15 are the number of hours in the nighttime and daytime periods respectively, and

$$K\ (night) = 1.2$$

The value assigned to C is 75. Choice of this value is based upon two considerations.

First, it is desirable that the number assigned to the NEF values be distinctly different in magnitude from the effective perceived noise level so that there is little likelihood of confusing effective perceived noise levels with NEF values. A second aspect is the desirability of selecting a normalization factor that will roughly indicate the size of the NEF value above some threshold value, indicating the emergence of the noise exposure from levels which would have little or no influence on most types of land usage.

With the above choices for values of K and C, Eq. (1) becomes

$$NEF\ (ij) = EPNL\ (ij) + 10 \log [N\ (day)\ (ij) + 16.67\ N\ (night)\ (ij)] - 88 \qquad \text{(Eq. 2)}$$

The total NEF at the given ground position may be determined by summation on an energy basis of all the individual NEF (ij) values:

$$NEF = 10 \log \sum_i \sum_j \text{antilog} \frac{NEF\ (ij)}{10} \qquad \text{(Eq. 3)}$$

APPENDIX C

FLOW DIAGRAMS AND PROGRAM LISTINGS

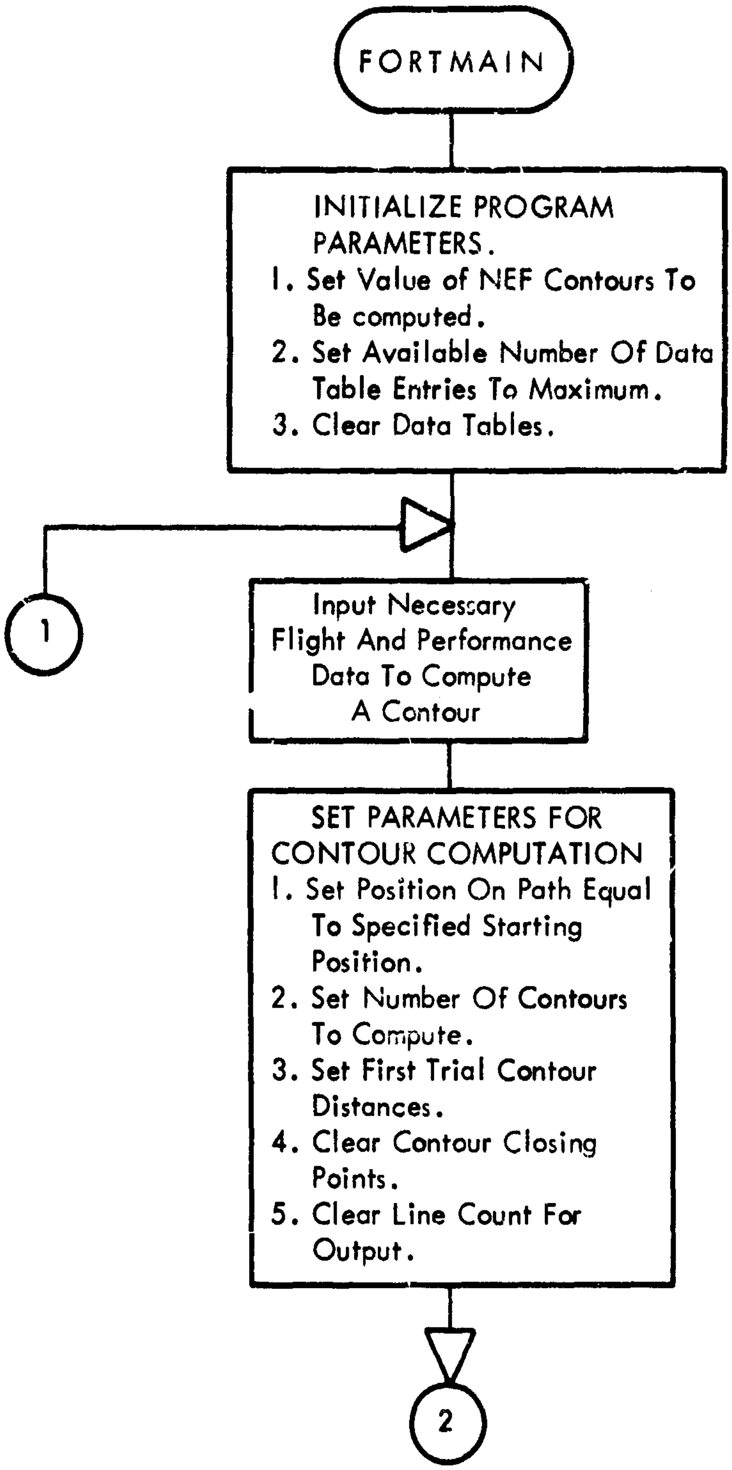


FIGURE C-1A. FLOW DIAGRAM - NEF CONTROL ROUTINE

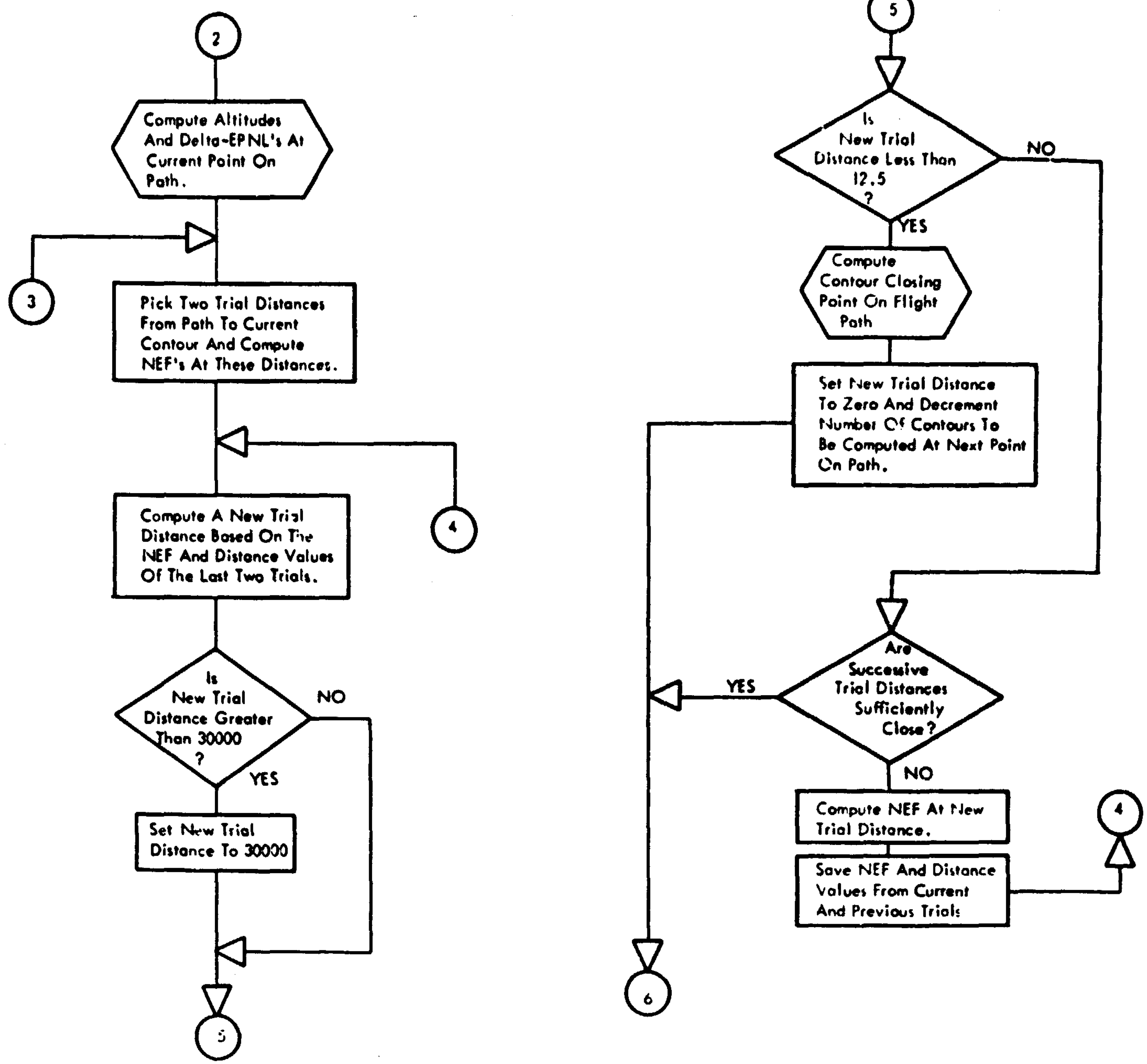


FIGURE C-1B. FLOW DIAGRAM - NEF CONTRO: ROUTINE

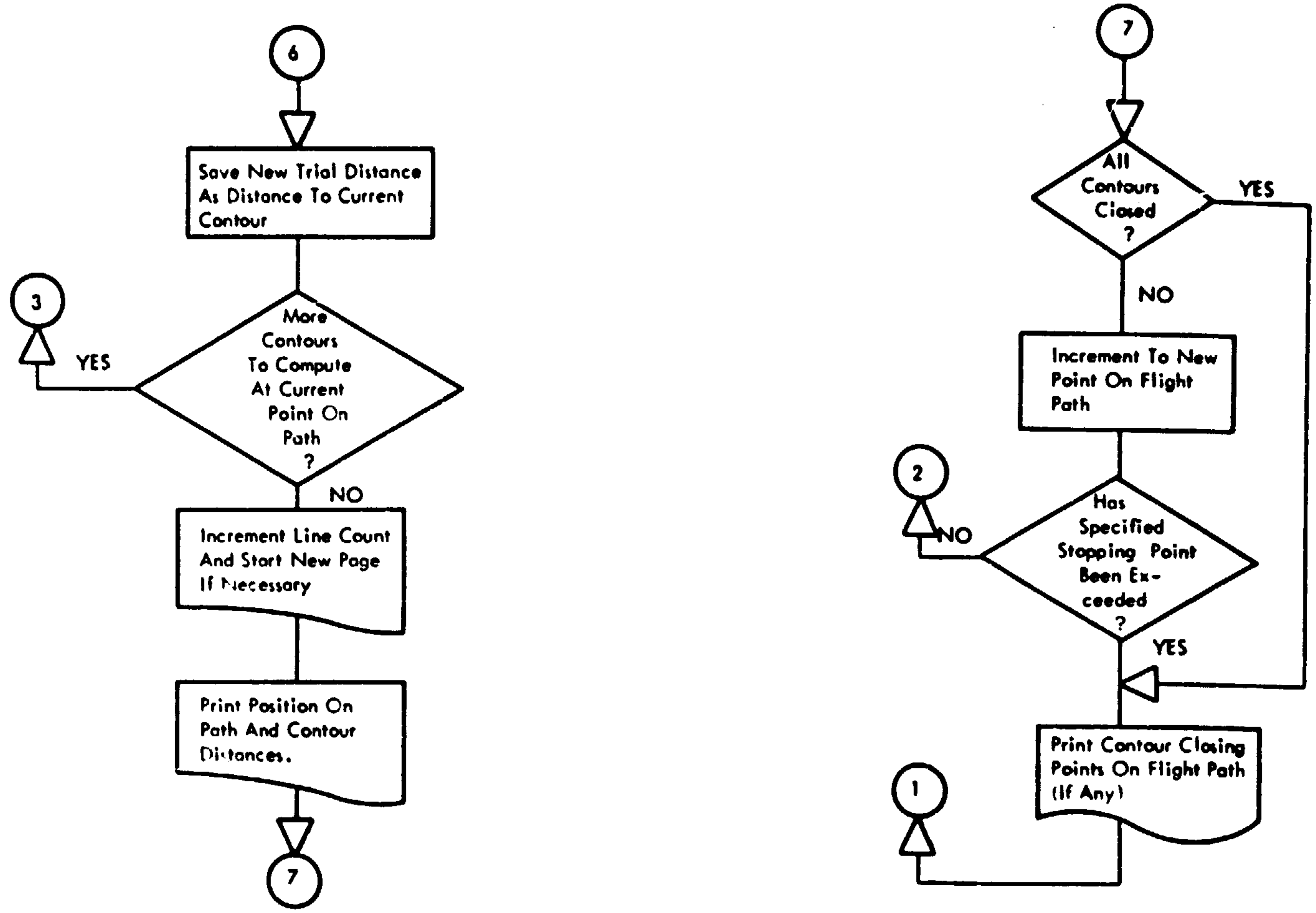


FIGURE C-1C. FLOW DIAGRAM - NEF CONTROL ROUTINE

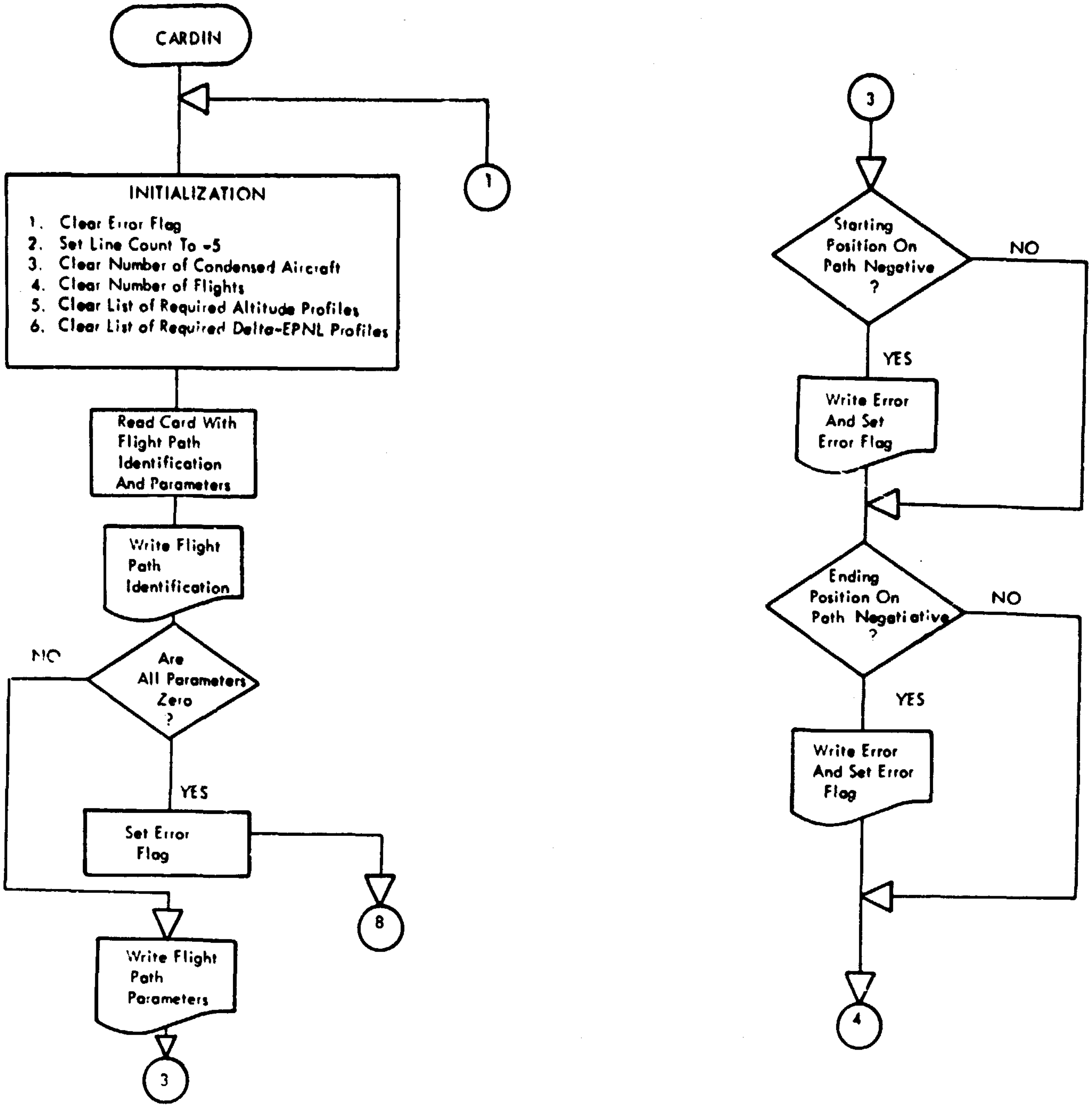


FIGURE C-2A. SUBROUTINE CARDIN.

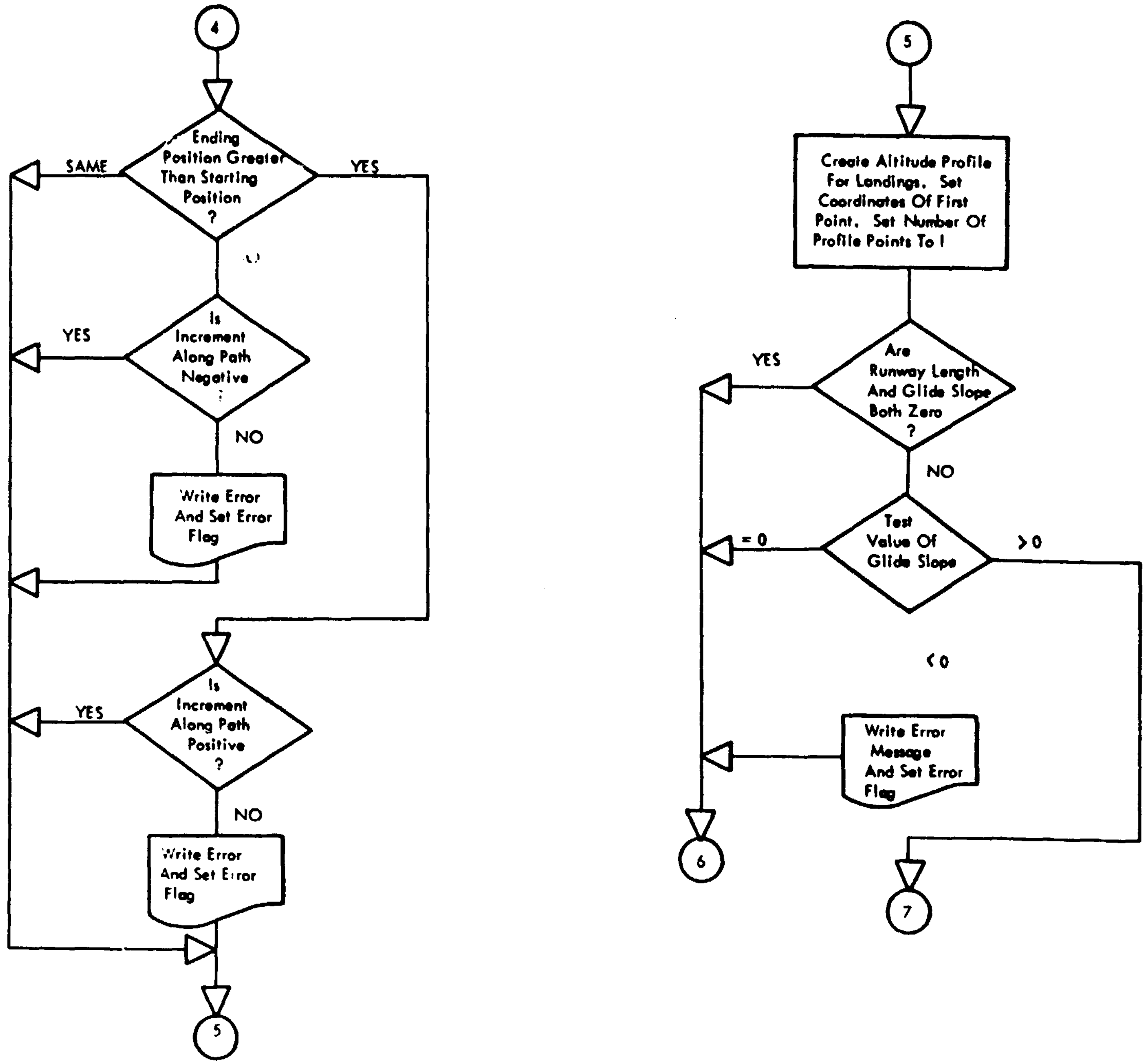


FIGURE C-2B. SUBROUTINE CARDIN

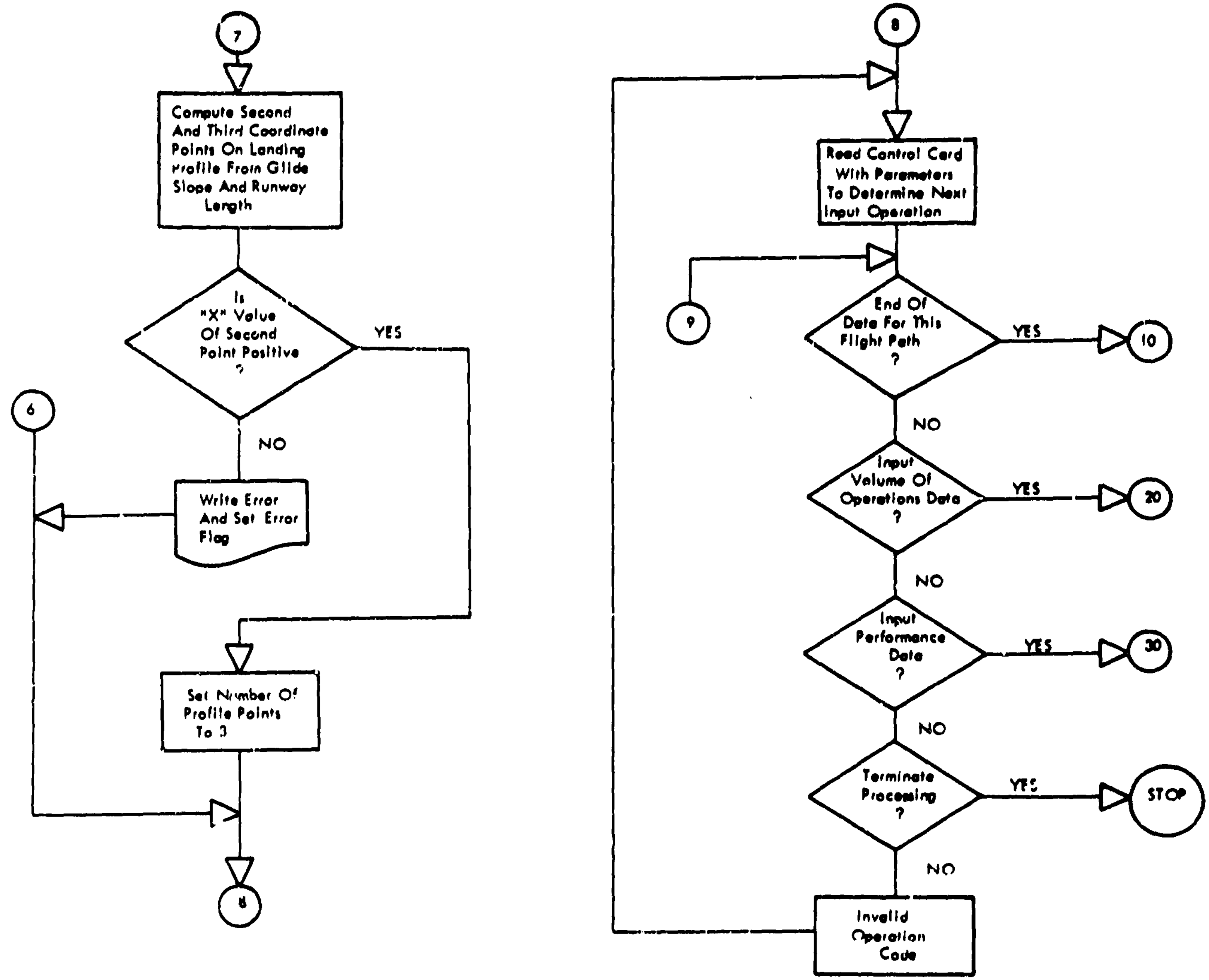


FIGURE C-2C. SUBROUTINE CARDIN

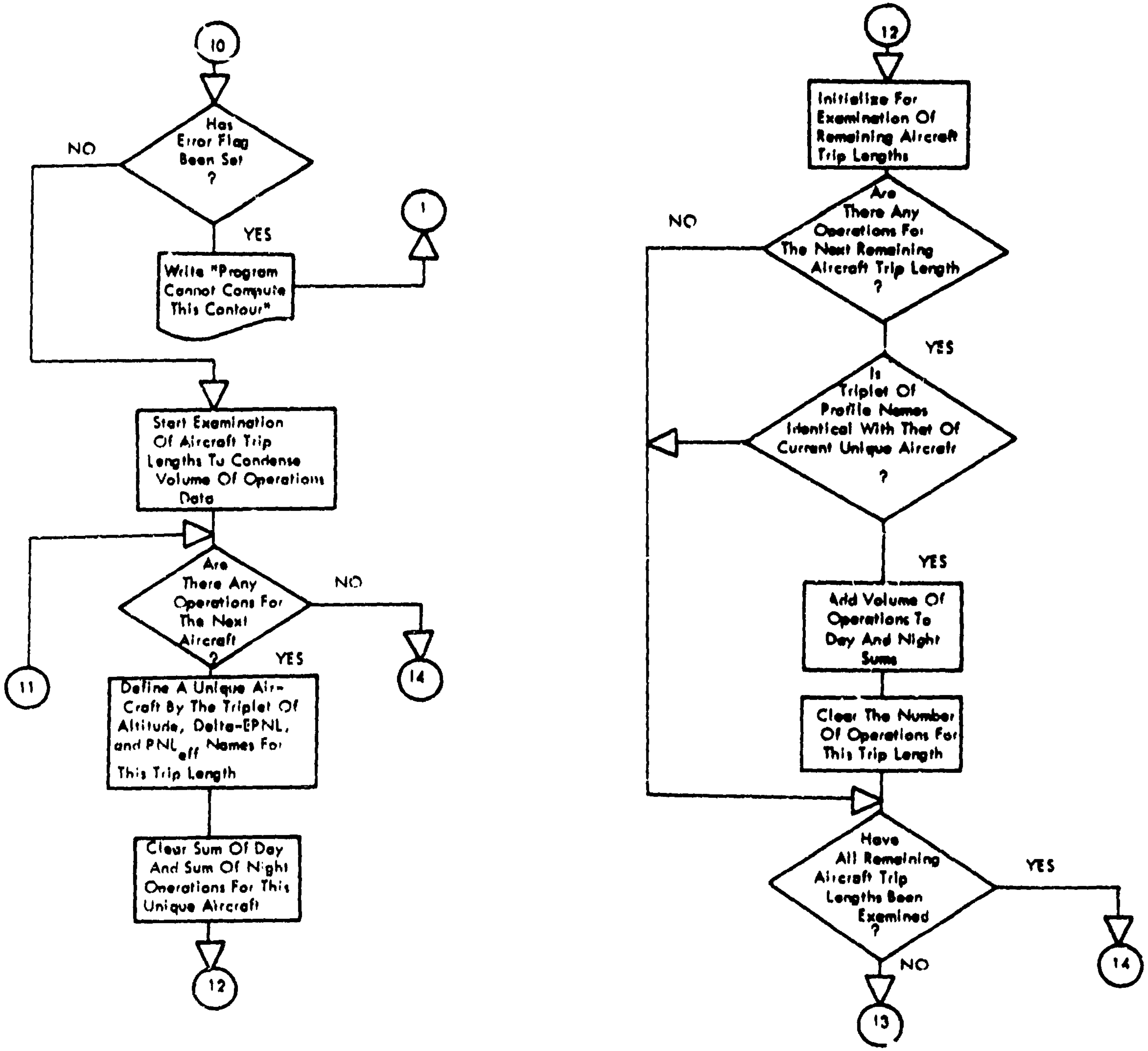


FIGURE C-2D. SUBROUTINE CARDIN

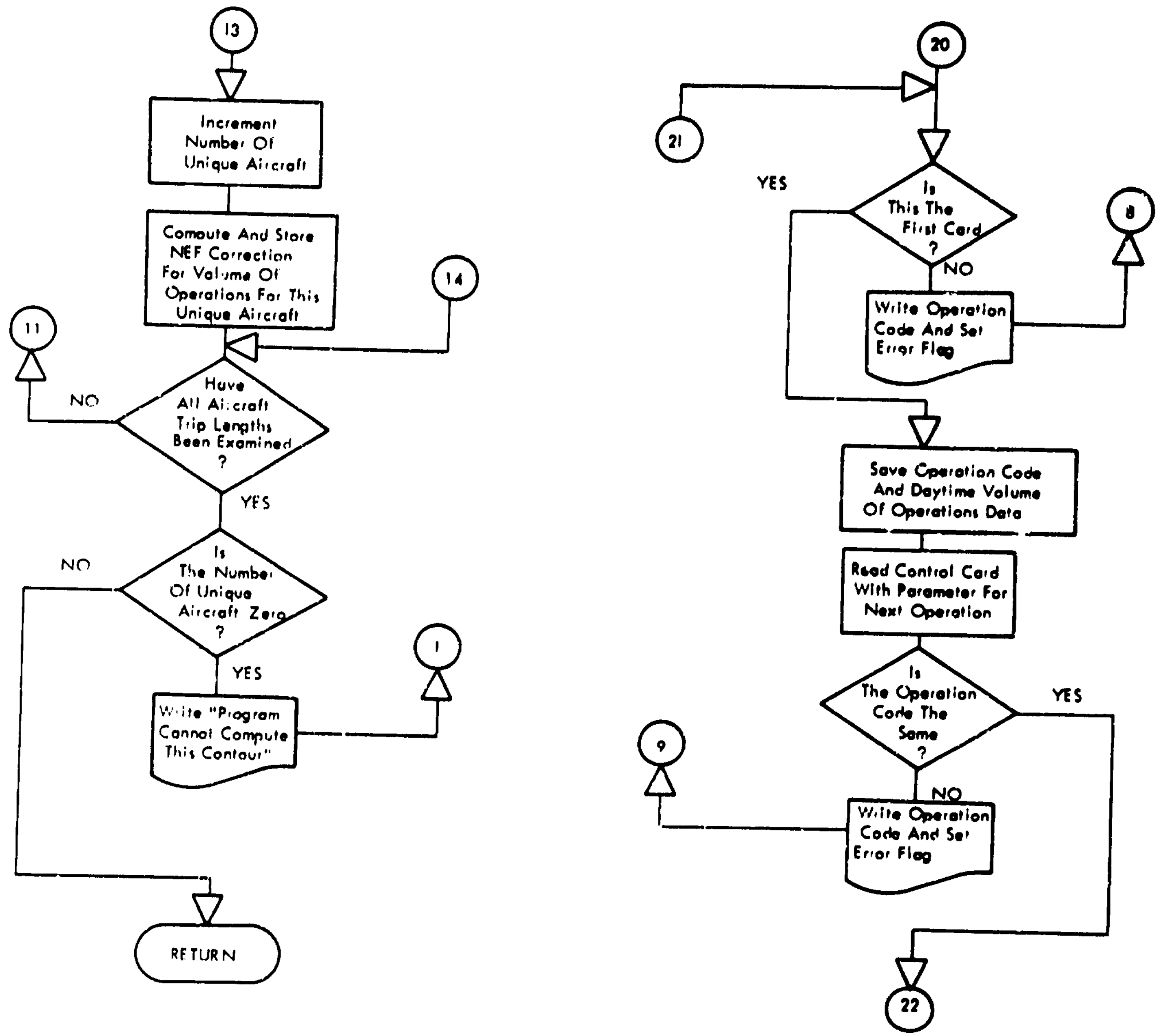


FIGURE C-25. SUBROUTINE CARDIN

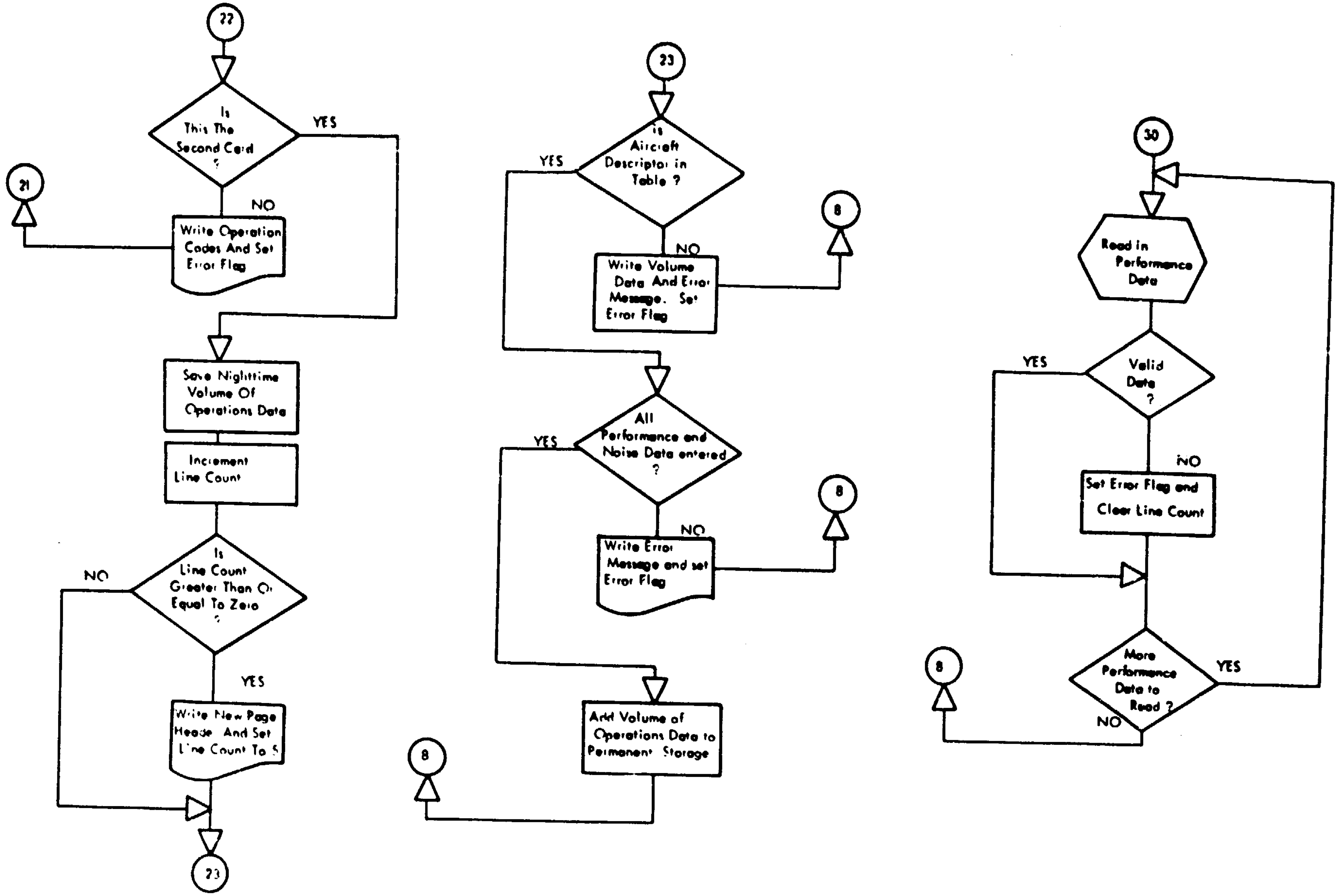
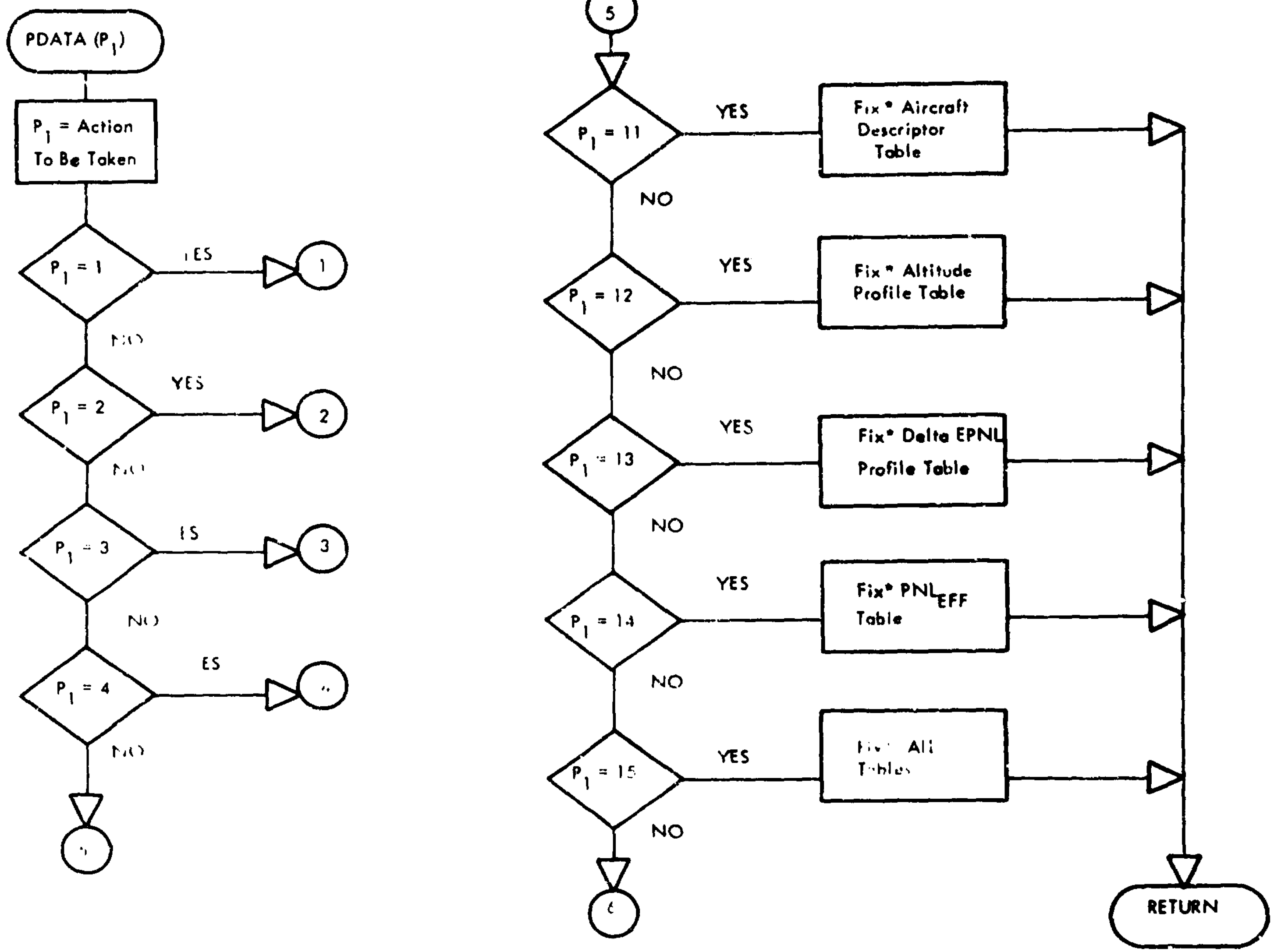
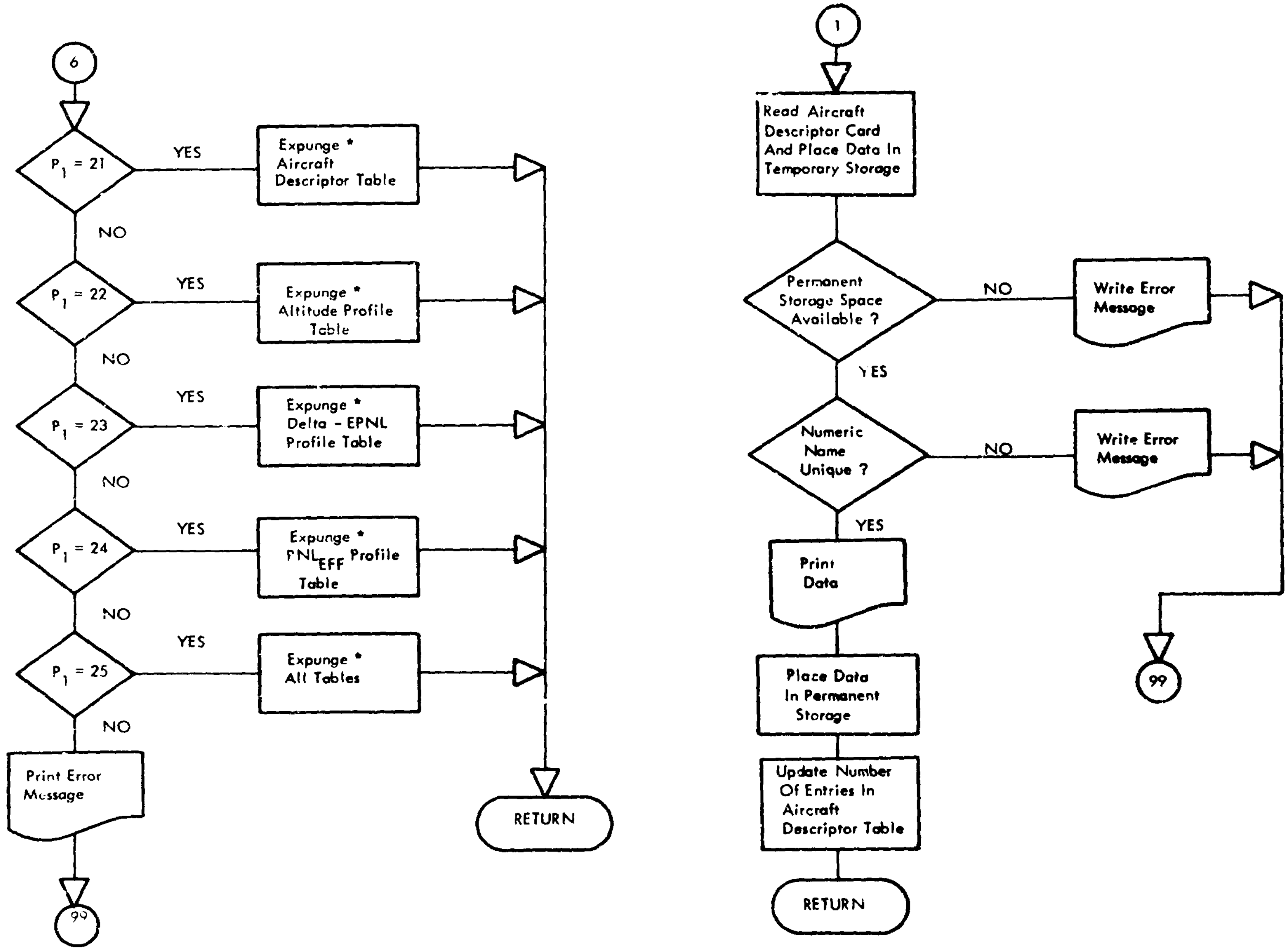


FIGURE C-2F. SUBROUTINE CARDIN



* Fix Command Prevents Data Currently In Tables From Being Deleted By Expunge Command.

FIGURE C-3A. FLOW DIAGRAM - SUBROUTINE PDATA



* Expunge Command Deletes Data Table Entries Not Fixed

FIGURE C-3B. FLOW DIAGRAM - SUBROUTINE PDATA

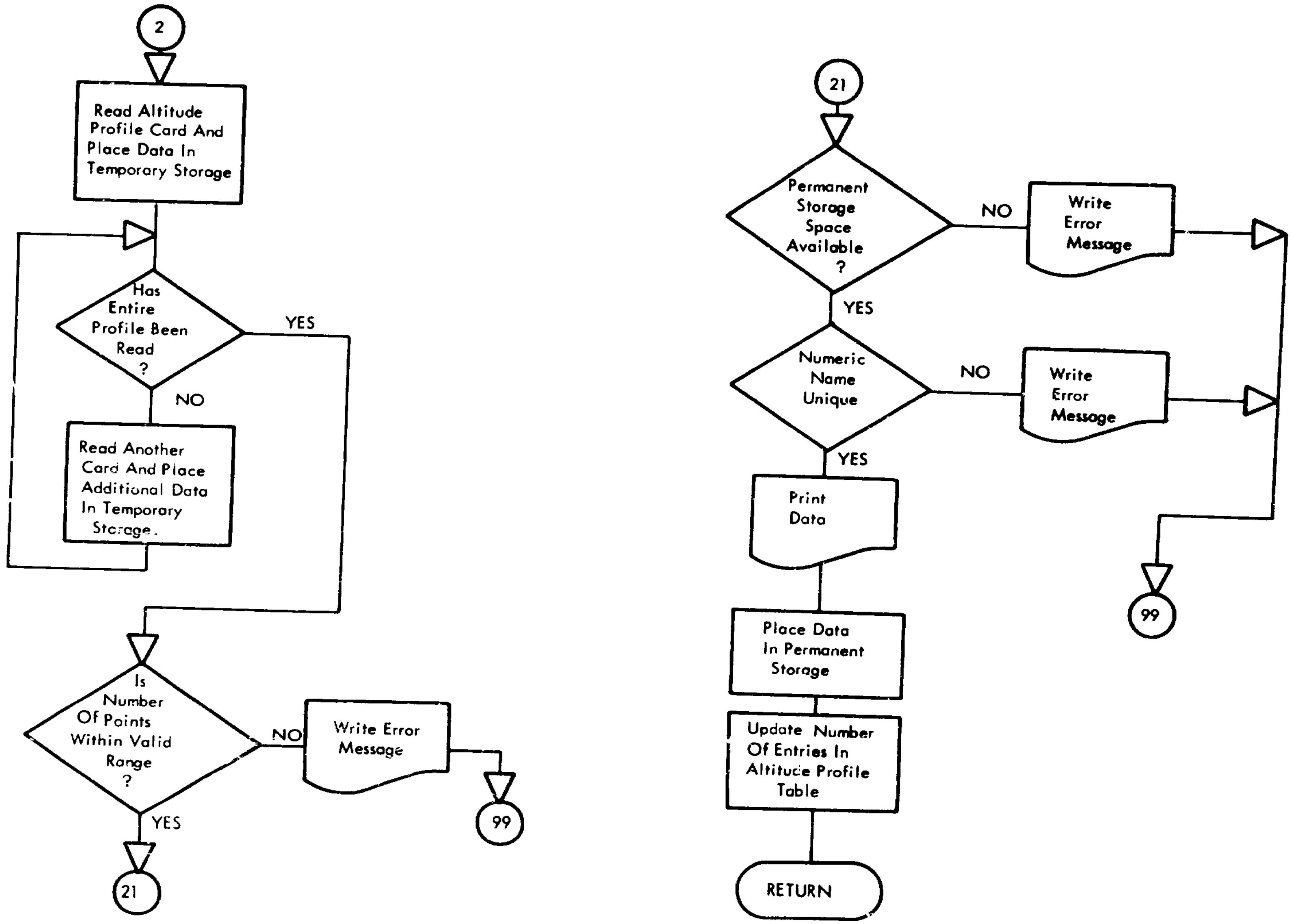


FIGURE C-3C. FLOW DIAGRAM - SUBROUTINE PDATA

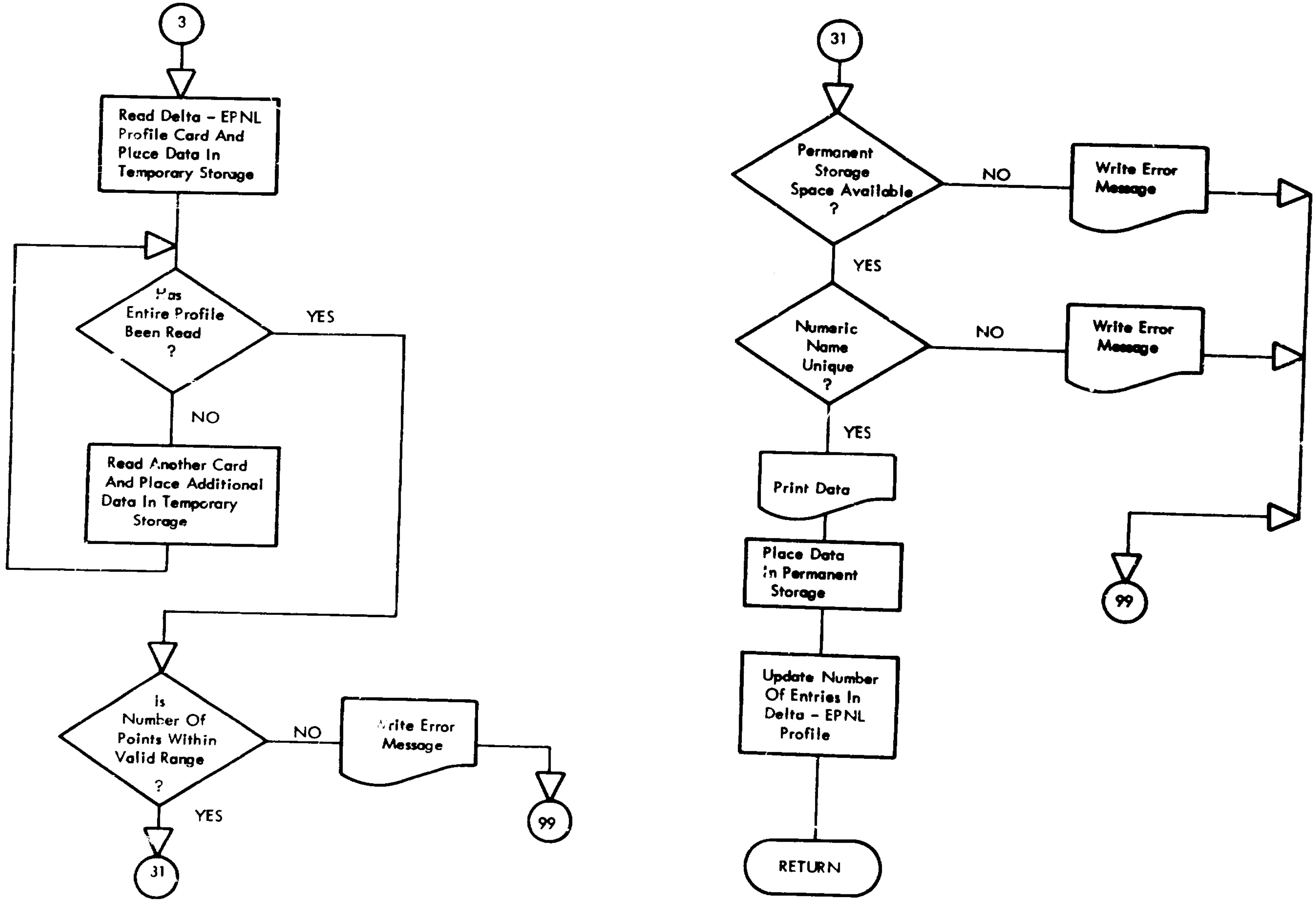


FIGURE C-3D. FLOW DIAGRAM - SUBROUTINE PDATA

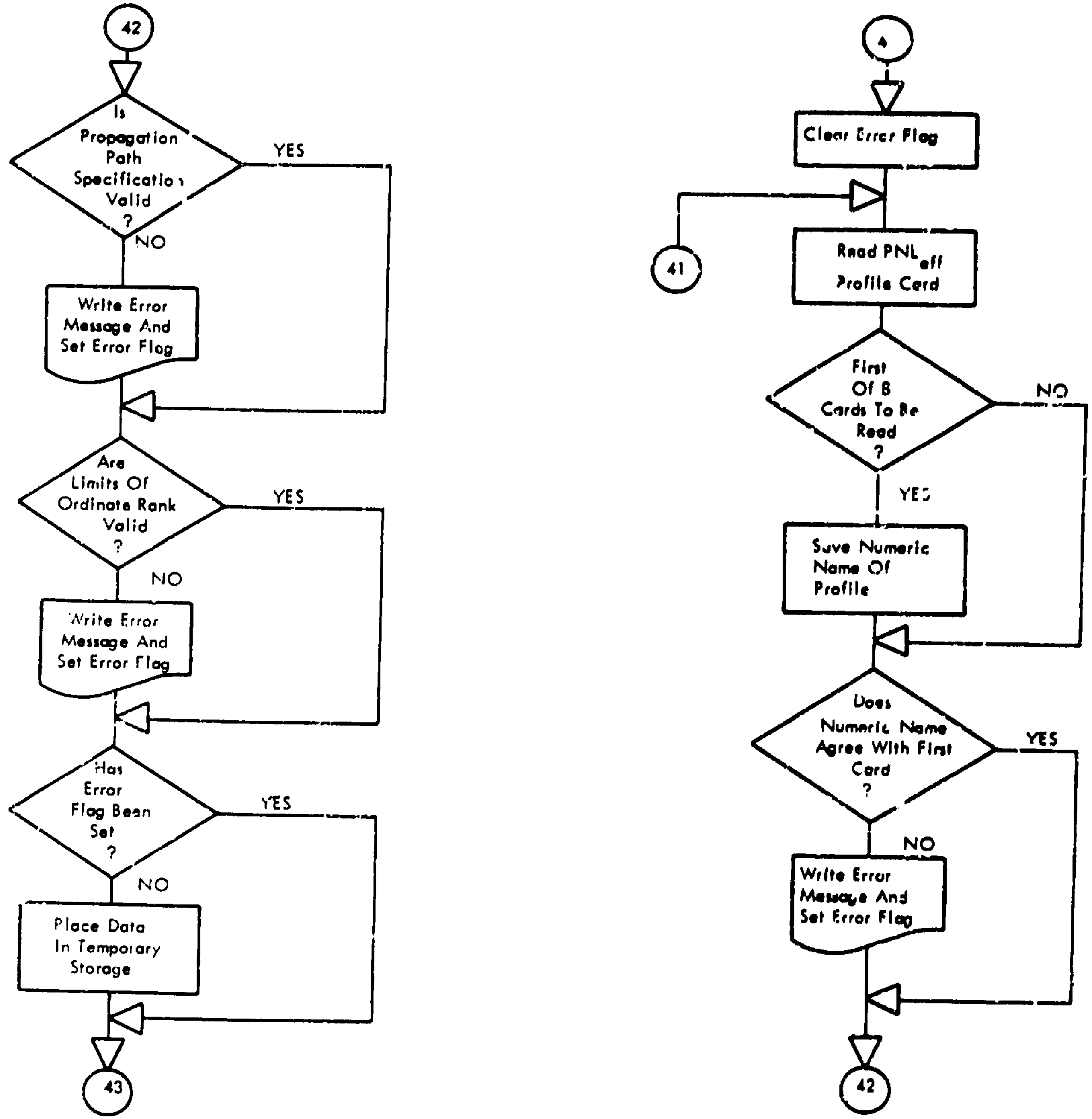


FIGURE C-3E. FLOW DIAGRAM - SUBROUTINE PDATA

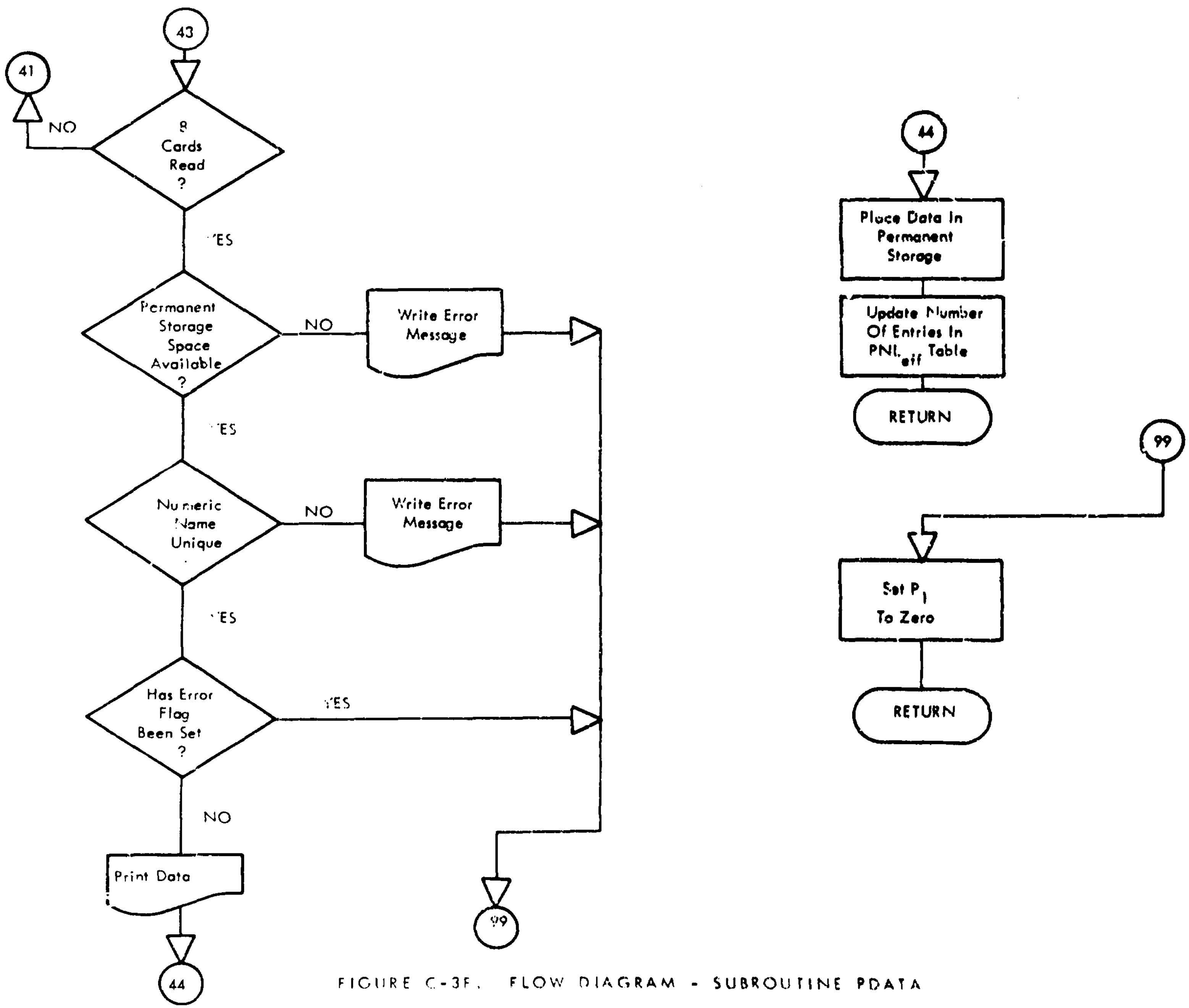


FIGURE C-3F. FLOW DIAGRAM - SUBROUTINE PDATA

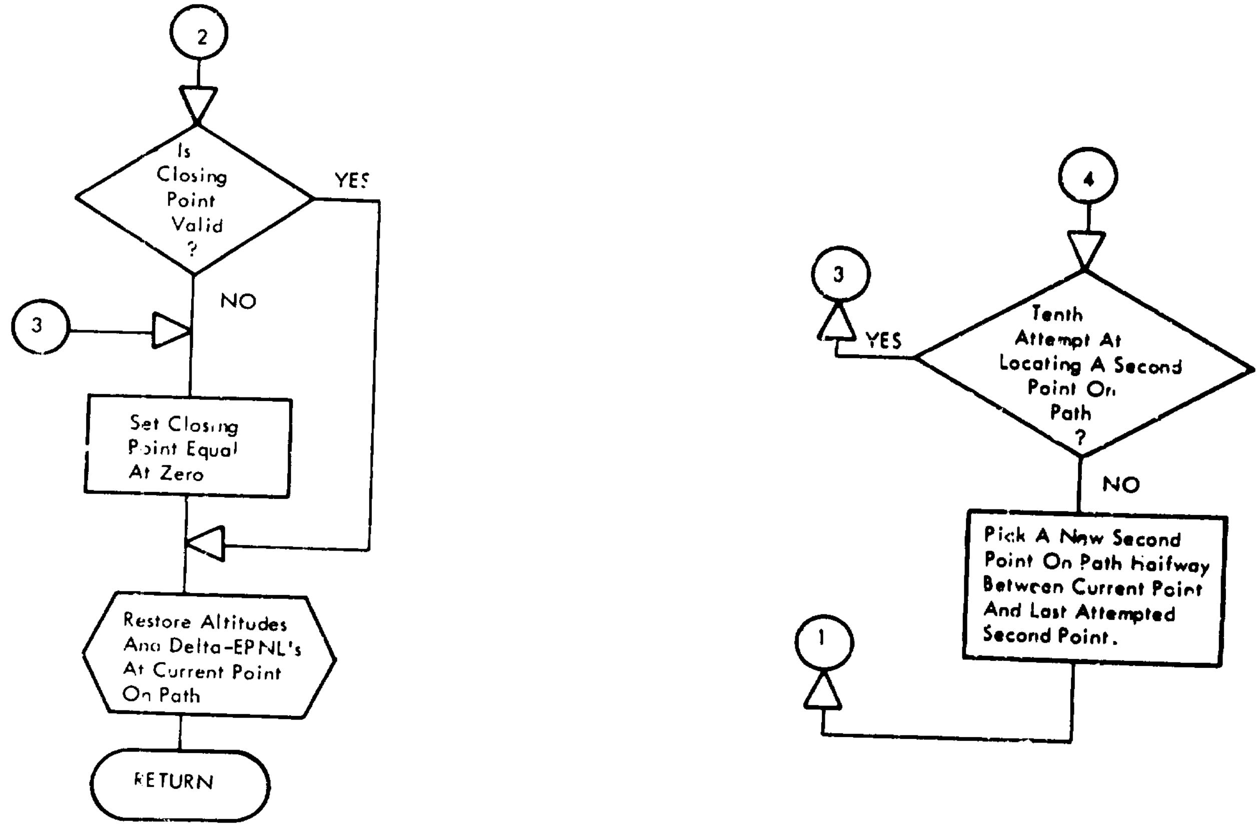


FIGURE C-4B. FLOW DIAGRAM - SUBROUTINE LASTPT

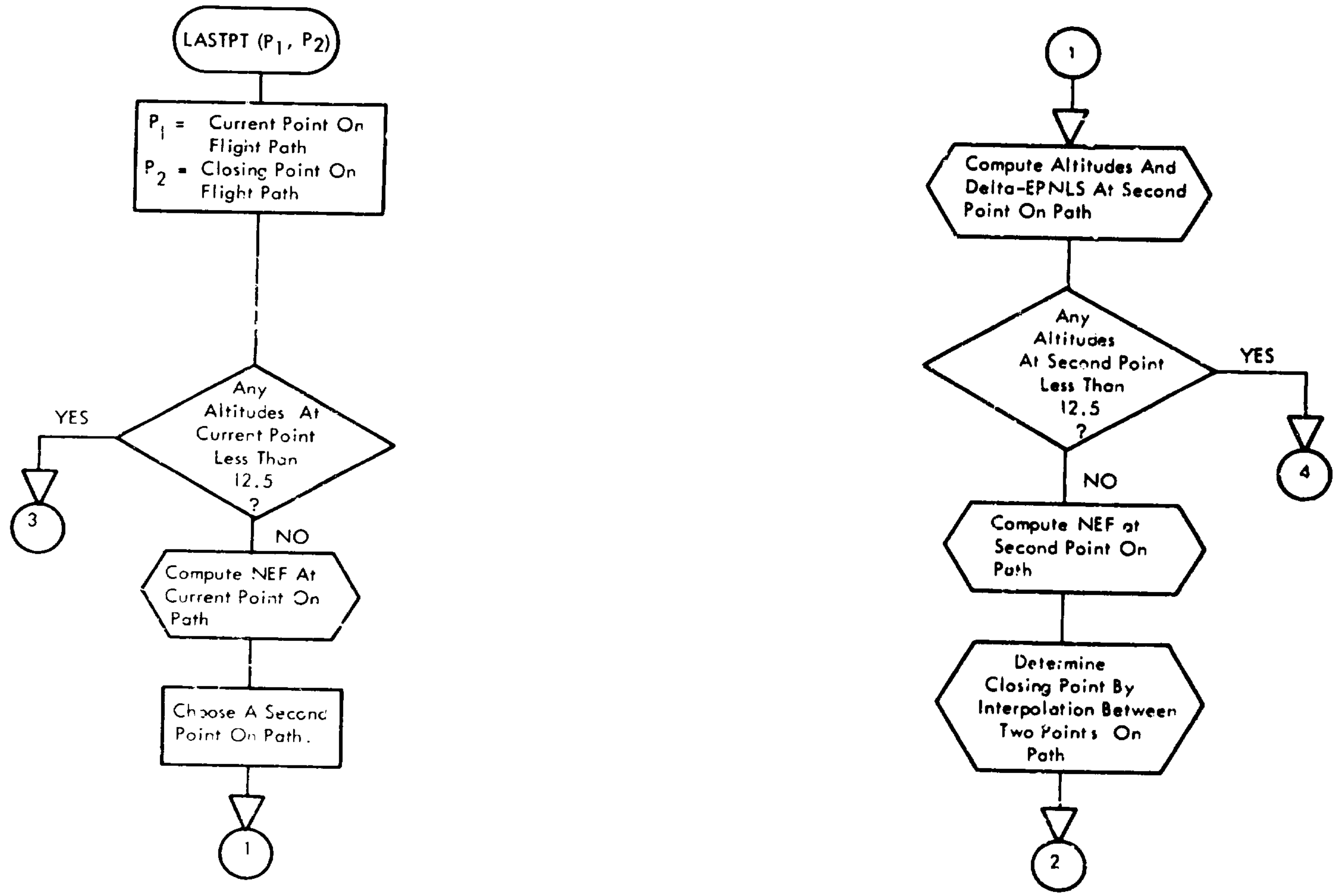


FIGURE C-4A. FLOW DIAGRAM - SUBROUTINE LASTPT

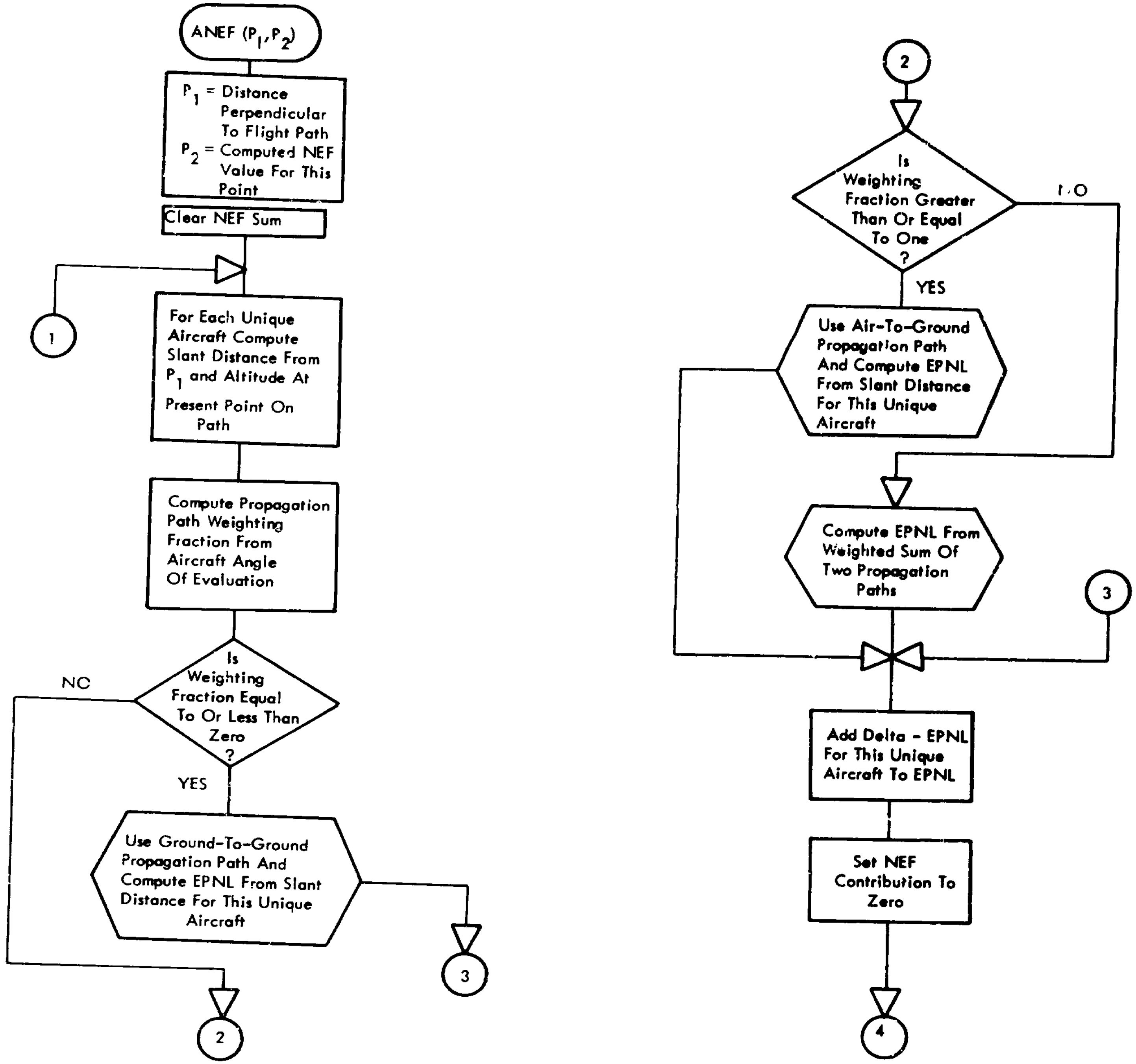


FIGURE C-5A. FLOW DIAGRAM - SUBROUTINE ANEF

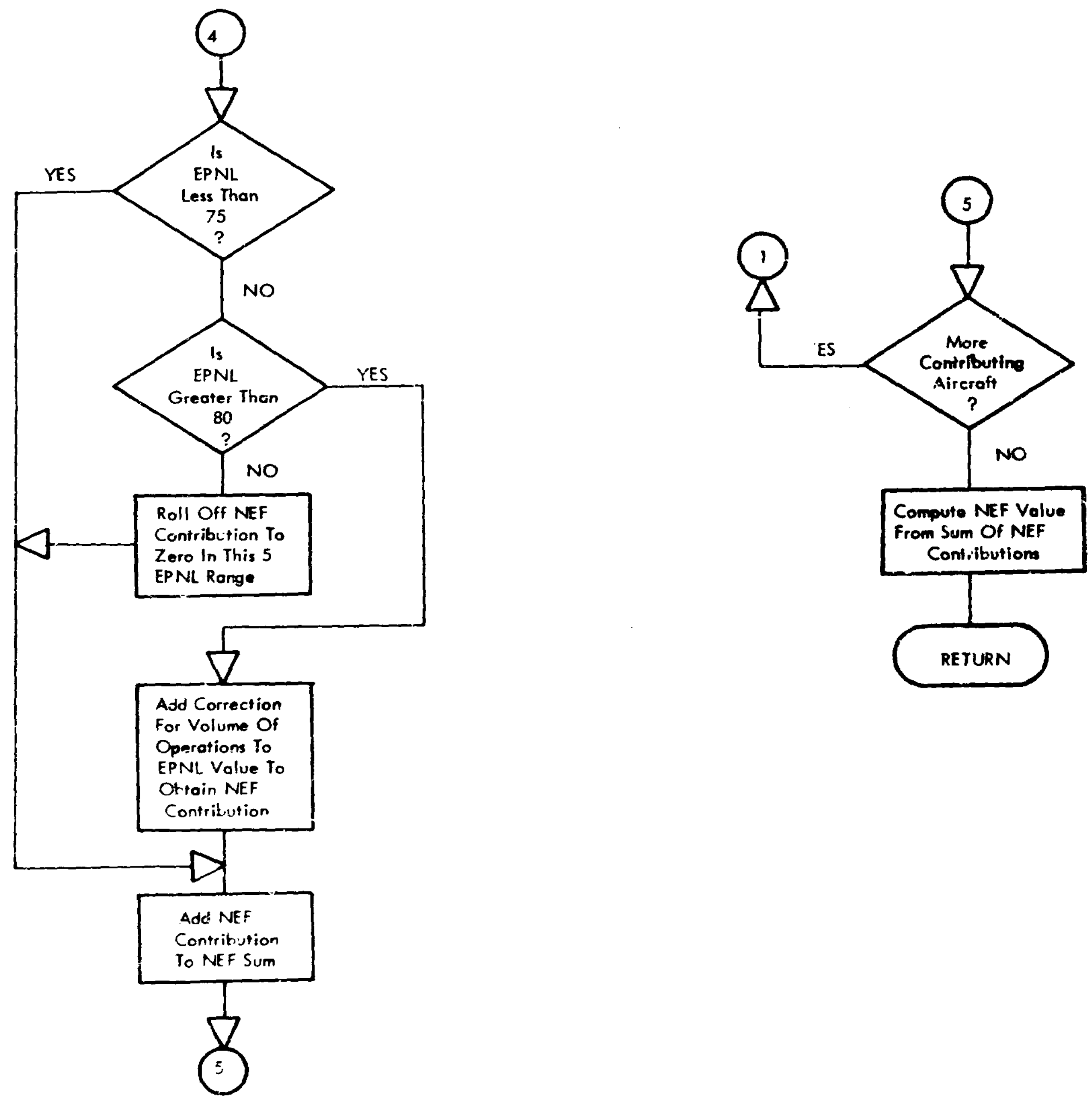


FIGURE C-5B FLOW DIAGRAM - SUBROUTINE ANEF

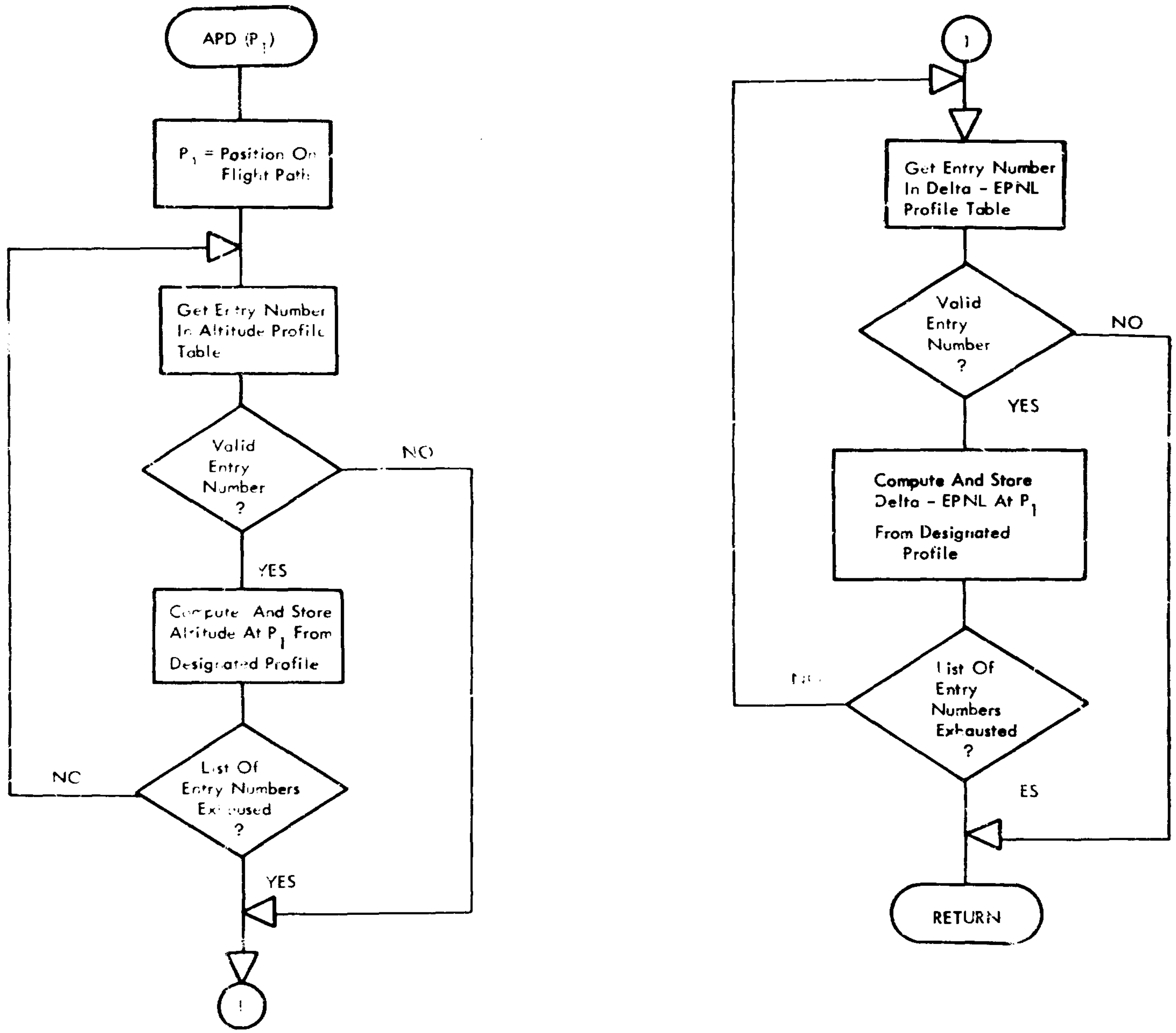


FIGURE C-5. FLOW DIAGRAM - SUBROUTINE APD

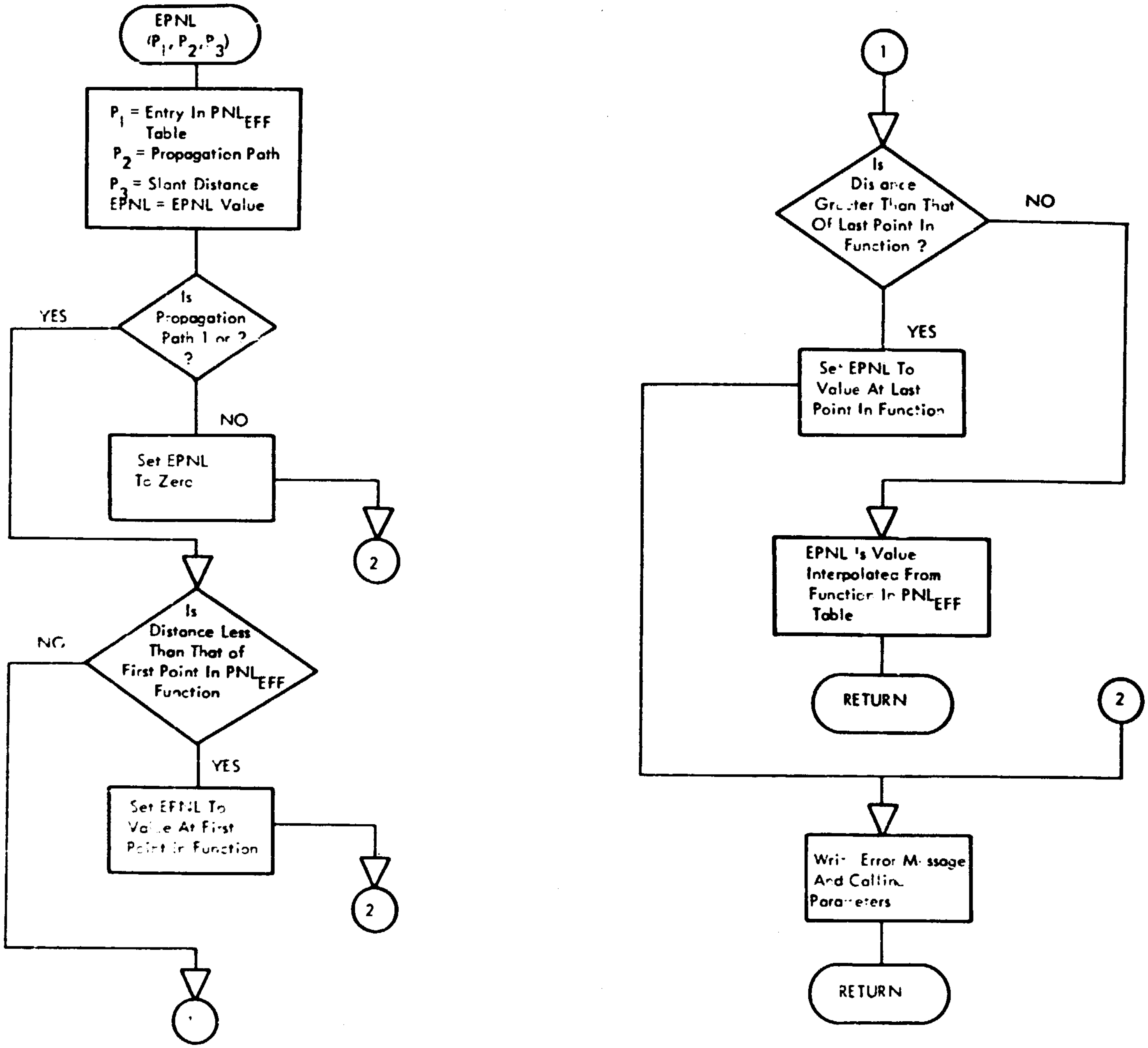


FIGURE - FLOW DIAGRAM - FUNCTION SUBPROGRAM EPNL

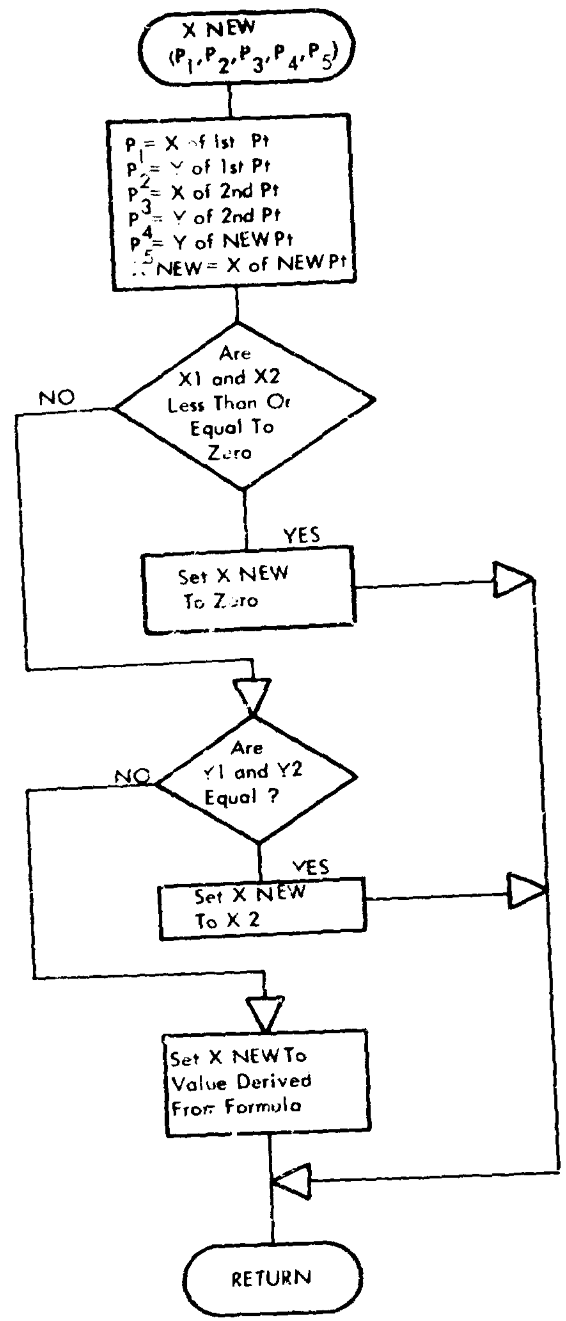


FIGURE C-8. FLOW DIAGRAM - FUNCTION SUBPROGRAM X NEW

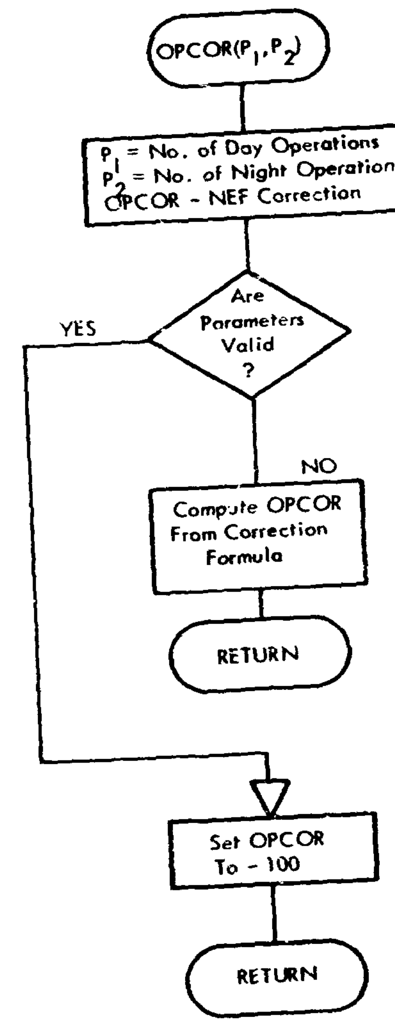


FIGURE C-9. FLOW DIAGRAM - FUNCTION SUBPROGRAM OPCOR

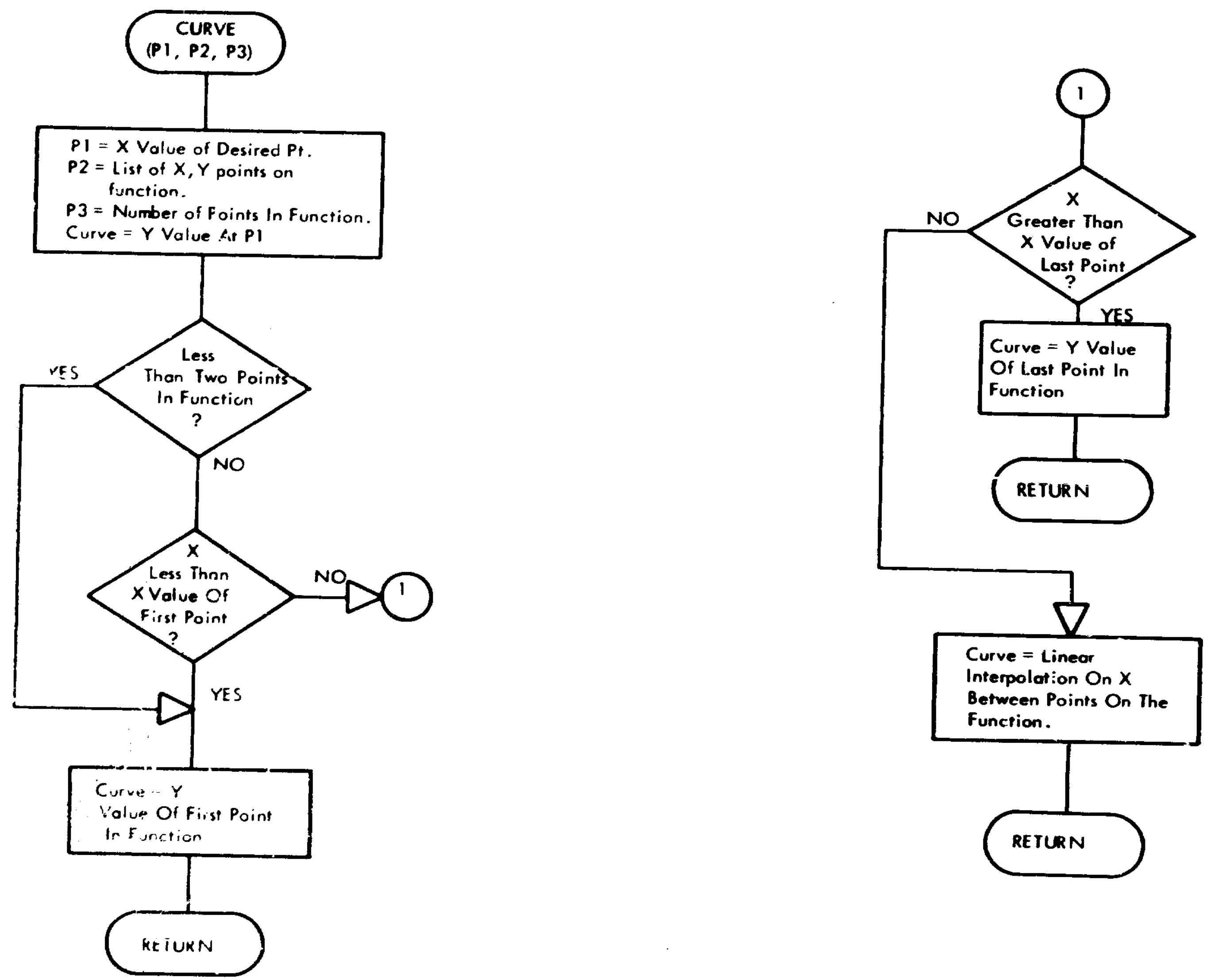


FIGURE C-10. FLOW DIAGRAM - FUNCTION SUBPROGRAM CURVE

```
C
C     ...........................................................
C
C        NEF CONTOUR CONTROL ROUTINE
C
C        PURPOSE
C           TO COMPUTE THE COORDINATES OF NEF CONTOURS AS A FUNCTION
C           OF AIRCRAFT FLIGHT PATH PARAMETERS AND AIRCRAFT PERFORMANCE
C
C        DESCRIPTION OF PARAMETERS
C           XSTART  -  LOCATION ON FLIGHT PATH TO START
C                      CONTOUR COMPUTATIONS  (REAL)
C        XEND  -  LOCATION ON FLIGHT PATH TO END
C                 CONTOUR COMPUTATIONS  (REAL)
C        DX  -  INTERVAL ALONG FLIGHT PATH BETWEEN
C               CONTOUR COMPUTATIONS  (REAL)
C        NAC  -  NUMBER OF UNIQUE AIRCRAFT TYPES  (INTEGER)
C        CONTUR  -  VALUE OF NEF CONTOUR CURRENTLY
C                   BEING COMPUTED  (REAL)
C        NTET(I)  -  TOTAL NUMBER OF CURRENT ENTRIES
C                    IN DATA TABLES  (INTEGER)
C           I =  1 - AIRCRAFT DESCRIPTOR TABLE
C                2 - ALTITUDE PROFILE TABLE
C                3 - DELTA-EPNL PROFILE TABLE
C                4 - EPNL PROFILE TABLES
C        NFET(I)  -  NUMBER OF ENTRIES IN DATA
C                    TABLES WHICH ARE FIXED AND MAY NOT
C                    BE DELETED  (INTEGER)
C           I =  1 - AIRCRAFT DESCRIPTOR TABLE
C                2 - ALTITUDE PROFILE TABLE
C                3 - DELTA-EPNL PROFILE TABLE
C                4 - EPNL PROFILE TABLES
C        NMAX(I)  -  MAXIMUM ALLOWABLE NUMBER OF ENTRIES
C                    IN THE VARIOUS DATA TABLES  (INTEGER)
C           I =  1 - AIRCRAFT DESCRIPTOR TABLE
C                2 - ALTITUDE PROFILE TABLE
C                3 - DELTA-EPNL PROFILE TABLE
C                4 - EPNL PROFILE TABLES
C        DSCNAM(I)  -  NUMBERIC NAME OF I-TH ENTRY IN
C                      THE AIRCRAFT DESCRIPTOR TABLE  (INTEGER)
C        DSCRPT(J,I)  -  12 CHARACTER ALPHANUMERIC
C                        AIRCRAFT DESCRIPTOR FOR THE I-TH
C                        ENTRY IN THE AIRCRAFT CLASS
C                        DESCRIPTOR TABLE. J  IS DIMENSIONED FOR 3 TO
C                        ALLOW FOR THE 12 CHARACTERS.  (INTEGER)
C        ALTDIR(J,I)  -  DIRECTORY WHICH CONTAINS THE NUMERIC NAMES OF
C                        ALTITUDE PROFILES USED BY THE I-TH ENTRY IN
C                        THE AIRCRAFT CLASS DESCRIPTOR TABLE FOR THE
C                        J-TH TRIP LENGTH CATEGORY.  (INTEGER)
C        POWDIR(J,I)  -  DIRECTORY WHICH CONTAINS THE NUMERIC NAMES OF
C                        DELTA-EPNL PROFILES USED BY THE I-TH ENTRY IN
C                        THE AIRCRAFT CLASS DESCRIPTOR TABLE FOR THE
C                        J-TH TRIP LENGTH CATEGORY.  (INTEGER)
C        EPNDIR(J,I)  -  DIRECTORY WHICH CONTAINS THE NUMERIC NAMES OF
```

FIGURE C-11A. PROGRAM LISTING - CONTROL ROUTINE

```
C                        EPNL PROFILES USED BY THE I-TH ENTRY IN THE
C                        AIRCRAFT CLASS DESCRIPTOR TABLE FOR THE J-TH
C                        TRIP LENGTH CATEGORY.  (INTEGER)
C     ALTNAM(-)  -  NUMERIC NAME OF THE I-TH ENTRY IN THE ALTITUDE
C                   PROFILE TABLE.  (INTEGER)
C     NALT(I)  -  NUMBER OF COORDINATES DEFINING THE I-TH ENTRY IN
C                 THE ALTITUDE TABLE  (INTEGER)
C     ALT(K,J,I)  -  THE ALTITUDE PROFILE VLAUE FOR THE I-TH ENTRY
C                    IN THE TABLE FOR THE J-TH COORDINATE(1=X,2=4)
C                    AND THE K-TH POINT IN THE PROFILE  (REAL)
C     Z(I)  -  THE ALTITUDE AT THE CURRENT POINT ON THE FLIGHT PATH,
C              EVALUATED FROM THE I-TH ENTRY IN THE ALTITUDE
C              PROFILE TABLE.  (REAL)
C     POWNAM(I)  -  NUMERIC AN
C     POWNAM(I)  -  NUMERIC NAME OF THE I-TH ENTRY IN THE DECTA-EPNL
C                   PROFILE TABLE.  (INTEGER)
C     NPOW(I)  -  NUMBER OF COORDINATES DEFINING THE I-TH ENTRY IN
C                 THE DELTA-EPNL TABLE  (INTEGER)
C     POWER(K,J,I)  -  THE DELTA-EPNL PROFILE VALUE FOR THE I-TH
C                      ENTRY IN THE TABLE FOR THE J-TH COORDINATE
C                      (1=X,2=Y)AND THE K-TH POINT IN THE PROFILE
C                      (REAL)
C     POWSPL(I)  -  THE DELTA-EPNL VALUE AT THE CURRENT POINT ON THE
C                   FLIGHT PATH, EVALUATED FROM THE I-TH ENTRY IN
C                   THE DELTA-EPNL PROFILE TABLE  (REAL)
C     PNLNAM(I)  -  NUMERIC NAME OF THE I-TH ENTRY IN THE EPNL
C                   PROFILE TABLE.  (INTEGER)
C     PNLEFF(K,J,I)  -  THE EPNL PROFILE VALUE FOR THE I-TH ENTRY IN
C                       THE TABLE FOR THE J-TH PROPAGATION PATH
C                       (1= GROUND-TO-GROUND,2=AIR-TO-GROUND) AND
C                       THE K-TH VALUE IN THE PROFILE
C     ACALC(I)  -  LIST OF ALTITUDE PROFILES REQUIRED FOR COMPUTING
C                  CONTOURS ON CURRENT FLIGHT PATH  (INTEGER)
C     PCALC(I)  -  LIST OF DELTA-EPNL PROFILES
C                  REQUIRED FOR COMPUTING CONTOURS ON CURRENT FLIGHT
C                  PATH  (INTEGER)
C     OPER(I)  -  NEF ADJUSTMENT FOR VOLUME OF OPERATIONS ASSOCIATED
C                 WITH THE I-TH UNIQUE AIRCRAFT TYPE  (REAL)
C     APTP(I)  -  POINTER TO THE ENTRY IN THE ALTITUDE PROFILE TABLE
C                 USED FOR THE I-TH UNIQUE AIRCRAFT TYPE.  (INTEGER)
C     PPTR(I)  -  POINTER TO THE ENTRY IN THE DELTA-EPNL PROFILE
C                 TABLE USED FOR THE I-TH UNIQUE AIRCRAFT TYPE
C                 (INTEGER)
C     EPTR(I)  -  POINTER TO THE ENTRY IN THE EPNL PROFILE TABLE USED
C                 FOR THE I-TH UNIQUE AIRCRAFT TYPE.  (INTEGER)
C     IDENT(I)  -  ALPHANUMERIC DESCRIPTOR STRING FOR FLIGHT
C                  PATH IDENTIFICATION.  (ALPHA)
C
C     SUBROUTINES AND FUNCTION SUBPROGRAMS CALLED
C        CARDIN, APD, ANEF, XNEW, LASTPT
C
C
      ...............................................................
      COMMON  XSTART, XEND, DX, NAC, CONTUR, NMAX(4)                    COM   1
      COMMON  NTEST(4), NEFT(4), NALT(12), NPOW(12), ACALC(12), PCALC(12) COM   2
      COMMON  DSCNAM(25), ALTNAM(12), POWNAM(12), PNLNAM(25), APTR(50)  COM   3
```

FIGURE C-11B. PROGRAM LISTING - CONTROL ROUTINE

FORTMAIN

```
      COMMON   PPTR(50), EPTR(50), OPER(50), Z(12), POWSPL(12), IDENT(13)   COM  4
      COMMON   DSCRPT(3,20), ALTDIR(8,20), POWDIR(8,20), EPNDIR(8,20)       COM  5
      COMMON   PNLEFF(35,2,25), ALT(10,2,12), POWER(10,2,12)                COM  6
      DIMENSION  NEFCON(7), IY(7), XCLOSE(7)
      REAL  NEF1, NEF2
C
      NEFCON(1) = 25
      NEFCON(2) = 30
      NEFCON(3) = 35
      NEFCON(4) = 40
      NEFCON(5) = 45
      NEFCON(6) = 50
      NEFCON(7) = 55
C
      NMAX(1) = 20
      NMAX(2) = 12
      NMAX(3) = 12
      NMAX(4) = 25
      DO 1  I=1,4
      NTET(I) = 0
    1 NEET(I) = 0
C
    2 CALL CARDIN
C
      X = XSTART
      NNEXT = 7
      LINCNT = 0
      DO 3  N=1,7
      XCLOSE(N) = 0.
    3 IY(N) = 4000
C
   11 CALL APD(X)
      NCON = NNEXT
C
      DO 40  N=1,NCON
      CONTUR = NEFCON(N)
      Y2 = IY(N)
      Y1 = Y2 * 0.707
      CALL ANEF (Y2, NEF2)
      CALL ANEF (Y1, NEF1)

   21 Y0 = XNEW(Y1, NEF1, Y2, NEF2, CONTUR)
      IF (Y0 - 30000.) 23, 23, 22
   22 Y0 = 30000.
   23 CONTINUE
      IF (Y0 - 12.6) 24, 25, 25

   24 CONTINUE
      CALL LASTPT(X, XCLOSE(N))
      Y0 = 0.
      NNEXT = NNEXT - 1
      GO TO 29

   25 IF (ABS(Y0 - Y1) - 2.) 28, 28, 26
   26 IF (ABS((Y0 - Y1) / Y0) - 0.001) 28, 28, 27
```

FIGURE C-11C. PROGRAM LISTING - CONTROL ROUTINE

```
   27 Y2 = Y1
      NEF2 = NEF1
      Y1 = Y0
      CALL ANEF (Y1, NEF1)
      GO TO 21
C
   28 IY(N) = Y0 + 0.5
   40 CONTINUE
C
C
      LINCNT = LINCNT + 1
      IF (LINCNT) 43, 42, 42
   42 LINCNT = -50
      WRITE (3, 3004)  IDENT
      WRITE (3, 3001)  NEFCON
C
   43 IX = X + 0.5
      WRITE (3, 3005) IX, (IY(N), N=1,NCON)
C
      IF (NNEXT) 55, 55, 53
C
   53 X = X + DX
C
      IF (X - XEND) 11, 11, 55
C
   55 IFLG = 2
      WRITE (3, 3006)
      DO 70  N=1,7
      IX = XCLOSE(N) + 0.5
      IF (IX) 70, 70, 60
   60 GO TO (61, 62, 63, 64, 65, 66, 67), N
   61 WRITE (3, 3031) IX
      GO TO 69
   62 WRITE (3, 3032) IX
      GO TO 69
   63 WRITE (3, 3033) IX
      GO TO 69
   64 WRITE (3, 3034) IX
      GO TO 69
   65 WRITE (3, 3035) IX
      GO TO 69
   66 WRITE (3, 3036) IX
      GO TO 69
   67 WRITE (3, 3037) IX
   69 IFLG = 1
   70 CONTINUE
      GO TO (71, 72), IFLG
   71 WRITE (3, 3030)
C
   72 GO TO 2
C
 3001 FORMAT ('0', 32X, 'DISTANCE TO CONTOUR IN FEET' / 11X, 'POINT ON'/
     1  9X, 'FLIGHT TRACK', 7I8 / 9X, '------------    ------  ------  -
     2-----  ------  ------  ------  ------')
 3004 FORMAT ('1', 15X, 13A4 / 9X, '-----------------------------------
```

FIGURE C-11D. PROGRAM LISTING - CONTROL ROUTINE

```
     1-----------------------------------')
[illegible] FORMAT (10X, I8, 4X, 7I 8)
[illegible] 6 FORMAT ('0')
[illegible] FORMAT ('+', 7X, 'CUNTOUR CLOSING' / 8X, 'PCINT ON FLIGHT'
     1  / 8X, 'TRACK')
 3031 FORMAT ('+', 21X, I8)
 3032 FORMAT ('+', 29X, I8)
 3033 FORMAT ('+', 37X, I8)
 3034 FORMAT ('+', 45X, I8)
 3035 FORMAT ('+', 53X, I8)
 3036 FORMAT ('+', 61X, I8)
 3037 FORMAT ('+', 69X, I8)
C
      END
```

FIGURE C-11E. PROGRAM LISTING - CONTROL ROUTINE

```
C
C
C     ..............................................................
C
C        SUBROUTINE CARDIN
C
C        PURPOSE
C           TO READ DATA NECESSARY TO COMPUTE
C           NEF CONTOURS FOR A FLIGHT PATH
C
C        USAGE
C           CALL CARDIN
C
C        REMARKS
C           THIS ROUTINE CONTROLS THE INPUT OF ALL DATA TO THE PROGRAM.
C           THE ROUTINE READS THE BLOCK OF DATA FOR A SINGLE FLIGHT
C           PATH AND PERFORMS VARIOUS CHECKS ON THE DATA TO TRAP LOGICAL
C           ERRORS WHICH WOULD RESULT IN ERRONEOUS COMPUTATIONS.
C           IF ERRORS ARE FOUND IN THE DATA, THE ROUTINE AUTOMATICALLY
C           READS THE BLOCK OF DATA FOR THE NEXT FLIGHT PATH. THE
C           ROUTINE DOES NOT RETURN UNTIL IT HAS SUCCESSFULLY READ A
C           COMPLETE BLOCK OF DATA.
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           SIN, COS, PDATA, OPCOR
C
C     ..............................................................
C
C
C
      SUBROUTINE CARDIN
C
      COMMON  XSTART, XEND, DX, NAC, CONTUR, NMAX(4)                    COM   1
      COMMON  NTET(4), NFET(4), NALT(12), NPOW(12), ACALC(12), PCALC(12) COM   2
      COMMON  DSCNAM(20), ALTNAM(12), POWNAM(12), PNLNAM(25), APTR(50)  COM   3
      COMMON  PPTR(50), EPTR(50), OPER(50), Z(12), POWSPL(12), IDENT(13) COM   4
      COMMON  DSCRPT(3,20), ALTDIR(8,20), POWDIR(8,20), EPNDIR(8,20)    COM   5
      COMMON  PNLEFF(35,2,25), ALT(10,2,12), POWER(10,2,12)             COM   6
C
      INTEGER   ERRFLG, ACALC, PCALC, EXPCRD, TRAP, ALTDIR, POWDIR
      INTEGER   EPNDIR, ALTNAM, POWNAM, PNLNAM, CARD, DSCNAM, DSCRPT
      INTEGER   APTR, PPTR, EPTR
      REAL NFLTS, NITFLT
      DIMENSION  BUFF(8), TEMP(2,8), NFLTS(2,8,20)
C
      NSEG = 8
C
C     ----- INITIALIZATION
C          1.  CLEAR ERROR FLAG
C          2.  SET LINE COUNT TO -5
C          3.  CLEAR NUMBER OF CONDENSED AIRCRAFT
C          4.  CLEAR NUMBER OF FLIGHTS
C          5.  CLEAR LIST OF REQUIRED ALTITUDE PROFILES
C          6.  CLEAR LIST OF REQUIRED DELTA-EPNL PROFILES
```

FIGURE C-12A. PROGRAM LISTING - SUBROUTINE CARDIN

```
C
    2 NAC = 0
      ERRFLG = 1
      LINCNT = -5
      DO 6  I=1,2
      DO 6  J=1,8
      DO 6  K=1,20
    6 NFLTS(I,J,K) = 0.
      DO 10 I=1,12
      ACALC(I) = 0
   10 PCALC(I) = 0
C
C  ----- READ CARD WITH FLIGHT PATH IDENTIFICATION AND PARAMETERS
C
      READ (1, 1001) IDENT, XSTART, XEND, DX, RWL, GS
C
C  ----- WRITE FLIGHT PATH IDENTIFICATION
C
      WRITE (3, 3001) IDENT
      WRITE (3, 3000)
C
C  ----- ARE ALL PARAMETERS ZERO
C
      IF (XSTART + XEND + DX + RWL + GS) 15, 15, 20
C
C  ----- SET ERROR FLAG
C
   15 ERRFLG = 0
      GO TO 50
C
C  ----- WRITE FLIGHT PATH PARAMETERS
C
   20 WRITE (3, 3002) XSTART, RWL, XEND, GS, DX
      WRITE (3, 3000)
      WRITE (3, 3003)
C
C  ----- STARTING POSITION OF PATH NEGATIVE
C
      IF (XSTART) 22, 23, 23
C
C  ----- WRITE ERROR AND SET FLAG
C
   22 WRITE (3, 3901)
      ERRFLG = 0
C
C  ----- ENDING POSITION ON PATH NEGATIVE
C
   23 IF (XEND) 24, 25, 25
C
C  ----- WRITE ERROR AND SET FLAG
C
   24 WRITE (3, 3902)
      ERRFLG = 0
C
C  ----- ENDING POSITION GREATER THAN STARTING POSITION
```

FIGURE C-12B. PROGRAM LISTING - SUBROUTINE CARDIN

```
C
   25 IF (XEND - XSTART) 26, 30, 28
C
C   ----- IS INCREMENT ALONG PATH NEGATIVE
C
   26 IF (DX) 30, 27, 27
C
C   ----- WRITE ERROR AND SET ERROR FLAG
C
   27 WRITE (3, 3903)
      ERRFLG = 0
      GO TO 30
C
C   ----- IS INCREMENT ALONG PATH POSITIVE
C
   28 IF (DX) 29, 29, 30
C
C   ----- WRITE ERROR AND SET ERROR FLAG
C
   29 WRITE (3, 3903)
      ERRFLG = 0
C
C   ----- CREATE ALTITUDE PROFILE FOR LANDINGS.  SET COORDINATES OF
C         FIRST POINT.  SET NUMBER OF PROFILE POINTS TO 1
C
   30 NALT(1) = 1
      ALT (1,1,1) = 0.
      ALT (1,2,1) = 0.
C
C   ----- ARE RUNWAY LENGTH AND GLIDE SLOPE BOTH ZERO
C
      IF (RWL + GS) 32, 50, 32
C
C   ----- TEST VALUE OF GLIDE SLOPE
C
   32 IF (GS) 33, 50, 35
C
C   ----- WRITE ERROR MESSAGE AND SET ERROR FLAG
C
   33 WRITE (3, 3907)
      ERRFLG = 0
      GO TO 50
C
C   ----- COMPUTE SECOND AND THIRD COORDINATE POINTS ON LANDING PROFILE
C         FROM GLIDE SLOPE AND RUNWAY LENGTH
C
   35 GS = 1.745329E-2 * GS
      TANGS = SIN(GS) / COS(GS)
      ALT (2,1,1) = RWL - 50./TANGS
      ALT(2,2,1) = 0.
      ALT (3,1,1) = RWL + 1.E5
      ALT(3,2,1) = 1.E5 * TANGS + 50.
C
C   ----- IS  X  VALUE OF SECOND POINT POSITIVE
C
```

FIGURE C-12C. PROGRAM LISTING - SUBROUTINE CARDIN

```
      IF (ALT(2,1,1)) 37, 37, 39
C
C   ----- WRITE ERROR AND SET ERROR FLAG
C
   37 WRITE (3, 3905)
      ERRFLG = 0
      GO TO 50
C
C   ----- SET NUMBER OF PROFILE POINTS TO 3
C
   39 NALT(1) = 3
C
   50 READ (1, 1002) IOP1, IOP2, BUFF
C
  501 IF (IOP1) 100, 100, 51
   51 IF (IOP1 - 99) 200, 200, 52
   52 EXPCRD = 1
      IF (IOP1 - 199) 300, 300, 53
   53 IF (IOP1 - 999) 54, 55, 55
   54 WRITE (3, 3908) IOP1
      GO TO 50
C
   55 STOP
C
C
C
  300 IOP = IOP1 - 100
      N = IOP2
      IF (N) 305, 305, 306
  305 N = 1
C
  306 DO 310  I=1,N
      CALL PDATA(IOP)
      IF (IOP) 307, 307, 310
  307 ERRFLG = 0
  310 CONTINUE
      LINCNT = 0
      GO TO 50
C
C
  200 CARD = IOP2
      IF (CARD - 1) 202, 206, 202
  202 WRITE (3, 3914) IOP1, IOP2
      ERRFLG = 0
      GO TO 50
C
  206 NAME = IOP1
      DO 204  I=1,NSEG
  204 TEMP(1,I) = BUFF(I)
C
      READ (1, 1002) IOP1, IOP2, BUFF
C
      IF (NAME - IOP1) 208, 205, 208
  208 WRITE (3, 3914) IOP1, IOP2
      ERRFLG = 0
```

FIGURE [illegible] PROGRAM LISTING - SUBROUTINE CARDIN

```
      GO TO 501
C
  205 IF (IOP2 - 2) 200, 209, 200
  209 DO 210  I=1,NSEG
  210 TEMP(2,I) = BUFF(I)
C
      LINCNT = LINCNT + 1
C
      IF (LINCNT) 218, 216, 216
  216 WRITE (3, 3001) IDENT
      WRITE (3, 3000)
      WRITE (3, 3003)
      LINCNT = -5
C
  218 N = NTET(1)
      DO 220  IENT = 1,N
      IF (NAME - DSCNAM(IENT))  220, 224, 220
  220 CONTINUE
C
      WRITE (3, 3909) NAME, (J, (TEMP(I,J), I=1,2), J=1,NSEG)
      ERRFLG = 0
      GO TO 50
C
  224 WRITE (3, 3011) NAME, (DSCRPT(I, IENT), I=1,3),
     1  (J, (TEMP(I,J), I=1,2), J=1,NSEG)
C
      TRAP = 1
      DO 238  I=1,NSEG
      NAME = ALTDIR(I, IENT)
      N = NTET(2)
      DO 228  J=1,N
      IF (NAME - ALTNAM(J)) 228, 230, 228
  228 CONTINUE
      WRITE (3, 3910) NAME
      TRAP = 0
  230 NAME = POWDIR(I, IENT)
      N = NTET(3)
      DO 232  J=1,N
      IF (NAME - POWNAM(J)) 232, 234, 232
  232 CONTINUE
      WRITE (3, 3911) NAME
      TRAP = 0
  234 NAME = FPNDIR(I, IENT)
      N = NTET(4)
      DO 236  J=1,N
      IF (NAME - PNLNAM(J)) 236, 238, 236
  236 CONTINUE
      WRITE (3, 3912) NAME
      TRAP = 0
  238 CONTINUE
C
      IF (TRAP) 239, 239, 244
  239 ERRFLG = 0
      GO TO 50
C
```

FIGURE C-12E. PROGRAM LISTING - SUBROUTINE CARDIN

```
  244 DO 246  J=1,2
      DO 246  I=1,NSEG
  246 NFLTS(J,I,IENT) = NFLTS(J,I,IENT) + TEMP(J,I)
      GO TO 50
C
C
  500 STOP
C
  100 CONTINUE
  102 IF (ERRFLG) 103, 103, 104
  103 WRITE (3, 3913)
      GO TO 2
  104 NN = NTET(1)
      DO 150  K=1,NN
      DO 150  J=1,NSEG
      IF (NFLTS(1,J,K) + NFLTS(2,J,K)) 150, 150, 105
  105 IA2 = ALTDIR(J,K)
      IA3 = POWDIR(J,K)
      IA4 = EPNDIR(J,K)
C
      TOTALD = 0.
      TOTALN = 0.
      DO 112  K1=K,NN
      DO 112  J1=1,NSEG
      DAYFLT = NFLTS(1,J1,K1)
      NITFLT = NFLTS(2,J1,K1)
      IF (DAYFLT + NITFLT) 110, 110, 106
  106 IF (IA2 - ALTDIR(J1,K1)) 112, 107, 112
  107 IF (IA3 - POWDIR(J1,K1)) 112, 108, 112
  108 IF (IA4 - EPNDIR(J1,K1)) 112, 110, 112
C
  110 TOTALD = TOTALD + DAYFLT
      TOTALN = TOTALN + NITFLT
      NFLTS(1,J1,K1) = 0.
      NFLTS(2,J1,K1) = 0.
  112 CONTINUE
      IF (NAC - 50) 118, 118, 115
  115 NAC = 1
      ERRFLG = 0
      WRITE (3, 3915)
  118 CONTINUE
      NAC = NAC + 1
C
      OPER(NAC) = OPCOR(TOTALD, TOTALN)
C
      N = NTET(2)
      DO 121  NENT=1,N
      IF (IA2 - ALTNAM(NENT))  121, 126, 121
  121 CONTINUE
      STOP1
  126 DO 125  L=1,N
      IF (ACALC(L)) 123, 123, 122
  122 IF (ACALC(L) - NENT) 125, 124, 125
  125 CONTINUE
  123 ACALC(L) = NENT
```

FIGURE C-12F. PROGRAM LISTING - SUBROUTINE CARDIN

```
  124 APTR(NAC) = NENT
C
      N = NTET(3)
      DO 131  NENT=1,N
      IF (IA3 - POWNAM(NENT)) 131, 132, 131
  131 CONTINUE
      STOP1
  132 DO 130  L=1,N
      IF (PCALC(L)) 128, 128, 127
  127 IF (PCALC(L) - NENT) 130, 129, 130
  130 CONTINUE
  128 PCALC(L) = NENT
  129 PPTR(NAC) = NENT
C
      N = NTET(4)
      DO 135  NENT=1,N
      IF (IA4 - PNLNAM(NENT))  135, 136, 135
  135 CONTINUE
      STOP1
  136 EPTR(NAC) = NENT
C
  150 CONTINUE
      IF (ERRFLG) 155, 155, 152
  152 IF (NAC) 2, 2, 153
  153 RETURN
  155 WRITE (3, 3913)
      GO TO 2
 4010 FORMAT('0', 20I5 / (1X,20I5))
      RETURN
C
C
 3000 FORMAT (9X, '-----------------------------------------------------
     1-------------------')
 3001 FORMAT ('1', 15X, 13A4)
 3002 FORMAT (18X, 'START AT', F8.0, 15X, 'RUNWAY LENGTH', F8.0 /
     1  19X, 'STOP AT', F8.0, 17X, 'GLIDE SLOPE', F8.1 /
     2  18X, 'INTERVAL', F8.0)
 3003 FORMAT ('0', 12X, 'AIRCRAFT CLASSIFICATION', 18X, 'VOLUME OF OPERA
     1TIONS' /  13X, '------------------------   TRIP LENGTH   ---------
     2-----------' / 14X, 'NAME     DESCRIPTION', 6X, 'CATEGORY',
     3  7X, 'DAYTIME   NIGHTTIME' /  14X, '----     -------------     ----
     4-------     -------    ---------')
 3011 FORMAT ('0', 6X, I10, 6X, 3A4, I10, 6X, 2F11.3 / (35X, I10, 6X,
     1  2F11.3))
C
 3901 FORMAT (19X, '* STARTING POINT CANNOT BE NEGATIVE')
 3902 FORMAT (19X, '* STOPPING POINT CANNOT BE NEGATIVE')
 3903 FORMAT (19X, '* INTERVAL IS OF WRONG SIGN')
 3907 FORMAT (19X, '* GLIDE SLOPE CANNOT BE NEGATIVE')
 3906 FORMAT (19X, '* RUNWAY LENGTH MUST BE GREATER THAN ZERO')
 3905 FORMAT (19X, '* RUNWAY TOO SHORT FOR GLIDE SLOPE')
 3908 FORMAT ('0', 18X, '* INVALID OPERATION CODE(', I10, ')')
 3909 FORMAT ('0', 6X, I10, 5X, '* UNDEFINED *', I10, 6X, 2F11.3 /
     1  (35X, I10, 6X, 2F11.3))
 3910 FORMAT (19X, '* ALTITUDE PROFILE UNDEFINED (', I10, ')')
```

FIGURE C-12G. PROGRAM LISTING - SUBROUTINE CARDIN

```
3911 FORMAT (19X, '* DELTA-EPNL PROFILE UNDEFINED (', I10, ')')
3912 FORMAT (19X, '* EPNL FUNCTION UNDEFINED (', I10, ')')
3913 FORMAT (24X, '*** PROGRAM CANNOT COMPUTE THIS CONTOUR ***')
3914 FORMAT ('0', 18X, '* CARD MISSING OR OUT OF ORDER,  A/C CLASS',I3,
    1  ', CARD NO', I3)
3915 FORMAT ('0', 18X, '* CONDENSED AIRCRAFT TABLE FULL')
C
1001 FORMAT (12A4, A3, 2F7.0, F5.0, F6.0, F4.0)
1002 FORMAT (2I3, 8F8.0)
     END
```

FIGURE C-12H. PROGRAM LISTING - SUBROUTINE CARDIN

```
C
C     ..................................................................
C
C        SUBROUTINE PDATA
C
C        PURPOSE
C           TO ENTER AIRCRAFT DESCRIPTORS AND PERFORMANCE DATA
C
C        USAGE
C           CALL PDATA (OPCODE)
C
C        DESCRIPTION OF PARAMETERS
C           OPCODE  -  INTEGER VALUE TO DENOTE ACTION TO BE TAKEN
C                       1 = ENTER NEW AIRCRAFT DESCRIPTOR
C                       2 = ENTER NEW ALTITUDE PROFILE
C                       3 = ENTER NEW DELTA EPNL PROFILE
C                       4 = ENTER NEW EPNL PROFILE
C
C                      11 = FIX AIRCRAFT DESCRIPTOR TABLE
C                      12 = FIX ALTITUDE TABLE
C                      13 = FIX DELTA EPNL TABLE
C                      14 = FIX EPNL TABLE
C                      15 = FIX ALL TABLES
C
C                      21 = EXPUNGE AIRCRAFT DESCRIPTOR TABLE
C                      22 = EXPUNGE ALTITUDE TABLE
C                      23 = EXPUNGE DELTA EPNL TABLE
C                      24 = EXPUNGE EPNL TABLE
C                      25 = EXPUNGE ALL TABLES
C
C        REMARKS
C           1.  EXPUNGE COMMAND DELETES ONLY THOSE TABLE ENTRIES NOT
C               FIXED.
C           2.  ERROR CONDITIONS
C               A.  INVALID OPCODE  -  NO ACTION TAKEN.  MESSAGE IS
C                       PRINTED ALONG WITH  OPCODE  VALUE.  ROUTINE
C                       RETURNS WITH  OPCODE = 0.
C               B.  INVALID DATA  -  NO ACTION TAKEN.  MESSAGE IS
C                       PRINTED ALONG WITH OFFENDING DATA.  ROUTINE
C                       RETURNS WITH OPCODE = 0.
C               C.  AVAILABLE TABLE AREA OVERFLOW  -  NO ACTION TAKEN.
C                       MESSAGE IS PRINTED.  ROUTINE RETURNS WITH
C                       OPCODE = 0.
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS CALLED
C           NONE
C
C     ..................................................................
C
C
      SUBROUTINE PDATA (OPCODE)
C
      COMMON  XSTART, XEND, DX, NAC, CONTUR, NMAX(4)                    COM   1
      COMMON  NTET(4), NFET(4), NALT(12), NPOW(12), ACALC(12), PCALC(12) COM   2
```

FIGURE C-13A. PROGRAM LISTING - SUBROUTINE PDATA

PDATA

```
      COMMON  DSCNAM(20), ALTNAM(12), POWNAM(12), PNLNAM(25), APTR(50)       COM   3
      COMMON  PPTR(50), EPTR(50), OPER(50), Z(12), POWSPL(12), IDENT(13)    COM   4
      COMMON  DSCRPT(3,20), ALTDIR(8,20), POWDIR(8,20), EPNDIR(3,20)        COM   5
      COMMON  PNLEFF(35,2,25), ALT(10,2,12), POWER(10,2,12)                  COM   6
      INTEGER  OPCODE, DSCNAM, DSCRPT, ALTNAM, POWNAM, PNLNAM
      INTEGER  PRP, ALTDIR, POWDIR, EPNDIR, DTEMP, ATEMP, PTEMP, ETEMP
      DIMENSION  DTEMP(3), ATEMP(8), PTEMP(8), ETEMP(8)
      DIMENSION  ALTTEM(10,2), POWTEM(10,2), EPNTEM(35,2), TEMP(10)
      EQUIVALENCE (ALTTEM(1,1), POWTEM(1,1), EPNTEM(1,1))

      NSEG = 8
      IF (OPCODE - 1) 900, 110, 10
   10 IF (OPCODE - 3) 120, 130, 11
   11 IF (OPCODE - 5) 140, 900, 12

   12 IF (OPCODE - 11) 900, 210, 13
   13 IF (OPCODE - 13) 220, 230, 14
   14 IF (OPCODE - 15) 240, 250, 15

   15 IF (OPCODE - 21) 900, 310, 16
   16 IF (OPCODE - 23) 320, 330, 17
   17 IF (OPCODE - 25) 340, 350, 18
   18 GO TO 900
  900 WRITE (3, 3900) OPCODE

  110 READ (1, 1001) NAME, DTEMP, (ATEMP(I),PTEMP(I),ETEMP(I), I=1,NSEG)

      M = NTET(1)
      N = M + 1

      IF (N-NMAX(1)) 111, 111, 910
  111 DO 112  I=1,M
      IF (DSCNAM(I) - NAME) 112, 911, 112
  112 CONTINUE
      DSCNAM(N) = NAME
      DO 114  I=1,3
  114 DSCRPT(I,N) = DTEMP(I)
      DO 116  I=1,NSEG
      ALTDIR(I,N) = ATEMP(I)
      POWDIR(I,N) = PTEMP(I)
  116 EPNDIR(I,N) = ETEMP(I)

      NTET(1) = N
      RETURN

  910 WRITE (3, 3910)
      WRITE (3, 3921) NAME
      GOTO 999
  911 WRITE (3, 3911)
      WRITE (3, 3921) NAME
      GO TO 999


  120 NN = 0
      READ (1, 1003) NAME, NP, ((ALTTEM(J,I), I=1,2), J=1,6)
```

FIGURE C-13B. PROGRAM LISTING - SUBROUTINE PDATA

```
  121 NN = NN + 6
      IF (NP - NN) 123, 123, 122
  122 READ (1, 1004) ((ALTTEM(J,I), I=1,2), J=7,10)
      GO TO 121
  123 M = NTET(2)
      N = M + 1
      IF (NP) 922, 922, 124
  124 IF (NP - 10) 125, 125, 922
  125 IF (N - NMAX(2)) 126, 126, 923
  126 DO 127  I=1,M
      IF (ALTNAM(I) - NAME) 127, 924, 127
  127 CONTINUE
C
      DO 128  I=1,2
      DO 128  J=1,NP
  128 ALT(J,I,N) = ALTTEM(J,I)
      NALT(N) = NP
      ALTNAM(N) = NAME
      NTET(2) = N
      RETURN
C
  922 WRITE (3, 3922)
      WRITE (3, 3901) NAME
      GO TO 999
  923 WRITE (3, 3923)
      WRITE (3, 3901) NAME
      GOTO 999
  924 WRITE (3, 3924)
      WRITE (3, 3901) NAME
      GO TO 999
C
C
  130 NN = 0
      READ (1, 1003) NAME, NP, ((POWTEM(J,I), I=1,2), J=1,6)
  131 NN = NN + 6
      IF (NP - NN) 133, 133, 132
  132 READ (1, 1004) ((POWTEM(J,I), I=1,2), J=7,10)
      GO TO 131
  133 M = NTET(3)
      N = M + 1
      IF (NP) 932, 932, 134
  134 IF (NP - 10) 135, 135, 932
  135 IF (N - NMAX(3)) 136, 136, 933
  136 DO 137  I=1,M
      IF (POWNAM(I) - NAME) 137, 934, 137
  137 CONTINUE
C
      DO 138  I=1,2
      DO 138  J=1,NP
  138 POWER(J,I,N) = POWTEM(J,I)
      NPOW(N) = NP
      POWNAM(N) = NAME
      NTET(3) = N
      RETURN
  932 WRITE (3, 3932)
```

FIGURE C-13C. PROGRAM LISTING - SUBROUTINE PDATA

```
      WRITE (3, 3901) NAME
      GO TO 999
  933 WRITE (3, 3933)
      WRITE (3, 3901) NAME
      GO TO 999
  934 WRITE (3, 3934)
      WRITE (3, 3901) NAME
      GO TO 999
C
C
  140 ERRFLG = 1
      DO 155  L=1,8
      READ (1, 1002) NAME, PRP, L1, L2, TEMP
      IF (L - 1) 141, 141, 142
  141 NAME1 = NAME
C
  142 IF (NAME1 - NAME) 942, 143, 942
  942 WRITE (3, 3942)
       ERRFLG = 0
  143 IF (PRP) 943, 943, 144
  144 IF (PRP - 2) 145, 145, 943
  943 WRITE (3, 3943)
      ERRFLG = 0
  145 IF (L1) 945, 945, 146
  146 IF (L2 - 35) 147, 147, 945
  147 IF (L2 - L1 - 10) 148, 148, 945
  148 IF (L2 - L1) 945, 945, 150
  945 WRITE (3, 3945)
      ERRFLG = 0
  150 IF (ERRFLG) 155, 155, 151
  151 I1 = 0
      DO 152  I=L1,L2
      I1 = I1 + 1
  152 EPNTEM(I,PRP) = TEMP(I1)
  155 CONTINUE
      IF (ERRFLG) 949, 949, 156
C
  156 M = NTET(4)
      N = M + 1
      IF (N - NMAX(4)) 157, 157, 947
  157 DO 158  I=1,M
      IF (PNLNAM(I) - NAME1) 158, 948, 158
  158 CONTINUE
C
  160 DO 162  I=1,35
      DO 162  J=1,2
  162 PNLEFF(I,J,N) = EPNTEM(I,J)
      PNLNAM(N) = NAME1
      NTET(4) = N
      RETURN
  947 WRITE (3, 3947)
      WRITE (3, 3901) NAME1
      GO TO 999
  948 WRITE (3, 3948)
      WRITE (3, 3901) NAME1
```

FIGURE C-14D. PROGRAM LISTING - SUBROUTINE PDATA

```
      GO TO 999
  949 WRITE (3, 3901) NAME1
      GO TO 999
  999 OPCODE = 0
      RETURN
C
C
  210 NFET(1) = NTET(1)
      RETURN
C
  220 NFET(2) = NTET(2)
      RETURN
C
  230 NFET(3) = NTET(3)
      RETURN
C
  240 NFET(4) = NTET(4)
      RETURN
C
  250 DO 252  I=1,4
  252 NFET(I) = NTET(I)
      RETURN
C
C
  310 NTET(1) = NFET(1)
      RETURN
C
  320 NTET(2) = NFET(2)
      RETURN
C
  330 NTET(3) = NFET(3)
      RETURN
C
  340 NTET(4) = NFET(4)
      RETURN
C
  350 DO 352  I=1,4
  352 NTET(I) = NFET(I)
      RETURN
C
C
 1001 FORMAT (I2, 1X, 3A4, 1X, 8(3I2,2X))
 1002 FORMAT (2I2, 2I3, 10F7.3)
 1003 FORMAT (2I3, 2X, 12F6.0)
 1004 FORMAT (8X, 12F6.0)
 3900 FORMAT ('0', 18X, '* INVALID OPERATION CODE (', I4, ')')
 3901 FORMAT (21X, 'CANNOT ENTER NAME =', I4)
 3910 FORMAT ('0', 18X, '* AIRCRAFT DESCRIPTOR TABLE FULL')
 3911 FORMAT ('0', 18X, '* DUPLICATE AIRCRAFT DESCRIPTOR NAME')
 3922 FORMAT ('0', 18X, '* INVALID NUMBER OF ALTITUDE POINTS')
 3923 FORMAT ('0', 18X, '* ALTITUDE PROFILE TABLE FULL')
 3924 FORMAT ('0', 18X, '* DUPLICATE ALTITUDE PROFILE NAME')
 3932 FORMAT ('0', 18X, '* INVALID NUMBER OF DELTA-EPNL POINTS')
 3933 FORMAT ('0', 18X, '* DELTA-EPNL PROFILE TABLE FULL')
 3934 FORMAT ('0', 18X, '* DUPLICATE DELTA-EPNL PROFILE NAME')
```

FIGURE C-15E. PROGRAM LISTING - SUBROUTINE PDATA

```
3942 FORMAT ('0', 18X, '* PNLEFF PROFILE NAMES DO NOT MATCH')
3943 FORMAT ('0', 18X, '* INVLAID PNLEFF PROPAGATION PATH')
3945 FORMAT ('0', 18X, '* INVALID PNLEFF RANK LIMITS')
3947 FORMAT ('0', 18X, '* PNLEFF PROFILE TABLE FULL')
3948 FORMAT ('0', 18X, '* DUPLICATE PNLEFF PROFILE NAME')
     END
```

FIGURE C-16F. PROGRAM LISTING - SUBROUTINE PDATA

```
C
C     ..................................................................
C
C        SUBROUTINE LASTPT
C
C        PURPOSE
C           TO COMPUTE THE POINT ON THE FLIGHT TRACK WHERE THE
C           CONTOUR CLOSES
C
C        USAGE
C           CALL LASTPT (X, XCLOSE)
C
C        DESCRIPTION OF PARAMETERS
C           X  -  THE CALLING VALUE OF THIS PARAMETER IS A POINT
C                 ON THE FLIGHT TRACK NEAR WHICH THE CONTOUR CLOSES .
C
C           XCLOSE  -  THE RETURNED VALUE OF THIS PARAMETER IS THE POINT
C                      ON THE FLIGHT PATH WHERE THE CONTOUR CLOSES
C                      (IE. THE NEF VALUE AT THIS POINT IS EQUAL TO THE
C                      NEF VALUE OF THE CONTOUR CURRENTLY BEING
C                      COMPUTED).
C
C        REMARKS
C           1.  A VALUE OF ZERO WILL BE RETURNED IF THE ROUTINE
C               CANNOT COMPUTE A CLOSING POINT.
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           ANEF, PARAM, XNEW
C
C        METHOD
C           THE ROUTINE PICKS TWO POINTS ON THE FLIGHT TRACK AND
C           COMPUTES THE NEF VALUES AT THESE POINTS.  INTERPOLATION
C           BETWEEN THESE POINTS USING THE VALUE OF THE CURRENT CONTOUR
C           BEING COMPUTED YIELDS THE CONTOUR CLOSING VALUE OF  X  TO
C           BE RETURNED.
C           ONE OF THE POINTS CHOSEN IS THE CALLING VALUE OF  X.  IF
C           THE ALTITUDE OF ANY AIRCRAFT IS LESS THAN 12.5 FEET AT
C           THIS POINT THE ROUTINE TERMINATES AND RETURNS WITH  X
C           EQUAL TO ZERO.
C           THE SECOND POINT IS VARIABLE AND IS CHOSEN IN THE FOLLOWING
C           MANNER.  A POINT ON THE TRACK ONE COMPUTATION INTERVAL
C           (DX) LESS THAN THE CURRENT VALUE OF  X  IS CHOSEN.  IF ANY
C           ALTITUDES ARE LESS THAN 12.5 FEET AT THIS POINT, ANOTHER
C           POINT IS CHOSEN HALF AGAIN THE DISTANCE BETWEEN THESE TWO
C           POINTS.  THE ALTITUDE TEST IS PERFORMED AGAIN AND IF
C           ANY ALTITUDE IS LESS THAN 12.5 FEET THE DISTANCE IS
C           HALVED ONCE AGAIN.  THIS SEQUENCE IS REPEATED UNTIL
C           A USABLE VALUE OF  X  IS OBTAINED.  IF, AFTER 10 TRIALS,
C           NO USABLE VALUE OF X IS OBTAINED THE ROUTINE TERMINATES
C           AND RETURNS WITH  X  EQUAL TO ZERO.
C
C     ..................................................................
C
C
```

FIGURE C-14A. PROGRAM LISTING - SUBROUTINE LASTPT

```
      SUBROUTINE LASTPT (X, XCLOSE)
C
      COMMON  XSTART, XEND, DX, NAC, CONTUR, NMAX(4)                    COM  1
      COMMON  NTET(4), NFET(4), NALT(12), NPOW(12), ACALC(12), PCALC(12) COM  2
      COMMON  DSCNAM(20), ALTNAM(12), POWNAM(12), PNLNAM(25), APTR(50)  COM  3
      COMMON  PPTR(50), EPTR(50), OPER(50), Z(12), POWSPL(12), IDENT(13) COM  4
      COMMON  DSCRPT(3,20), ALTDIR(8,20), POWDIR(8,20), EPNDIR(8,20)    COM  5
      COMMON  PNLEFF(35,2,25), ALT(10,2,12), POWER(10,2,12)             COM  6
C
      INTEGER APTR
C
C        ----- CHECK AIRCRAFT ALTITUDES AT CALLING VALUE OF X.  IF ANY
C              ARE LESS THAN 12.5 FEET TERMINATE COMPUTATIONS FOR
C              THIS CONTOUR -----
C
C
      DO 5  I=1,NAC
      J = APTR(I)
      IF (Z(J) - 12.5) 30, 5, 5
    5 CONTINUE
C
C        ----- IF ALTITUDES ARE OK COMPUTE NEF AT THIS POINT -----
C
      CALL ANEF (0., NEF2)
C
C        ----- NOW COMPUTE NEF DIRECTLY UNDER PATH AT SOME POINT
C              WITH A LESSER VALUE OF  X -----
C
      XLESS = X - DX
      XINT = DX
C
      DO 10  IT = 1,10
C
C        ----- CALCULATE PERFORMANCE PARAMETERS -----
C
      CALL APD (XLESS)
C
C        ----- CHECK FOR ANY ALTITUDES LESS THAN 12.5 FEET -----
C
      DO 7  I=1,NAC
      J = APTR(I)
      IF (Z(J) - 12.5) 8, 7, 7
    7 CONTINUE
C
C        ----- COMPUTE NEF AT THIS POINT -----
C
      CALL ANEF (0., NEF1)
C
C        ----- INTERPOLATE TO FIND VALUE OF  X  WHERE CONTOUR CLOSES
C
      XCLOSE = XNEW (XLESS, NEF1, X, NEF2, CONTUR)
      IF (ABS(X - XCLOSE) - ABS(DX)) 40, 40, 30
C
C
C        ----- TRY ANOTHER VLAUE OF  X -----
```

FIGURE C-14B. PROGRAM LISTING - SUBROUTINE LASTPT

LASTPT

```
C
    8 XINT = XINT/2.
      XLESS = XLESS + XINT
   10 CONTINUE
C
C
C         ----- ROUTINE CANNOT COMPUTE CLOSING VALUE OF  X -----
C
   30 XCLOSE = 0.
C
C
C         ----- RESTORE ALTITUDES AND DELTA-EPNL'S -----
C
   40 CALL APD(X)
      RETURN
C
      END
```

FIGURE C-14C. PROGRAM LISTING - SUBROUTINE LASTPT

```
C
C     ..............................................................
C
C        SUBROUTINE ANEF
C
C        PURPOSE
C           TO COMPUTE THE NEF VALUE AT A POINT  P ,  A GIVEN
C           DISTANCE PERPENDICULAR TO THE FLIGHT TRACK.
C
C        USAGE
C           CALL ANEF (Y, NEFVAL)
C
C        DESCRIPTION OF PARAMETERS
C           Y  -  PERPENDICULAR DISTANCE FROM POINT TO FLIGHT TRACK
C           NEFVAL  -  NEF VALUE AT POINT  P
C
C        REMARKS
C           1.  IT IS ASSUMED THAT
C                A.  NAC IS GREATER THAN ZERO
C                B.  ALTITUDES AND EPNL CORRECTIONS HAVE PREVIOUSLY
C                    BEEN TABULATED.
C                C.  NO ALTITUDE IS EQUAL TO ZERO IF THE CALLING
C                    VALUE OF  Y  IS EQUAL TO ZERO.
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           EPNL, SQRT, ALOG10
C
C     ..............................................................
C
      SUBROUTINE ANEF (Y, NEFVAL)
C
      COMMON  XSTART, XEND, DX, NAC, CONTUR, NMAX(4)                    COM   1
      COMMON  NTET(4), NFET(4), NALT(12), NPOW(12), ACALC(12), PCALC(12) COM   2
      COMMON  DSCNAM(20), ALTNAM(12), POWNAM(12), PNLNAM(25), APTR(50)  COM   3
      COMMON  PPTR(50), EPTR(50), OPER(50), Z(12), POWSPL(12), IDENT(13) COM   4
      COMMON  DSCRPT(3,20), ALTDIR(8,20), POWDIR(8,20), EPNDIR(8,20)    COM   5
      COMMON  PNLEFF(35,2,25), ALT(10,2,12), POWER(10,2,12)             COM   6
C
      INTEGER  APTR, PPTR, EPTR, PTR
      REAL     NEFLIN, NEF, NEFVAL
C
      NEFLIN = 0.
C
C        ----- COMPUTE NEF FOR EACH AIRCRAFT AND COLLECT CUMULATIVE SUM
C
      DO 20 I=1,NAC
C
C        ----- PICK UP POINTER TO ALTITUDE FOR THIS AIRCRAFT AND
C              CALCULATE SLANT DISTANCE -----
C
      PTR = APTR(I)
      DIST = SQRT(Y*Y + Z(PTR)*Z(PTR))
C
C        ----- CALCULATE PROPAGATION FACTOR -----
```

FIGURE C-15A. PROGRAM LISTING - SUBROUTINE ANEF

```
C                1.  IF LESS THAN OR EQUAL TO ZERO USE GROUND TO
C                    GROUND PROPAGATION.
C                2.  IF BETWEEN ZERO AND ONE INTERPOLATE BETWEEN
C                    GROUND TO GROUND  AND  AIR TO GROUND PROPAGATION.
C                3.  IF GREATER THAN OR EQUAL TO ONE USE AIR TO
C                    GROUND PROPAGATION
C
C        ----- NOTE
C                PROPAGATION FACTOR IS CALCULATED ON THE BASIS OF THE
C                ANGLE OF ELEVATION OF THE AIRCRAFT WITH RESPECT TO
C                THE OBSERVER AND THE GROUND PLANE.  PF = 0 AT
C                ARCSIN(0.075) AND PF = 1 AT ARCSIN(0.125),
C                4 DEG 18 MIN  AND  7 DEG 11 MIN  RESPECTIVELY.
C
      PF = (Z(PTR) / DIST - 0.075) / 0.05
C
C        ----- PICK UP POINTER TO PNLEFF LIST FOR THIS AIRCRAFT -----
C
      PTR = EPTR(I)
C
      IF (PF) 5, 5, 3
    3 IF (PF - 1.) 6, 7, 7
C
C        ----- GET EPNL -----
C
    5 P = EPNL(PTR, 1, DIST)
      GO TO 10
C
    6 P = (1. - PF)*EPNL(PTR,1,DIST) + PF*EPNL(PTR,2,DIST)
      GO TO 10
C
    7 P = EPNL(PTR,2,DIST)
C
C        ----- PICK UP POINTER TO EPNL CORRECTION FOR THIS AIRCRAFT
C              AND COMPUTE TOTAL EPNL FOR THIS AIRCRAFT -----
C
   10 PTR = PPTR(I)
      P = P + POWSPL(PTR)
C
C        ----- TRUNCATE EPNL IF LESS THAN 80 -----
C
      NEF = 0.
      IF (P - 75.) 19, 19, 13
   13 IF (P - 80.) 14, 14, 15
   14 P = (80. + OPER(I)) * (P - 75.) * 0.2
C
C        ----- COMPUTE NEF FOR THIS AIRCRAFT -----
C
   15 NEF = P + OPER(I)
C
C        ----- CONVERT NEF FROM  LOG TO LINEAR AND ADD TO SUM -----
C
   19 NEFLIN = NEFLIN + 10.**(NEF/10.)
   20 CONTINUE
C
C        ----- CONVERT SUM FROM LINEAR TO LOG -----
C
      NEFVAL = 10. * ALOG10(NEFLIN) - 75.
C
      RETURN
C
      END
```

FIGURE C-15B. PROGRAM LISTING - SUBROUTINE ANEF

```
C
C     ***********************************************************
C
C        SUBROUTINE APD
C
C        PURPOSE
C           TO COMPUTE THE NECESSARY ALTITUDES AND DELTA-EPNLS
C           AT A POINT ALONG THE FLIGHT TRACK.
C
C        USAGE
C           CALL APD (X)
C
C        DESCRIPTION OF PARAMETERS
C           X  -  DISTANCE ALONG FLIGHT TRACK FROM START OF
C                 TAKEOFF ROLL.
C
C        REMARKS
C           1.  ROUTINE ASSUMES THAN TWO PROFILE POINTER TABLES
C               (ACALC AND PCALC) HAVE BEEN BUILT CONTAINING THE
C               PROFILE ENTRIES FROM WHICH TO CALCULATE THE
C               PARAMETER VALUES.
C           2.  ROUTINE ASSUMES THAT SPECIFIED PROFILES HAVE
C               BEEN DEFINED.
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           CURVE
C
C        METHOD
C           ROUTINE PICKS UP PROFILE ENTRY FROM AFOREMENTIONED
C           PREVIOUSLY BUILT TABLES AND COMPUTES THE PARAMETER VALUE
C           FROM THE SPECIFIED PROFILE FOR THE GIVEN VALUE OF  X.
C           PARAMETER VALUE IS THEN PLACED IN TABLE WHICH IS PARALLEL TO
C           THE PROFILE POINTER TABLES.
C
C
C     ***********************************************************
C
      SUBROUTINE APD (X)
C
      COMMON  XSTART, XEND, DX, NAC, CONTUR, NMAX(4)                         COM  1
      COMMON  NTET(4), NFET(4), NALT(12), NPOW(12), ACALC(12), PCALC(12)     COM  2
      COMMON  DSCNAM(20), ALTNAM(12), POWNAM(12), PNLNAM(25), APTR(50)       COM  3
      COMMON  PPTR(50), EPTR(50), OPER(50), Z(12), POWSPL(12), IDENT(13)     COM  4
      COMMON  DSCRPT(3,20), ALTDIR(8,20), PCWDIR(8,20), EPNDIR(8,20)         COM  5
      COMMON  PNLEFF(35,2,25), ALT(10,2,12), POWER(10,2,12)                  COM  6
C
      DIMENSION DUMY(2,10)
      INTEGER  ACALC, PCALC
C
C        ----- COMPUTE ALTITUDES -----
C
      M = NTET(2)
      DO 19 I=1,M
C
C        ----- PICK UP PROFILE NUMBER -----
```

FIGURE C-16A. PROGRAM LISTING - SUBROUTINE APD

```
      J = ACALC(I)
C
C         ----- TEST FOR ZERO (IE. END OF STRING) -----
C
      IF (J) 20, 20, 15
C
C         ----- PICK UP NUMBER OF POINTS DEFINING THIS PROFILE -----
C
   15 N = NALT(J)
C
C         ----- LOAD THIS PROFILE INTO DUMMY ARRAY -----
C
      DO 18 K=1,2
      DO 18 L=1,N
   18 DUMY(K,L) = ALT(L,K,J)
C
C         ----- CALCULATE PARAMETER VALUE -----
C
   19 Z(J) = CURVE(X,DUMY,N)
C
C
C         ----- COMPUTE EPNL ADJUSTMENTS -----
C
      M = NTET(3)
   20 DO 29 I=1,M
      J = PCALC(I)
      IF (J) 30, 30, 25
   25 N = NPOW(J)
      DO 28 K=1,2
      DO 28 L=1,N
   28 DUMY(K,L) = POWER(L,K,J)
   29 POWSPL(J) = CURVE(X,DUMY,N)
C
C
   30 RETURN
C
      END
```

FIGURE C-16B. PROGRAM LISTING - SUBROUTINE APD

```
C
C    ...................................................................
C
C       FUNCTION EPNL
C
C       PURPOSE
C          TO COMPUTE AN EFFECTIVE PERCEIVED NOISE LEVEL AS A
C          FUNCTION OF DISTANCE FROM AIRCRAFT TO OBSERVER.
C
C       USAGE
C          VARIABLE  =  EPNL(LIST, PRPGTN, DIST)
C
C       DESCRIPTION OF PARAMETERS
C          LIST - INTEGER TO SPECIFY WHICH OF 18 PNL VS. DISTANCE
C                 FUNCTIONS TO USE FROM PNLEFF TABLE (INTEGER)
C          PRPGTN  -  PROPAGATION PATH     1 - GROUND TO GROUND
C                      (INTEGER)           2 - AIR TO GROUND
C          EPNL   -  EFFECTIVE PERCEIVED NOISE LEVEL (REAL)
C          DIST   -  DISTANCE FROM AIRCRAFT TO OBSERVER  (REAL)
C
C       REMARKS
C          1.  FUNCTION IS DEFINED FOR ONLY TWO PROPAGATION PATHS
C              AND FOR DISTANCES FROM 12.5 TO 31500 FEET.  IF DIST
C              IS OUTSIDE OF DEFINED RANGE OR PRPGTN IS NOT EQUAL
C              TO 1 OR 2, AN ERROR MESSAGE IS PRINTED ALONG WITH THE
C              VALUES OF THE THREE CALLING PARAMETERS.
C          2.  EPNL VALUES RETURNED FOR INVALID DATA
C              A.  IF DISTANCE IS LESS THAN 12.5 FEET THE EPNL
C                  AT 12.5 FEET WILL BE RETURNED.
C              B.  IF DISTANCE IS GREATER THAN 31500 FEET THE EPNL
C                  AT 31500 FEET WILL BE RETURNED.
C              C.  IF PRPGTN IS NOT EQUAL TO 1 OR 2  A VALUE OF
C                  ZERO WILL BE RETURNED
C
C       SUBROUTINES AND FUNCTION SUBPROGRAMS CALLED
C          ALOG10
C
C       METHOD
C          ROUTINE FIRST LOOKS UP TWO PNLEFF VALUES IN TABLE FOR
C          DISTANCES BOUNDING CALLING DISTANCE.  INTERPOLATION
C          BETWEEN THESE VALUES IS PERFORMED TO YIELD DESIRED EPNL.
C
C    ...................................................................
C
      REAL FUNCTION EPNL(LIST,PRPGTN,DIST)
C
      COMMON  XSTART, XEND, DX, NAC, CONTUR, NMAX(4)                    COM  1
      COMMON  NTET(4), NFET(4), NALT(12), NPOW(12), ACALC(12), PCALC(12) COM  2
      COMMON  DSCNAM(20), ALTNAM(12), POWNAM(12), PNLNAM(25), APTR(50)  COM  3
      COMMON  PPTR(50), EPTR(50), OPER(50), Z(12), POWSPL(12), IDENT(13) COM  4
      COMMON  DSCRPT(3,20), ALTDIR(8,20), POWDIR(8,20), EPNDIR(8,20)    COM  5
      COMMON  PNLEFF(35,2,25), ALT(10,2,12), POWER(10,2,12)             COM  6
C
      INTEGER  PRPGTN
```

FIGURE C-17A. PROGRAM LISTING - FUNCTION EPNL

EPNL

```
      REAL      LOGD, LOGD1
C
C          ----- CONVERT DISTANCE TO LOGARITHMIC SCALE UPON WHICH PNLEFF
C                TABLES ARE BASED -----
C
      LOGD = 10. * ALOG10(DIST) - 10.
C
C          ----- CHECK FOR INVALID PARAMETERS -----
C
      IF (PRPGTN) 14, 14, 2
    2 IF (PRPGTN - 2) 11, 11, 14
C
   11 IF (LOGD - 1.) 15, 12, 12
   12 IF (LOGD - 35.) 13, 16, 16
C
C          ----- PARAMETERS ARE VALID -----
C          ----- TRUNCATE FRACTION PART OF LOG DISTANCE -----
C
   13 ILOGD = LOGD
      LOGD1 = ILOGD
C
C          ----- LOOK UP BOUNDING PNLEFF VALUES -----
C
      C1 = PNLEFF(ILOGD,PRPGTN,LIST)
      C2 = PNLEFF(ILOGD+1,PRPGTN,LIST)
C
C          ----- INTERPOLATE -----
C
      EPNL = C1 + (C2-C1)*(LOGD-LOGD1)
      RETURN
C
C
C          ----- PARAMETERS ARE INVALID -----
C
   14 EPNL = 0.
      GO TO 17
C
   15 EPNL = PNLEFF(1,PRPGTN,LIST)
      GO TO 17
C
   16 EPNL = PNLEFF(35,PRPGTN,LIST)
C
   17 WRITE (3, 3001) LIST, PRPGTN, DIST
      RETURN
C
 3001 FORMAT ('0', 20X, 'FUNCTION EPNL  -  INVALID INPUT PARAMETER'
     1       21X, 'LIST =', I3,     PRPGTN =', I3, ',  DIST =', F9.0)
C
      END
```

FIGURE C-17B. PROGRAM LISTING - FUNCTION EPNL

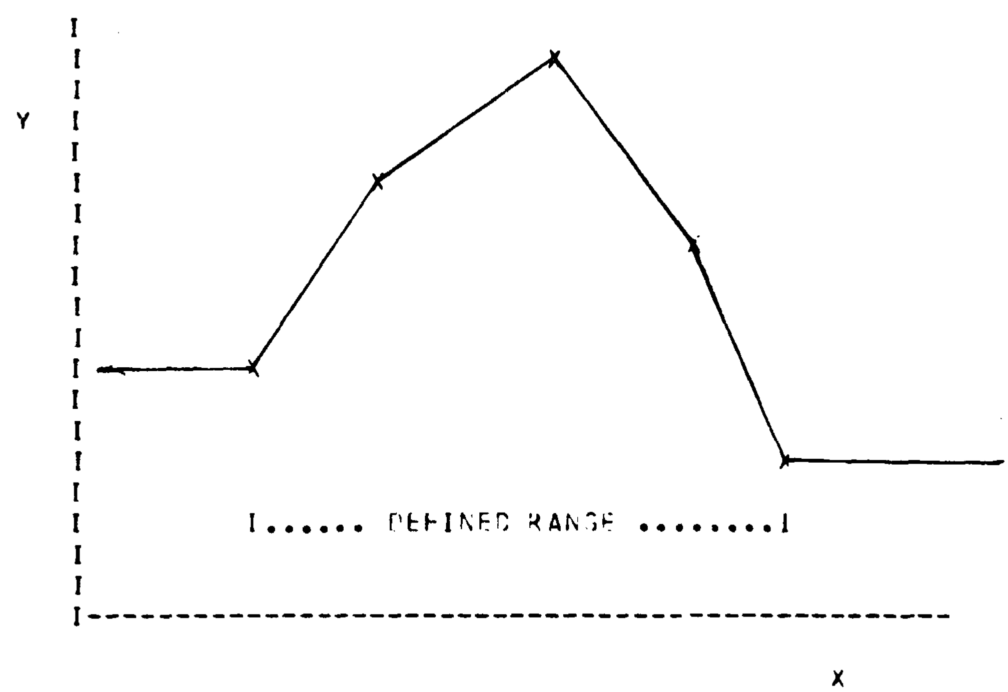
```
C
C   ...............................................................
C
C        FUNCTION XNEW
C
C        PURPOSE
C           COMPUTE A VALUE OF X ON THE CURVE   Y = M * LOG(X) + B
C           GIVEN THE VALUE OF Y FOR WHICH X IS TO BE COMPUTED AND
C           TWO X,Y COORDINATES ON THE CURVE
C
C        USAGE
C           VARIABLE = XNEW(X1, Y1, X2, Y2, YNEW)
C
C        DESCRIPTION OF PARAMETERS
C           X1   -  VALUE OF X AT FIRST COORDINATE POINT
C           Y1   -  VALUE OF Y AT FIRST COORDINATE POINT
C           X2   -  VALUE OF X AT SECOND COORDINATE POINT
C           Y2   -  VALUE OF Y AT SECOND COORDINATE POINT
C           YNEW   -  VALUE OF Y FOR WHICH X IS TO BE COMPUTED
C           XNEW   -  VALUE OF X AT  Y = YNEW
C
C        REMARKS
C           1.  IF Y1 AND Y2 ARE THE SAME, THEN XNEW IS SET
C               EQUAL TO X2.
C           2.  IF EITHER X1 OR X2 ARE EQUAL TO OR LESS THAN ZERO,
C               THEN XNEW IS SET EQUAL TO ZERO.
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           THE EQUATION USED IS GIVEN IN STATEMENT 30.
C
C   ...............................................................
C
      REAL FUNCTION XNEW (X1, Y1, X2, Y2, YNEW)
C
C          ----- CHECK FOR INVALID PARAMETERS -----
C
      IF (Y1 - Y2) 2, 10, 2
    2 IF (X1) 20, 20, 3
    3 IF (X2) 20, 20, 30
C
C          ----- IF Y1 AND Y2 ARE THE SAME -----
C
   10 XNEW = X2
      RETURN
C
C          ----- IF X1 AND X2 ARE LESS THAN OR EQUAL TO ZERO -----
C
   20 XNEW = 0.
      RETURN
C
C          ----- IF PARAMETERS ARE OK -----
C
   30 XNEW = X2 * (X2/X1)**((YNEW-Y2)/(Y2-Y1))
      RETURN
C
      END
```

FIGURE C-18A. PROGRAM LISTING - FUNCTION XNEW

```
C
C     ...............................................................
C
C        FUNCTION OPCOR
C
C        PURPOSE
C           COMPUTE THE NEF CORRECTION FOR A GIVEN NUMBER OF
C           DAYTIME(0700 - 2200) AND NIGHTTIME (2200 - 0700)
C           AIRCRAFT OPERATIONS.
C
C        USAGE
C           VARIABLE = OPCOR(NDAY, NNITE)
C
C        DESCRIPTION OF PARAMETERS
C           NDAY  -  NUMBER OF DAYTIME OPERATION (REAL)
C           NNITE  -  NUMBER OF NIGHTTIME OPERATIONS (REAL)
C           OPCOR  -  VALUE OF NEF CORRECTION (REAL)
C
C        REMARKS
C           1.  IF THE TOTAL NUMBER OF OPERATIONS (DAY AND NIGHT)
C               IS LESS THAN OR EQUAL TO ZERO, A VALUE OF -100 WILL
C               BE RETURNED AS THE NEF CORRECTION.
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           ALOG10
C
C        METHOD
C           THE DEFINITION OF THE NEF CORRECTION FOR A GIVEN NUMBER
C           OF DAYTIME AND NIGHTTIME OPERATIONS IS GIVEN IN STATEMENT 1.
C
C     ...............................................................
C
      REAL FUNCTION OPCOR(NDAY, NNITE)
      REAL NDAY, NNITE
C
C        ----- CHECK FOR INVALID PARAMETERS -----
C
      A = NDAY/20. + NNITE/1.2
      IF (A) 5, 5, 1
C
C        ----- PARAMETERS ARE VALID.  CALCULATE CORRECTION -----
C
    1 OPCOR = 10. * ALOG10(A)
      RETURN
C
C        ----- PARAMETERS ARE INVALID.  SET CORRECTION EQUAL TO -100 ---
C
    5 OPCOR = -100.
      RETURN
C
      END
```

FIGURE C-19. PROGRAM LISTING - FUNCTION OPCOR

```
C         REAL FUNCTION CURVE
C
C         PURPOSE
C            COMPUTES A Y VALUE FOR A GIVEN VALUE OF X, WHERE
C            Y = F(X) IS DEFINED BY X,Y POINTS CONNECTED BY STRAIGHT
C            LINE SEGMENTS.       (SEE BELOW)
C
C         USAGE
C            VARIABLE = CURVE(XL, DUMY, NPTS)
C
C         DESCRIPTION OF PARAMETERS
C            XL        - VALUE OF X FOR WHICH Y IS TO BE COMPUTED
C            DUMY(I,J)- TABLE OF X,Y POINTS TO DEFINE FUNCTION.
C                        I = 1 FOR X, 2 FOR Y
C            NPTS      - NUMBER OF X,Y POINTS DEFINING FUNCTION
C
C         REMARKS
C            1.   DUMY ARRAY MUST BE ORDERED SUCH THAT VALUES OF X
C                 ARE CONSTANTLY INCREASING
C            2.   IF NPTS IS EQUAL TO OR LESS THAN ZERO, A VALUE
C                 OF ZERO WILL BE RETURNED
C            3.   IF THE VALUE OF XL DOES NOT LIE WITHIN THE DEFINED
C                 RANGE OF THE FUNCTION, THEN THE VALUE OF Y FOR THE
C                 DEFINED VALUE OF X WHICH IS CLOSEST TO XL WILL BE
C                 RETURNED.  (IE THE FUNCTION LOOKS LIKE THIS.)
C
```




```
C
C         SUBROUTINES AND FUNCTIONS REQUIRED
C            NONE
C
C
```

FIGURE C-20A. PROGRAM LISTING - FUNCTION CURVE

```
      REAL FUNCTION CURVE (XL, DUMY, NPTS)
      DIMENSION DUMY(2, 10)
      J=1
      IF (NPTS - 1) 6, 5, 1
    1 IF (XL - DUMY(1,J)) 5, 5, 2
    2 DO 3 J=2,NPTS
      IF (XL - DUMY (1,J)) 4, 5, 3
    3 CONTINUE
      GO TO 5
    4 I = J - 1
      CURVE = (DUMY(2,J) - DUMY(2,I))*(XL - DUMY(1,I))/(DUMY(1,J) -
     1 DUMY(1,I))+ DUMY(2,I)
      RETURN
    5 CURVE = DUMY(2,J)
      RETURN
    6 CURVE = 0
      RETURN
      END
```

FIGURE C-208. PROGRAM LISTING - FUNCTION CURVE

APPENDIX D

COMPUTATION OF CONTOURS FOR PARALLEL RUNWAYS

APPENDIX D

COMPUTATION OF CONTOURS FOR PARALLEL RUNWAYS

The NEF program computes the loci of points of equal NEF value resulting from operations along a single flight path. Although it is possible to compute the NEF contours for many different flight paths in sequence, the program does not take into account the possible influence of operations on other nearby flight paths.

Larger airports in particular, are likely to have runways parallel to each other. Consequently the operations on one runway may influence the location of the contours about the other. There are two ways to solve this problem. One is to compute the NEF contours for the two flight paths directly, using a modification of the "normal" NEF program. Although conceptually very simple, this approach requires considerable more main storage, because the two paths may well have different profiles associated with them. As all efforts have been made to keep storage requirements to a minimum, so that users of small computers may use the program, expanding storage requirements for this purpose seems inconsistent.

The path chosen was to use the output of the NEF program as input for a merging program, which constructs a summed contour from the digital form of the NEF contour output. This program is completely separate from the NEF program although it needs the output of the NEF routine as input. In this report the merging program has been named MERGENEF. Dependent upon the needs of the user, however, the program can be incorporated in the main NEF program. Input and output media for MERGENEF as well as the particular way in which it may be called will be dependent upon the operating system and the preference of the programmer.

The MERGENEF program will accept the perpendicular contour distances to the flight tracks of two parallel, arbitrarily displaced tracks, and will compute the proper NEF contour distances for the operations on the two paths together. In order to perform this task the following input is required:

a. Title Card. This card is for identification and can contain up to 80 characters of identifying information.

TABLE VIII

MERGED CONTOUR CONTROL CARD #1 FORMAT

Columns	Contents	Mode	Range
1-7	Distance along flight track to commence merging	Integer	≥ 0 "
8-14	Increment along flight track between successive computations	"	"
15-18	NEF value of 1st merged contour	Real	"
19-22	" " " 2nd " "	"	"
23-26	" " " 3rd " "	"	"
27-30	" " " 4th " "	"	"
31-34	" " " 5th " "	"	"
35-38	" " " 6th " "	"	"
39-42	" " " 7th " "	"	"
43-46	" " " 8th " "	"	"
47-50	" " " 9th " "	"	"
51-54	" " " 10th " "	"	"

TABLE IX

MERGED CONTOUR CONTROL CARD #2

Columns	Contents	Mode	Range
1-7	Distance along the flight track to terminate computation	Integer	≥ 0
8-14	Offset in the X-direction	"	"
15-21	Offset in the Y-direction	"	"

TABLE X

CONTOUR CONTROL CARD FORMAT

Columns	Contents	Mode	Range
1-7	Distance along flight track for the first data point	Integer	≥ 0
8-14	Increment along flight track between successive data points	"	"
15-18	NEF value of the 1st NEF contour	Real	"
19-22	" " " " 2nd " "	"	"
23-26	" " " " 3rd " "	"	"
27-30	" " " " 4th " "	"	"
31-34	" " " " 5th " "	"	"
35-38	" " " " 6th " "	"	"
39-42	" " " " 7th " "	"	"
43-46	" " " " 8th " "	"	"
47-50	" " " " 9th " "	"	"
51-54	" " " " 10th " "	"	"

TABLE XI

DATA CARD FORMAT

Columns	Contents					Mode	Range
1-7			x-distance			Integer	>0
8-14	Distance	to	1st	NEF	contour	"	"
15-21	"	"	2nd	"	"	"	"
22-28	"	"	3rd	"	"	"	"
29-35	"	"	4th	"	"	"	"
36-42	"	"	5th	"	"	"	"
43-49	"	"	6th	"	"	"	"
50-56	"	"	7th	"	"	"	"
57-63	"	"	8th	"	"	"	"
64-70	"	"	9th	"	"	"	"
71-77	"	"	10th	"	"	"	"

TABLE XII

CONTOUR CLOSING CARD FORMAT

Columns	Contents						Mode	Range
1-7	Closing	distance	of	1st	NEF	contour	Integer	>0
8-14	"	"	"	2nd	"	"	"	"
15-21	"	"	"	3rd	"	"	"	"
22-28	"	"	"	4th	"	"	"	"
29-35	"	"	"	5th	"	"	"	"
36-42	"	"	"	6th	"	"	"	"
43-49	"	"	"	7th	"	"	"	"
50-56	"	"	"	8th	"	"	"	"
57-63	"	"	"	9th	"	"	"	"
64-70	"	"	"	10th	"	"	"	"

b. Merged contour control card #1. This card specifies the location along the flight path where contour calculations are to commence and the increment along the flight track for which the contour distances are to be computed. The set of NEF contour values for which the contour distances are to be computed must also be specified on the card.

c. Merged contour control card #2. This card specifies the point at which the merging process is to be terminated as well as the offset of the runways (flight paths).

For each of the two contours to be merged the data to be processed must appear in the following sequence:

a. A title card. This card identifies one of the contours to be merged (up to 80 characters are allowed).

b. A contour control card. This card details the starting location and the interval along the track of successive data cards and the NEF values of the successive data points on each card.

c. Data cards containing the distance along the flight track and the distances to the contours for the given track.

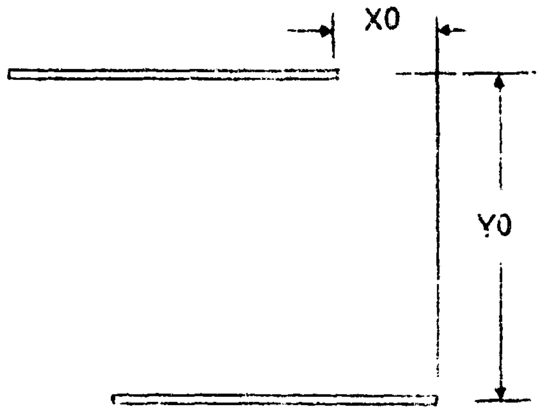
d. A blank card.

e. A card with the closing point of the various NEF contours. The columns pertaining to contours which do not close in the interval under consideration are left blank.

f. The same sequence as a through e but now for the second track.

The various card formats are described in Tables VIII through XII.

All distances, which are read into or generated by the program must be uniquely defined. In order to make input to MERGENEF compatible with the output of the NEF program the data describing the contours for each runway are accepted in the same form as the output generated by the NEF program. For purposes of output, however, a unique origin must be specified. MERGENEF assumes that the point of takeoff roll of the <u>first</u> runway for which data are entered is the new origin. As the contours obtained will generally be plotted, the output of MERGENEF has been organized with this use for the data in mind.



To find the correct offset distances, do the following:

1. Choose one of the two runways to be the datum line.

2. Standing at the beginning of this runway and looking along the flighttrack, is the other runway on the left or on the right?

3. If the other runway is on the left, Y0 is negative.
   If the other runway is on the right, Y0 is positive.

4. Still looking down the flighttrack, is the beginning of the other runway ahead of you or behind you?

5. If the other runway is ahead, X0 is positive.
   If the other runway is behind, X0 is negative.
   If the other runway is right next to you, X0 is zero.

EXAMPLE: Runways 25-L and 25-R East Mythical Municipal Airport 1970.

1. Choose 1st. runway to be 25-L.
2. 25-R is to the right.
3. Y0 = + 7000'
4. Beginning of 25-R is ahead.
5. X0 = + 2000'.
   For results compare Fig. D-2.

1. Choose 1st. runway to be 25-R.
2. 25-L is to the left.
3. Y0 = - 7000'
4. Beginning of 25-L is behind.
5. X0 = - 2000'.
   For results compare Fig. D-3.

FIGURE D-1. ALGORITHM TO FIND CORRECT OFFSET DISTANCES.

The output for each point along the flight track, which is considered by the program as the datum line, consists of 4 sets of numbers. The first row gives the contour distances to the right of the right hand runway. The next two rows give the contour distances between the two runways, those closest to the right hand runway first, and the final row gives the contour distances to the left of the lefthand runway. All references to left and right are for a person looking down the flight track. All distances, which are to the left of the datum are given a negative sign.

The offset of the runways is also referenced to the same origin, and a displacement in the X-direction (along the track) given by the appropriately signed distance. The offset in the y-direction follows the same convention as is used in specifying the output. If the second runway is to the right of the first, the sign of the y-offset is positive, on the other hand, if the second runway is to the left of the first the y-offset is negative. A simple algorithm to find the correct offset values is given in Fig. D-1.

The examples, which we have carried through in Appendix A, can also serve here to illustrate the procedure. We assume that each of the two examples gives the operations on only one of two parallel runways. The desired contours are then obtained by using the MERGENEF program. The results are shown in Figs. D-2 and D-4 for the 1970 and 1990 forecasts respectively. Fig. D-3 shows an alternate computation of the 1970 contour to demonstrate how either one of the runways can be chosen for the datum.

A few words of caution are in order. The MERGENEF program is in essence an interpolation scheme which generates a summation curve from two overlapping curves. It is clear, that such a program is most accurate when a large number of points are known between which the desired curve must be fitted. The routine becomes therefore inaccurate when only a few contours are still existence at a particular X-value. In particular if only one or two contours are left (at fairly large X-values) the program cannot compute the contours accurately and the resultant output shows a diverging rather than a closing contour. The reason is that at those points the program performs on extrapolation, rather than an interpolation. Similarly the contour with the smallest NEF value is always extrapolated (it is always further away than the contours it was generated from) and therefore it is better to supply the program with one lower value NEF contour than is of interest (i.e. if the 25 NEF contour is desired MERGENEF will compute its location more accurately if the 20 NEF contour is also known to the program).

As it is necessary to punch the NEF contour information into cards for input to MERGENEF it is more convenient (and it eliminates errors) to have the computer punch out the necessary information while the NEF contours are computed.

One way to achieve this is shown in Fig. D-5. The modified NEF routine MODNEF will punch out the cards needed by MERGENEF in the right sequence. It is up to the user, however, to supply the identifying and control cards needed to run the merging program. This can be added after all cards are punched out and sorted into parallel runway sets. If cards are not the medium chosen to punch the MODNEF output (but for example a tape or disk), the user will have to make sure that the needed control information is inserted in the MODNEF output stream. Such extensive modifications are not within the scope of this text.

A flowchart of the MERGENEFF program is given in Fig. D-6, a program listing in Fig. D-7.

```
DOUBLE RUNWAYSET 25L-25R      1970
    START=        0     INTERVAL=     2000     CONTOURS    25. 30. 35. 40. 45. 50. 55.
    STOP  =   16000     X,Y OFFSET OF 2ND RUNWAY=     2000,     7000

EAST MYTHICAL MUNICIPAL AIRPORT  - FORECAST 1970  25-L
    START=        0     INTERVAL=     2000     CONTOURS    25. 30. 35. 40. 45. 50. 55.

EAST MYTHICAL MUNICIPAL AIRPORT  - FORECAST 1970  25-R
    START=        0     INTERVAL=     2000     CONTOURS    25. 30. 35. 40. 45. 50. 55.
```

DISTANCE TO MERGED CONTOUR (IN FEET) FROM FIRST RUNWAY

CONTOUR VALUES	25.	30.	35.	40.	45.	50.	55.
X VALUE= 2000.							
TO RIGHT	12117	10845	9799	8907	8173	7646	7321
BETWEEN			4057	5091	5829	6346	6773
RUNWAYS			2575	1594	923	510	87
TO LEFT	-4556	-3401	-2423	-1592	-920	-497	-236
X VALUE= 4000.							
TO RIGHT	11540	10394	9421	8591	7920	7437	7236
BETWEEN			4531	5415	6072	6489	6912
RUNWAYS			2113	1307	736	390	36
TO LEFT	-4083	-2987	-2065	-1378	-732	-384	-178
X VALUE= 6000.							
TO RIGHT	11071	9983	9063	8307	7732	7384	7178
BETWEEN		3725	4913	5696	6264	6609	6963
RUNWAYS		2912	1769	1057	604	289	13
TO LEFT	-3631	-2617	-1745	-1054	-590	-302	-137
X VALUE= 8000.							
TO RIGHT	10624	9614	8745	8053	7590	7302	7137
BETWEEN		4277	5242	5944	6395	6710	6986
RUNWAYS		2517	1592	1012	686	366	41
TO LEFT	-3381	-2401	-1577	-994	-689	-365	-164
X VALUE= 10000.							
TO RIGHT	10365	9395	8575	7994	7589	7365	7164
BETWEEN		4530	5415	5989	6313	6633	6958
RUNWAYS		2991	1973	1213	754	447	171
TO LEFT	-3613	-2890	-1973	-1192	-746	-463	-201
X VALUE= 12000.							
TO RIGHT	10663	9509	8978	8195	7746	7463	7201
BETWEEN			4576	5779	6244	6551	
RUNWAYS			2263	1457	802	392	6800
TO LEFT	-4722	-3147	-2269	-1466	-909	-392	0
X VALUE= 14000.							
TO RIGHT	11753	10162	9264	8467	7935	7362	0
BETWEEN			4640	5530	6136		
RUNWAYS			2620	1408	631	6648	
TO LEFT	-5076	-3967	-2500	-1398	-587	0	0
X VALUE= 16000.							
TO RIGHT	12119	10985	9502	8401	7647	0	0
BETWEEN			4171	5576	6360		
RUNWAYS			2673	1277	369		
TO LEFT	-5911	-4049	-2424	-1288	-472	0	0

FIGURE D-2. EXAMPLE OF COMPUTER OUTPUT - MERGED CONTOUR DISTANCE LISTING, EXAMPLE 1

DOUBLE RUNWAYSET 25L-25R 1970 ALTERNATE COMPUTATION
START= 0 INTERVAL= 2000 CONTOURS 25. 30. 35. 40. 45. 50. 55.
STOP = 16000 X,Y OFFSET OF 2ND RUNWAY= -2000, -7000

EAST MYTHICAL MUNICIPAL AIRPORT - FORECAST 1970 25-R
START= 0 INTERVAL= 2000 CONTOURS 25. 30. 35. 40. 45. 50. 55.

EAST MYTHICAL MUNICIPAL AIRPORT - FORECAST 1970 25-L
START= 0 INTERVAL= 2000 CONTOURS 25. 30. 35. 40. 45. 50. 55.

DISTANCE TO MERGED CONTOUR (IN FEET) FROM FIRST RUNWAY

CONTOUR VALUES	25.	30.	35.	40.	45.	50.	55.
X VALUE= 0.							
TO RIGHT	5117	3845	2799	1907	1173	646	321
BETWEEN			-4424	-5405	-6070	-6489	-6912
RUNWAYS			-2942	-1908	-1170	-653	-226
TO LEFT	-11556	-10401	-9423	-8502	-7920	-7497	-7236
X VALUE= 2000.							
TO RIGHT	4540	3394	2421	1591	920	497	236
BETWEEN			-4886	-5692	-6263	-6609	-5763
RUNWAYS			-2466	-1584	-927	-510	-87
TO LEFT	-11080	-9987	-9065	-8309	-7732	-7384	-7179
X VALUE= 4000.							
TO RIGHT	4071	2983	2064	1307	732	384	178
BETWEEN		-4067	-5233	-5942	-6395	-6710	-5986
RUNWAYS		-3274	-2391	-1303	-735	-399	-36
TO LEFT	-10631	-9617	-8744	-8054	-7533	-7302	-7137
X VALUE= 6000.							
TO RIGHT	4424	2514	1749	1053	590	302	137
BETWEEN		-4472	-5407	-5987	-6313	-6633	-6958
RUNWAYS		-2722	-1757	-1055	-624	-283	-13
TO LEFT	-10381	-9401	-8577	-7994	-7689	-7365	-7164
X VALUE= 8000.							
TO RIGHT	3365	2455	1575	994	689	365	164
BETWEEN		-4004	-5025	-5786	-6245	-6552	-6928
RUNWAYS		-2463	-1594	-1011	-685	-366	-41
TO LEFT	-10613	-9659	-8973	-8192	-7746	-7463	-7201
X VALUE= 10000.							
TO RIGHT	3663	2906	1478	1195	746	463	201
BETWEEN			-4735	-5542	-6147	-6647	
RUNWAYS			-2023	-1220	-755	-448	-171
TO LEFT	-11722	-10147	-9263	-8466	-7990	-7362	0
X VALUE= 12000.							
TO RIGHT	4753	3162	2264	1457	909	352	0
BETWEEN			-4379	-5591	-6370		
RUNWAYS			-2351	-1460	-803	-352	
TO LEFT	-12076	-10967	-9503	-8394	-7687	0	0
X VALUE= 14000.							
TO RIGHT	5119	3885	2505	1401	597	0	0
BETWEEN			-4325	-5722	-6631		
RUNWAYS			-2424	-1423	-639		
TO LEFT	-12411	-11345	-9428	-8298	-7472	0	0
X VALUE= 16000.							
TO RIGHT	6314	4204	2443	1391	475	0	0
BETWEEN			-4233	-5906			
RUNWAYS			-2914	-1335	-474		
TO LEFT	-14075	-11377	-9351	-8142	0	0	0

FIGURE D-3. EXAMPLE OF COMPUTER OUTPUT - MERGED CONTOUR DISTANCE LISTING, ALTERNATE COMPUTATION OF EXAMPLE 1

```
DOUBLE RUNWAYSET 25L-25R      1990
    START=        0     INTERVAL=     2000     CONTOURS    25. 30. 35. 40. 45. 50. 55.
    STOP =   100000     X,Y OFFSET OF 2ND RUNWAY=      2000,     7000

EAST MYTHICAL MUNICIPAL AIRPORT  -  FORECAST 1990  25-L
    START=        0     INTERVAL=     2000     CONTOURS    25. 30. 35. 40. 45. 50. 55.

EAST MYTHICAL MUNICIPAL AIRPORT  -  FORECAST 1990  25-R
    START=        0     INTERVAL=     2000     CONTOURS    25. 30. 35. 40. 45. 50. 55.
```

DISTANCE TO MERGED CONTOUR (IN FEET) FROM FIRST RUNWAY

CONTOUR VALUES	25.	30.	35.	40.	45.	50.	55.
X VALUE= 2000.							
TO RIGHT	14770	13179	11776	10699	9810	9021	8333
BETWEEN					4117	4979	5690
RUNWAYS					2352	1550	910
TO LEFT	-6715	-5253	-4062	-3117	-2279	-1550	-951
X VALUE= 4000.							
TO RIGHT	13665	12228	11051	10113	9277	8550	7951
BETWEEN				3672	4716	5458	5091
RUNWAYS				2771	1804	1141	612
TO LEFT	-5714	-4457	-3431	-2570	-1788	-1144	-654
X VALUE= 6000.							
TO RIGHT	12682	11442	10425	9568	8797	8144	7654
BETWEEN				4382	5209	5860	6387
RUNWAYS				2417	1527	1076	738
TO LEFT	-5325	-4167	-3209	-2363	-1619	-1065	-803
X VALUE= 8000.							
TO RIGHT	12300	11156	10204	9362	8619	8065	7803
BETWEEN				4567	5375	5924	6261
RUNWAYS				3469	2324	1446	793
TO LEFT	-5277	-4600	-3952	-3367	-2322	-1442	-980
X VALUE= 10000.							
TO RIGHT	12269	11597	10951	10367	9322	8442	7880
BETWEEN					4504	5529	6201
RUNWAYS					2346	1362	703
TO LEFT	-7369	-6622	-5902	-3685	-2321	-1363	-711
X VALUE= 12000.							
TO RIGHT	14273	13624	12901	10697	9335	8370	7712
BETWEEN					4432	5611	6291
RUNWAYS					2438	1179	362
TO LEFT	-9647	-9409	-6161	-3729	-2220	-1195	-394
X VALUE= 14000.							
TO RIGHT	16661	15917	13221	10898	9266	9197	7396
BETWEEN					4539	5814	
RUNWAYS					2305	853	6634
TO LEFT	-11972	-10867	-6352	-3694	-2090	-857	-91
X VALUE= 16000.							
TO RIGHT	19265	17990	13731	10867	9117	7901	7091
BETWEEN					4703	6144	
RUNWAYS					2043	406	6920
TO LEFT	-14301	-10872	-6389	-3751	-1893	-484	0
X VALUE= 18000.							
TO RIGHT	22685	18582	13883	10764	8895	7484	0
BETWEEN					4977		
RUNWAYS					1688	6600	
TO LEFT	-17181	-11626	-6709	-3600	-1594	0	0

FIGURE D-4. EXAMPLE OF COMPUTER OUTPUT - MERGED CONTOUR DISTANCE LISTING, EXAMPLE 2

DISTANCE TO MERGED CONTOUR (IN FEET) FROM FIRST RUNWAY

CONTOUR VALUES	25.	30.	35.	40.	45.	50.	55.
X VALUE= 20000.							
TO RIGHT	26219	19570	14009	10617	8596	0	0
BETWEEN					5334		
RUNWAYS					1251		
TO LEFT	-20293	-12450	-6926	-3426	-1205	0	0
X VALUE= 22000.							
TO RIGHT	28429	19955	13930	10427	8203	0	0
BETWEEN					5765		
RUNWAYS					738		
TO LEFT	-21410	-12929	-6828	-3204	-697	0	0
X VALUE= 24000.							
TO RIGHT	28373	19924	13835	10203	7694	0	0
BETWEEN					6271		
RUNWAYS					9		
TO LEFT	-21393	-13013	-6715	-2928	-49	0	0
X VALUE= 26000.							
TO RIGHT	28848	20175	13748	9925	7037	0	0
BETWEEN							
RUNWAYS					6993		
TO LEFT	-22117	-13051	-6587	-2598	0	0	0
X VALUE= 28000.							
TO RIGHT	29790	20271	13604	9585	0	0	0
BETWEEN							
RUNWAYS							
TO LEFT	-22861	-13193	-6444	-2184	0	0	0
X VALUE= 30000.							
TO RIGHT	30038	20230	13428	9168	0	0	0
BETWEEN				3868			
RUNWAYS				2490			
TO LEFT	-23015	-13161	-6259	-1663	0	0	0
X VALUE= 32000.							
TO RIGHT	30072	20153	13232	8644	0	0	0
BETWEEN				4797			
RUNWAYS				1325			
TO LEFT	-23073	-13072	-6040	-1000	0	0	0
X VALUE= 34000.							
TO RIGHT	30183	20078	13012	7979	0	0	0
BETWEEN				5794			
RUNWAYS				78			
TO LEFT	-23183	-12992	-5794	-496	0	0	0
X VALUE= 36000.							
TO RIGHT	30246	19979	12759	7405	0	0	0
BETWEEN							
RUNWAYS				6935			
TO LEFT	-23219	-12891	-5517	0	0	0	0

FIGURE D -4. EXAMPLE OF COMPUTER OUTPUT - MERGED CONTOUR DISTANCE LISTING, EXAMPLE 2 (CONTINUED)

DISTANCE TO MERGED CONTOUR (IN FEET) FROM FIRST RUNWAY

CONTOUR VALUES	25.	30.	35.	40.	45.	50.	55.
X VALUE= 38000.							
TO RIGHT	30242	19862	12476	0	0	0	0
BETWEEN							
RUNWAYS							
TO LEFT	-23219	-12766	-5200	0	0	0	0
X VALUE= 40000.							
TO RIGHT	30251	19736	12154	0	0	0	0
BETWEEN							
RUNWAYS							
TO LEFT	-23226	-12633	-4839	0	0	0	0
X VALUE= 42000.							
TO RIGHT	30251	19597	11786	0	0	0	0
BETWEEN							
RUNWAYS							
TO LEFT	-23221	-12488	-4424	0	0	0	0
X VALUE= 44000.							
TO RIGHT	30242	19447	11359	0	0	0	0
BETWEEN							
RUNWAYS							
TO LEFT	-23211	-12330	-3955	0	0	0	0
X VALUE= 46000.							
TO RIGHT	30235	19285	10856	0	0	0	0
BETWEEN							
RUNWAYS							
TO LEFT	-23203	-12161	-3397	0	0	0	0
X VALUE= 48000.							
TO RIGHT	30232	19112	10273	0	0	0	0
BETWEEN							
RUNWAYS							
TO LEFT	-23200	-11979	-2637	0	0	0	0
X VALUE= 50000.							
TO RIGHT	30234	18927	9533	0	0	0	0
BETWEEN			4159				
RUNWAYS			242				
TO LEFT	-23203	-11785	-1199	0	0	0	0
X VALUE= 52000.							
TO RIGHT	30244	18730	7927	0	0	0	0
BETWEEN			6805				
RUNWAYS			163				
TO LEFT	-23213	-11578	-843	0	0	0	0
X VALUE= 54000.							
TO RIGHT	30263	18519	7680	0	0	0	0
BETWEEN			6839				
RUNWAYS			123				
TO LEFT	-23232	-11355	-585	0	0	0	0

FIGURE D-4. EXAMPLE OF COMPUTER OUTPUT - MERGED CONTOUR DISTANCE LISTING, EXAMPLE 2 (CONTINUED)

DISTANCE TO MERGED CONTOUR (IN FEET) FROM FIRST RUNWAY

CONTOUR VALUES	25.	30.	35.	40.	45.	50.	55.
X VALUE= 56000.							
TO RIGHT	30292	18292	7445	0	0	0	0
BETWEEN			6879				
RUNWAYS			78				
TO LEFT	-23263	-11115	-329	0	0	0	0
X VALUE= 58000.							
TO RIGHT	30336	18049	7230	0	0	0	0
BETWEEN			6925				
RUNWAYS			24				
TO LEFT	-23309	-10859	-88	0	0	0	0
X VALUE= 60000.							
TO RIGHT	30397	1778[illegible]	7050	0	0	0	0
BETWEEN							
RUNWAYS			6979				
TO LEFT	-23374	-10583	0	0	0	0	0
X VALUE= 62000.							
TO RIGHT	30480	17508	0	0	0	0	0
BETWEEN							
RUNWAYS							
TO LEFT	-23462	-10285	0	0	0	0	0
X VALUE= 64000.							
TO RIGHT	30593	17205	0	0	0	0	0
BETWEEN							
RUNWAYS							
TO LEFT	-23580	-9962	0	0	0	0	0
X VALUE= 66000.							
TO RIGHT	30743	16876	0	0	0	0	0
BETWEEN							
RUNWAYS							
TO LEFT	-23740	-9610	0	0	0	0	0
X VALUE= 68000.							
TO RIGHT	30944	16517	0	0	0	0	0
BETWEEN							
RUNWAYS							
TO LEFT	-23951	-9223	0	0	0	0	0
X VALUE= 70000.							
TO RIGHT	31210	16121	0	0	0	0	0
BETWEEN							
RUNWAYS							
TO LEFT	-24234	-8793	0	0	0	0	0
X VALUE= 72000.							
TO RIGHT	31574	15682	0	0	0	0	0
BETWEEN							
RUNWAYS							
TO LEFT	-24622	-8306	0	0	0	0	0

FIGURE D-4. EXAMPLE OF COMPUTER OUTPUT - MERGED CONTOUR DISTANCE LISTING, EXAMPLE 2 (CONTINUED)

DISTANCE TO MERGED CONTOUR (IN FEET) FROM FIRST RUNWAY

CONTOUR VALUES	25.	30.	35.	40.	45.	50.	55.
X VALUE= 74000.							
TO RIGHT	32061	15183	0	0	0	0	0
BETWEEN							
RUNWAYS							
TO LEFT	-25139	-7766	0	0	0	0	0
X VALUE= 76000.							
TO RIGHT	32757	14626	0	0	0	0	0
BETWEEN							
RUNWAYS							
TO LEFT	-25899	-7098	0	0	0	0	0
X VALUE= 78000.							
TO RIGHT	33878	13930	0	0	0	0	0
BETWEEN							
RUNWAYS							
TO LEFT	-27133	-6242	0	0	0	0	0
X VALUE= 80000.							
TO RIGHT	35871	13024	0	0	0	0	0
BETWEEN							
RUNWAYS							
TO LEFT	-29412	-4996	0	0	0	0	0
X VALUE= 82000.							
TO RIGHT	45890	11480	0	0	0	0	0
BETWEEN		1373					
RUNWAYS		672					
TO LEFT	-46585	-4387	0	0	0	0	0
X VALUE= 84000.							
TO RIGHT	105704	10465	0	0	0	0	0
BETWEEN		6778					
RUNWAYS		203					
TO LEFT	-99896	-3245	0	0	0	0	0
X VALUE= 86000.							
TO RIGHT	108015	10044	0	0	0	0	0
BETWEEN		6799					
RUNWAYS		181					
TO LEFT	-102314	-2902	0	0	0	0	0
X VALUE= 88000.							
TO RIGHT	110409	9613	0	0	0	0	0
BETWEEN		6822					
RUNWAYS		156					
TO LEFT	-104830	-2346	0	0	0	0	0
X VALUE= 90000.							
TO RIGHT	112899	9172	0	0	0	0	0
BETWEEN		6847					
RUNWAYS		129					
TO LEFT	-107453	-1876	0	0	0	0	0

FIGURE D-4. EXAMPLE OF COMPUTER OUTPUT - MERGED CONTOUR DISTANCE LISTING, EXAMPLE 2 (CONTINUED)

DISTANCE TO MERGED CONTOUR (IN FEET) FROM FIRST RUNWAY

CONTOUR VALUES		25.	30.	35.	40.	45.	50.	55.
X VALUE= 2000.								
	TO RIGHT	115500	8720	0	0	0	0	0
	BETWEEN		6874					
	RUNWAYS		60					
	TO LEFT	-110198	-1392	0	0	0	0	0
X VALUE= 94000.								
	TO RIGHT	118243	8259	0	0	0	0	0
	BETWEEN		6904					
	RUNWAYS		65					
	TO LEFT	-113119	-894	0	0	0	0	0
X VALUE= 96000.								
	TO RIGHT	121156	7768	0	0	0	0	0
	BETWEEN		6938					
	RUNWAYS		27					
	TO LEFT	-116236	-382	0	0	0	0	0
X VALUE= 98000.								
	TO RIGHT	124282	7367	0		0	0	0
	BETWEEN							
	RUNWAYS		6974					
	TO LEFT	-119587	0	0	0	0	0	0

FIGURE D-4. EXAMPLE OF COMPUTER OUTPUT - MERGED CONTOUR DISTANCE LISTING, EXAMPLE 2 (CONTINUED)

```
C     ---------------------------------------------------------------------
C
C
C     THIS IS A SIMPLE MODIFICATION OF THE NEF CONTROL ROUTINE
C     IN THIS FORM THE ROUTINE PUNCHES OUT THE CARDS
C     REQUIRED FOR THE MERGENEF CONTOUR MERGING ROUTINE
C     ALL OTHER FUNCTIONS OF THE ROUTINE ARE UNCHANGED
C
C     ---------------------------------------------------------------------
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS CALLED
C           CARDIN, APD, ANEF, XNEW, LASTPT
C
C   ..................................................................
      COMMON  XSTART, XEND, DX, NAC, CONTUR, NMAX(4)                    COM  1
      COMMON  NTET(4), NFET(4), NALT(12), NPOW(12), ACALC(12), PCALC(12) COM  2
      COMMON  DSCNAM(20), ALTNAM(12), POWNAM(12), PNLNAM(25), APTR(50)  COM  3
      COMMON  PPTR(50), EPTR(50), OPER(50), Z(12), POWSPL(12), IDENT(13) COM  4
      COMMON  DSCRPT(3,20), ALTDIR(3,20), POWDIR(3,20), EPNDIR(8,20)    COM  5
      COMMON  PNLEFF(35,2,25), ALT(10,2,12), POWER(10,2,12)             COM  6
      DIMENSION  NEFCON(7), IY(7), XCLOSE(7)
C     ---------------------------------------------------------------------
C
C     THE ARRAY IXCLS IS INTRODUCED FOR USE WITH THE PUNCHING OF THE LAST CARD.
C
      DIMENSION IXCLS(7)
C
C     ---------------------------------------------------------------------
      REAL  NEF1, NEF2
C
      NEFCON(1) = 25
      NEFCON(2) = 30
      NEFCON(3) = 35
      NEFCON(4) = 40
      NEFCON(5) = 45
      NEFCON(6) = 50
      NEFCON(7) = 55
C
      NMAX(1) = 25
      NMAX(2) = 51
      NMAX(3) = 51
      NMAX(4) = 30
      DO 1  I=1,4
      NTET(I) = 0
    1 NFET(I) = 0
C
    2 CALL CARDIN
C
C     ---------------------------------------------------------------------
C
C     THE IDENTIFYING INFORMATION OF THE DATA IS PUNCHED HERE.
C
      WRITE(2,2010) IDENT
 2010 FORMAT(13A4)
C
```

FIGURE D-5A. PROGRAM LISTING - MODNEF CONTROL ROUTINE

```
C
C     JST AND JDX ARE INTRODUCED TO CREATE THE DESIRED PUNCHED OUTPUT.
C
      JST = XSTART
      JDX = DX
      WRITE(2,2001) JST, JDX, (NEFCON(I),I=1,7)
 2001 FORMAT(2I7,10I4)
C
C     ------------------------------------------------------------------------
      X = XSTART
      NNEXT = 7
      LINCNT = 0
      DO 3  N=1,7
      XCLOSE(N) = 0.
    3 IY(N) = 4000
C
   11 CALL APD(X)
      NCON = NNEXT
C
      DO 40  N=1,NCON
      CONTUR = NEFCON(N)
      Y2 = IY(N)
      Y1 = Y2 * 0.707
      CALL ANEF (Y2, NEF2)
      CALL ANEF (Y1, NEF1)
C
   21 YC = XNEW(Y1, NEF1, Y2, NEF2, CONTUR)
      IF (YC - 30000.) 23, 23, 22
   22 YO = 30000.
   23 CONTINUE
      IF (YO - 12.E) 24, 25, 25
C
   24 CONTINUE
      CALL LASTPT(X, XCLOSE(N))
      YO = 0.
      NNEXT = NNEXT - 1
      GO TO 28
C
   25 IF (ABS(YO - Y1) - 2.) 28, 28, 26
   26 IF (ABS((YO - Y1) / YO) - 0.001) 28, 28, 27
   27 Y2 = Y1
      NEF2 = NEF1
      Y1 = YO
      CALL ANEF (Y1, NEF1)
      GO TO 21
C
   28 IY(N) = YO + 0.5
   40 CONTINUE
C
C
      LINCNT = LINCNT + 1
      IF (LINCNT) 43, 42, 42
   42 LINCNT = -50
      WRITE (3, 3004)  IDENT
      WRITE (3, 3001)  NEFCON
```

FIGURE D-5b. PROGRAM LISTING - MODNEF CONTROL ROUTINE

```
C
   43 IX = X + 0.5
      WRITE (3, 3005) IX, (IY(N), N=1,NCON)
C     -----------------------------------------------------------------
C
C     THE DISTANCE ALONG THE TRACK AND THE CONTOUR DISTANCES ARE PUNCHED HERE.
C
      WRITE(2,2002) IX, (IY(I), I=1,NCON)
 2002 FORMAT(11I7)
C
C     -----------------------------------------------------------------
C
      IF (NNEXT) 55, 55, 53
C
   53 X = X + DX
C
      IF (X - XEND) 11, 11, 55
C
   55 IFLG = 2
      WRITE (3, 3006)
      DO 70  N=1,7
      IX = XCLOSE(N) + 0.5
      IF (IX) 70, 70, 60
   60 GO TO (61, 62, 63, 64, 65, 66, 67), N
   61 WRITE (3, 3031) IX
      GO TO 69
   62 WRITE (3, 3032) IX
      GO TO 69
   63 WRITE (3, 3033) IX
      GO TO 69
   64 WRITE (3, 3034) IX
      GO TO 69
   65 WRITE (3, 3035) IX
      GO TO 69
   66 WRITE (3, 3036) IX
      GO TO 69
   67 WRITE (3, 3037) IX
   69 IFLG = 1
   70 CONTINUE
C     -----------------------------------------------------------------
C
C     THE BLANK CARD AT THE END OF THE CONTOUR DISTANCE DATASET IS PUNCHED HERE
C
      WRITE(2,2000)
 2000 FORMAT(' ')
C
C     THE CARD CONTAINING THE CLOSING POINTS OF THE CONTOURS IS PUNCHED HERE
C
      DO 2004  I =1, 7
 2004 IXCLS(I) = XCLOSE(I) + 0.5
      WRITE(2,2002) (IXCLS(I), I = 1,7)
C
C     -----------------------------------------------------------------
      GO TO (71, 72), IFLG
   71 WRITE (3, 3030)
```

FIGURE D-5C. PROGRAM LISTING - MODNEF CONTROL ROUTINE

```
C
   72 GO TO 2
C
 3001 FORMAT ('0', 32X, 'DISTANCE TO CONTOUR IN FEET' / 114, 'POINT ON'/
     1  9X, 'FLIGHT TRACK', 7I8 / 9X, '------------    ------  ------  -
     2-----  ------  ------  ------  ------')
 3004 FORMAT ('1', 15X, 13A4 / 9X, '----------------------------------------
     1----------------------------------')
 3005 FORMAT (10X, I8, 4X, 7I8)
 3006 FORMAT ('0')
 3030 FORMAT ('+', 7X, 'CONTOUR CLOSING' / 8X, 'POINT ON FLIGHT'
     1  / 8X, 'TRACK')
 3031 FORMAT ('+', 21X, I8)
 3032 FORMAT ('+', 29X, I8)
 3033 FORMAT ('+', 37X, I8)
 3034 FORMAT ('+', 45X, I8)
 3035 FORMAT ('+', 53X, I8)
 3036 FORMAT ('+', 61X, I8)
 3037 FORMAT ('+', 69X, I8)
C
      END
```

FIGURE D-5D. PROGRAM LISTING - MODNEF CONTROL ROUTINE

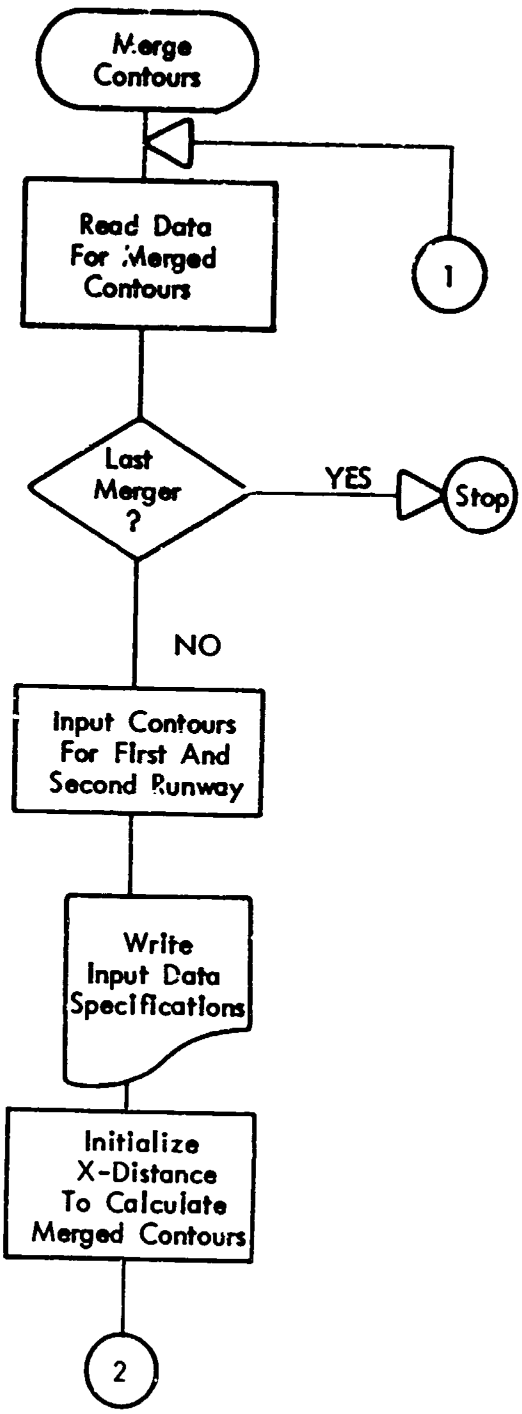


FIGURE D-6B. MERGENEF ROUTINE

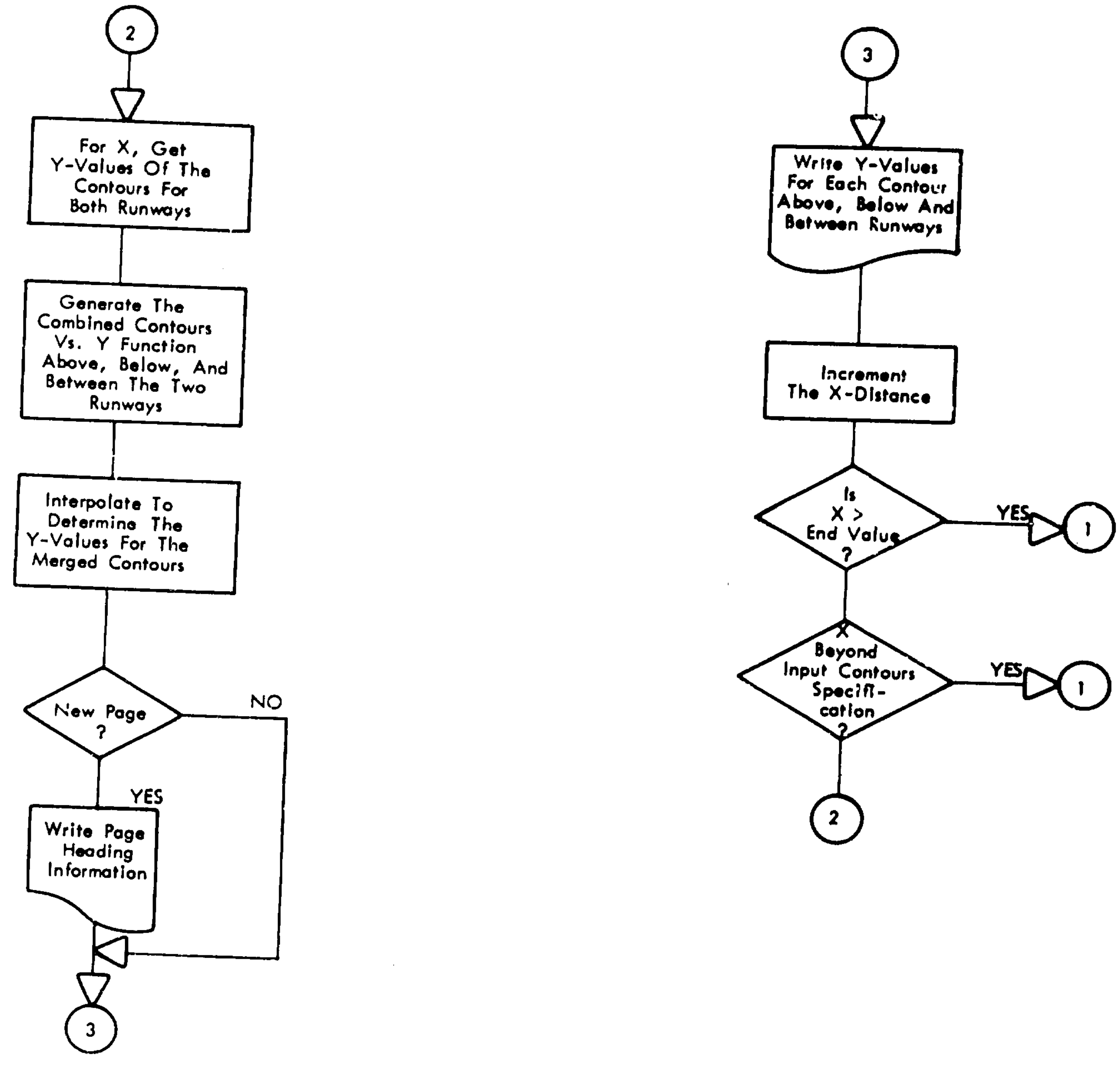


FIGURE D-38. MERGENEF ROUTINE

MERGENEF

```
      DIMENSION T1(10),T2(10),TP(22),TM(22),TB(21)
      DIMENSION CM(10),C1(10),C2(10),CP(22),CN(22),CB(21)
      DIMENSION RX1CLS(10),RX2CLS(10)
      INTEGER X1CLS(10), X2CLS(10), IRRAY(200,10)
      INTEGER DXM, DX1, DX2, XO, YO, XEND
       INTEGER XBEG, XBEG1, XBEG2
       DIMENSION  LBLM(20), LBL1(20), LBL2(20)
      DIMENSION MYP(10), MYM(10), MYA(10), MYB(10)
C  .......................................................
C
C
C       NEF CONTOUR MERGING ROUTINE
C
C
C  .......................................................
C      PURPOSE.  TO ACCEPT CONTOURS FROM TWO PARALLEL RUNWAYS
C                 AND GENERATE THE MERGED CONTOURS.
C
C  .......................................................
C      INPUT.
C      THE DATA FOR THE MERGER
C
C      1. LABEL FOR THE MERGER
C      2. X-START, X-INTERVAL, AND THE CONTOUR VALUES FOR THE MERGER
C      3. X-END, X-OFFSET, Y-OFFSET OF THE 2ND RUNWAY FROM THE 1ST
C
C      THE DATA FOR THE FIRST RUNWAY
C      1. LABEL FOR THE FIRST RUNWAY
C      2. X-START, X-INTERVAL, CONTOUR VALUES
C      3.  X-VALUE, Y-VALUES FOR EACH CONTOUR
C      4.  THE CONTOURS ARE TERMINATED WITH A BLANK CARD
C      5. THE SET OF DATA FOR THE FIRST RUNWAY IS TERMINATED
C        WITH THE X-VALUES FOR THE CLOSURE OF EACH CONTOUR
C
C      THE DATA FOR THE SECOND RUNWAY IS SIMILAR TO THAT OF THE FIRST
C
C  .......................................................
C
C      READ THE DATA FOR THE MERGER
C
  300  READ (1,1003)  LBLM
       READ (1,1000) XBEG, DXM, (CM(I), I=1,10)
      IF (DXM)  205,205,301
  301  CONTINUE
C
C      THE PROGRAM STOPS WHEN THE X-INTERVAL FOR MERGER IS 0
C
      DO 1  I = 1,10
      IF  (CM(I)) 2,2,1
    1  CONTINUE
      GOTO 3
    2 I = I -1
    3 NCM = I
      READ (1,1001)  XEND, XO, YO
C
```

FIGURE D-7A. PROGRAM LISTING - MERGENEF ROUTINE

```
                    MERGENEF

C       READ THE DATA FOR THE FIRST RUNWAY
        READ (1,1003)  LBL1
        READ (1,1000)  XBEG1, DX1, (C1(I), I=1,10)
       DO 4 I = 1,10
       IF (C1(I)) 5,5,4
    4  CONTINUE
       GOTO 6
    5   I = I -1
    6   NC1 = I
       J = 1
    7   READ (1,1002) IX, (IRRAY(J,I), I=1,NC1)
        IF (IRRAY(J,1))  302,302,8
  302   IF (IX)  9,9,8
    8   J = J + 1
       GOTO 7
    9   NX1 = J - 1
       READ (1,1002) (X1CLS(I), I=1,NC1)
       DO 700 NR=1,NC1
  700 RX1CLS(NR) = X1CLS(NR)
C
C       READ THE DATA FOR THE SECOND RUNWAY
        READ (1,1003)  LBL2
       READ (1,1000)  XBEG2, DX2, (C2(I), I=1,10)
       DO 10 I = 1,10
       IF (C2(I)) 11,11,10
   10   CONTINUE
       GOTO 12
   11   I = I - 1
   12   NC2 = I
       J = NX1 + 1
   13   READ (1,1002)  IX, (IRRAY(J,I), I=1,NC2)
        IF (IRRAY(J,1))  303,303,14
  303   IF (IX)  15,15,14
   14   J = J + 1
       GOTO 13
   15   NX2 = J - NX1 - 1
       READ (1,1002)  (X2CLS(I), I=1,NC2)
       DO 701 NR=1,NC2
  701   RX2CLS(NR) = X2CLS(NR)
        WRITE (3,3010)
C
C       WRITE THE INPUT DATA FOR IDENTIFICATION PURPOSES
C
        WRITE (3,3011)  LBLM
        WRITE (3,3012)  XBEG, DXM, (CM(I), I=1,NCM)
        WRITE (3,3013)  XEND, X0, Y0
        WRITE (3,3020)
        WRITE (3,3011)  LBL1
        WRITE (3,3012)  XBEG1, DX1, (C1(I), I=1,NC1)
        WRITE (3,3020)
        WRITE (3,3011)  LBL2
        WRITE (3,3012)  XBEG2, DX2, (C2(I), I=1,NC2)
        LINES = 60
C
C       FIND THE FIRST X-VALUE WHERE BOTH SETS OF CONTOURS ARE DEFINED
```

FIGURE D-7B. PROGRAM LISTING - MERGENEF ROUTINE

```
C
      XT = XBEG1
      IF (XT - XBEG2 - XO)  16,17,17
   16 XT = XBEG2 + XO
   17 X = XBEG
      RX = X
      IF(X) 18,702,18
  702 RX = 1.
   18 IF (X - XT)  19,20,20
   19 X = X + DXM
      RX = X
      GOTO 18
C
C       CHECK FOR CLOSURE OF THE CONTOUR
C
   20 CONTINUE
      J = (X - XBEG1)/DX1 + 1
      XP = (J - 1) * DX1 + XBEG1
      DO 26  I = 1,NC1
      IF (IRRAY(J,I))  22,22,28
   28 CONTINUE
      IF (X1CLS(I))  25,25,21
   21 IF(X - X1CLS(I))  23,22,22
   22 T1(I) = 0.
      GOTO 26
   23 IF (XP + DX1 - X1CLS(I)) 25,24,24
   24 T1(I) =  (( X1CLS(I)-X)/(X1CLS(I) - XP))*IRRAY(J,I)
      GOTO 26
   25 T1(I)=((X-XP)/DX1)*(IRRAY(J+1,I)-IRRAY(J,I))+IRRAY(J,I)
   26 CONTINUE
C
C       INTERPOLATE TO FIND THE Y-VALUES FOR THE TWO CONTOURS SETS AT X
C
      J = (X - XO - XBEG2)/DX2 + 1
      XP = (J-1) * DX2 + XO + XBEG2
      K = J + NX1
      DO 36  I=1,NC2
      IF (IRRAY(K,I))32,32,38
   38 CONTINUE
      IF (X2CLS(I))  35,35,31
   31 IF (X - X2CLS(I) - XO)  33,32,32
   32 T2(I) = 0.
      GOTO 36
   33 IF (XP + DX2 - X2CLS(I) - XO)  35,34,34
   34 T2(I) = ((X2CLS(I)-X)/(X2CLS(I)-XP))*IRRAY(K,I)
      GOTO 36
   35 T2(I) = ((X - XP)/DX2) *
     C  (IRRAY(K+1,I) - IRRAY(K,I)) + IRRAY(K,I)
   36 CONTINUE
C
C       T1, T2 ARE THE SET OF INTERPOLATED Y-VALUES FOR THE CONTOUR
C       VALUES OF THE FIRST AND SECOND RUNWAYS AT X
C
C       INTERPOLATE UNDER FLIGHT PATH
C
```

FIGURE D-7C. PROGRAM LISTING - MERGENEF ROUTINE

```
C       CX1 IS THE CONTOUR VALUE ON FIRST RUNWAY (UNDER PATH)
C
  350   NJ =  NC1-1
        DO 351  L=1,NJ
        IF (X1CLS(L))  351,351,352
  351   CONTINUE
        CX1= (0. - T1(NC1-1))/(T1(NC1) - T1(NC1-1))*
     C   (C1(NC1) - C1(NC1-1)) + C1(NC1-1)
        GOTO 360
  352   JS = L+1
        DO 353  J=JS,NC1
      IF(X1CLS(J))355,355,356
  356   IF (X - X1CLS(J))  353,354,354
  353   CONTINUE
 1356 CX1 = T1(J-2)/(T1(J-2)-T1(J-1))*(C1(J-1)-C1(J-2)) + C1(J-2)
      GOTO 360
  355 IF(J-2) 1355,1355,1356
 1355 WRITE(3,3048) X,C1(J)
      WRITE(3,3011) LBL1
      LINES = LINES + 5
      GOTO 399
  354 CX1 = ALOG(RX1CLS(J-1)/RX) / ALOG(RX1CLS(J-1)/RX1CLS(J)) *
     C   (C1(J) - C1(J-1)) + C1(J-1)
C
C       CX2 IS THE CONTOUR VALUE ON SECOND RUNWAY
C
  360   NJ = NC2-1
        DO 361  L=1,NJ
        IF (X2CLS(L))  361,361,362
  361   CONTINUE
        CX2= (0. - T2(NC2-1))/(T2(NC2) - T2(NC2-1))*
     C   (C2(NC2) - C2(NC2-1)) + C2(NC2-1)
        GOTO 357
  362   JS = L+1
        DO 363  J=JS,NC2
      IF(X2CLS(J)) 365,365,366
  366   IF (X - X2CLS(J))  363,364,364
  363   CONTINUE
 1366 CX2 = T2(J-2)/(T2(J-2)-T2(J-1))*(C2(J-1)-C2(J-2))+C2(J-2)
      GOTO 357
  365 IF(J-2) 1365,1365,1366
 1365 WRITE(3,3048) X, C2(J)
      WRITE(3,3011) LBL2
      LINES = LINES + 6
      GOTO 399
  364 CX2 = ALOG(RX2CLS(J-1)/ABS(RX-X0)) / ALOG(RX2CLS(J-1)/RX2CLS(J))*
     1 (C2(J)-C2(J-1))+C2(J-1)
  357   Y = 0.
        K = 4
        GOTO 520
  358   CCX1 = 10.*ALOG10(10.**(CX1/10.) + 10.**(CC2/10.))
        Y =  Y0
        K = 4
        GOTO 510
  368   CCX2 = 10.*ALOG10(10.**(CX2/10.) + 10.**(CC1/10.))
```

FIGURE D-7D. PROGRAM LISTING - MERGENEF ROUTINE

```
C
C
C      GENERATE THE CONTOUR VALUE VS. Y FUNCTION AT THE GIVEN X
C      ALONG THE LINE TO THE RIGHT OF THE TWO CONTOURS AND TO THE LEFT.
C
C
C      CP,TP ARE THE ASSOCIATED LISTS OF CONTOUR AND Y-VALUES TO THE RIGHT OF
C      THE TWO RUNWAYS.
C
C
C      CN,TM ARE THE ASSOCIATED LISTS OF CONTOUR AND Y-VALUES TO THE LEFT OF
C      THE TWO RUNWAYS.
C
C
C      THE CASE WHERE THE FIRST RUNWAY IS THE RIGHTMOST
C     OR WHERE TWO PATHS ARE SUPERPOSED
C
      IF (YO)  401,401,411
 401  NP = 0
      DO 404  I = 1,NC1
      Y = T1(I)
      IF (T1(I))  404,404,402
 402  K = 1
      GOTO 520
 403  NP = NP + 1
      CP(NP) =  10.*ALOG10(10.**(C1(I)/10.) + 10.**(CC2/10.))
      TP(NP) = Y
 404  CONTINUE
      DO 1410 I = 1, NC2
      Y = T2(I) + YO
      IF(Y) 1410,1410,1405
1405  K=5
      GOTO 510
1406  NP = NP +1
      CP(NP)=10.*ALOG10(10.**(C2(I)/10.)+10.**(CC1/10.))
      TP(NP) = Y
1410  CONTINUE
      NO = 1
      GOTO 600
1411   NM = 0
      DO 407  I=1,NC2
       Y = -T2(I) + YO
       IF (T2(I))  407,407,405
 405   K = 1
       GOTO 510
 406   NM = NM + 1
       CN(NM) = 10.*ALOG10(10.**(C2(I)/10.) + 10.**(CC1/10.))
       TM(NM) = -T2(I)
 407   CONTINUE
      DO 1450 I=1,NC1
      Y = -T1(I)
      IF(Y-YO) 1415,1450,1450
1415  K = 5
      GOTO 520
1416  NM = NM +  1
```

FIGURE D-7E. PROGRAM LISTING - MERGENEF ROUTINE

```
      CN(NM) = 10.*ALOG10(10.**(C1(I)/10.) + 10.**(CC2/10.))
      TM(NM) = Y - YO
 1450 CONTINUE
      NO = 1
      GOTO 650
 1451  NP = NP + 1
       CP(NP) = CCX1
       TP(NP) = 1.
       NM = NM + 1
       CN(NM) = CCX2
       TM(NM) = -1.
       YP = 0.
       YM = YO
       GOTO 420
C
C      THE CASE WHERE THE SECOND RUNWAY IS THE RIGHTMOST
C
  411  NP = 0
       DO 414  I=1,NC2
       Y = T2(I) + YO
       IF (T2(I))  414,414,412
  412  K = 2
       GOTO 510
  413  NP = NP + 1
       CP(NP) = 10.*ALOG10(10.**(C2(I)/10.) + 10.**(CC1/10.))
       TP(NP) = T2(I)
  414  CONTINUE
      DO 1460  I = 1,NC1
      Y = T1(I)
      IF(Y-YO) 1460,1460,1425
 1425 K=6
      GOTO 520
 1426 NP = NP + 1
      CP(NP) = 10.*(ALOG10(10.**(C1(I)/10.) + 10.**(CC2/10)) )
      TP(NP) = Y - YO
 1460 CONTINUE
      NO = 2
      GOTO 600
 1461  NM = 0
       DO 417  I=1,NC1
       Y = -T1(I)
       IF (T1(I))  417,417,415
  415  K = 2
       GOTO 520
  416  NM = NM + 1
       CN(NM) = 10.*ALOG10(10.**(C1(I)/10.) + 10.**(CC2/10.))
       TM(NM) = Y
  417  CONTINUE
      DO 1470 I = 1,NC2
      Y = -T2(I) + YO
      IF(Y) 1435,1470,1470
 1435 K=6
      GOTO 510
 1436 NM = NM + 1
      CN(NM) = 10. * ALOG10(10.**(C2(I)/10 ) + 10.**(CC1/10.))
```

FIGURE D-7F. PROGRAM LISTING - MERGENEF ROUTINE

```
      TM(NM) = Y
 1470 CONTINUE
      NC = 2
      GOTO 650
 1471  NP = NP + 1
       CP(NP) = CCX2
       TP(NP) = 1.
       NM = NM + 1
       CN(NM) = CCX1
       TM(NM) = -1.
       YP = YO
       YM = 0.
C
C      CB, TB ARE THE ASSOCIATED LISTS OF CONTOUR VALUES AND Y-VALUES
C      FOR THE REGION BETWEEN THE TWO RUNWAYS AT THE GIVEN X-DISTANCE
C
  420  Y = 0.
      DY = YO/20.
       DO 423  I=1,21
       K = 3
       GOTO 510
  421  K = 3
       GOTO 520
  422  CB(I) = 10.*ALOG10(10.**(CC1/10.) + 10.**(CC2/10.))
       TB(I) = Y
       Y = Y + DY
  423  CONTINUE
C
C      INTERPOLATE BETWEEN CONTOURS TO GET THE Y-VALUES FOR THE MERGED CONTOURS.
C
C      MYA, MYB ARE TWO POSSIBLE Y-VALUES FOR EACH CONTOUR BETWEEN
C
  430  DO 442  I=1,NCM
       MYA(I) = 0
       MYB(I) = 0
       IF (YO)  431,442,431
  431  DO 441  J=1,20
       IF (CM(I) - CB(J))  433,434,434
  433  IF (CM(I) - CB(J+1))  441,435,435
  434  IF (CM(I) - CB(J+1)) '435,435,441
  435  IF (CB(J+1) - CB(J))  437,436,437
  436 TY = (TB(J+1)+TB(J))/2.
       GOTO 438
  437  TY = (CM(I) - CB(J)) / (CB(J+1) - CB(J))  *
     C (TB(J+1) - TB(J)) + TB(J)
  438  IF (MYB(I))  440,439,440
  439  MYB(I) = TY
       GOTO 441
  440  MYA(I) = TY
  441  CONTINUE
  442 CONTINUE
C
C      MYP IS THE LIST OF Y-VALUES FOR THE CONTOURS TO THE RIGHT OF THE RUNWAYS.
C
       DO 455  I=1,NCM
```

FIGURE D-7G. PROGRAM LISTING - MERGENEF ROUTINE

```
      MYP(I) = 0
      J = 2
      IF (NP -1)  455,455,451
  451 IF (CM(I) - CP(1))  458,452,452
  452 DO 453  J = 1,NP
      IF (CM(I) - CP(J))  454,453,453
  453 CONTINUE
      GOTO  455
  454 IF (J - NP)  459,458,458
  459 TT = TP(J-1)
      IF (TT)  457,456,457
  456 TT = TP(J)/ABS(TP(J))
  457 MYP(I) = 10.**((CM(I) - CP(J-1))/(CP(J) - CP(J-1))*
     C ALOG10(ABS(TP(J)/TT)))*TT + YP
      GOTO 455
  458 MYP(I) = (CM(I) - CP(J-1))/(CP(J) - CP(J-1))*
     C (TP(J) - TP(J-1)) + TP(J-1)+ YP
  455 CONTINUE
C
C     MYM IS THE LIST OF Y-VALUES FOR THE CONTOURS TO THE LEFT OF THE RUNWAYS.
C
      DO 465  I = 1,NCM
      MYM(I) = 0
      J = 2
      IF (NM -1)  465,465,461
  461 IF (CM(I) - CN(1))  468,462,462
  462 DO 463  J = 1,NM
      IF (CM(I) - CN(J))  464,463,463
  463 CONTINUE
      GOTO  465
  464 IF (J-NM)  [illegible],468,[illegible]
  469 TT = TM(J-1)
      IF (TT)  467,466,467
  466 TT = TM(J)/ABS(TM(J))
  467 MYM(I) = 10.**((CM(I) - CN(J-1))/(CN(J) - CN(J-1))*
     C ALOG10(ABS(TM(J)/TT)))*TT + YM
      GOTO 465
  468 MYM(I) = (CM(I) - CN(J-1))/(CN(J) - CN(J-1))*
     C (TM(J) - TM(J-1)) + TM(J-1)+ YM
  465 CONTINUE
C
C     WRITE THE OUTPUT
C
      LINES = LINES + 6
C
C     54 LINES ARE PUT ON EACH PAGE OF OUTPUT
C
      IF (LINES - 48)  212,212,211
  211 LINES = 0
      WRITE (3,3021)
      WRITE (3,3020)
      WRITE (3,3022) (CM(I), I=1,NCM)
      WRITE (3,3020)
C
C     THE CONTOUR VALUES FOR THE MERGER ARE WRITTEN AT THE TOP
```

FIGURE D-7H. PROGRAM LISTING - MERGENEF ROUTINE

```
C       THE X-VALUE FOR THE MERGED CONTOUR Y-VALUES IS WRITTEN
C       THE Y-VALUES ARE WRITTEN
C
C
  212   WRITE (3,3023)  X
        WRITE (3,3024)  (MYP(I), I=1,NCM)
        WRITE (3,3025)
        DO 221  I=1,NCM
        K = 1
        M = MYA(I)
        IF (M)  240,230,240
  221   CONTINUE
       WRITE (3,3028)
        DO 222  I=1,NCM
        K =  2
        M = MYB(I)
        IF (M)  240,230,240
  222   CONTINUE
        WRITE (3,3029)  (MYM(I), I=1,NCM)
        WRITE (3,3020)
C
C       THE X-VALUE IS INCREMENTED
C       AND END CHECKS ARE PERFORMED
C
  399 X = X + DXM
      IF (X - XEND)  201,201,300
  201 XP = DX1*(NX1-1) + XBEG1
      IF(X-XP) 202,300,300
  202 XP = DX2*(NX2 - 1) + X0 + XBEG2
      IF ( X - XP) 203,300,300
  203 RX = X
      GOTO 20
C
C     IF THIS MERGED CONTOUR IS DONE, A NEW SET OF DATA IS READ IN.
C
  205   STOP
  230   GOTO (231,232,233,234,235,236,237),I
  231   WRITE (3,3031)
        GOTO 238
  232   WRITE (3,3032)
        GOTO 238
  233   WRITE (3,3033)
        GOTO 238
  234   WRITE (3,3034)
        GOTO 238
  235   WRITE (3,3035)
        GOTO 238
  236   WRITE (3,3036)
        GOTO 238
  237   WRITE (3,3037)
  238   GOTO (221,222),K
  240   GOTO (241,242,243,244,245,246,247),I
  241   WRITE (3,3041)  M
      GOTO 248
  242   WRITE (3,3042) M
```

FIGURE D-71. PROGRAM LISTING - MERGENEF ROUTINE

```
      GOTO 248
 243  WRITE (3,3043) M
      GOTO 248
 244  WRITE (3,3044) M
      GOTO 248
 245  WRITE (3,3045)  M
      GOTO 248
 246  WRITE (3,3046) M
      GOTO 248
 247  WRITE (3,3047)  M
 248  GOTO (221,222),K
C
C
C      TO GET THE CONTOUR VALUE FOR THE FIRST RUNWAY AT A GIVEN Y-VALUE
C
 510  CC1 = 0.
      TY = Y
      IF (TY)  511,512,512
 511  TY = -TY
 512  IF (T1(1))  530,530,513
 513  J = 1
      IF (TY - T1(1))  514,514,517
 514  NJ = NC1 - 1
      DO 516  J = 1,NJ
      IF (TY   T1(J))  515,515,516
 515  IF (TY - T1(J+1))  516,517,517
 516  CONTINUE
      CC1 = (TY/T1(NC1))*(C1(NC1) - CX1) + CX1
      GOTO 530
 517 IF(TY) 1517,1516,1517
1516 TY = 1.
1517  IF (T1(J+1))  518,518,519
 518 CC1 = ALOG(T1(J)/TY) / ALOG(T1(J))*(CX1 - C1(J)) + C1(J)
      GOTO 530
 519 CC1 =  ALOG(T1(J)/TY)/ALOG(T1(J)/T1(J+1))*(C1(J+1)-C1(J))+C1(J)
 530 GOTO(406,413,421,368,1406,1436),K
C
C      TO GET THE CONTOUR VALUE FOR THE SECOND RUNWAY AT A GIVEN Y
C
 520  CC2 = 0.
      TY = Y - Y0
      IF (TY)  521,522,522
 521  TY = -TY
 522  IF (T2(1))  540,540,523
 523  J  = 1
      IF (TY - T2(1))  524,524,527
 524  NJ = NC2 - 1
      DO 526  J=1,NJ
      IF (TY - T2(J))  525,525,526
 525  IF (TY - T2(J+1))  526,527,527
 526  CONTINUE
     CC2 = (TY/T2(NC2))*(C2(NC2) - CX2) + CX2
      GOTO 540
 527 IF(TY) 1527,1526,1527
1526 TY = 1.
```

FIGURE D-7J. PROGRAM LISTING - MERGENEF ROUTINE

```
1527  IF (T2(J+1))  528,528,529
 528 CC2 = ALOG(T2(J)/TY) / ALOG(T2(J))*(CX2 - C2(J)) + C2(J)
      GOTO 540
 529 CC2 =  ALOG(T2(J)/TY)/ALOG(T2(J)/T2(J+1))*(C2(J+1)-C2(J))+C2(J)
 540 GOTO (403,416,422,358,1416,1426),K
 600 NC = 0
     NM1 = NP - 1
     DO 607 I = 1,NM1
     IP1 = I + 1
     DO 607 J=IP1,NP
     IF(CP(J) - CP(I)) 601,602,608
 601 CT = CP(I)
     CP(I) = CP(J)
     CP(J) = CT
     CT = TP(I)
     TP(I) = TP(J)
     TP(J) = CT
     GOTO 607
 602 IF(ABS(TP(I)) - ABS(TP(J))) 603,604,606
 603 CP(I) = CP(I) + 0.01
     CP(J) = CP(J) - 0.01
     GOTO 601
 604 DO 605 K=J,NM1
     CP(K) = CP(K+1)
 605 TP(K) = TP(K+1)
     NC = NC + 1
     GOTO 607
 606 CP(I) = CP(I) - 0.01
     CP(J) = CP(J) + 0.01
     GOTO 607
 608 IF(ABS(TP(I)) - ABS(TP(J))) 607,609,607
 609 TP(I) = TP(I) + 1.
     TP(J) = TP(J) - 1.
 607 CONTINUE
     NP = NP - NC
     GOTO (1411,1461), NO
 650 NC = 0
     NM1 = NM - 1
     DO 657 I=1,NM1
     IP1 = I + 1
     DO 657 J=IP1,NM
     IF(CN(J) - CN(I)) 651,652,658
 651 CT = CN(I)
     CN(I) = CN(J)
     CN(J) = CT
     CT = TM(I)
     TM(I) = TM(J)
     TM(J) = CT
     GOTO 657
 652 IF(ABS(TM(I)) - ABS(TM(J))) 653,654,656
 653 CN(I) = CN(I) + 0.01
     CN(J) = CN(J) - 0.01
     GOTO 651
 654 DO 655 K=J,NM1
     CN(K) = CN(K+1)
```

FIGURE D-7K. PROGRAM LISTING - MERGENEF ROUTINE

```
  655 TM(K) = TM(K+1)
      NC = NC + 1
      GOTO 657
  656 CN(I) = CN(I) - 0.01
      CN(J) = CN(J) + 0.01
      GOTO 657
  658 IF(ABS(TM(I)) - ABS(TM(J))) 657,659,657
  659 TM(I) = TM(I) + 1.
      TM(J) = TM(J) - 1.
  657 CONTINUE
      NM = NM - NC
      GOTO (1451,1471) , NO
C
 1000  FORMAT (2I7, 10F4.0)
 1001  FORMAT (3I7)
 1002  FORMAT (11I7)
 1003  FORMAT (20A4)
 3000  FORMAT ( 10I7)
 3001  FORMAT (' X VALUE = ', 1F8.0)
 3002  FORMAT (' TS', 10F7.1)
 3003  FORMAT (' CS', 10F7.1)
 3004  FORMAT (' NS', 4I8)
 3005  FORMAT (' ', 5I7)
 3006  FORMAT (' ', I7)
 3010  FORMAT ('1')
 3011  FORMAT (' ', 20A4)
 3012  FORMAT (' ', 4X, 'START= ', I7, 4X, 'INTERVAL= ', I7,
     C  4X, 'CONTOURS  ', 10F4.0)
 3013  FORMAT (' ', 4X, 'STOP = ', I7, 4X,
     C  'X,Y OFFSET OF 2ND RUNWAY= ', I7, ', ', I7)
 3020  FORMAT (' ')
 3021  FORMAT ('1', 16X,
     C  'DISTANCE TO MERGED CONTOUR (IN FEET) FROM FIRST RUNWAY')
 3022  FORMAT (' ', 'CONTOUR VALUES  ', 10F7.0)
 3023  FORMAT (' ', 'X VALUE= ', 1F7.0)
 3024  FORMAT (' ', 6X, 'TO RIGHT  ', 10I7)
 3025  FORMAT (' ', 6X, 'BETWEEN', 3X)
 3028  FORMAT (' ', 6X, 'RUNWAYS', 3X)
 3029  FORMAT (' ', 6X, 'TO LEFT   ', 10I7)
 3031  FORMAT ('+', 23X)
 3032  FORMAT ('+', 30X)
 3033  FORMAT ('+', 37X)
 3034  FORMAT ('+', 44X)
 3035  FORMAT ('+', 51X)
 3036  FORMAT ('+', 58X)
 3037  FORMAT ('+', 65X)
 3041  FORMAT ('+', 16X, I7)
 3042  FORMAT ('+', 23X, I7)
 3043  FORMAT ('+', 30X, I7)
 3044  FORMAT ('+', 37X, I7)
 3045  FORMAT ('+', 44X, I7)
 3046  FORMAT ('+', 51X, I7)
 3047  FORMAT ('+', 58X, I7)
 3048 FORMAT(' X VALUE= ', 1F8.0,'**** CLOSURE OF ',F7.0,
     1'  CONTOUR NEEDED.')
      END
```

FIGURE D-7L. PROGRAM LISTING - MERGENEF ROUTINE