AIRPORT-NOISE LEVELS AND ANNOYANCE MODEL
SYSTEM'S REFERENCE MANUAL

Richard DeLoach
James Donaldson
Margaret Johnson

January 1986
CONTENTS

1.0 INTRODUCTION .................................................................1-1
2.0 SYSTEM DESCRIPTION .......................................................2-1
3.0 MODULE DESCRIPTION .......................................................3-1
   3.1 INTEGRATED NOISE MODEL ............................................3-2
      3.1.1 Input ..........................................................3-2
      3.1.2 Processing ...................................................3-5
      3.1.3 Output ......................................................3-6
      3.1.4 Utilities ....................................................3-7
   3.2 CONTOUR PROCESSOR ....................................................3-8
   3.3 SITE II DEMOGRAPHIC MODULES .......................................3-12
      3.3.1 Phase 1 ......................................................3-12
      3.3.2 Phase 2 ......................................................3-16
   3.4 REPORT GENERATOR ....................................................3-18
4.0 SYSTEM USAGE ...............................................................4-1
   4.1 SITE II DATABASE RETRIEVAL .........................................4-2
   4.2 ALAMO JOB FLOW .....................................................4-9
      4.2.1 Integrated Noise Model .....................................4-9
      4.2.2 Contour Processor ..........................................4-13
      4.2.3 SITE II Demographic Processor ..............................4-13
      4.2.4 Report Generator ...........................................4-14
5.0 APPENDICES .................................................................A-1
   5.1 APPENDIX A - SOURCE DOCUMENTATION .............................A-1
      Program AUPHS1 ..................................................A-2
Subroutine CHKREC ............................................. A-6
Subroutine RDCONTR ............................................. A-8
Subroutine INIT .................................................. A-9
Subroutine SITE2 .................................................. A-10
Subroutine NEWCRV .............................................. A-11
Subroutine NTHSTH ............................................... A-13
Subroutine TRNSLT ............................................... A-15
Function AREA ..................................................... A-16
Subroutine OCTANT ............................................... A-17
Subroutine ENRICH ............................................... A-18
Function INTABLE ............................................... A-20
Subroutine ADDPTS ............................................... A-21
Subroutine OCTTAB ............................................... A-23
Function IOCTNT ............................................... A-25
Subroutine POLY .................................................. A-26
Subroutine DSTNCE ............................................... A-30
Subroutine INTSCT ............................................... A-32
Subroutine PLOT .................................................. A-34
Subroutine ROSE .................................................. A-35
Subroutine LABELS ............................................... A-37
Subroutine BOUND ............................................... A-39
Subroutine PLTPOL ............................................... A-40
Subroutine BLDARR ............................................... A-41
Subroutine RESTPLT ............................................... A-44
Program AUREPT ............................................. A-46
Subroutine RDREPT ........................................... A-49
Subroutine BLDTAB ............................................. A-52
Subroutine HEADER ............................................. A-58
Subroutine CHART .............................................. A-59
Subroutine CHART1 ............................................. A-60
Subroutine BREAK .............................................. A-62
Subroutine SUMMARY ........................................... A-63
Function INTERP .............................................. A-66
Subroutine LASTPAG ........................................... A-68
Function PERCNT .............................................. A-69

5.2 APPENDIX B - SITE II DATABASE INSTALLATION .............. B-1

6.0 FIGURES ......................................................
   6.1 FIGURE 1 Baseline System Tab Flow .................. 2-2
   6.2 FIGURE 2 Integrated Noise Model Flow ................. 3-3
   6.3 FIGURE 3 Contour Processor Flow ...................... 3-9
   6.4 FIGURE 4 SITE II - Phase 1 Processor Flow .......... 3-13
   6.5 FIGURE 5 SITE II - Phase 2 Processor Flow .......... 3-14
   6.6 FIGURE 6 Report Generator Flow ...................... 3-19
   6.7 FIGURE 7 Submit File for SITE II Retrieval .......... 4-3
   6.8 FIGURE 8 SITE II Procedure File SETUP ............... 4-4
   6.9 FIGURE 9 SITE II Procedure File T .................... 4-6
   6.10 FIGURE 10 SITE II Procedure File BT ................. 4-7
   6.11 FIGURE 11 Submit File for Baseline Execution ...... 4-10
   6.12 FIGURE 12 Baseline Procedure File AI ............... 4-11
   6.13 FIGURE 13 INM and Contour Processor Execution ..... 4-12
   6.14 FIGURE 14 Report Generator Execution ............... 4-15
   6.15 FIGURE 15 Baseline Procedure File Bl ............... 4-16
   6.16 FIGURE 16 Step 1 job for the SITE II database conversion . B-3

iii
1.0 INTRODUCTION

The purpose of this document is to describe the Airport-noise Levels and Annoyance MOdel (ALAMO) in terms of a detailed description of the constituent modules, to describe the execution of ALAMO procedure files, necessary for system execution, and to present the source code documentation associated with code development at Langley Research Center. The modules constituting ALAMO are presented both in flow graph form, and through a description of the subroutines and functions that comprise them.
2.0 SYSTEM DESCRIPTION

The Airport-noise Levels and Annoyance Model (ALAMO) is a system of five processor modules which when executed in sequence evaluate the effect of aircraft noise on a surrounding airport community. The five processors contained in the baseline ALAMO system are: the Integrated Noise Model, the Contour Processor, the SITE II Phase 1 Processor, the SITE II Phase 2 Processor and the Report Generator. These processors use, in addition to the INM and SITE II databases, ten user named files to produce noise exposure and impact reports, as well as noise impacted population distribution plots overlaid on an octant line compass rose. These files are named as follows in the ALAMO submit file: XXX is the Airport Auxiliary file and is input to the Contour Processor and the Report Generator; XXX001 is Airport Description Data and is input to the Integrated Noise Model; XXX002 contains airport statistics generated from the Integrated Noise Model; XXX003 is noise contour data generated from the Integrated Noise Model to serve as input to the Contour Processor; XXX004 is the SITE II Phase 1 input file generated from the Contour Processor; XXX005 is the plot vector file generated in the Contour Processor; XXX006 is used as a temporary accumulator for the output generated from both the SITE II Phase 1 and Phase 2 processors; XXX007 contains the complete output from both Phase 1 and Phase 2, and serves as input to the Report Generator; XXX008 contains the Noise Impact Summary Report tabulated by the Report Generator; XXXR contains the Noise Impact Summary Report associated with a countermeasure case executed subsequent to the baseline case.
Figure 1.-Baseline System Job Flow.
The logical unit numbers assigned by a particular processor is illustrated in figure 1. The integrated Noise Model (INM) Processor reads as input the INM database and Airport Description Data. The INM database contains standard information on aircraft noise and performance for commercial, general aviation, and military aircraft. The Airport Description Data (named file XXX001) contains runway, track, approach profile and traffic mix data for any particular airport. The INM Processor reads these files, generates an airport statistics file (named file XXX002), and computes noise contour data for noise levels 65 to 85 decibels. The noise contour data file (named file XXX003) is then input to the Contour Processor.

The Contour Processor reads, in addition to the contour data file, the Airport Auxiliary File (named file XXX). This file contains the latitude, longitude, displacement and translation values, and a user designated population growth rate for a given airport. The Contour Processor then generates noise footprint contour plots and writes them to the plot vector file (named file XXX005), and computes a sequential file (named file XXX004) containing noise contour coordinates which is input to the SITE II Phase 1 Processor.
The SITE II Phase 1 Processor reads, in addition to noise contour coordinates the SITE II database. The SITE II database is contained on magnetic tape and can be accessed from the volume serial numbers ND0775, ND1062, NC0773 and NC0938, respectively. The census data for the states Alabama through Mississippi (in alphabetical order) are contained on tape number ND0775; the census data for the states Missouri through Wyoming are contained on tape number ND1062; the block group data which describes a physical area surrounding a landmark is contained on tape number NC0773; and the census tract data is contained on tape number NC0938. The zipcode data contained on tape number NC0753 may be optionally processed. The data is retrieved (see fig. 8) before SITE II processing and written to the direct access file NSF1, NSF2 and NSF3 to be input to the Phase 1 processor via the logical files TAPE21, TAPE22 and TAPE23. Tape number NC0773 or NC0938 is selected for retrieval depending on whether the user wants to process block groups or census tract data. The census data for each state is retrieved from tape numbers ND0775 and ND1062 and copied to the direct access files MNF1, MNF2 and MNF3 to be input to the Phase 2 processor via the logical files TAPE31, TAPE32 and TAPE33. The census information retrieved is demographic and housing data obtained from the 1980 census, updated to 1981; additional census data was obtained from the 1970 census. The Phase 1 Processor selects records containing population values that are within the noise level boundary computed in the Contour Processor.
The Phase 2 processor compiles the contour and demographic data, arranges it and writes it to a file (named file XXX006). The contents of the file XXX006 are transferred to the named file XXX007; this final file is input to the Report Generator.

The Report Generator reads the data file written in SITE II and the Airport Auxiliary File (named file XXX). Ten reports are generated, the first eight reports contain noise exposure correlations for each octant in the compass rose; the ninth report contains demographic noise exposure values and the tenth report (written on the named file XXX008) is the Noise Impact Summary Report.
3.0 MODULE DESCRIPTION

The baseline ALAMO system was assembled by linking four major subsystems together to comprehensively assess the noise impact on various airport communities. The system is a combination of software developed by outside interest, and software developed here at Langley Research Center (LaRC) to augment ALAMO for use on the Network Operation System (NOS). The Integrated Noise Model (INM), which computes noise exposure values, was developed by the Federal Aviation Administration; the Contour Plot Processor, which plots the contour noise footprints and generates contour descriptions in SITE II format, was developed at LaRC by Computer Sciences Corporation (CSC); the SITE II demographic program, which provides access to the United States Census database, was developed by CACI, Inc.; and the Report Program, which generates noise impact statistics was developed at LaRC by CSC. The following four sections describe the logic flow of each subsystem program illustrated by a tree diagram of source modules.
3.1 INTEGRATED NOISE MODEL

The Integrated Noise Model program VERSUN6 is logically divided into four distinct areas of code as illustrated in figure 2, input, processing, output and general utility routines. The Airport Description Data Deck and the INM database are the input files to VERSUN6. The Airport Description Data Deck is read from logical unit 7 in subroutine READIN. Since Fortran input/output routines add the characters TAPE as a prefix to the logical unit to form the file name, TAPE7 is the file name assigned to logical unit 7. From this point on all of the logical units will be referred to by their file names.

Input

Refer to the input area of routines in figure 2. Runway and track data are read in subroutine RWYRD, track data is input beginning at entry point TRAKRD in RWYRD. Subroutine HEIG is referenced by RWYRD to compute circular segment variables. Aircraft approach parameters, altitude data, wind information and noise parameters are read in subroutines APPTRD, ALTRRD, WINDRD and NOISRD respectively. Aircraft performance profiles are read in and stored in subroutine PROFD. The aircraft takeoff performance parameters are corrected by airport altitude and ambient temperature in subroutine TPROF. User defined replacement values for the tolerances used for checking
Figure 2.-Integrated Noise Model Flow
contour looping conditions are read in subroutine TOLRD. The aircraft mix data describing the type, magnitude and arrangement of aircraft activity at an airport is read in subroutine MIXRD. Refer to the Integrated Noise Model Version 2 User's Guide (Section 2) for a detailed description of the Airport Description Data Deck. The INM database contains aircraft noise and performance data for each aircraft type. If the user decides to enter new aircraft definitions a call is made to subroutine LOAD to read the INM database. In the event that this occurs, a call is made to subroutine INFORM to inform anyone reading the output that the user has modified the aircraft data in the database or has created his/her own aircraft definitions.

A call is made to subroutine METFIX to initialize the noise metric variables. The metric identification (NEF, LDN, CNEL, LEQ, ASDS or DOSE) is read from the second record in the Airport Description Data Deck. A call is also made to subroutine CHEKTHR to compare the thrust noise settings in the database. This call is made if the user has opted to modify the database.
3.1.2 Processing

Each noise level contour is processed point by point. Initially subroutine FIRST is called from VERSUN6 to take the first guess at the new contour point position. A call is made to subroutine GRADIE which computes the new point by determining the gradient. If the first point has been computed subroutine NEWPNT is called to compute the next contour point. If the metrics NEF, LDN, CNEL or LEQ are used then NEWPNT is referenced, if the metrics ASDS or DOSE are used then a call is made to subroutine NEWASDS to compute the next contour point. After the position of each point is computed a call is made to subroutine EXPOSE to calculate the noise exposure value at that point. The first call in EXPOSE is made to subroutine SIFT to determine which flights create a significant contribution to the overall noise exposure level. Noise exposure correction values for track geometry are computed in subroutine HBT. The shortest distance from a contour point to a ground track segment are computed to determine the correction values. The distance to curved and straight segments are computed in subroutines CURVE and STRAIGHT respectively. The corrections are then computed for curved segments in the function CONTF and for straight segments in the function CONTG. For comparison purposes a call is made to subroutine ACNOIS to compute the noise exposure level for one single flight from any combination of metrics. To determine the noise exposure levels
for the ASDS and DOSE metrics calls are made to subroutines ASD2 and ASD3 respectively. For the ASDS metric a call is made to the time history routine TIHISI to determine the times that the selected threshold noise levels are exceeded by one flight. Subroutine OVERLAY is referenced to overlay the flight performance profile over the ground track segments. The aircraft position coordinates, the ground attenuation and the noise level at a point is computed in subroutines PSCOO, EGA and AL respectively.

3.1.3 Output

The output area of the code writes two files for output, the Airport Statistics file on TAPE6 and the Noise Contour Data file on TAPE2. Subroutine HEADER prints the column headers for the contour information. The formats for PLINE, which is an entry point in HEADER, are also set up. The actual contour points and values are printed a line at a time at the entry point PLINE. The Noise Contour Data file is written to TAPE2 (named file XXX003) in subroutine TPGRM. Subroutine INSIDE is called to determine if point is within the defined contour or not, if not, that point is not written to the output file. Subroutine SKFIL is called to skip pass the end of file mark to start a new file. The entry points SKPFF
3.1.4 Utilities

The system utilities are subroutines and function subprograms that are used to perform routine vector, sort and arithmetic operations. Vector operations such as addition, subtraction, dot product, vector to vector transfer, unit vector computations, vector magnitude computations and vector by scalar multiplication are performed by the functions VADD, VSUB, VDOT, VTRN, VUNT, VMAG and VSCL respectively. The subroutine ZERO is used to clear vectors. The sort utility SORT2 sorts three arrays of numerical data in ascending order. The entry point SORT3 in SORT2 sorts two arrays in ascending order. The subroutine SORT4 sorts numerical data from four arrays or tables in descending order. SORT4 is referenced from subroutine MIXRD to assemble the flight identification table and the associated equivalent operations for the noise metrics NEF, LDN, CNEL or LEQ. The degrees to radians conversion is performed by subroutine DGTRD; value normalization and sign determination are performed by the routines SGNR and SGNI respectively.
3.2 CONTOUR PROCESSOR

The contour processor program AUPHS1 converts the Noise Contour Data file (named file XXX003) generated in the INM processor, from x-y coordinate points to a set of polygons for each noise level. These polygons are in a format compatible for input to the SITE II Phase 1 processor (see fig. 1). The subroutine INIT is called to read the Airport Auxiliary File (named file XXX) from TAPE2. This file contains the latitude, longitude, and optional north-south and east-west distances from the airport latitude and longitude. Axis translation data, axis length information and the airport title are also contained in the Airport Auxiliary File. Subroutine SYSTEM is called to issue error messages if the file doesn't process properly.

The Noise Contour Data file on TAPE5 is read and evaluated to determine the number of records between the metric card and the x-y coordinate points. The header data and the x-y points for the noise contour are read from TAPE5 in subroutine RDCONTR and the axis translation is performed in subroutine TRNSLT (see fig. 3). The point reduction software is driven by subroutine SITE2. There is a 150 point limit for each contour. Subroutine NEWCRV is called to delete all unnecessary points below a given maximum on a noise contour. The area of the new contour is then computed to make sure that there is no more than a one percent difference in area between the old contour and the new reduced point contour. The plot vector file is then initialized in subroutine PLOT if this is the first.
Figure 3.-Contour Processor Flow
contour processed. The entry point GETCON in the COMPASS routine JOBCON is referenced to sample job control register three. This is done to determine if this is a baseline case run. Plot information for the compass rose, plot boundaries and labels are written to the plot vector file in subroutine ROSE.

The SITE II input file contains four types of card images containing either the characters 'A', 'B', 'C', or 'D'. These characters indicate the following:

'A' Run Identification
'B' Site Identification
'C' Case Identification
'D' Case Identification Continuation

The reader is referred to the SITE II User's Manual, section IV for a detailed description of these card images. After the plot vector file is generated SITE II input file is written to TAPE4. The run number is written on card type 'A' and the airport name along with the airport latitude, longitude, the north-south displacement value and the east-west displacement value. The subroutine OCTANT is called to coordinate the polygon definitions. Subroutine OCTTAB is called from OCTANT to build a table of start and end points for each line segment in each octant. Subroutine IOCTNT is called from OCTTAB to determine which octant an x-y point lies in. After the start/end table is built a call is made to subroutine ENRICH if two
are made to the functions INTABLE and IOCTNT to determine if the ordinal is in the start/end table and which octant the x-y point lies in, respectively. If two adjacent points that are not in the same octant are not in adjacent octants subroutine ADDPTS is called to add points. Subroutine POLY is then called from OCTANT to build polygons from the line segment table. Subroutine DSTNCE is called from POLY to find the distance from start/end points to the origin. Subroutine INTSCT is referenced from DSTNCE to compute the line equation for each segment, determine the octant of each point to find the octant line they straddle and to find the point of intersection on the octant line. Upon return to POLY 'C' type cards are written to TAPE4 containing the metric name, a table of LDN metrics and the area of the polygon in square miles. Subroutine NTHSTH is then called from POLY to convert x-y points from feet to statue miles and to write 'D' type cards containing there converted points to TAPE4. A final call is made to subroutine PLTPOL to plot the polygons given the x-y points and a scale factor. Execution control is then returned to program PHASE1. A final call is then made to BLDARR to extract information from the scratch unit TAPE4 and build the SITE II input file on TAPE1 (named file XXX004). If any deck contains more than 150 points then that deck is subdivided in this routine. Since each 'A' type card indicates the beginning of a new SITE II input deck an end-of-file mark is written to separate decks and the run counter is bumped.
3.3 SITE II DEMOGRAPHIC MODULES

There are two SITE II processor modules, the phase 1 program MAIN1 (see fig. 4) and the phase 2 program MAIN2 (see fig. 5). The phase 1 module reads the SITE II input file (named file XXX004) and generates an intermediate file of selected SITE II records for sorting. The phase 2 module reads the sorted file and generates the input file (named file XXX006). The information on XXX006 is aggregated on the named file XXX007, which serves as input to the Report Generator Module.

Phase 1

The phase 1 program MAIN1 initially calls subroutine INITIAL to set default values for data item counts, maximum number of polygon points per run, population estimates, logical unit assignments, record length and block sizes for file structure. A call is then made to subroutine INPUT to read the SITE II input file (named file XXX004) and insure that there is a valid sequence of card types. Calls are initially made from INPUT to the subroutines WTCTRL and CDCBLK, to open the selected record file on TAPE1 and to encode hollerith blanks to blanks configured in "A" format, respectively.
Figure 4.-SITE II Phase 1 Processor flow
Figure 5.-SITE II Phase 2 Processor Flow
The SITE II input file is then read a record at a time and the first character on the record is checked for the card type. A call is then made to subroutine PROCES to define the structure for each card type. Calls are then made from PROCES to subroutines PREFOR and EDIT, to convert free form strings to fixed format card images and to edit each record to insure that each field has valid characters, respectively. Subroutine VERIFY is called to check specific data items in the SITE II input file. A call is then made to subroutine PUTDAT to create a sort key for the selected record file (TAPE1). Execution control is then passed back to subroutine INPUT to read another input record.

After the SITE II input has been read a call is made to subroutine GETZIP from MAIN1. This subroutine takes in array of requested zip codes and searches the zip file. For each zip found tract inclusion records are written to the selected record file on TAPE1. Subroutine RDZIP is called from GETZIP to open, read, rewind or close the zip file on TAPE1. A call is then made to subroutine LIMITS to find the latitude and longitude limits for each site and each run. Subroutine CREBND is referenced from LIMITS to convert sectors, chords and polygons to a form useable by subroutine SEARCH. Subroutine SEARCH reads the north/south records to determine the cases for which the record is a hit. For each hit a control record is written to TAPE1. Calls are made to subroutine NSPOSN, RDNTOS and NTCTRL, to get the first record below a given latitude, to get the next record and to write to the control file, respectively.
Before the phase 2 processor is invoked, the track inclusion control file written by the phase 2 processor on TAPE1 is sorted using the CDC utility SORTMRG. The sorted control file is written on TAPE2 for input to the phase 2 processor. Subroutine INITAL is referenced to initialize operational variables the same as the phase 1 processor. Subroutine ACCUM1 is then called to accumulate the area data for each case. Subroutine RDCTRL is called from ACCUM1 to read the control file which contains area information for each site. Subroutine MNFIND is called to read the main file. The file header is read to check for the state and to determine the number of records for that state. Upon return, ACCUM1 adds the percentage of area data to the total and writes the component area to TAPE3.

There are three levels of data on the SITE control file; level 1 is RUN data which is the population summation of all the areas formed around the sites under study; level 2 is SITE data, this is the reference point around which one or more boundaries are defined; and level 3 is CASE data which represents a physical area around the site. Subroutine OUTA controls the production of growth, profile and component area report data by referencing subroutine GETDAT to read the specified level of data from TAPE2. A call is then made to subroutine CNTROL to resolve specific requests for output. Calls
made to COMPON to generate values for all the reports, accumulate
data for SITE and RUN totals, to write summary report data in
condensed form, to produce profile report data and to write com-
ponent area report data, respectively to TAPE6 for input to the
Report Generator. Upon return to MAIN2 a final call is made to
subroutine OUTB to control the production of columnar and comparison
reports data through calls to subroutine COLUMN and COMPAR. This
data is also written to TAPE6 for input to the Report Generator.
3.4 REPORT GENERATOR

The SITE II report file (named file XXX007) generated following the SITE II phase 2 processor is input to the Report Generator on TAPE5. The Airport Auxiliary File (named file XXX) is also input on TAPE1 to the Report Generator program AUREPT (see fig. 6) to obtain the year of population, the population growth rate and the airport name and address. The SITE II case title is then read from the first record on the SITE II report file. The first call is made to subroutine BLDTAB to build a 3-dimensional table of sixteen demographic variables for each octant of each LDN band. The starting LDN index is defined by calling the entry point GETCON (in subroutine JOBCON) to fetch the job control register R3. A call is made to subroutine RDREPT to read the following demographic information from the SITE II report file:

- Male and female age groups
- Family income, home values and occupations
- The number of single dwellings households
- Households with T.V.'s and air-conditioners

This information is stored in the labelled common block /COMM/ and a return is made back to subroutine BLDTAB. Beginning at the LDN value determined from job control register R3, the 3-dimensional table is generated, stored in the array TABLE and passed back to REPORTS.
Figure 6.-Report Generator Flow
to print the heading for each octant. The TABLE array is checked for a void octant and a call is made to subroutine CHART. Subroutine CHART prints sixteen demographic variables by LDN band values and calls subroutine BREAK to transfer seven columns out of the 3-dimensional table to a vector array called LINE. An alternate call is made to subroutine CHART1 for special printing if an LDN contour is missing. A call is then made to subroutine SUMARY to print the Noise Impact Summary Report. The functions INTERP and PERCNT are referenced by SUMARY to compute the equivalent noise level value and the weighting factor for a given LDN value. Finally subroutine LASTPAG is referenced to write a separate Noise Impact Summary to TAPE2 (named file XXX008).
4.0 SYSTEM USAGE

The baseline ALAMO system can be used by submitting two submit files, the SITE II file retrieval file SITEJOB and the baseline ALAMO execution submit file BASJOB. These submit files along with the procedure files they invoke will be described in the following sections to give an overview of the logic flow of the system.
4.1 SITE II DATABASE RETRIEVAL

The SITE II database resides on five magnetic tapes. These tapes are organized in the following manner:

<table>
<thead>
<tr>
<th>Volume Serial No.</th>
<th>Tape Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ND0775</td>
<td>State Code less than or equal to 28</td>
</tr>
<tr>
<td>ND1062</td>
<td>State Code greater than 28</td>
</tr>
<tr>
<td>NC0773</td>
<td>Physical area surrounding a landmark</td>
</tr>
<tr>
<td>NC0938</td>
<td>Census Tract Data</td>
</tr>
<tr>
<td>NC0753</td>
<td>Zip codes</td>
</tr>
</tbody>
</table>

The submit file illustrated in figure 7 is used to retrieve the SITE II database. The procedure file SETUP which resides on the permanent file SET is invoked by BEGIN statement. The state integer codes 17, 21 and 29, the state character codes IL, KY and MO along with the AREAS and ZIPS parameters are passed to the procedure file SETUP. In this example AREAS=1 indicates that the tape containing physical areas will be retrieved instead of the census tract data, ZIPS=0 indicates that the zip codes tape will not be retrieved. The procedure file SETUP is illustrated in figure 8. The parameters passed to SETUP through the PROC statement are as follows:

\[
\begin{align*}
A &= 17 & E &= \text{KY} \\
B &= 21 & F &= \text{MO} \\
C &= 29 & G &= 1 \\
D &= \text{IL} & H &= 0
\end{align*}
\]
SUBFIL, T100, CM180000.
USER,........
CHARGE,.....,LRC.
GET,SET.
BEGIN, SETUP, SET, 17, 21, 29, IL, KY, MO, 1, 0.
EXIT.

EDITING NOTES:
- REPLACE INTEGER CODES 17, 21, AND 29 WITH THE APPROPRIATE INTEGER CODES FOR THE STATES OF INTEREST
- REPLACE CHARACTER CODES IL, KY, AND MO WITH THE APPROPRIATE CHARACTER CODES FOR THE STATES OF INTEREST

Figure 7.-Submit File for SITE II Retrieval
PROC, SETUP, A, B, C, D, E, F, G, H.
BEGIN, T, SET, A, A, MD0775, MD1062, D, MNF1.
BEGIN, T, SET, A, B, MD0775, MD1062, E, MNF2.
BEGIN, T, SET, B, C, MD0775, MD1062, F, MNF3.
RETURN, TAPE1, MNF1, MNF2, MNF3.
IFE, (G .NE. 0), LABELA.
VSN, TAPE2 = NC0773.
BEGIN, BT, SET, BLKGRP, D, NSF1.
BEGIN, BT, SET, BLKGRP, E, NSF2.
BEGIN, BT, SET, BLKGRP, F, NSF3.
SKIP (LABEL B)
ENDIF, LABELA.
VSN, TAPE2 = NC0773.
BEGIN, BT, SET, TRACKS, D, NSF1.
BEGIN, BT, SET, TRACKS, E, NSF2.
BEGIN, BT, SET, TRACKS, F, NSF3.
ENDIF, LABELB.
RETURN, TAPE2.
IFE, (H .EQ. 1), LABELC.
VSN, TAPE3 = NC0753.
LABEL, TAPE3, PO=R, SI=ZIPCDS, FI=ZIPALL.
COPYBF, TAPE3, ZIPP1.
REPLACE, ZIPP1.
ENDIF, LABELC.
REVERT. MYCOL SETUP

Figure 8.-SITE II Procedure File SETUP
Since at least three states from the database must be retrieved for an ALAMO run, the procedure file T is invoked three times as indicated by the three BEGIN,T,... statements. In the first statement the integer code 09 is passed twice through the A and B parameters and the state character code CT is passed through the D parameter. Since all of the database tapes are multi-file files containing a file per state, the D parameter is the file identifier (FI) parameters on the LABEL card. The procedure file T illustrated in figure 9 is invoked by the BEGIN statement and the A and B parameters are checked to determine which tape to load. If the state integer code is less than or equal to 28 then the tape with the volume serial number ND0775 is retrieved, if the integer code is greater than 28 then the tape numbered ND1062 is retrieved. The file containing data for the state of Connecticut (integer code 09, character code CT) is read from ND0775 and written to the direct access file MNF1; the file for the state of New Jersey (integer code 24, character code NJ) is read from ND0775 and written to the direct access file MNF2; and the New York file (integer code 26, character code NY) is read from ND0775 and written to the direct access file MNF3. The G parameter is checked to determine if the block group or the census track tape will be read. In this example G = 1 indicates that the block group tape (NC0773) will be retrieved. Notice that the IFE statement does not work like an IF statement. If the expression is true, the statements following the IFE are executed until an ELSE or ENDF with a matching label is encountered. The procedure file BT (see fig. 10) is invoked three times to write block group data from the states, Connecticut, New Jersey and New York to the direct access files NSF1, NSF2 and NSF3.
.PROC T, A, B, T1, T2, C, FF.
PURGE, FF/NA, ST-LPF.
DEFINE, FF.
IFE, (A.LE.28. AND. B.LE.28), LABELA.
LABEL, TAPE1, PO-R, VSN-T1, SI=MN1, FI=C.
SKIP, LABELD.
ENDIF, LABELA.
IFE, (A.GT.28. AND. B.GT.28), LABELB.
LABEL, TAPE1, PO-R, VSN-T2, SI=MN2, FI=C.
SKIP, LABELD.
ENDIF, LABELB.
RETURN, TAPE1.
IFE, (B.GT.28), LABELC.
LABEL, TAPE1, PO-R, VSN-T2, SI=MN2, FI=C.
SKIP, LABELD.
ENDIF, LABELC.
LABEL, TAPE1, PO-R, VSN-T1, SI=MN1, FI=C.
ENDIF, LABELD.
COPYFF, TAPE1, FF.
RETURN, FF.
REVERT, MYCCL T

Figure 9. - SITE II Procedure File T
PROC BT, SID, X, MN1.
PURGE, MN1/NA, ST=LPF.
DEFINE, MN1.
LABEL, TAPE2, SI=SID, FI=X, PO=R.
COPYBF, TAPE2, MN1.
RETURN, MN1.
REVERT. MYCO. BT

Figure 10.-SITE II Procedure File BT
The reader should note that the direct access files MNF1, MNF2, MNF3, NSF1, NSF2 and NSF3 must be created on the WFS cluster. The H parameter is checked to determine whether the zip codes tape (NC0753) is to be retrieved. If $H = 1$ then the tape file is written to the indirect access file ZIPI. After these files have been created a baseline ALAMO case can now be executed.
4.2 ALAMO JOB FLOW

To run a baseline ALAMO case the submit file BASJOB (see fig. 11) must be submitted. Due to the large amount of execution time required the user should submit BASJOB to a CYBER 175 (R or T) machine. The job control register R3 is set to zero indicating a baseline run.

4.2.1 Integrated Noise Model

The procedure file Al is invoked using the BEGIN statement. There are 10 parameters naming the ALAMO system files. The XXX should be replaced with the F.A.A. airport identifier. The procedure file Al illustrated in figure 12 describes each subsequent procedure file invoked and the parameters used. The direct access file CONTORA containing the absolutes for the INM processor program VERSUN6 is attached. The INM data base DATABIN is also attached and copied to TAPE7 for processing. The Airport Description Data file (named file XXX001) is loaded from the permanent file A. Figure 13 illustrates the execution of the INM processor. The local file A is the Airport Description Data file input to the INM processor CONTORA on TAPE5, the parameter B is the Airport Statistics file (named file XXX002) output from CONTORA on TAPE6 and the Noise Contour Data file (named file XXX003) generated on TAPE2 is copied to the local file C for input to the Contour Processor.
BASJOB, T12345, CM27083.
USER,........
CHARGE,.......LRC.
VSN(TAPE1=YYYYY,TAPE2=ZZZZZZ)
SET(F3=0)
BEGIN,A1,A2,XXXXXX,XXXXXX,XXXXXX,XXXXXX,XXXXXX,XXXXXX,XXXXXX,XXXXXX,XXXXXX,XXXXXX,XXXXXX.
REWIND,TAPE6.
ROUTE,TAPE5,DC=LP.
EXIT.

EDITING NOTES:
- INSERT ACTUAL REEL NUMBERS IN VSN STATEMENT
- REPLACE XXX WITH F.A.A. IDENTIFIER

Figure 11.—Submit File for Baseline Execution
**FILE A2 CONTAINS THE 5 FOLLOWING PROCEDURE FILES:**

* A1 : DRIVER FOR A COMPLETE ALAMO RUN
* B1 : EXECUTES PHASE 2 OF ALAMO
* C1 : WRITES FILES PRODUCED BY ALAMO RUN TO TAPE (TAPE1 FOR SUCCESSFUL RUN, TAPE2 FOR RUN WITH ERRORS)
* D1 : Initializes TAPES USED IN ALAMO RUN
* G1 : GENERATES POPULATION GRID AND STANDARD TRACKS

**PROCEDURE PARAMETER IDENTIFICATION:**

* A : INM INPUT CASE
* B : LISTABLE INM OUTPUT
* C : INM OUTPUT - CONTOUR POINTS (PHASE 2 INPUT)
* D : PHASE 1 OUTPUT (SITE II INPUT)
* E : SAVPLT FILE - OUTPUT FROM PHASE 1
* F : SITE II OUTPUT (PARTIAL)
* G : SITE II OUTPUT (COMPLETE) - INPUT FOR PHASE 2
* H : 18TH PAGE ALAMO REPORT (BASELINE CASE)
* J : 18TH PAGE ALAMO REPORT (COUNTERMEASURE CASE)
* K : ALAMO AUXILIARY FILE (INPUT FOR PHASES 1 AND 2)

**IF R3=1, NOISE REDUCTION SIMULATION RUN IS IN PROGRESS; A MODIFIED VERSION OF INM DATABASE ALREADY EXISTS**

* ATTACH, CONTORA.
  IFE, (R3 NE. 1), LABEL1.
* ATTACH, DATABIN.
* COPYEI, DATABIN, TAPE7.
* REWIND, TAPE7.
* RETURN, DATABIN.
* ENDIF, LABEL1.
* GET, A.
* EXECUTE THE INM MODULE OF ALAMO
* CONTORA, A, B.
* SEND, B, DC=LP.
* RETURN, CONTORA.
* REWIND, TAPE2.
* COPYEI, TAPE2, C.
* EXIT.
* RETURN, TAPES, TAPE6, TAPE2, TAPEB.
* RETURN, TAPES, TAPE3, TAPE7.

*Figure 12.-Baseline Procedure File A1*
ATTACH, CONTORA.
IFE, (R3, NE. 1), LABEL1.
ATTACH, DATABIN.
COPYEI, DATABIN, TAPE7.
REWIND, TAPE7.
RETURN, DATABIN.
ENDIF, LABEL1.
GET, A.
*
* EXECUTE THE INM MODULE OF ALAMO
*
CONTORA, A, B.
SEND, B, DC+LP.
RETURN, CONTORA.
REWIND, TAPE2.
COPYEI, TAPE2, C.
SKIP(LABELA)
EXIT.
SKIP(LABEL99)
ENDIF(LABELA)
RETURN, TAPE5, TAPE6, TAPE2, TAPE9.
RETURN, TAPE9, TAPE3, TAPE7.
*
* EXECUTE PHASE 1
*
REWIND, C.
GET, ALPHA51A, K.
ALPHA51A, C, .... K.
MAP, OFF.
IFE, (R3, EQ. 8), LABEL6.
REPLACE, K.
SET(R1G+R2)

Figure 13.-INM and Contour Processor Execution
4.2.2 Contour Processor

The indirect access file AUPHSL1A containing the absolutes for the Contour Processor program PHASE1 is loaded along with the Airport Auxiliary File (named file XXX). The noise contour data (named file XXX003) is input through the C parameter to the Contour Processor AUPHSL1A, along with the Airport Auxiliary File through the K parameter. A plot vector file SAVPLT is generated to plot the noise polygons. The SITE II (named file XXX004) input file is also generated on TAPE1 and copied to the D file to serve as input to the SITE II phase 1 processor.

4.2.3 SITE II Demographic Processor

Prior to running the SITE II phase 1 processor, the direct access files containing the block group or census track portion of the SITE II database are attached as logical unit files TAPE21, TAPE22, and TAPE23. The indirect access file PH1A containing the SITE II phase 1 processor program MAIN1 is loaded. The SITE II input file is input to PH1A through the D parameter and the input reflection file (named file XXX006) is output through the F parameter after execution. The SITE II control file is generated on TAPE1 to be sorted using the CDC sort/merge utility SORTRG.
The direct access files containing the state census data portion of the SITE II database are attached as logical unit files TAPE31, TAPE32 and TAPE33. The indirect access file PH2A containing the SITE II phase 2 processor program MAIN2 is loaded and executed. The SITE II report file (named file XXX007) is output through the F parameter. The F file is then copied to the G file for input to the Report Generator.

4.2.3 Report Generator

The procedure file B1 is invoked using the BEGIN statement to process the Report Generator (see fig. 15). The listing of B1 describes the parameters used in the procedure file as illustrated in figure 16. The indirect access file AUREPTA containing the Report Generator program REPORTS is loaded, along with the Airport Auxiliary File (named file XXX). The SITE II report file is input to AUREPTA through the G parameter and the Airport Auxiliary File is input through the K parameter. After execution, the noise exposure correlation and demographic noise exposure reports are written on TAPE6 and the 10th page ALAMO report is written and saved on the permanent file J (named file XXXR). The reports on TAPE6 are printed before job control is returned to the procedure Al.
Figure 14.—Report Generator Execution
.PROC, B1, G, H, J, K.

* B1 EXECUTES PHASE 2 OF ALAMO

* B1 PROCEDURE PARAMETER IDENTIFICATION:
  * G : SITE II REPORTS (PHASE 2 INPUT)
  * H : 10TH PAGE ALAMO REPORT (COUNTERMEASURE CASE)
  * J : 10TH PAGE ALAMO REPORT (BASELINE CASE)
  * K : ALAMO AUXILIARY FILE

GET, AUTEPTA.
GET, K.
REWIND, G.
PACK, G.
MAP, OFF.
AUTEPTA, G, TAPE6, , , K, H.
REWIND, H, TAPE6.
COPYEI, TAPE6.
IFE, (R3.EQ.0), LABEL4.
REPLACE, G.
COPYEI, H, J.
REWIND, J.
SAVE, J.
REWIND, H, K, TAPE6.
PACK, K.
REPLACE, K.
ELSE, LABEL4.
APPEND, J, H.
ENDIF, LABEL4.
REVERT, MYCOL.

Figure 15.—Baseline Procedure File B1
APPENDIX A

SOURCE DOCUMENTATION

The following listings are the source code documentation for the auxiliary programs developed here at LaRC for the ALAMO System. The documentation consist of two program modules and their corresponding subroutines and function subprograms; the contour data processor program AUPHS1 and the report generator program AUREPT. Each routine description contains the objective, the method used (where applicable), the input/output parameters, the calling routines, the routines referenced, a local variable dictionary, a description of common block variables and a description of the files used.
PROGRAM AUPHS1

OBJECTIVE:

This program processes the Integrated Noise Model (INM) contour output file. The main objective is to convert the X,Y contour data into a set of polygons in an input form which is compatible with Site II, a demographics program. As the polygons are defined they are plotted using a standard X,Y pen-plotter (CALCOMP). The program also performs some secondary functions as required by the limitations of the programs that are dependent on the data it produces. A notable secondary function is the reduction of data points on the INM contours to conform to the Site II restriction of no more than 150 points per Site II execution.

METHOD:

The program reads the X,Y data representing a constant level noise metric contour and reduces the number of X,Y points to less than 150. An octant line compass rose is superimposed over the contours and polygons are defined for each octant by the intersection of the octant lines and the contour line. The polygon X,Y points are converted to an 80 character per line input deck that meets the input specifications for Site II. The X,Y polygons are also plotted via the LACGOSF graphics routines. Supporting data such as airport centroid latitude and longitude and airport name and address are read in from an external file named TAPE2.

SUBPROGRAMS USED:

BLDARR : evaluates the Site II cards for 150+ points violation and adjusts if necessary.
CALPLT : plotter pen control routine; used here to terminate the plot file.
CHKREC : looks for a message on the contour input file.
INIT : reads the supplemental data file, TAPE2 and initializes certain critical variables.
OCTANT : drives the construction of an octant ordinal table and the definition of the polygons.
PLOT : initializes the plot vector file, computes the plot scale factor and causes the compass rose to be drawn.
RDCONTH : inputs the data for 1 noise contour.
RESTPLT: routine used to plot the range of the Ldn values found.
SETCON : COMPASS ROUTINE ENTRY POINT FOR SETTING THE 3
JOB CONTROL REGISTERS.
SITE2 : DRIVES THE POINT REDUCTION PROCESS.
TRNSLT : PERFORMS X,Y POINT TRANSLATION ON X,Y CONTOUR
POINTS BASED ON DATA ON FILE TAPE2.

FUNCTIONS USED:
NONE

LOCAL VARIABLE DICTIONARY:
EWDIS : EAST/WEST DISPLACEMENT FROM LATITUDE POINT.
FACT : THE LENGTH OF THE POSITIVE X AND Y AXIS IN
INCHES.
H : X DISTANCE TRANSLATION.
ICASE : STORAGE FOR CASE NAME INFORMATION.
ISW : INTEGER VARIABLE USED AS AN INDICATOR OF RECORD
STRUCTURE OF TAPES.
ITITLE : 8 WORD ARRAY STORING THE AIRPORT TITLE.
JJ : INTEGER USED AS LOOP COUNTER.
K : Y DISTANCE TRANSLATION.
KEOF : END-OF-FILE FLAG SET TO 1 BY RDCONTR TO
INDICATE E-O-F ON CONTOUR INPUT FILE.
LAT : LATITUDE OF AIRPORT CENTROID IN CHARACTER FORM.
LDN : NOISE METRIC VALUE IN CHARACTER FORM.
LINE : CHARACTER STORAGE FOR NOISE METRICS AS THEY
ACCUMULATE.
LONG : LONGITUDE OF AIRPORT CENTROID IN CHARACTER FORM.
MET : NOISE METRIC IN CHARACTER FORM; EG: "LDN".
NCON : 18 BIT INTEGER VALUE TO SET JCR R1 VIA SETCON.
NEWPTS : THE NUMBER OF X,Y POINTS AFTER POINT REDUCTION.
NPTS : THE NUMBER OF X,Y POINTS BEFORE POINT REDUCTION.
NR3 : JOB CONTROL REGISTER, R3.
NSDIS : NORTH/SOUTH DISPLACEMENT FROM LONGITUDE POINT.
NSTCON : 18 BIT INTEGER VALUE TO SET JCR R2 VIA SETCON.
RUNCNT : COUNTER OF CONTOURS CORRESPONDING TO ONE SITE
RUN PER COUNT OF RUNCNT.
SCALE : THE PLOTTING SCALE FACTOR.
X : THE X POINTS OF THE CONTOUR BEFORE POINT REDUCT.
XNEW : THE X POINTS OF THE CONTOUR AFTER POINT REDUCT.
Y : THE Y POINTS BEFORE REDUCTION.
YNEW : THE Y POINTS AFTER REDUCTION.

COMMON BLOCK CROSS-REFERENCE LIST:
NONE
COMMON VARIABLES:

/ITITL/
ICASE1 : 4-WORD ARRAY CONTAINING CASE TITLE

/PLTINFO/
CENT : THE CENTER X AND Y VALUE OF THE COMPASS ROSE
HT : THE HEIGHT IN INCHES OF CHARACTERS TO BE NOTATED
ISTRIN : CHARACTER STRING (2 WORDS) USED TO NOTATE RANGE OF LDN VALUES
S : THE DISTANCE FROM THE LEADING EDGE OF A SYMBOL TO THE LEADING EDGE OF THE NEXT SYMBOL
W : REAL VARIABLE REPRESENTING THE WIDTH OF ONE SYMBOL

FILES USED:

INPUT:
TAPE5 - SEQUENTIAL INPUT FILE CONTAINING THE CONTOUR X, Y POINTS AND SUPPORTING DATA IN DISPLAY CODE FORMAT.
CARD1 DATE IN YY/MM/DD FORM.
CARD2 INN BANNER AND VERSION ID.
CARD3 NOISE METRIC AND VALUES.
CARD4 CASE TITLE
CARD5 NULL RECORD OR SPECIAL MESSAGE. IF A SPECIAL MESSAGE EXISTS, IT IS FOLLOWED BY A VARIABLE NUMBER OF EMPTY RECORDS (CARD5B, CARD5C, ETC.).
CARD6 6111,6E11.5 WHERE THE 6TH INTEGER IS THE NUMBER OF CONTOUR POINTS.
CARD7-16 (10E14.7 FORMAT) THE 1ST 10 X POINTS.
CARD17-25 THE 2ND 10 X POINTS.

CARD K-K+9 (10E14.7 FORMAT) THE 1ST 10 Y POINTS.

E-O-F

NEXT CONTOUR AS ABOVE
TAPE2 - SEQUENTIAL INPUT FILE CONTAINING
SUPPLEMENTAL DATA NOT GIVEN ON TAPE5.
DISPLAY CODE FORMAT:
CARDS 1-6 - INFORMATION USED BY OTHER ALAMO
PROGRAMS THAT SHARE THIS FILE
CARD7  LATITUDE, LONGITUDE, NORTH/SOUTH AND
EAST/WEST DISPLACEMENT.
CARD8  H,K TRANSLATION VALUES.
CARD9  POSITIVE X,Y AXIS LENGTH FOR PLOT.
CARD10 AIRPORT TITLE
CARD11 AIRPORT ADDRESS.

OUTPUT:
TAPE6 - CONVENTIONAL FORTRAN STANDARD OUTPUT FILE;
CURRENTLY NOT USED.
TAPE1 - SEQUENTIAL FILE CONTAINING SITE II INPUT
DECK, EACH DESCRIBING 150 POINTS OR LESS,
WITH INPUT DECKS SEPARATED BY E-O-F MARKS.
SEE SUBROUTINE BLDARR.

SCRATCH:
TAPE4 - SEQUENTIAL FILE CONTAINING CARD IMAGES OF
SITE II INPUT DECKS BEFORE PROCESSING BY
BLDARR.
SUBROUTINE CHKREC

OBJECTIVE:
   TO DETERMINE IF A MESSAGE, REGARDING USER-DEFINED PROFILES, IS WRITTEN ON THE INPUT FILE (TAPE5)

METHOD:
   SUBROUTINE CHKREC EVALUATES TAPE5 TO DETERMINE THE NUMBER OF RECORDS BETWEEN THE CARD IMAGE CONTAINING THE NOISE METRIC VALUE AND THE BEGINNING OF THE XY-DATA POINTS

PARAMETERS:
   INPUT:
      NONE
   OUTPUT:
      ICASE: 8-WORD ARRAY CONTAINING CASE NAME IN WORDS 5 THROUGH 8
      ISW: INTEGER VARIABLE USED TO INDICATE THE NUMBER OF RECORDS TO BE SKIPPED PRIOR TO THE READING OF THE XY DATA POINTS

CALLING ROUTINE:
   AUPHS1 (MAIN PROGRAM)

SUBPROGRAMS USED:
   NONE

FUNCTIONS USED:
   NONE

LOCAL VARIABLE DICTIONARY:
   IBLANK: INTEGER ARRAY USED TO CONTAIN THE POTENTIAL SPECIAL MESSAGE
   J: INTEGER VARIABLE USED AS LOOP COUNTER
   KEOF: INTEGER VARIABLE USED AS A FLAG TO INDICATE AN EARLY END-OF-INPUT
FILES USED:

INPUT:
TAPE5 - SEQUENTIAL FILE CONSISTING OF VARIABLE-LENGTH RECORDS; SUBROUTINE CHKREC IS CONCERNED WITH THE FORMAT OF THE FIFTH Z-TYPE RECORD ON TAPE5; WITH NO SPECIAL MESSAGE, THIS RECORD IS BLANK; WITH SPECIAL MESSAGE, RECORD CONSISTS OF A SPECIAL MESSAGE, FOLLOWED BY A VARIABLE NUMBER OF EMPTY RECORDS

OUTPUT:
NONE

SCRATCH:
NONE
SUBROUTINE RDCONTR

OBJECTIVE:
THIS ROUTINE READS THE HEADER DATA AND THE X, Y POINTS FOR A NOISE CONTOUR AS OUTPUT BY INM.

PARAMETERS:

INPUT:
ISW: INTEGER VARIABLE USED TO POSITION CORRECTLY TO XY DATA POINTS ON TAPE5

OUTPUT:
LINE: STORAGE FOR NOISE METRIC AND VALUES.
X: X POINTS FOR CONTOUR
Y: Y POINTS FOR CONTOUR
NPTS: THE NUMBER OF X,Y POINTS.
KEOF: SET TO 1 TO INDICATE EOF ON INPUT FILE.

CALLING ROUTINE:
AUPHS1

SUBPROGRAMS USED:
NONE

FUNCTIONS USED:
EOF : SYSTEM END-OF-FILE CHECK.

LOCAL VARIABLE DICTIONARY:
A,B,C,D
E,F,I,J
K,L,M : DUMMY VARIABLES FOR I/O LIST ONLY.
IJ : IMPLIED DO-LOOP INDEX.

COMMON VARIABLES:
NONE

FILES USED:
INPUT:
TAPE5 - SEQUENTIAL INPUT FILE CONTAINING THE CONTOUR X,Y POINTS AND HEADER DATA. (SEE MAIN PROGRAM)

OUTPUT:
NONE

A-8
SUBROUTINE INIT

OBJECTIVE:
THIS ROUTINE READS THE SUPPLEMENTAL DATA FILE, TAPE2.

PARAMETERS:

INPUT:
NONE

OUTPUT:
LAT : AIRPORT CENTROID LATITUDE.
LONG : AIRPORT CENTROID LONGITUDE.
NSDIS : NORTH/SOUTH DISPLACEMENT.
EWDIS : EAST/WEST DISPLACEMENT.
H : X AXIS TRANSLATION DISTANCE.
K : Y AXIS TRANSLATION DISTANCE.
FACT : PLOT AXIS LENGTH.
ITITLE : AIRPORT TITLE.

CALLING ROUTINE:
AUPHS1

SUBPROGRAMS USED:
NONE

FUNCTIONS USED:
EOF : SYSTEM END-OF-FILE TEST.

LOCAL VARIABLE DICTIONARY:
NO LOCAL VARIABLES

COMMON VARIABLES:
NONE

FILES USED:
INPUT:
TAPE2 - SEQUENTIAL INPUT FILE IN DISPLAY CODE FORMAT
SEE MAIN PROGRAM FOR CARD BY CARD FORMAT.
SUBROUTINE SI'l'E2

OBJECTIVE:
THIS ROUTINE DRIVES THE POINT REDUCTION SOFTWARE. ITS
OBJECTIVE IS TO DELETE AS MANY POINTS AS POSSIBLE FROM
A NOISE CONTOUR WITHOUT CHANGING THE TOTAL AREA THAT THE
CONTOUR ENCLOSES BY MORE THAN 1 PERCENT. THE MINIMUM
OBJECTIVE IS TO REDUCE THE CONTOUR TO LESS THAN 150 PTS.

METHOD:
THE ROUTINE SETS THE REDUCTION TARGET TO 65 POINTS AND
CALCULATES THE AREA OF THE ORIGINAL CONTOUR. THE POINT
REDUCTION ROUTINE IS CALLED ITERATIVELY UNTIL THE AREA
RESTRICTION IS MET. THE TARGET NUMBER OF POINTS IS
INCREASED BY 5 ON EACH ITERATION.

PARAMETERS:
INPUT:
X : THE ORIGINAL CONTOUR X POINTS.
Y : THE ORIGINAL CONTOUR Y POINTS.
N : THE NUMBER OF POINTS IN THE ORIG. CONTOUR.

OUTPUT:
XNEW : THE X POINTS AFTER REDUCTION.
YNEW : THE Y POINTS AFTER REDUCTION.
NEWNUM : THE NUMBER OF POINTS AFTER POINT REDUCT.

CALLING ROUTINE:
PHASE1

SUBPROGRAMS USED:
NEWCRV : POINT REDUCTION ALGORITHM.

FUNCTIONS USED:
AREA : CALCULATES THE AREA WITHIN A CLOSED POLYGON.
ABS : ABSOLUTE VALUE FOR REAL NUMBERS.

LOCAL VARIABLE DICTIONARY:
AREA1 : THE ORIGINAL CONTOUR AREA.
AREA2 : THE AREA FOR THE CONTOUR AFTER POINT REDUCTION.
DIF : THE DIFFERENCE IN AREA FOR THE ORIGINAL AND
POINT REDUCED CONTOURS.
INT : THE NUMBER OF POINTS TO RELAX POINT REDUCTION
BY TO MEET THE 1 PERCENT CHANGE IN AREA LIMIT.
ITER : THE NUMBER OF ITERATIONS THAT SUBROUTINE NEWCRV
PERFORMED TO DELETE POINTS.
PRCNT : THE CALCULATED PERCENT CHANGE IN AREA AFTER
POINT REDUCTION.
SUBROUTINE NEWCRV

OBJECTIVE:
TO DELETE THE MOST UNNECESSARY POINTS BELOW A GIVEN
MAXIMUM ON A NOISE CONTOUR. POINTS DELETED WILL BE THOSE
WHICH TEND TO BE ON STRAIGHT LINE SEGMENTS OR BROAD
CURVES.

METHOD:
OF 3 CONTIGUOUS POINTS A, B AND C, POINT B WILL BE DELETED
WHEN THE ABSOLUTE VALUE OF THE DIFFERENCE IN SLOPES FOR
LINES AB AND BC IS LESS THAN A GIVEN LIMIT. THE SIGNED
DIFFERENCES ARE ACCUMULATED SUCH THAT WHEN THE MAGNITUDE
OF THE ACCUMULATION BECOMES GREATER THAN THE LIMIT THE
POINT TO BE DELETED IS SAVED. THIS PROHIBITS EXCESSIVE
CONTOUR DISTORTION OVER BROAD CURVES IN ONE DIRECTION.
IF ENOUGH POINTS CANNOT BE DELETED AFTER PROCESSING THE
CONTOUR THE LIMIT IS RELAXED AND A NEW ITERATION IS
BEGUN. AS THE LIMIT IS RELAXED MORE POINTS WILL BE
DELETED AND MORE DISTORTION OF THE CONTOUR WILL BE
INTRODUCED.

PARAMETERS:
INPUT:
X : X POINTS OF THE CONTOUR TO BE REDUCED.
Y : Y POINTS OF THE CONTOUR TO BE REDUCED.
N : THE NUMBER OF X,Y POINTS IN CONTOUR TO BE
REDUCED.
MAXUM : THE MAXIMUM NUMBER OF POINTS TO BE LEFT IN
THE REDUCED CONTOUR.

OUTPUT:
XNEW : THE REDUCED X POINTS.
YNEW : THE REDUCED Y POINTS.
NEWNUM : THE NUMBER OF X,Y POINTS AFTER REDUCTION.
ITER : THE NUMBER OF SEPARATE COMPLETE LOOPS
AROUND THE CONTOUR. IE, THE NUMBER OF
RELAXATIONS OF THE SLOPE TOLERANCE.

CALLING ROUTINE:
SITE2

FUNCTIONS USED:
IOCTINT : USED TO DETERMINE WHAT COMPASS OCTANT AN X,Y
POINT IS IN.
LOCAL VARIABLE DICTIONARY:
B : ORDINAL POINTER TO THE 1ST OF THE 3 POINTS USED TO CALCULATE THE SLOPES.
DELPTS : ARRAY STORAGE FOR ORDINALS OF DELETED POINTS.
DIF : THE DIFFERENCE IN THE SLOPES, BT AND TTPl.
I : DO LOOP INDEX OVER THE TOTAL PT. ORDINALS FOR THE ORIGINAL CONTOUR.
INC : THE RELAXATION INCREMENT VALUE FOR THE SLOPE DIFFERENCE LIMIT.
KI : THE OCTANT NUMBER OF THE MIDPOINT OF THE 3 POINTS BEING EVALUATED. (B,T,TP1)
KJ : THE OCTANT NUMBER OF THE 3RD POINT, TP1 OF THE 3 POINTS BEING EVALUATED.
KPT : ORDINAL POINTER FOR THE ARRAY OF NEW POINTS AFTER REDUCTION.
LAST : THE LAST ORDINAL ON THE ORIGINAL CONTOUR.
MAXCHG : THE LIMIT FOR DIFFERENCE IN SLOPES WHEN A POINT IS OR IS NOT DELETED.
M1 : THE SLOPE OF LINE SEGMENT BT.
M2 : THE SLOPE OF LINE SEGMENT TTPl. (TP1= T + 1)
NEXT : NEXT AVAILABLE SLOT PTR FOR DELETED POINTS ARRAY, DELPTS.
NPT : NEXT AVAILABLE SLOT PTR FOR NEW POINTS ARRAY.
SUM : ACCUMULATOR FOR SIGNED SLOPE DIFFERENCES.
T : THE MIDDLE POINT ORDINAL FOR THE 3 POINTS UNDER SCRUTINY. T IS THE DELETION POINT.
TP1 : THE 3RD POINT ORDINAL, T + 1.
XZ : DELTA X FOR THE BT SLOPE.
XZl : DELTA X FOR THE TTPl SLOPE.

COMMON VARIABLES:
NONE

FILES USED:
NONE

A-12
SUBROUTINE NTHSTH

OBJECTIVE:
To convert X, Y points from feet to statute miles in a Site II compatible format using North/South/East/West designators and to output the converted points as Site II "D" cards.

METHOD:
The points are converted from feet to statute miles by dividing the feet units by 5280. Y points that are positive are given an "N" (North) indicator; negative Y's are given an "S" (South) indicator. The same logic applies to X points using E/W (East/West) indicators.

PARAMETERS:
INPUT:
X : The X pts to be converted.
Y : The Y pts to be converted.
N : The number of X,Y points.

CALLING ROUTINE:
POLY

SUBPROGRAMS USED:
NONE

FUNCTIONS USED:
ABS : Absolute value for real numbers.
LOCAL VARIABLE DICTIONARY:
I : DO LOOP INDEX TO COLUMNS OF ARRAY JBUF, A 3
   CO-ORDINATE BUFFER ("D" CARD COMPLEMENT).
J : IMPLIED DO LOOP INDEX FOR COLUMNS OF JBUF.
JBUF : BUFFER ARRAY FOR 3 CO-ORDINATES FOR A "D" CARD.
K : IMPLIED DO LOOP INDEX FOR ROWS OF JBUF WHEN
   THERE ARE LESS THAN 3 POINTS (LAST "D" CARD).
KPT : TOTAL CO-ORDINATE COUNTER TO PREVENT WRITING
      ONE OR TWO EXTRA POINTS ON LAST "D" CARDS.
KTRI : FORMAT RE PattION COUNT.
L : IMPLIED DO LOOP INDEX FOR COLUMNS OF JBUF.
XMILES : THE ABSOLUTE VALUE IN MILES OF THE CURRENT
         X POINT.
YMILES : SAME AS XMILES BUT FOR Y POINTS.

COMMON VARIABLES:
NONE

FILES USED:
INPUT:
NONE
OUTPUT:
NONE
SCRATCH:
TAPE4 - SEQUENTIAL FILE FOR SITE II INPUT DECKS.
       IT IS SUBSEQUENTLY REPROCESSED FOR 150 POINT
       RESTRICTION VIOLATIONS.
SUBROUTINE TKNSLT

OBJECTIVE:
TO TRANSLATE THE AXIS FOR THE NOISE CONTOURS.

METHOD:
THE TRANSLATION VALUES H AND K ARE SUBTRACTED FROM EACH X AND Y POINT RESPECTIVELY.

PARAMETERS:
  INPUT:
  X : X PTS FOR CONTOUR.
  Y : Y PTS FOR CONTOUR.
  NPTS : THE NUMBER OF X,Y PTS.
  H : THE X TRANSLATION FACTOR.
  K : THE Y TRANSLATION FACTOR.

CALLING ROUTINE:
AUPHS1

SUBPROGRAMS USED:
NONE

FUNCTIONS USED:
NONE

LOCAL VARIABLE DICTIONARY:
  I : DO LOOP INDEX OVER THE NUMBER OF PTS.

COMMON VARIABLES:
NONE

FILES USED:
NONE
FUNCTION AREA

OBJECTIVE:
TO CALCULATE THE AREA IN SQ UNITS OF A CLOSED POLYGON.

METHOD:
A SPECIAL FORM OF GREEN'S THEOREM IS EMPLOYED:

\[
\text{AREA} = \frac{1}{2} \sum_{i=0}^{N} (X_i Y_{i+1} - X_{i+1} Y_i)
\]

WHERE \( X = X_{n+1} \quad Y = Y_0 \)

PARAMETERS:

INPUT:
\( X \) : THE CONTOUR X POINTS.
\( Y \) : THE CONTOUR Y POINTS.
\( N \) : THE NUMBER OF X,Y POINTS.

OUTPUT:
\( \text{AREA} \) : FUNCTION VALUE RETURNED FOR THE CONTOUR AREA.

CALLING ROUTINE:
SITE2

SUBPROGRAMS USED:
NONE

FUNCTIONS USED:
\( \text{ABS} \) : ABSOLUTE VALUE.

LOCAL VARIABLE DICTIONARY:
\( \text{AREA} \) : SQUARE UNITS INSIDE CONTOUR.
\( I \) : DO LOOP INDEX OVER RANGE OF X,Y POINTS.
\( K \) : THE I+1 ORDINAL.
\( \text{SUM} \) : REAL VARIABLE USED IN COMPUTING CONTOUR AREA
SUBROUTINE OCTANT

OBJECTIVE:
THIS ROUTINE CO-ORDINATES THE POLYGON DEFINITIONS.

PARAMETERS:
INPUT:
X : THE CONTOUR X POINTS.
Y : THE CONTOUR Y POINTS.
NEWPTS : THE NUMBER OF X,Y POINTS AFTER POINT DEL.
LDN : THE NOISE METRIC OF THE CONTOUR. (LDN)
MET : THE NOISE METRIC VALUE. (55 - 85)
INDEX : THE CONTOUR COUNTER.
SCALE : THE PLOTTING SCALE FACTOR.

CALLING ROUTINE:
AUPHS1

SUBPROGRAMS USED:
OCTTAB : BUILDS A TABLE OF START AND END POINTS OF LINE SEGMENTS FOR EACH OCTANT.
POLY : BUILDS THE POLYGONS USING THE TABLE BUILT BY OCTTAB.
ENRICH : CHECKS FOR OCTANTS SPANned WITH NO X,Y POINTS AND PUTS IN POINTS WHEN NEEDED.

FUNCTIONS USED:
NONE

LOCAL VARIABLE DICTIONARY:
ITABLE : THE START POINT/END POINT TABLE FOR LINE SEGMENTS IN EACH OCTANT.
KCHANGE : FLAG THAT INDICATES POINTS WERE PUT IN AND ITABLE MUST BE REDEFINED BY OCTTAB ROUTINE.

COMMON VARIABLES:
NONE

FILES USED:
NONE
SUBROUTINE ENRICH

OBJECTIVE:
SUBROUTINE ENRICH IS USED TO EVALUATE CERTAIN POINTS TO DETERMINE IF ADJACENT POINTS ARE IN EITHER (1) THE SAME OCTANT OR (2) ADJACENT OCTANTS; IF 2 ADJACENT POINTS NOT IN THE SAME OCTANT ARE NOT IN ADJACENT OCTANTS, ROUTINE ADDPTS IS INVOKED TO ADD POINTS IN ORDER TO DEAL WITH THE SITUATION

PARAMETERS:
INPUT:
ITABLE : INTEGER ARRAY USED TO STORE THE START/END POINTS FOR LINE SEGMENTS IN EACH OCTANT
NEWPTS : THE NUMBER OF XY POINTS AFTER POINT DELETION
X    : REAL ARRAY CONTAINING THE X-COORDINATES OF CONTOUR POINTS PRIOR TO POSSIBLE POINT ENRICHMENT
Y    : REAL ARRAY CONTAINING THE Y-COORDINATES OF CONTOUR POINTS PRIOR TO POSSIBLE ENRICHMENT

OUTPUT:
KCHANGE : INTEGER VARIABLE USED AS FLAG TO INDICATE THAT POINTS WERE PUT IN AND ITABLE MUST BE REDEFINED BY OCTTAB ROUTINE
NEWPTS : INTEGER VARIABLE REPRESENTING THE NUMBER OF XY POINTS AFTER POINT ENRICHMENT
X    : REAL ARRAY CONTAINING X-COORDINATES OF CONTOUR POINTS AFTER POSSIBLE ENRICHMENT
Y    : REAL ARRAY CONTAINING Y-COORDINATES OF CONTOUR POINTS AFTER POSSIBLE ENRICHMENT

CALLING ROUTINE:
OCTANT

SUBPROGRAMS USED:
ADDPTS: ROUTINE USED TO ADD POINTS ON A CONTOUR WHEN A SPECIFIED CONDITION, AS DESCRIBED IN "OBJECTIVE" SECTION

SYSTEM: STANDARD FTN SUBROUTINE USED TO TERMINATE PROGRAM EXECUTION AND WRITE A DAYFILE MESSAGE
FUNCTIONS USED:
INTABLE : FUNCTION USED TO DETERMINE THE OCTANT OF AN ORDINAL, IF THE ORDINAL IS IN THE START/END TABLE
IOCTNT : FUNCTION USED TO DETERMINE WHAT OCTANT AN X,Y POINT RESIDES IN

LOCAL VARIABLE DICTIONARY:
A : REAL ARRAY USED FOR TEMPORARY STORAGE OF THE X-COORDINATES OF CONTOUR POINTS
B : REAL ARRAY USED FOR TEMPORARY STORAGE OF THE Y-COORDINATES OF CONTOUR POINTS
I : INTEGER VARIABLE USED AS LOOP COUNTER AND ARRAY INDEX
IOCT : INTEGER VARIABLE USED TO INDICATE IF ORDINAL IS IN THE START/END TABLE
IOCTM1 : INTEGER VARIABLE REPRESENTING AN ADJACENT OCTANT (COUNTERCLOCKWISE)
IOCTP1 : INTEGER VARIABLE REPRESENTING AN ADJACENT OCTANT (CLOCKWISE)
IPTR : INTEGER USED TO COUNT POINTS IN CONTOUR
J : INTEGER VARIABLE USED TO REPRESENT THE ORDINAL OF THE NEXT CLOCKWISE POINT
KOCT : INTEGER VARIABLE REPRESENTING THE OCTANT OF THE ADJACENT POINT
NPTS :

COMMON VARIABLES:
NONE

FILES USED:
INPUT:
NONE
OUTPUT:
NONE
SCRATCH:
NONE
FUNCTION INTABLE

OBJECTIVE:
FUNCTION INTABLE IS USED TO DETERMINE THE OCTANT OF AN ORDINAL

PARAMETERS:
INPUT:
   KORD : ORDINAL OF XY POINT
   ITABLE : THE START/END POINT TABLE FOR LINE SEGMENTS IN EACH OCTANT

OUTPUT:
   NONE

CALLING ROUTINE:
   ENRICH

SUBPROGRAMS USED:
   NONE

FUNCTIONS USED:
   NONE

LOCAL VARIABLE DICTIONARY:
   I : INTEGER VARIABLE USED AS LOOP COUNTER AND ARRAY INDEX
   J : INTEGER VARIABLE USED AS LOOP COUNTER AND ARRAY INDEX

COMMON VARIABLES:
   NONE

FILES USED:
   INPUT:
     NONE
   OUTPUT:
     NONE
   SCRATCH:
     NONE

A-20
SUBROUTINE ADDPTS

OBJECTIVE:
SUBROUTINE USED TO ADD POINTS ON A CONTOUR WHEN A
SPECIFIC CONDITION, AS DESCRIBED IN ENRICH, IS ENCOUNTERED

PARAMETERS:
INPUT:
A : REAL ARRAY CONTAINING X VALUES OF CONTOUR
   POINTS
B : REAL ARRAY CONTAINING Y VALUES OF CONTOUR
   POINTS
IPTR : INTEGER USED TO COUNT POINTS IN A CONTOUR
NPTS : INTEGER SET TO VALUE OF 2
XST : X-COORDINATE OF POINT PREceding ANOTHER
   POINT THAT IS NOT IN THE SAME, OR ADJACENT,
   OCTANT
YST : Y-COORDINATE OF POINT PREceding ANOTHER
   POINT THAT IS NOT IN THE SAME, OR ADJACENT,
   OCTANTS
XEND : X-COORDINATE OF POINT THAT HAS A PRECEDING
   POINT THAT SPANS MORE THAN 1 OCTANT
YEND : Y-COORDINATE OF POINT THAT HAS A PRECEDING
   POINT THAT SPANS MORE THAN 1 OCTANT

OUTPUT:
A : REAL ARRAY CONTAINING X VALUES OF CONTOUR
   POINTS AFTER POINT ENRICHMENT
B : REAL ARRAY CONTAINING Y VALUES OF CONTOUR
   POINTS AFTER POINT ENRICHMENT

CALLING ROUTINE:
ENRICH

SUBPROGRAMS USED:
NONE

FUNCTIONS USED:
ABS : STANDARD FTN FUNCTION USED TO CALCULATE ABSOLUTE
   VALUE
FLOAT : STANDARD FTN FUNCTION USED TO CONVERT INTEGER
   TO REAL NUMBER

A-21
LOCAL VARIABLE DICTIONARY:

DX : DIFFERENCE IN THE VALUES OF THE X-COORDINATES
     OF THE POINTS UNDER EVALUATION

DY : DIFFERENCE IN THE VALUES OF THE Y-COORDINATES
     OF THE POINTS UNDER EVALUATION

I  : INTEGER VARIABLE USED AS LOOP COUNTER

X  : X-COORDINATE OF NEW CONTOUR POINT FOUND

XINT : REAL VARIABLE REPRESENTING INCREMENT ON THE
       STARTING X VALUE, WHEN COMPUTING NEW VALUE

Y  : Y-COORDINATE OF NEW CONTOUR POINT FOUND

YINT : REAL VARIABLE REPRESENTING INCREMENT OF
       STARTING X-VALUE WHEN COMPUTING NEW POINT
SUBROUTINE OCTTAB

OBJECTIVE:
THIS ROUTINE BUILDS A TABLE OF START AND END POINTS FOR EACH LINE SEGMENT IN EACH OCTANT FOR THE SUBSEQUENT POLYGON DEFINITIONS.

METHOD:
THE CONTOUR POINTS ARE ARRANGED IN A CLOCKWISE FASHION IN ASCENDING ORDINAL ORDER. IT IS TYPICAL THAT THE 1ST POINT FALLS IN THE MIDDLE OF AN OCTANT. THIS PROCESSOR LOOKS BACKWARD (COUNTERCLOCKWISE) FROM THE 1ST POINT UNTIL IT FINDS A POINT OUTSIDE THE OCTANT THAT THE 1ST POINT OCCUPIES. THE LAST COUNTERCLOCKWISE POINT THAT IS INSIDE THE STARTING OCTANT IS INSERTED IN THE TABLE AS INDEXED BY THE OCTANT (1-8). THE COLUMNS OF THE TABLE ARE ARRANGED AS ODD FOR START POINTS AND EVEN FOR END POINTS. FOR EXAMPLE, THE SECOND LINE SEGMENT IN OCTANT 4 WOULD BE STORED IN ROW 4, WITH THE START POINT IN COLUMN 3 AND THE END POINT IN COLUMN 4. THE ARRAY NXTCOL IS USED TO POINT TO THE NEXT AVAILABLE COLUMN ORDINAL IN EACH OCTANT OF THE TABLE. HAVING FOUND THE REAL STARTING POINT BY BACKING UP AS DESCRIBED ABOVE, THE END POINT IS FOUND IN THE SAME MANNER BY MOVING AHEAD (CLOCKWISE) UNTIL THE END POINT IS FOUND AND STORED IN THE TABLE. AT THIS TIME THE CURRENT OCTANT BECOMES THE OCTANT THAT THE 1ST OUTSIDE POINT WAS FOUND WITHIN. THE POINT IS STORED IN THE TABLE AS A START POINT AND THE CLOCKWISE PROCESS IS BEGUN AGAIN BY MOVING CLOCKWISE UNTIL THE NEXT OUTSIDE POINT IS FOUND. THIS PROCESSING IS CONTINUED UNTIL THE ORIGINAL STARTING POINT IS ENCOUNTERED (NOT NECESSARILY ORDINAL 1) INDICATING 1 COMPLETE CLOCKWISE TRIP AROUND THE CONTOUR.

PARAMETERS:

INPUT:
X : THE X POINTS FOR THE POINT REDUCED CONTOUR
Y : THE Y POINTS FOR THE POINT REDUCED CONTOUR
NPTS : THE NUMBER OF X,Y POINTS. NOTE THAT POINT 1 = POINT NPTS.

OUTPUT:
TABLE : THE STARTPOINT/ENDPOINT LINE SEGMENT, OCTANT TABLE.
CALLING ROUTINE:
  OCTANT

SUBPROGRAMS USED:
  NONE

FUNCTIONS USED:
  IOCTNT : DETERMINES THE OCTANT WHERE AN X,Y POINT RESIDES.

LOCAL VARIABLE DICTIONARY:
  BEGOCT : THE ORDINAL OF THE BEGINNING OCTANT (WHERE PT NUMBER 1 RESIDES).
  CUROCT : THE ORDINAL OF THE OCTANT WHERE PTS ARE BEING PROCESSED TO FIND AN OUTSIDE PT.
  I : DO LOOP INDEX
  INDEX : THE CONTOUR ORDINAL POINTER FOR CLOCKWISE AND COUNTERCLOCKWISE PROCESSING.
  LSTPT : THE ORDINAL OF THE LAST POINT IN A LINE SEGMENT WHICH IS IN THE SAME OCTANT AS THE OTHER POINTS
  NXTCOL : THE VECTOR OF PTRS TO THE NEXT AVAILABLE PAIR OF COLUMNS FOR A ST/END POINT PAIR FOR THE ITH OCTANT.
  NXTOCT : THE OCTANT ORDINAL FOR THE NEXT CLOCKWISE PT ON A LINE SEGMENT WHICH IS COMPARED WITH THE VARIABLE CUROCT.
  STRTPT : THE TRUE STARTING POINT OF THE 1ST LINE SEGMENT FOUND BY BACKING UP FROM PT NUMBER 1.

COMMON VARIABLES:
  NONE

FILES USED:
  NONE
FUNCTION IOCTNT

OBJECTIVE:
TO DETERMINE WHAT OCTANT AN X,Y POINT RESIDES IN.

METHOD:
THE OCTANTS ARE ORDERED CLOCKWISE AS FOLLOWS USING THE
STANDARD CARTESIAN QUADRANTS:
OCTANT 1 THE UPPER HALF OF QUADRANT IV
2 THE LOWER HALF OF QUADRANT IV
3 THE LOWER HALF OF QUADRANT III
4 THE UPPER HALF OF QUADRANT III
5 THE LOWER HALF OF QUADRANT II
6 THE UPPER HALF OF QUADRANT II
7 THE UPPER HALF OF QUADRANT I
8 THE LOWER HALF OF QUADRANT I

SPECIAL CASES ARE TESTED FOR (X = Y OR X = Y = 0) FIRST
THEN THE QUADRANT IS TESTED FOR. ONCE THE QUADRANT IS
DETERMINED, IT IS DETERMINED ON WHAT SIDE OF THE LINE
Y = X OR Y = -X THE POINT FALLS. THE SPECIAL CASES WHERE
THE POINT FALLS ON AN OCTANT LINE OR ON THE ORIGIN CAUSE
THE OCTANT TO BE DEFINED AS THE LAST OCTANT VALUE.

PARAMETERS:
INPUT:
X : THE X POINTS OF THE CONTOUR.
Y : THE Y POINTS OF THE CONTOUR.
N : THE NUMBER OF POINTS IN THE CONTOUR.
INDEX : THE ORDINAL OF THE POINT TO BE EXAMINED.
LSTOCT : THE CURRENT OCTANT ORDINAL.

OUTPUT:
NONE

CALLING ROUTINE:
NEWCRV
OCTTAB
POLY
DISTNCE
INTSCT

FUNCTIONS USED:
ABS : ABSOLUTE VALUE ROUTINE.
SUBROUTINE POLY

OBJECTIVE:

This routine forms polygons from the start/end point table defined by subroutine OCTTAB. The polygons are formed by the contour lines and their intersection with the octant lines. When a polygon is defined it is output as a site II "C" card followed by the required number of "D" cards. Lastly the polygon is plotted if the contour counter is odd.

METHOD:

The table of start and end points is first processed to compute the distance from the origin to each start and end point on an octant line. To do this the equation of a line is defined using a start/end point and its neighboring outside point. This line must intersect an octant line and it is this intersection point that is used for distance to origin calculations. The distances are stored in the table DSTS. Another table, IUSED, is used to mark a start/end point as used or unused since the algorithm will do things like search away from the origin for the 1st unused start pt or towards the origin for the 1st unused end pt. The distances were calculated to afford this capability. The polygon generation algorithm can be stated as the following 4 rules:

Note: The current octant line is the octant line whose ordinal = the octant number being processed. The positive x axis is ordinal 1. The other octant lines are numbered ascendingly from no.1 in a clockwise direction.

1. Start processing an octant by taking points starting at the 1st start point in the table until an end point is encountered.

2. When an endpoint is encountered on the current octant line find the next start point moving away from the origin (a start pt whose distance is greater than the end pt just encountered and whose distance is the minimum of other qualifying start pts). If the qualifying start pt has already been used we have closure (a complete polygon). If it has not been used it is selected as a member of the current polygon and the "used" table is updated.
3. When an endpoint is encountered on the next octant line find the next start point moving toward the origin (a start pt whose distance is less than the end pt just encountered and whose distance is the maximum of other qualifying start pts). If a start point cannot be found then the origin is taken as the next pt. and rule number 2 is applied.

4. When a start/end point is selected it is no longer available for later selection.

Parameters:

Input:
- Table: the start/end point octant table.
- X: the X points of the contour.
- Y: the Y points of the contour.
- N: the number of X, Y points.
- Met: the table of LDN metrics.
- LDNPT: ptr to current LDN metric.
- LDN: the metric name. ("LDN").
- Scale: the plotting scale factor.

Calling routine:
- OCTANT

Subprograms used:
- DISTNE: calculates the distance from the origin of all start/end points.
- INTSCT: finds the intersection X, Y point of an octant line and a line defined by 2 X, Y points which straddle the octant line.
- NTSTH: converts the polygon points to site II north/south/east/west displacements in miles.
- PLTPOL: routine which plots polygon X, Y points.

Functions used:
- AREA: calculates the area of a polygon.
- IOCTNT: determines the octant number for a given X, Y point.
LOCAL VARIABLE DICTIONARY:

A : THE POLYGON AREA IN SQUARE MILES, OUTPUT AS F6.1 ON THE SITE II "C" CARD.

DELTA X : WHEN AN INTERSECTION PT IS CALCULATED ON AN OCTANT LINE, ITS PROXIMITY TO THE ALREADY EXISTING NEIGHBOR PT IS CALCULATED IN TERMS OF DELTA X AND DELTA Y (THE DIFFERENCE IN X'S AND Y'S).

DELTA Y : SEE DELTA X.

DIST : A TABLE OF START/END POINT DISTANCES FROM THE ORIGIN.

ENDDIS : A DISTANCE TO A SELECTED END POINT (FROM THE ORIGIN).

I : DO LOOP VARIABLE FOR INITIALIZATION.

ICOL : COLUMN INDEX TO LINE SEGMENT ST/END POINT TABLE.

ICOUNT : COUNTER FOR THE NUMBER OF TIMES THIS ROUTINE IS CALLED. USED TO PLOT POLYGONS ON ODD COUNTS.

IFLAG : SET TO 1 TO INDICATE THAT THIS IS AN ODD COUNT CALL.

INPT : THE "INSIDE POINT" WHEN 2 POINTS STRADDLE AN OCTANT LINE. THE INSIDE POINT IS THE POINT WITHIN THE OCTANT BEING ANALYSED.

IOCT : THE ARRAY INDEX CORRESPONDING TO OCTANTS.

IPTR : THE NEXT AVAILABLE SLOT FOR X,Y POINT STORAGE AS THE POLYGON IS BEING FORMED.

IRES : THE RESULT OF DIVIDING ICOUNT BY 2. WHEN ICOUNT IS ODD THE REMAINDER IS TRUNCATED SUCH THAT 2 * IRES WILL NOT = ICOUNT.

IUSED : AN ARRAY USED TO MARK ST/END POINTS AS USED OR UNUSED DURING POLYGON DEFINITION.

J : MISCELLANEOUS DO LOOP COUNTER.

JPTR : THE NEXT AVAILABLE SLOT FOR A VECTOR OF QUALIFYING ST/END POINTS TO BE COMPARED FOR DISTANCES FROM THE ORIGIN.

KOCT : THE OCTANT ORDINAL OF AN INTERSECTION POINT ON AN OCTANT LINE.

KPT : WORKING POINTER TO THE ST/END POINTS COLUMNS.

KEK : VECTOR FOR STORAGE OF ST/END POINTS TO BE COMPARED FOR MINS AND MAXES.

LAST : LAST POINT (END POINT) IN A LINE SEGMENT.

MVEK : SAME AS KEK.

NPTR : WORKING POINTER TO EITHER AN INTERSECTION PT OR AN EXISTING POINT EXTREMELY CLOSE TO THE INTERSECTION POINT.

NUM : ON AN EVEN VALUE OF ICOUNT NUM WILL BE EQUAL TO ICOUNT. ON ODD VALUE NUM WILL NOT EQUAL ICOUNT.
COMMON VARIABLES:
  NONE

FILES USED:
  SCRATCH:
    TAPE4 : COLLECTS SITE II CARDS FOR SUBSEQUENT
            POST PROCESSING.
SUBROUTINE DSTNCE

OBJECTIVE:
TO COMPUTE THE DISTANCE FROM THE ORIGIN TO EVERY START
AND END POINT IN THE START/END POINT OCTANT TABLE AND
TO LOCATE THE OCTANT ORDINAL FOR EACH START/END POINT.

METHOD:
GIVEN THE ORDINAL OF A START POINT WHICH IS INSIDE OR ON
OCTANT LINE I, IT IS POSSIBLE TO CALCULATE A NEW POINT ON
THE OCTANT LINE BY FORMING THE INTERSECTION OF THE OCTANT
LINE AND THE LINE FORMED BY THE START POINT AND ITS PRED-
CESSOR. ONCE THE INTERSECTION POINT IS OBTAINED THE
DISTANCE FORMULA IS USED WITH THE ORIGIN AS THE 2ND POINT
TO FIND THE DISTANCE TO THE ORIGIN. THE INTERSECTION
POINT IS ALSO PASSED TO FUNCTION IOCNTNT WHICH RETURNS THE
ORDINAL OF THE OCTANT LINE THAT THE START/END POINT
RESIDES ON.

PARAMETERS:
INPUT:
  TABLE : THE START/END POINT OCTANT TABLE.
  X    : THE X POINTS OF THE CONTOUR BEING
         PROCESSED.
  Y    : THE Y POINTS OF THE CONTOUR BEING
         PROCESSED.
  N    : THE NUMBER OF X,Y POINTS.
OUTPUT:
  DSTS : THE TABLE OF START/END POINT DISTANCES
         TO THE ORIGIN.
  WHERE : THE TABLE OF OCTANT ORDINALS FOR EACH
         START/END POINT.

CALLING ROUTINE:
  POLY
SUBPROGRAMS USED:
INTSCT: CALCULATES THE INTERSECTION OF A LINE WITH AN OCTANT LINE GIVEN 2 POINTS THAT ARE KNOWN TO STRADDLE AN OCTANT LINE.

FUNCTIONS USED:
IOCTNT: DETERMINES THE ORDINAL OF THE OCTANT WHERE A POINT RESIDES.
SQR: FORTRAN SQUARE ROOT FUNCTION FOR REALS.

LOCAL VARIABLE DICTIONARY:
I: DO LOOP INDEX FOR OCTANT ORDINALS.
INPT: A POINT WHICH IS LOCATED INSIDE A CERTAIN OCTANT.
J: DO LOOP INDEX TO START/END POINTS.
OUTPT: A POINT WHICH IS IN AN ADJACENT OCTANT RELATIVE TO EITHER ITS PREDECESSOR OR SUCCESSOR POINT.
RAD: SPECIAL FORM OF THE DISTANCE FORMULA WHEN ONE POINT IS THE ORIGIN: D = SQRT(RAD) WHERE RAD = X**2 + Y**2.
XNEW: X COORDINATE OF INTERSECTION POINT WHOSE DISTANCE IS TO BE COMPUTED.
YNEW: Y COORDINATE (SEE XNEW).

COMMON VARIABLES:
NONE

FILES USED:
NONE
SUBROUTINE INTSCT

OBJECTIVE:
GIVEN TWO POINTS, CALCULATE THE EQUATION OF THE LINE THEY
LIE ON, DETERMINE THE UNIQUE OCTANT OF EACH POINT TO FIND
THE OCTANT LINE THEY STRADDLE AND FIND THE POINT OF
INTERSECTION ON THE OCTANT LINE.

METHOD:
The OCTANT LINES ARE THE SPECIAL FORMS: Y = X, Y = -X,
Y = 0, X = 0. GIVEN THE 2 POINTS IT IS POSSIBLE TO
DETERMINE THE STRADDLED OCTANT LINE USING THE FUNCTION
IOCTNT WHICH DETERMINES THE OCTANT ORDINAL FOR A GIVEN
POINT. ONCE THE OCTANT LINE IS KNOWN ITS EQUATION IS
ALSO KNOWN. THE EQUATION OF THE LINE FOR THE 2 GIVEN
POINTS IS OBTAINED USING THE POINT-SLOPE FORMULA. THEN
CRAMER'S RULE IS USED TO CALCULATE THE INTERSECTION X,Y
VALUES. SPECIAL CASES (INFINITE SLOPE) ARE HANDLED.

PARAMETERS:
INPUT:
XIN : X VALUE OF INSIDE POINT.
YIN : Y VALUE OF INSIDE POINT.
XOUT : X VALUE OF OUTSIDE POINT.
YOUT : Y VALUE OF OUTSIDE POINT.

OUTPUT:
X : X VALUE OF INTERSECTION POINT.
Y : Y VALUE OF INTERSECTION POINT.

CALLING ROUTINE:
POLY
DISTANCE

SUBPROGRAMS USED:
NONE

FUNCTIONS USED:
IOCTNT : DETERMINES ORDINAL OF OCTANT WHERE A PT RESIDES
LOCAL VARIABLE DICTIONARY:

A : THE X COEFFICIENT FOR THE EQUATION OF AN OCTANT LINE.
B : THE Y COEFFICIENT AS IN A ABOVE.
C : THE CONSTANT AS IN A ABOVE.
D : THE NEGATIVE SLOPE OF THE LINE FORMED BY (XIN,YIN) AND (XOUT,YOUT).
DELTA : THE DENOMINATOR OR "DELTA" FOR CRAMER'S RULE.
DX : DETERMINANT AFTER SUBSTITUTING FOR X.
DY : DETERMINANT AFTER SUBSTITUTING FOR Y.
E : Y COEFFICIENT FOR THE LINE GIVEN 2 PTS.
F : THE CONSTANT FOR THE LINE GIVEN 2 PTS.
KI : THE OCTANT ORDINAL FOR THE INSIDE POINT.
KJ : THE OCTANT ORDINAL FOR THE OUTSIDE POINT.
LFLAG : IDENTIFIES OCTANT LINE EQUATION WHEN SLOPE IS INFINITE FOR EQUATION OF A LINE.
LSTOCT : PARAMETER TO IOCTNT FUNCTION WHICH TELLS WHAT THE PREVIOUS OCTANT ORDINAL WAS. SET TO ZERO FOR THIS USAGE.
M : THE SLOPE OF THE LINE FOR THE INSIDE/OUTSIDE POINTS.

COMMON VARIABLES:
NONE

FILES USED:
NONE
SUBROUTINE PLOT

OBJECTIVE:
TO INITIALIZE THE PLOT VECTOR FILE, CALCULATE THE SCALE
FACTOR, DRAW THE COMPASS ROSE AND RESET THE PLOT ORIGIN.

PARAMETERS:
INPUT:
XPTS : THE X VALUES OF THE POLYGON TO BE PLOTTED.
YPTS : THE Y VALUES OF THE POLYGON TO BE PLOTTED.
N : THE NUMBER OF X, Y POINTS.
ITITLE : THE 30 CHARACTER AIRPORT TITLE.
LINE : THE LDN METRIC TITLE INFORMATION.
FACT : HALF THE LENGTH OF THE X, Y AXIS.

OUTPUT:
SCALE : THE PLOTTING SCALE FACTOR.

CALLING ROUTINE:
AUPHS1

SUBPROGRAMS USED:
CALPLT : GRAPHICS ROUTINE FOR MOVING THE "PEN".
PSEUDO : GRAPHICS ROUTINE FOR INITIALIZING THE PLOT
VECTOR FILE (PVF).
ROSE : DRAWS THE COMPASS ROSE AND TITLES.

FUNCTIONS USED:
ABS : ABSOLUTE VALUE FOR REAL NUMBERS.

LOCAL VARIABLE DICTIONARY:
I : DO LOOP INDEX OVER THE RANGE OF X, Y POINTS.
ZMAX : THE MAX VALUE OF ALL X, Y DATA POINTS.

COMMON VARIABLES:
NONE

FILES USED:
INPUT:
TAPE2 - SEQUENTIAL INPUT FILE USED WITH
COUNTERMEASURES FOR READING IN THE
PLOT SCALE OF THE BASELINE CASE

OUTPUT:
TAPE2 - SEQUENTIAL FILE USED WITH THE BASELINE
CASE FOR WRITING THE PLOT SCALE OF THE
BASELINE CASE
SUBROUTINE ROSE

OBJECTIVE:
   TO PLOT THE COMPASS ROSE, PLOT BOUNDARIES AND LABELS.

METHOD:
   DRAW A SQUARE USING THE GIVEN HALF-LENGTH FOR THE X, Y
   AXES; THEN DRAW + AND - X, Y AXES AND THE 2 LINES Y = X
   AND Y = -X (THE DIAGONALS OF THE SQUARE). ANNOTATE EACH
   OCTANT WITH ITS 3-LETTER COMPASS MNEUMONIC.

PARAMETERS:
   INPUT:
   AXISLEN: THE HALF-LENGTH OF THE X, Y AXES.
   SCALE : THE SCALE FACTOR; USED TO DRAW THE SCALE
            CHART.
   ITITLE : THE 30-CHARACTER AIRPORT TITLE.
   LINE : THE LDN METRIC VALUE.

CALLING ROUTINE:
   PLOT

SUBPROGRAMS USED:
   BOUND : DRAWS BOUNDARY LINES (BOX) AND 3-LETTER
            COMPASS MNEUMONICS.
   CALPLT : GRAPHICS ROUTINE FOR PEN POSITIONING.
   LABELS : PARTIALLY NOTATES THE PLOT LABELS.
   NOTATE : GRAPHICS CHARACTER PLOTTING ROUTINE.

FUNCTIONS USED:
   NONE
LOCAL VARIABLE DICTIONARY:

CENTER : X COORDINATE FOR THE CENTER OF THE SCALE CHART.
HT : THE HEIGHT OF CHARACTERS IN FLOATING PT INCHES.
IHOL : HOLLERITH STRING ARRAY CONTAINING THE 3-LETTER
      COMPASS DIRECTION MNEUMONICS.
S : THE DISTANCE FROM THE LEFT MARGIN OF A SYMBOL
    TO THE LEFT MARGIN OF THE NEXT SYMBOL.
TOIW : THE TOTAL WIDTH OF THE SCALE LEDGER.
W : THE WIDTH OF 1 CHARACTER OR SYMBOL.
WADJ : THE WIDTH ADJUSTMENT WHICH IS USED TO
       COMPENSATE FOR CHARACTER HEIGHT.
X : THE X COORDINATE PASSED TO THE BOUNDARY LINE
    ROUTINE.
XLEN : THE LENGTH OF THE 2 MILE SCALE LINE IN INCHES.
X1 : THE LEFT MOST X COORDINATE OF THE SCALE LINE.
X2 : THE RIGHT MOST COORDINATE OF THE SCALE LINE.
X3 : THE CENTER PT X COORDINATE OF THE SCALE LINE.
Y : THE COORDINATE PASSED TO THE BOUNDARY LINE
   ROUTINE.
Y1 : THE Y COORDINATE OF THE LEFT MOST PART OF THE
    SCALE LINE.
Y2 : THE COORDINATE OF THE POSITIVE END OF THE
    VERTICAL SCALE LINE TIC MARKS.
Y3 : LIKE Y2 ONLY THE NEGATIVE END.

COMMON VARIABLES:
 N O N E

FILES USED:
 N O N E
SUBROUTINE LABELS

OBJECTIVE:
TO DRAW THE AIRPORT NAME AND "LDN FOOTPRINT" AT THE TOP
OF THE PLOT; TO ESTABLISH VALUES, LATER USED BY RESTPLT,
TO COMPLETE THE PLOT LABELING.

PARAMETERS:

INPUT:
CENTER : THE CENTER X AND Y VALUE OF THE COMPASS
ROSE.
ITITLE : THE AIRPORT NAME
LDNST : THE STARTING LDN METRIC.

CALLING ROUTINE:
ROSE

SUBPROGRAMS USED:
NOTATE : GRAPHICS ROUTINE USED TO DRAW CHARACTER STRINGS

FUNCTIONS USED:
FLOAT : INTEGER TO REAL CONVERSION
OR : LOGICAL OR OF 2 OR MORE ARGUMENTS.
SHIFT : LEFT CIRCULAR(LOGICAL)/RT ARITHMETIC WITH SIGN
EXTENSION SHIFT.

LOCAL VARIABLE DICTIONARY:
HT : THE HEIGHT IN INCHES OF CHARACTERS TO BE
NOTATED.
I : DO LOOP INDEX USED FOR FINDING A.P. NAME WIDTH.
ICHAR : COMPUTED CHARACTER PACKED INTO THE RANGE OF
LDN CONTOUR TITLE.
IDUM : DUMMY LIST VARIABLE USED FOR DECODE.
IREM : VARIABLE USED TO DO DIGIT BY DIGIT ENCODE OF
LDN METRIC VALUES FOR LDN RANGE TITLE.
ISTRIN : CHARACTER STRING (2 WORDS) WITH HOLES FOR LOW
AND HIGH LDN RANGE VALUES:
" 55 DB TO 85 DB " IS THE IDEAL FINAL FORM.
J : BACKWARD COUNTER FROM 80 AND 40 IN CONTRAST WITH I.
LINE : DECODE TARGET (8 WORDS ONTO 80) FOR A.P. TITLE.
LINE2 : 40-WORD ARRAY CONTAINING CASE TITLE.
MET : THE ALPHA FORM OF THE LDN METRIC VALUE IN F5.1
METST : THE INTEGER FORM OF MET. (THE STARTING VALUE).
S : THE DISTANCE FROM THE LEADING EDGE OF A SYMBOL TO THE LEADING EDGE OF THE NEXT SYMBOL.
TOTW : THE TOTAL WIDTH OF A CHARACTER STRING IN INCHES
W : THE WIDTH OF ONE SYMBOL.
WADJ : WIDTH ADJUSTMENT FACTOR BASED ON SYMBOL HEIGHT.
X : THE X COORDINATE OF THE LOWER LEFT HAND CORNER OF THE 1ST CHARACTER OF A STRING TO BE DRAWN SUCH THAT IT IS CENTERED.
XMET : THE REAL FORM OF THE LDN STARTING CONTOUR.
Y : Y COORDINATE - SEE X.

COMMON VARIABLES:
/ITITL/
ICASE1 : 4-WORD ARRAY CONTAINING CASE TITLE
/PLTINFO/
   CENT : THE CENTER X AND Y VALUE OF THE COMPASS ROSE
   HT : THE HEIGHT IN INCHES OF CHARACTERS TO BE NOTATED
   ISTRIN : CHARACTER STRING (2 WORDS) USED TO NOTATE RANGE OF LDN VALUES
   S : THE DISTANCE FROM THE LEADING EDGE OF A SYMBOL TO THE LEADING EDGE OF THE NEXT SYMBOL
   W : REAL VARIABLE REPRESENTING THE WIDTH OF ONE SYMBOL

FILES USED:
NONE

A-38
SUBROUTINE BOUND

OBJECTIVE:

TO DRAW THE PLOTTING BOUNDARY LINES AND ANNOTATE THE COMPASS ROSE DIRECTION MNEUMONICS.

PARAMETERS:

INPUT:

AXLEN : HALF THE LENGTH OF THE X, Y AXES.
X : THE X PT OF ORIGIN FOR THE BOUNDARY LINE TO BE DRAWN.
Y : Y COORDINATE (SEE X).
INDEX : THE INDEX TO THE 3-LETTER DIRECTION MNEUMONICS TO BE NOTATED ON THIS CALL.
IANGL : 0 FOR VERTICAL BOUNDARY, 1 FOR HORIZONTAL.
IHOL : THE ARRAY OF 3 LETTER COMPASS HEADING MNEUMONICS.

CALLING ROUTINE:

ROSE

SUBPROGRAMS USED:

CALPLOT : X, Y PEN MOVEMENT FOR GRAPHICS.
NOTATE : NOTATES CHARACTER STRINGS ON GRAPHIC DISPLAY.

FUNCTIONS USED:

NONE

LOCAL VARIABLE DICTIONARY:

HT : CHARACTER HEIGHT.
THETA : ANGLE OF ORIENTATION FOR CHARACTER STRINGS.
TOWDT : TOTAL WIDTH OF 3 LETTER MNEUMONIC.
WADJ : CHARACTER WIDTH ADJUSTMENT FACTOR.
X4 : THE DESTINATION PT FOR PLOT PEN IN DOWN POS.
X6 : X COORD WHEN IANGL = 1, Y COORD WHEN IANGL = 0.
X7 : SAME AS X6 BUT FOR 2ND MNEUMONIC PLOTTED (THERE ARE 2 MNEUMONICS FOR EACH BOUNDARY LINE).
X8 : THE X COORDINATE FOR THE HORIZONTAL BOUNDARY LINE MNEUMONICS.
Y3 : THE Y COORDINATE OF HORIZONTAL BOUNDARY LINE MNEUMONICS.
SUBROUTINE PLTPOL

OBJECTIVE:
   TO DRAW A POLYGON GIVEN THE X, Y POINTS AND A SCALE FACTOR.

METHOD:
   CONVERT THE SCALE FACTOR TO MILES UNITS AND DIVIDE ALL DATA VALUES BY NEW SCALE FACTOR. THEN USE MULTIPLE POINT DRAW GRAPHICS ROUTINE.

PARAMETERS:
   X : X POINTS FOR POLYGON TO BE PLOTTED.
   Y : Y POINTS (SEE X).
   N : NUMBER OF X, Y POINTS IN POLYGON.
   SCALE : THE SCALE FACTOR.

CALLING ROUTINE:
   POLY

SUBPROGRAMS USED:
   DRAW : DRAWS A LINE GIVEN A SERIES OF SCALED DATA PTS.

FUNCTIONS USED:
   NONE

LOCAL VARIABLE DICTIONARY:
   DIV : SCALE FACTOR AFTER CONVERSION TO STATUTE MILES.
   I : DO LOOP VARIABLE OVER RANGE OF POINTS.
   XS : SCALED X POINTS.
   YS : SCALED Y POINTS.

COMMON VARIABLES:
   NONE

FILES USED:
   NONE
SUBROUTINE BLDARR

OBJECTIVE:
To evaluate the Site II input deck, subdividing any deck with more than 150 points into 2 Site II runs; to put end-of-file marks at the end of input decks for each Site II run. Additionally, to compute values for register settings, with NCON indicating the total number of Site II runs, and NSTCON providing an index to the LDN value of the first contour found.

METHOD:
For each Site II run (indicated by an 'A' card), count the corresponding number of points on the associated 'D' cards; store the total number of points in array IPTS; for each run exceeding 150 points, the associated groups of 'C' and 'D' cards are subdivided.

PARAMETERS:
INPUT:
ICASE: 8-WORD array with words 5 through 8 containing the case name

OUTPUT:
NCON: Integer variable, to be used for register setting, indicating the total number of Site II executions required
NSTCON: Integer variable, to be used for register setting, that is an index to the LDN value of the first contour found

CALLING ROUTINE:
AUPHS1 (MAIN PROGRAM)

SUBPROGRAMS USED:
NONE

FUNCTIONS USED:
NONE
LOCAL VARIABLE DICTIONARY:

IA: INTEGER ARRAY USED TO CONTAIN THE CARD IMAGE OF A SITE II 'A' CARD

IARRAY: 2-DIMENSION INTEGER ARRAY, WITH THE FIRST DIMENSION USED TO REPRESENT A SINGLE LDN CONTOUR VALUE, AND THE SECOND DIMENSION USED TO INDICATE THE NUMBER OF POINTS IN A SINGLE POLYGON FOR THE ASSOCIATED LDN CONTOUR VALUE

IB: INTEGER ARRAY USED TO CONTAIN THE CARD IMAGE OF A SITE II 'B' CARD

IBLK: INTEGER ARRAY USED TO CONTAIN THE FIRST SITE II CARD IMAGE

ICHAR: INTEGER VARIABLE CONTAINING, IN THE LAST 6 BITS, THE FIRST 6 BITS OF A SITE II CARD IMAGE

ICHAR1: INTEGER VARIABLE CONTAINING, IN THE LAST 6 BITS, THE FIRST 6 BITS OF A SITE II CARD IMAGE

ICHAR2: INTEGER VARIABLE CONTAINING, IN THE LAST 6 BITS, THE FIRST 6 BITS OF A SITE II CARD IMAGE

ICOUNT: INTEGER VARIABLE USED TO INDICATE A SPECIFIC POLYGON ASSOCIATED WITH A SPECIFIC LDN CONTOUR

ICTR: INTEGER VARIABLE USED TO INDICATE THE NUMBER OF POINTS IN A PARTICULAR POLYGON

IFACT: INTEGER VARIABLE INDICATING THE TOTAL NUMBER OF LDN CONTOURS FOUND

IJ: INTEGER VARIABLE USED AS POLYGON INDEX

IJK: INTEGER VARIABLE USED AS LDN CONTOUR INDEX

IPTS: INTEGER ARRAY USED TO CONTAIN THE TOTAL NUMBER OF POINTS FOR EACH LDN CONTOUR

I: INTEGER VARIABLE, CONTAINING IN THE LAST 6 BITS, THE FIRST 6 BITS OF A SITE II CARD IMAGE

ISA: INTEGER USED AS TEMP. STORAGE FOR PART OF CASE TITLE

ISB: INTEGER USED AS TEMP. STORAGE FOR PART OF CASE TITLE

I6A: INTEGER USED AS TEMP. STORAGE FOR PART OF CASE TITLE

I6B: INTEGER USED AS TEMP. STORAGE FOR PART OF CASE TITLE

I7A: INTEGER USED AS TEMP. STORAGE FOR PART OF CASE TITLE

I8A: INTEGER USED AS TEMP. STORAGE FOR PART OF CASE TITLE

I8B: INTEGER USED AS TEMP. STORAGE FOR PART OF CASE TITLE

LINE: INTEGER ARRAY USED TO CONTAIN A SITE II CARD IMAGE

SPACE: INTEGER ARRAY USED TO CONTAIN THE FIRST FOUR WORDS OF A SITE II CARD IMAGE
COMMON VARIABLES:
NONE

FILES USED:
INPUT:
NONE
OUTPUT:
TAPE1 - SEQUENTIAL FILE CONTAINING SITE II INPUT DECK, EACH DESCRIBING 150 POINTS OR LESS, WITH INPUT DECKS SEPARATED BY END-OF-FILE MARKS
SCRATCH:
TAPE4 - SEQUENTIAL FILE CONTAINING CARD IMAGES OF UNPROCESSED SITE II INPUT DECKS
SUBROUTINE RESTPLT

OBJECTIVE:
SUBROUTINE RESTPLT PLOTS A LABEL INDICATING THE RANGE
OF DB VALUES WHOSE CONTOUR POINTS ARE PLOTTED; ADDITION-
ALLY, IF MORE THAN ONE CONTOUR IS PLOTTED, THE STEP SIZE
IS NOTATED ON THE PLOT

PARAMETERS:
INPUT:
FACT : THE LENGTH OF THE POSITIVE X AND Y AXES IN
INCHES
MET : NOISE METRIC IN CHARACTER FORM
RUNCNT: INTEGER VARIABLE USED TO COUNT CONTOURS
OUTPUT:
NONE

CALLING ROUTINE:
AUPHS1

SUBPROGRAMS USED:
CALPLT : LRCGOSF ROUTINE USED TO RESET ORIGIN
NOTATE : LRCGOSF ROUTINE USED TO PLOT A CHARACTER
STRING

FUNCTIONS USED:
AND : STANDARD FIN AND-ING FUNCTION
OR : STANDARD FIN OR-ING FUNCTION
SHIFT : STANDARD FIN SHIFT-ING FUNCTION

LOCAL VARIABLE DICTIONARY:
CENTER : REAL VARIABLE REPRESENTING THE CENTER X AND Y
VALUE OF THE COMPASS ROSE
ICHAR : VARIABLE USED IN THE FORMATION OF THE UPPERMOST
LDN CONTOUR VALUE PLOTTED
IREM : VARIABLE USED IN THE FORMATION OF THE UPPERMOST
LDN CONTOUR VALUE PLOTTED
ITEMP1 : VARIABLE USED IN THE FORMATION OF CHARACTER
STRING WHEN ONLY ONE CONTOUR IS PLOTTED
ITEMP2 : VARIABLE USED TO STORE THE INTEGER VALUE OF
THE UPPERMOST LDN VALUE
TOW : REAL VARIABLE REPRESENTING THE LENGTH IN INCHES
OF THE CHARACTER STRING TO BE PLOTTED
X : REAL VARIABLE REPRESENTING THE X-COORDINATE OF THE STARTING POINT OF THE CHARACTER STRING TO BE NOTATED

Y : REAL VARIABLE REPRESENTING THE Y-COORDINATE OF THE STARTING POINT OF THE CHARACTER STRING TO BE NOTATED

COMMON VARIABLES:
/PLTINFO/
  CENT : THE CENTER X AND Y VALUE OF THE COMPASS ROSE
  HT : THE HEIGHT IN INCHES OF CHARACTERS TO BE NOTATED
  ISTRING : CHARACTER STRING (2 WORDS) USED TO NOTATE RANGE OF LDN VALUES
  S : THE DISTANCE FROM THE LEADING EDGE OF A SYMBOL TO THE LEADING EDGE OF THE NEXT SYMBOL
  W : REAL VARIABLE REPRESENTING THE WIDTH OF ONE SYMBOL

FILES USED:
  INPUT:
      NONE
  OUTPUT:
      NONE
  SCRATCH:
      NONE
PROGRAM AUREPT

OBJECTIVE:
PROGRAM AUREPT CONSOLIDATES THE DEMOGRAPHIC VARIABLES FROM A SERIES OF SITE II DEMOGRAPHIC PROFILE REPORTS FOR AN AIRPORT COMMUNITY WITHIN THE 65 DB LDN NOISE CONTOUR. TEN PRINTED REPORTS ARE GENERATED; ONE REPORT FOR EACH OCTANT OF THE IMPACTED AREA, A SUMMARY OF ALL OCTANT REPORTS AND A NOISE IMPACT SUMMARY FOR THE IMPACTED AREA.

METHOD:
INPUT TO PROGRAM AUREPT CONSISTS OF A SITE II DEMOGRAPHIC PROFILE REPORT FOR EACH OF A SERIES OF POLYGONAL AREAS (PA). EACH PA REPRESENTS THE AREA ENCLOSED BY THE COORDINATES OF A CONSTANT LEVEL NOISE CONTOUR INTERSECTED WITH TWO COMPASS ROSE OCTANT LINES. THE DEMOGRAPHIC VARIABLES FROM THE PROFILE REPORTS ARE SUMMED FOR EACH OCTANT OF EACH CONTOUR. DIFFERENCES ARE CALCULATED BETWEEN EQUIVALENT OCTANTS OF ADJACENT CONTOURS SUCH THAT DEMOGRAPHIC VARIABLES ARE STORED FOR THE AREAS (BANDS) BETWEEN EACH CONTOUR FOR EACH OCTANT. THE DEMOGRAPHIC VARIABLES ARE ACCUMULATED IN A 3-DIMENSIONAL ARRAY SUCH THAT EACH PLANE REPRESENTS AN OCTANT WITH THE 9TH PLANE REPRESENTING THE SUM OF ALL OCTANTS. THE ROWS OF EACH PLANE REPRESENT THE DEMOGRAPHIC VARIABLES WHILE THE COLUMNS REPRESENT THE CONTOUR BANDS. THE AGGREGATED POPULATION DATA AS SUMMED AND DIFFERENCED FROM THE DEMOGRAPHIC DATA IS USED TO CALCULATE A NOISE IMPACT SUMMARY WHICH INCLUDES THE CALCULATION OF A LEVEL WEIGHTED POPULATION, NOISE IMPACT INDEX, EQUIVALENT NOISE LEVEL, PERCENT HIGHLY ANNOYED AND POPULATION HIGHLY ANNOYED.

SUBPROGRAMS USED:
BLDTAB : READS THE PA REPORTS, CALCULATES THE BAND DATA AND STORES IT IN THE 3-DIMENSIONAL TABLE.
CHART : OUTPUTS THE OCTANT REPORTS ON PRINTER PAPER USING THE CONTENTS OF THE 3-DIMENSIONAL TABLE.
HEADER : PERFORMS PAGE EJECT AND PRINTS PAGE HEADINGS AND TITLES FOR THE OCTANT REPORTS.
SUMMARY : CALCULATES AND PRINTS THE VARIABLES FOR THE NOISE IMPACT SUMMARY.

FUNCTIONS USED:
NONE
LOCAL VARIABLE DICTIONARY:

ALINE : 2-DIMENSIONAL ARRAY USED TO STORE AREAS
OF 65, 75, 85 DB CONTOURS.

GR2 : USER-DESIGNATED GROWTH RATE

I : USED AS A DO LOOP INDEX IN 50 LOOP AND 100 LOOP
IN THE 50 LOOP IT INDEXES THE ARRAY TABLE FOR
PRESET TO ZEROES AND IN THE 100 LOOP IT IS THE
OCTANT COUNTER.

ICASE : 60-WORD ARRAY CONTAINING CASE TITLE

IR3 : VALUE OF REGISTER 3, AS DETERMINED BY GETCON

ISIG1 : FLAG USED TO INDICATE THE ABSENCE OF THE 75 DB
(OR GREATER) CONTOUR

ISIG2 : FLAG USED TO INDICATE THE ABSENCE OF THE 65 DB
(OR GREATER) CONTOUR

IYEAR : USER-DESIGNATED YEAR FOR POPULATION UPDATES

J : USED AS A DO LOOP INDEX IN 75 LOOP. IT IS THE
COLUMN(BAND) POINTER IN THE ARRAY TABLE.

KADRS : AN 80 WORD ARRAY THAT STORES THE AIRPORT
ADDRESS IN 80R1 FORMAT.

KEMPTY : A COUNTER WHOSE VALUE DETECTS THE PRESENCE OF
AN OCTANT WITH NO POPULATION IN ANY BAND.

LDNS : A 8 WORD ARRAY THAT STORES THE ASCENDING LDN
VALUES FOR EACH CONTOUR. DEFINED IN SUBROUTINE
BLDTAB.

MINUS : INTEGER VARIABLE USED FOR LANDUSE COUNTER-
MEASURES TO DETERMINE DB INSULATION FACTOR

NAME : AN 80 WORD ARRAY THAT STORES THE AIRPORT NAME
IN 80R1 FORMAT.

NCON : THE INDEX TO THE STARTING LDN CONTOUR. DEFINED
IN SUBROUTINE BLDTAB BY SAMPLING THE JOB
CONTROL REGISTER, R2.

TABLE : THE 3-DIMENSIONAL TABLE BUILT UP BY SUBROUTINE
BLDTAB.

COMMON BLOCK CROSS-REFERENCE LIST:

NONE

COMMON VARIABLES:

NONE

A-47
FILES USED:

INPUT:

TAPE5 - SEQUENTIAL INPUT FILE CONSISTING OF ALL THE SITE II PRINTED OUTPUT. FORMATTED MODE.
TAPE1 - SEQUENTIAL INPUT FILE CONSISTING OF AIRPORT DESCRIPTIVE DATA. FORMATTED MODE.
CARDS 1 - 6: INFORMATION USED BY ALAMO PROGRAMS THAT SHARE THIS FILE
CARD 7: LATITUDE, LONGITUDE, DISPLACEMENTS.
   COL1 - COL2 = LATITUDE DEGREES
   COL4 - COL5 = LATITUDE MINUTES
   COL7 - COL8 = LATITUDE SECONDS
   COL11 - COL13 = LONGITUDE DEGREES
   COL15 - COL16 = LONGITUDE MINUTES
   COL18 - COL19 = LONGITUDE SECONDS
   COL25 = N OR S (NORTH/SOUTH)
   COL26 - COL30 = F5.2 WITH NO SIGN
   COL32 = E OR W (EAST/WEST)
   COL33 - COL37 = F5.2 WITH NO SIGN
CARD 8: 2F10.2 - X AND Y AXES TRANSLATION CONSTANTS.
CARD 9: COL1 - COL5: F5.2 - SCALE FACTOR FOR CONTOUR GRAPHIC EQUALS HALF THE LENGTH OF X AND Y AXES
   COL7 - COL10: YEAR FOR POPULATION UPDATES
   COL11 - COL16: USER DESIGNATED GROWTH RATE EQUALS HALF THE LENGTH OF X AND Y AXES.
CARD 10: AIRPORT NAME IN 8A10 FORMAT.
CARD 11: AIRPORT ADDRESS IN 8A10 FORMAT.

OUTPUT:

TAPE6 - LINE PRINTER FILE FOR ALL 10 REPORTS.
TAPE2 - LINE PRINTER FILE FOR NOISE IMPACT SUMMARY REPORT ONLY

SCRATCH:
NONE
SUBROUTINE RDREPT

OBJECTIVE:
TO FIND AND READ THE NEXT SITE II POLYGON REPORT ON TAPE5

METHOD:
RDREPT READS TAPE5, LINE BY LINE, UNTIL IT FINDS THE NEXT OCCURRENCE OF THE CHARACTER STRING " PROFILE R" STARTING IN COLUMN 41. WHEN THE STRING IS FOUND RDREPT WILL BE POSITIONED AT THE TOP OF A SITE II DEMOGRAPHIC PROFILE REPORT. THE DEMOGRAPHIC VARIABLES FOR THE NEXT POLYGON ARE THEN READ THROUGH A SERIES OF FORMATTED READS.

ASSUMPTIONS:
RDREPT ASSUMES THAT THE SITE II REPORTS ARE IN THE PROPER ORDER FOR LATER PROCESSING. IT MAKES NO ATTEMPT TO RECOGNIZE EITHER THE OCTANT NUMBER OR THE LDN VALUE.

PARAMETERS:
INPUT:
GR2 : USER-DESIGNATED GROWTH RATE
IYEAR : USER-DESIGNATED YEAR OF POPULATION UPDATE

OUTPUT:
IR3 : VALUE OF REG. 3 AS DETERMINED BY GETCON
ISIG1 : FLAG USED TO INDICATE ABSENCE OF 75 DB (OR GREATER) CONTOUR
ISIG2 : FLAG USED TO INDICATE ABSENCE OF 65 DB (OR GREATER) CONTOUR

CALLING ROUTINE:
BLDTAB

SUBPROGRAMS USED:
GETCON : COMPASS ROUTINE THAT SAMPLES REGISTERS

FUNCTIONS USED:
EOF: END OF FILE CHECK ON TAPE5.

LOCAL VARIABLE DICTIONARY:
I : USED AS INDEX VARIABLE IN IMPLIED DO I/O LISTS.
IR1 : INTEGER VARIABLE REPRESENTING VALUE OF REGISTER 1
IR2 : INTEGER VARIABLE REPRESENTING VALUE OF REGISTER 2
IR3 : INTEGER VARIABLE REPRESENTING VALUE OF REGISTER 3
KEY : STORES 10 CHARACTER STRING WHICH UNIQUELY IDENTIFIES A SITE II DEMOGRAPHIC PROFILE REPORT
COMMON VARIABLES:

ACHH: THE NUMBER OF HOUSEHOLDS WITH AIR CONDITIONING.

AREA: THE AREA IN SQUARE MILES OF THE POLYGON BEING READ.

COL16P: NOT USED

EDUC: 5 WORD VECTOR STORING THE EDUCATED ADULTS GREATER THAN 25 COLUMN INFO AS FOLLOWS:
- EL1 0 - 8 GRADES
- EL2 9 - 11 GRADES
- EL3 12TH GRADE
- EL4 13 - 15 GRADES
- EL5 16+ GRADES

FAMPOP: THE NUMBER OF PEOPLE WITH FAMILY STATUS.

FEM: 6 WORD VECTOR STORING THE NUMBER OF WOMEN IN THE FOLLOWING AGE CATEGORIES:
- EL1 18 - 20
- EL2 21 - 29
- EL3 30 - 39
- EL4 40 - 49
- EL5 50 - 64
- EL6 65 +

FMIN: 7 WORD VECTOR STORING THE NUMBER OF PEOPLE IN THE FOLLOWING FAMILY INCOME BRACKETS:
- EL1 $0 - 5 (THOUSANDS)
- EL2 $5 - 7
- EL3 $7 - 10
- EL4 $10 - 15
- EL5 $15 - 25
- EL6 $25 - 50
- EL7 $50 +

HOMVAL: 7 WORD VECTOR STORING THE NUMBER OF PEOPLE IN THE FOLLOWING HOME VALUE BRACKETS:
- EL1 $0 - 10 (THOUSANDS)
- EL2 $10 - 15
- EL3 $15 - 20
- EL4 $20 - 25
- EL5 $25 - 35
- EL6 $35 - 50
- EL7 $50 +

IOCT: THE OCTANT NUMBER OF THE POLYGON BEING READ.

KEOF: END OF FILE FLAG - SET TO 1 TO INDICATE AN EOF WAS ENCOUNTERED. SENSED BY THE CALLING ROUTINE.

LINE: AN 8 WORD VECTOR FOR STORING 80 CHARACTER LINE IMAGES OF THE SITE II REPORTS.

MALE: 6 WORD VECTOR STORING MALE AGE INFO AS DESCRIBED UNDER VARIABLE FEM.
MGRPRO : 9 WORD VECTOR STORING THE NUMBER OF PEOPLE
IN THE FOLLOWING OCCUPATIONAL CATEGORIES:
EL1 MANAGERIAL/PROFESSIONAL
EL2 SALES
EL3 CLERICAL
EL4 CRAFT
EL5 OPERATIVES
EL6 LABORER
EL7 FARM
EL8 SERVICE
EL9 PRIVATE

NOHH : THE NUMBER OF HOUSEHOLDS.
POP70 : THE 1970 CENSUS POPULATION FOR THIS POLY.
POP81 : THE 1981 CENSUS POPULATION FOR THIS POLY.
SNGDWL : THE NUMBER OF PEOPLE IN SINGLE FAMILY
Dwellings.
TOTOWN : THE TOTAL NUMBER OF HOMEOWNERS.
TOTRNT : NOT USED.
TVHH : THE NUMBER OF HOUSEHOLDS WITH TELEVISION.
XLON : THE NOISE CONTOUR METRIC (DBLON) FOR THIS
POLYGON.
ZERO : A FLAG SET TO 1 TO INDICATE A ZERO
POPULATION POLYGON.

FILES USED:
INPUT:
TAP5 - SEQUENTIAL INPUT FILE CONTAINING ALL OF THE
SITE II OUTPUT FOR AN ALAMO CASE RUN. THIS
CONSISTS OF A REPTITION OF THE FOLLOWING:
1. SITE II REFLECTION OF A,B,C AND D CARDS
   FOR A PARTICULAR RUN.
2. REFLECTION OF THE SORT/MERGE DIRECTIVES
   USED BY SITE II.
3. SITE II DEMOGRAPHIC PROFILE REPORTS FOR
   INPUT DESCRIBED IN 1 ABOVE.
SUBROUTINE BLDTAB

OBJECTIVE:

To build a 3-dimensional table which stores the values of the 16 demographic variables to be reported for each octant of each LDN band.

METHOD:

BLDTAB calls subroutine RDEPT to read each successive polygon report. RDEPT defines and redefines the demographic variables in named common block, COMM. BLDTAB accumulates the values for all polygons of each octant for each LDN value and stores the accumulations in local variables. When all polygons have been read and values are locally stored, BLDTAB begins to build the 3-dimensional table by finding the differences of the accumulated values of the demographic variables between LDN contours. This is done for each octant. The row dimension of the table corresponds to the demographic variables; the column dimension of the table corresponds to the LDN contour bands and the plane dimension corresponds to the 8 compass rose octants. The 9th plane represents a summary of all the octants. The capacity of the table accounts for the following bands: 65-70, 70-75, 75-80, 80-85 and over 85. It is possible that low end and/or top end LDN contours will be missing. In any case, the bands will be stored in the upper dimension of the columns of the table. This means that column 5 will always be the "over xx" band where xx corresponds to the highest LDN contour found by the integrated noise model. BLDTAB samples the job control register, R2 for the starting column index to account for missing contours. R2 is set by a separate program which defines the polygon areas and therefore knows the LDN range of values.

RESTRICTIONS AND ASSUMPTIONS:

BLDTAB assumes that:

1. Register R2 has been set previous to program startup.
2. The polygon reports are in ascending order of LDN value.
PARAMETERS:

INPUT:
GR2 : USER-DESIGNATED GROWTH RATE
IYEAR : USER-DESIGNATED YEAR OF POPULATION UPDATE

OUTPUT:
ALINE : 2-DIMENSIONAL ARRAY CONTAINING AREAS OF
        65, 75, 85 DB CONTOURS.
IR3 : VALUE OF REG. 3 AS DETERMINED BY GETCON
ISIG1 : FLAG USED TO INDICATE ABSENCE OF 75 DB
        (OR GREATER) CONTOUR
ISIG2 : FLAG USED TO INDICATE ABSENCE OF 65 DB
        (OR GREATER) CONTOUR

TABLE : THE 3 DIMENSIONAL TABLE - TABLE(16, 5, 9)
        EACH PLANE REPRESENTS AN OCTANT EXCEPT THE
        9TH PLANE WHICH REPRESENTS THE SUMMARY OF
        ALL THE OCTANTS. THE 5 COLUMNS OF EACH
        PLANE CORRESPOND TO THE CONTOUR BANDS AS
        FOLLOWS:
        65-70 70-75 75-80 80-85 85+
        EACH ROW CORRESPONDS TO A DEMOGRAPHIC
        VARIABLE AS FOLLOWS:
        1. SELF-NOISE (AMBIENT NOISE IN DBLDN)
        2. IMPACTED POPULATION
        3. AREA IN SQUARE KILOMETERS
        4. TOTAL POPULATION PER SQUARE KILOMETER
        5. AVERAGE GROWTH RATE, APR
        6. PERCENT FAMILY POPULATION
        7. AVERAGE AGE, ADULTS OVER 17
        8. PERCENT AGE OVER 65
        9. PERCENT OF PEOPLE WITH 16+ YEARS EDUC.
        10. PERCENT MANAGER/PROFESSIONALS
        11. AVERAGE FAMILY INCOME
        12. PERCENT SINGLE FAMILY DWELLINGS
        13. PERCENT HOME OWNERS
        14. AVERAGE HOME VALUE
        15. PERCENT HOUSEHOLDS WITH AIRCONDITIONER
        16. PERCENT HOUSEHOLDS WITH TV

LDNS : A 6 WORD ARRAY WHICH STORES THE 5
POSSIBLE LDN CONTOUR VALUES INTENDED TO BE
65 THROUGH 85 IN 5DB INTERVALS. THE 6TH
WORD STORES THE MAXIMUM LDN CONTOUR FOUND
WHICH WILL BE THE CONTENTS OF WORD 5.

ICON : STORAGE FOR THE VALUE OF JOB CONTROL
REGISTER, R2. DEFINED BY COMPASS
SUBROUTINE GETCON.
CALLING ROUTINE:
AUREPT

SUBPROGRAMS USED:
GETCON : READS JOB CONTROL REGISTERS.
RDEPT  : READS THE NEXT POLYGON REPORT ON THE INPUT FILE

FUNCTIONS USED:
ALOG10 : CALCULATES THE LOGARITHM, BASE 10

LOCAL VARIABLE DICTIONARY:
AC    : AN 8X6 MATRIX WHOSE ROWS CORRESPOND TO THE 8 OCTANTS AND WHOSE COLUMNS CORRESPOND TO THE 5 POSSIBLE LDN VALUES. COLUMN 6 WILL ALWAYS BE ZEROES TO ALLOW FOR THE CALCULATION OF OVER 85 VALUES. AC ACCUMULATES THE AIR CONDITIONER DATA.
ALLGR : REAL ARRAY USED TO STORE GROWTH RATES OF THE OCTANTS OF THE CONTOURS
AVGAGE : A 3-DIMENSIONAL ARRAY WHOSE ROWS CORRESPOND TO THE 6 AGE CATEGORIES; COLUMNS CORRESPOND TO THE 8 OCTANTS AND PLANES CORRESPOND TO THE LDN VALUES.
AVGFIN : A 3-DIMENSIONAL ARRAY WHOSE ROWS CORRESPOND TO THE 7 FAMILY INCOME CATEGORIES; COLUMNS CORRESPOND TO THE 8 OCTANTS AND PLANES CORRESPOND TO THE LDN VALUES.
AVGHMV : A 3-DIMENSIONAL ARRAY WHOSE ROWS CORRESPOND TO THE 7 HOME VALUE CATEGORIES; COLUMNS CORRESPOND TO THE 8 OCTANTS AND PLANES CORRESPOND TO THE LDN VALUES.
BANDGRO : REAL ARRAY REPRESENTING BAND GROWTH RATES WITH THE FINAL 8 OCTANT VARIABLES REPRESENTING THE INNER CONTOUR (8 OCTANTS, 5 BANDS, WHERE LAST BAND IS "OVER XX"
DIF    : TEMPORARY VARIABLE USED TO ACCUMULATE DIFFERENCES THAT ARE MULTIPLIED BY A FACTOR.
DIFHHS : TEMPORY USED FOR DIFFERENCING THE NO. OF HOUSEHOLDS.
DIF1: TEMPORARY FOR DIFFERENCING THE AVG. HOME VALUES
DIF2: TEMPORARY FOR DIFFERENCING THE YRS EDUCATION.
DIF3: TEMPORARY FOR DIFFERENCING THE OCCUPATIONS.
DIF4: TEMPORARY FOR DIFFERENCING THE YRS EDUCATION OVER 16.
DIF5: TEMPORARY FOR DIFFERENCING THE MANAGER/PROF.
DIF6: TEMPORARY FOR DIFFERENCING THE SINGLE DWELLINGS
DIF7: TEMPORARY FOR DIFFERENCING THE AC UNITS.
DIF8: TEMPORARY FOR DIFFERENCING THE TVS.
DENSITY: TEMPORARY FOR COMPUTING POPULATION DENSITY.
DWL1: ACCUMULATION ARRAY FOR SINGLE DWELLING DATA;
ROWS ARE OCTANTS, COLUMNS ARE LDN VALUES.
ED16: 3-DIMENSIONAL ARRAY WHICH ACCUMULATES THE
EDUCATIONAL CATEGORIES; ROW 1 IS 16YRS+ AND
ROW 2 IS ALL CATEGORIES, COLUMNS ARE
OCTANTS, PLANES ARE LDN VALUES.
FACT: ARRAY OF MULTIPLYING FACTORS USED FOR COMPUTING
THE AVERAGE AGE DATA; FOR EXAMPLE, THE 1ST
ELEMENT IS 19 WHICH CORRESPONDS TO THE AVERAGE
AGE OF THE 1ST AGE CATEGORY (18-20) BEING
SUMMED FOR AVERAGING.
FACTHM: SIMILAR TO FACT ABOVE BUT USED FOR AVERAGING
THE HOME VALUES.
FACTIN: SIMILAR TO FACT ABOVE BUT USED FOR AVERAGING
THE FAMILY INCOMES.
FPOP: TEMPORARY USED FOR DIFFERENCING THE FAMILY
POPULATION.
GRATE: REAL VARIABLE REPRESENTING THE GROWTH RATE,
EITHER EXTRACTED FROM ARRAY BANDGro, OR
USER-DESIGNATED; THIS VARIABLE IS THE GROWTH
RATE USED TO PROJECT A CORRESPONDING POPULATION
GROCT: REAL VARIABLE REPRESENTING THE GROWTH RATE FOR
A PARTICULAR OCTANT
GROTH: REAL ARRAY USED TO STORE THE GROWTH RATES FOR
AN OCTANT WITH MULTIPLE POLYGONS
GR9: REAL VARIABLE, REPRESENTING THE GROWTH RATE FOR
THE "ALL OCTANTS" CASE, COMPUTED USING THE ORIGINAL
CUMULATIVE POPULATION AND THE GROWN CUMULATIVE
POPULATION
HHS: TEMPORARY USED FOR ACCUMULATING THE HOUSEHOLD
DATA; ROWS ARE OCTANTS, COLUMNS ARE LDN VALUES.
HMVALT: TEMPORARY USE FOR TOTALING THE HOME VALUES.
I: GENERAL PURPOSE DO LOOP INDEX.
IALLPOP: INTEGER ARRAY USED TO STORE POPULATION VALUES OF
THE OCTANTS OF THE CONTOURS, NEEDED TO COMPUTE
BAND GROWTH RATES
ICT: COUNTER USED AS INDEX TO ACCUMULATE GROWTH RATES
AND AND POPULATIONS IN SINGLE OCTANT
IDIF: INTEGER VARIABLE USED FOR TEMPORARY STORAGE TO
REPRESENT THE NUMBER OF PEOPLE FOR A PARTICULAR
OCTANT AND A PARTICULAR CONTOUR BAND
IGRPOP : INTEGER ARRAY USED TO STORE THE POPULATIONS OF AN OCTANT WITH MULTIPLE POLYGONS.

IOCT2 : INTEGER VARIABLE USED TO IDENTIFY OCTANT, THUS CAN ACCOUNT FOR MULTIPLE POLYGONS IN 1 OCTANT.

IOCTSUM: INTEGER VARIABLE REPRESENTING THE TOTAL VALUE OF THE POPULATION IN A GIVEN OCTANT.

ITEMP : INTEGER VARIABLE USED TO INDEX INTO ARRAY IALLPOP TO FIND POPULATION OF INNER CONTOUR.

J : DO LOOP INDEX FOR 600 LOOP (OCTANT INDEX).

JJ : INDEX USED INTO ARRAY ALINE, USED TO STORE AREA INFORMATION.

JJJ : INDEX INTO ARRAY ALINE, USED TO STORE AREA INFORMATION.

JPTR : LDN VALUE INDEX.

K : GENERAL PURPOSE DO LOOP INDEX.

KK : INDEX INTO ARRAY POPDEN, USED TO EXTRACT AREA INFORMATION.

KDUM1 : DUMMY PARAMETER FOR GETCON ROUTINE.

KDUM2 : DUMMY PARAMETER FOR GETCON ROUTINE.

LDN : INTEGER REPRESENTATION FOR THE CURRENT LDN VALUE.

LOC : INTEGER VARIABLE USED TO LOCATE STORAGE OF CONTOUR AREAS, ON AN OCTANT BY OCTANT BASIS.

N : INTEGER VARIABLE REPRESENTING THE NUMBER OF YEARS OF POPULATION UPDATE.

POP : TEMPORARY USED FOR DIFFERENCING THE POPULATION DENSITIES.

POPDEN : 3-DIMENSIONAL ARRAY USED FOR ACCUMULATING THE 1981 POPULATIONS AND CORRESPONDING AREAS; COL 1 1981 POPULATION, COL 2 IS AREA, ROWS ARE OCTANTS AND COLUMNS ARE LDNS.

POPIMP : ARRAY FOR ACCUMULATING THE IMPACTED POPULATIONS ROWS ARE OCTANTS AND COLUMNS ARE LDNS.

POPO : REAL VARIABLE USED FOR DIFFERENCING POPULATION VALUES.

POP1 : REAL VARIABLE USED TO STORE THE POPULATION OF A POLYGON WITHIN AN OCTANT OF A CONTOUR.

PRATIO : TEMPORARY FOR CALCULATING THE POPULATION RATIOS.

SAREA : TEMPORARY FOR SUM OF AREAS.

SHHS : TEMPORARY FOR SUM OF HOUSEHOLDS.

SHMV : TEMPORARY FOR SUM OF HOME VALUES.

SPOP : TEMPORARY USED FOR SUMMING POPULATIONS.

SPOPO : REAL VARIABLE USED TO STORE SUMMED POPULATIONS; CONTAINS THE SUMMED POPULATIONS WITHOUT APPLICATION OF THE GROWTH RATE.

SPOP70 : TEMPORARY USED FOR SUMMING 1970 POPULATIONS.

STIN : TEMPORARY USED FOR SUMMING TOTAL INCOME.
COMMON VARIABLES:

\texttt{/COMM/}

SEE SUBROUTINE \texttt{RDREPT} COMMON BLOCK DESCRIPTION.
SUBROUTINE HEADER

OBJECTIVE:
  TO PRINT OCTANT REPORT HEADERS

PARAMETERS:
  INPUT:
    KOCT : THE OCTANT NUMBER OF THE OCTANT BEING PRINTED. (1 - 8)
    NAME : THE AIRPORT NAME FOR THE HEADING (80 R1)
    KADRS : THE AIRPORT ADDRESS (80 R1)
  OUTPUT:
    ICASE : 60-WORD ARRAY CONTAINING CASE NAME
    MINUS : INTEGER VARIABLE, NON-ZERO FOR LANDUSE INSULATION COUNTERMEASURES, THAT REPRESENTS THE DB DECREMENT FOR THE 65 LDN (AND GREATER) VALUE

CALLING ROUTINE:
  AUREPT

SUBPROGRAMS USED:
  GETCON : COMPASS SUBROUTINE THAT SAMPLES REGISTER VALUES

FUNCTIONS USED:
  NONE

LOCAL VARIABLE DICTIONARY:
  IOCT : 3 LETTER MNEUMONIC FOR OCTANT AS FOLLOWS:
         1 = ESE  2 = SSE  3 = SSW  4 = WSW  5 = WNW
         6 = NW  7 = NNE  8 = ENE
  IR1 : INTEGER VALUE OF REGISTER 1
  IR2 : INTEGER VALUE OF REGISTER 2
  IR3 : INTEGER VALUE OF REGISTER 3
  K : GENERAL PURPOSE IMPLIED DO LOOP INDEX.

COMMON VARIABLES:
  NONE

FILES USED:
  INPUT:
    NONE
  OUTPUT:
    TAPE6 - LINE PRINTER FILE WITH CARRIAGE CONTROL

A-58
SUBROUTINE CHART

OBJECTIVE:
TO PRINT THE 16 DEMOGRAPHIC VARIABLES BY LDN BAND VALUES
IN REPORT FORM USING THE LINE PRINTER.

PARAMETERS:
INPUT:
TABLE: THE 3-DIMENSIONAL TABLE STORING THE
DEMOGRAPHIC VARIABLES TO BE PRINTED.
LDNS: A VECTOR STORING THE LDN VALUES IN
ASCENDING ORDER.
N: THE OCTANT INDEX TO TABLE
NCON: STARTING INDEX TO THE LDN TABLE. IF NCON
IS GREATER THAN ONE THEN 1 OR MORE LDN
CONTOURS WERE MISSING.
IYEAR: USER-DESIGNATED YEAR OF POPULATION UPDATE
GR2: USER-DESIGNATED GROWTH RATE

CALLING ROUTINE:
AUREPT

SUBPROGRAMS USED:
BREAK: BREAK BREAKS OUT 5 COLUMNS OUT OF THE
3-DIMENSIONAL TABLE TO A VECTOR CALLED LINE.
CHART1: ALTERNATIVE OCTANT PRINTING ROUTINE USED WHEN
LDN CONTOURS ARE MISSING (NCON GT 1)

FUNCTIONS USED:
NONE

LOCAL VARIABLE DICTIONARY:
I: IMPLIED DO LOOP INDEX.
J: IMPLIED DO LOOP INDEX.
K: DO LOOP INDEX FOR REPLACING VERY SMALL AREA
QUANTITIES WITH A "-".
LINE: GENERAL PURPOSE 5 WORD VECTOR USED TO ENCODE
BAND VALUES FOR CONVENIENT PRINTING.

COMMON VARIABLES:
NONE

FILES USED:
OUTPUT:
TAPE6: SEQUENTIAL LINE PRINTER FILE FOR OCTANT
REPORT.
SUBROUTINE CHART1

OBJECTIVE:
TO PRINT OCTANT REPORTS TO THE PRINTER AS AN ALTERNATIVE
TO SUBROUTINE CHART WHEN LDN CONTOUR ARE MISSING.

PARAMETERS:
INPUT:
TABLE : 3-DIMENSIONAL TABLE STORING THE
         DEMOGRAPHIC VARIABLES.
LDNS  : 6 WORD VECTOR STORING THE RANGE OF LDN
         CONTOUR VALUES.
N     : THE OCTANT NUMBER TO BE PRINTED.
NCON  : THE STARTING INDEX IN THE TABLE OF LDNS
         WHICH COMPENSATES FOR MISSING CONTOURS.
IYEAR : USER-DESIGNATED YEAR OF POPULATION UPDATES
GR2   : USER-DESIGNATED GROWTH RATE

OUTPUT:
NONE

CALLING ROUTINE:
CHART

SUBPROGRAMS USED:
BREAK : BREAK BREAKS OUT 5 COLUMNS OUT OF THE
        3-DIMENSIONAL TABLE TO A VECTOR CALLED LINE.

FUNCTIONS USED:
NONE
LOCAL VARIABLE DICTIONARY:

I : IMPLIED DO LOOP INDEX.
INDEX : ARRAY INDEX TO LDN TABLE USED TO COMPUTE MISSING CONTOUR VALUES AT LOW END OF CONTOUR RANGE.
INT : THE INTERVAL BETWEEN CONTOUR VALUES; EXPECTED VALUE IS 5.
J : IMPLIED DO LOOP INDEX.
JCON : THE NUMBER OF CONTOUR BANDS THAT WERE FOUND.
K : DO LOOP INDEX USED TO CHECK FOR SMALL AREAS.
KCON : THE NUMBER OF CONTOUR BANDS THAT WERE NOT FOUND.
LINE : GENERAL PURPOSE 5 WORD VECTOR USED TO ENCODE BAND VALUES FOR CONVENIENT PRINTING.

COMMON VARIABLES:

NONE

FILES USED:

INPUT:
NONE
OUTPUT:
SEQUENTIAL LINE PRINTER FILE FOR OCTANT REPORT.
SCRATCH:
NONE
SUBROUTINE BREAK

OBJECTIVE:
BREAK CREATES A VECTOR FROM A SELECTED ROW OF THE
3-DIMENSIONAL TABLE.

PARAMETERS:
INPUT:
   TABLE : THE 3-DIMENSIONAL TABLE OF DEMOGRAPHIC
           VARIABLES.
   KROW : THE SELECTED ROW TO BE VECTORIZED.
   KPLANE : THE SELECTED PLANE WHERE THE SELECTED ROW
            RESIDES.
OUTPUT:
   LINE : THE VECTOR REMOVED FROM THE 3-DIMENSIONAL
           TABLE.

CALLING ROUTINE:
   CHART, CHART1

SUBPROGRAMS USED:
   NONE

FUNCTIONS USED:
   NONE

LOCAL VARIABLE DICTIONARY:
   I : DO LOOP INDEX USED FOR COLUMNS OF TABLE.

COMMON VARIABLES:
   NONE

FILES USED:
   NONE
SUBROUTINE SUMMARY

OBJECTIVE:
SUMMARY COMPUTES AND PRINTS A NOISE IMPACT SUMMARY REPORT BASED ON THE POPULATION DATA AND THE LDN VALUES.

METHOD:
SIX VALUES ARE CALCULATED FOR EACH OCTANT AND A TOTAL FOR EACH VALUE IS COMPUTED FOR ALL OCTANTS. THE VALUES ARE NAMED AND COMPUTED AS FOLLOWS:

IMPACTED POPULATION   THE SUM OF THE 1981 POPULATIONS FOR ALL BANDS OF AN OCTANT.
LEVEL WEIGHTED POP.    THE POPULATION EXPOSED TO A CERTAIN LDN BAND IS WEIGHTED.
                       THE FOLLOWING FORMULA IS USED TO COMPUTE THE WEIGHT CORRESPONDING TO A GIVEN LDN VALUE:
                       
                       \[ 0.103 \cdot \text{LDN} + 0.000124(10^\text{LDN}) \]

NOISE IMPACT INDEX    THE LEVEL WEIGHTED POPULATION IS THE SUM OF THE PRODUCTS OF THE POPULATIONS AND THEIR WEIGHS.
                       THE LEVEL WEIGHTED POPULATION DIVIDED BY THE TOTAL POPULATION.
                       A TABLE OF WEIGHTS IS CALCULATED FOR EACH LDN VALUE USING THE FORMULA GIVEN ABOVE. THE LDN VALUES ARE THE BAND AVERAGES; EG 65 - 70 HAS VALUE 67. THE ENL IS OBTAINED BY INTERPOLATING THE NII IN THE TABLE OF WEIGHTS. ONCE A WEIGHT IS KNOWN A CORRESPONDING LDN CAN BE INTERPOLATED. THIS LDN IS THE ENL.
EQUIVALENT NOISE LEVEL

POP. HIGHLY ANNOYED     THE PRODUCT OF THE LWP AND THE CONSTANT .369
PARAMETERS:

INPUT:
ALINE : 2-DIMENSIONAL ARRAY CONTAINING AREAS OF 65, 75, 85 DB CONTOURS.
ICASE : 60-WORD ARRAY CONTAINING CASE TITLE
MINUS : INTEGER VALUE, NON-ZERO FOR LANDUSE INSULATION COUNTERMEASURES, REPRESENTING FOR THOSE CASES, THE DB DECREMENT FOR LDN VALUES GREATER THAN OR EQUAL TO 65 LDN
TABLE : THE 3-DIMENSIONAL TABLE USED HERE TO SUPPLY THE POPULATION DATA.
LDNS : 8 WORD VECTOR CONTAINING THE CONTOUR LDN VALUES.
LINE1 : 80 WORD ARRAY FOR STORAGE OF THE AIRPORT NAME IN 80 R1 FORMAT.
LINE2 : 80 WORD ARRAY FOR STORAGE OF THE AIRPORT ADDRESS IN 80 R1 FORMAT.

OUTPUT:
NONE

CALLING ROUTINE:
AUREPT

SUBPROGRAMS USED:
NONE

FUNCTIONS USED:
INTERP : USED TO FIND A WEIGHTING FACTOR GIVEN AN NII; THEN INTERPOLATES AN LDN FROM THE WEIGHT.
PERCNT : USED TO CALCULATE THE WEIGHTS USING THE FORMULA DESCRIBED ABOVE.
LOCAL VARIABLE DICTIONARY:

DBVAL : INTEGER VARIABLE USED TO REPRESENT CONTOUR VALUE WHEN PRINTING OUT AREA OF CONTOUR.
ENL : A 9 WORD VECTOR USED TO STORE THE COMPUTED EQUIVALENT NOISE LEVEL FOR PRINTING BY OCTANT AND TOTAL OF ALL OCTANTS.
I : GENERAL PURPOSE DO LOOP INDEX.
IDASH : WORD USED TO STORE THE SYMBOL "-", TO BE USED AS SUBSTITUTE VALUE FOR NII, ENL, PCTHA, AND POPHA WHEN A CORRESPONDING BASELINE CASE POPULATION VALUE EQUALS ZERO (UTILIZED ONLY IN COUNTERMEASURE CASES).
IR1 : VALUE OF REGISTER 1
IR2 : VALUE OF REGISTER 2
IR3 : VALUE OF REGISTER 3
J : DO LOOP INDEX FOR OCTANTS.
JJ : INDEX USED FOR DECODING NII, ENL, PCTHA, AND POPHA
K : INDIRECT DO LOOP INDEX BASED ON J TO REVERSE INDEX THE ARRAYS.
LDNTAB : A 5 WORD VECTOR USED TO STORE THE LDN BAND AVERAGE VALUES; 67,72,77,82,85 THE LAST VALUE IS FOR OVER 85 AREA.
LINE : 9-WORD ARRAY USED IN PRINTING OF NII, ENL, PCTHA, AND POPHA
LWP : 9 WORD VECTOR USED TO STORE THE COMPUTED LEVEL WEIGHTED POPULATIONS BY OCTANT AND TOTAL.
NII : 9 WORD VECTOR USED TO STORE THE COMPUTED NOISE IMPACT INDEX BY OCTANT AND TOTAL.
PCTHA : 9 WORD VECTOR USED TO STORE THE PERCENT HIGHLY ANNOYED BY OCTANT AND TOTAL.
POPHA : 9 WORD VECTOR USED TO STORE THE POPULATION HIGHLY ANNOYED BY OCTANT AND TOTAL.
TOTPOP : USED TO SUM THE RUNNING TOTAL BY OCTANTS.
WTS : A 5 WORD VECTOR USED TO STORE THE WEIGHTING FACTORS FOR THE LDN BANDS.
XLDN : USED FOR FLOATING PT. STORAGE OF THE INTEGER VALUES IN THE ARRAY LDNTAB.

COMMON VARIABLES:
NONE

FILES USED:

INPUT:
NONE

OUTPUT:
TAPE6 - SEQUENTIAL - LINE PRINTER FILE FOR SUMMARY REPORT.
FUNCTION INTERP

OBJECTIVE:
To return an equivalent noise level value in LDN form given a noise impact index.

METHOD:
INTERP uses the given noise impact index to find an entry in the weights table that is greater than the noise impact index. The weight table stores the weights in ascending order corresponding to the LDN band average values. INTERP then computes the proportion of the noise impact index between the weight entries in the table. This proportion is then used to compute the appropriate LDN which is proportionately located between the two corresponding LDN entries in the LDN table. If the noise impact index is less than the first entry in the weight table, the floor values for weights and LDNs are zeroes. If no entry in the weights table is greater than the noise impact index then INTERP returns the greatest LDN value in the LDN table as the equivalent noise level.

PARAMETERS:
INPUT:
LDNS : A 6 word vector storing the contour LDN values in ascending order.
WIS : A 6 word vector of weighting factors corresponding to the elements of the vector LDNS.
NII : The given noise impact index for which an equivalent noise level is to be returned.

OUTPUT:
NONE

CALLING ROUTINE:
SUMARY

SUBPROGRAMS USED:
NONE

FUNCTIONS USED:
FLOAT : Inline function for integer to real conversion.
LOCAL VARIABLE DICTIONARY:

- **BOT**: The floor value of the two weights values that enclose the noise impact index.
- **I**: Do loop index for weights and LDNS tables.
- **INTERP**: The function variable used to return the ENL.
- **XBOT**: The floor value of the two LDN values that enclose the ENL.
- **XLDNS**: The ceiling value of the two LDN values that enclose the ENL.
SUBROUTINE LASTPAG

OBJECTIVE: TO WRITE A SEPARATE REPORT CONSISTING OF THE NOISE IMPACT SUMMARY REPORT

PARAMETERS: NONE

CALLING ROUTINE: AUREPT

SUBPROGRAMS USED: NONE

LOCAL VARIABLE DICTIONARY:
   IPAGE : INTEGER ARRAY USED TO READ IN AND WRITE OUT REPORT
FUNCTION PERCNT

OBJECTIVE:
TO CALCULATE A WEIGHTING FACTOR GIVEN AN LDN VALUE.

METHOD:
PERCENT IMPLEMENTS THE FOLLOWING FORMULA WHICH IS THE CURVE OF PERCENT OF POPULATION "HIGHLY ANNOYED" AS A FUNCTION OF NOISE LEVEL IN LDN. THE PERCENT HIGHLY ANNOYED CURVE IS NORMALIZED TO UNITY AT 75LDN BY DIVIDING THE PERCENT HIGHLY ANNOYED BY 36.9, THE PERCENT OF POPULATION HIGHLY ANNOYED AT 75LDN.

\[
\text{PERCENT HIGHLY ANNOYED} = \frac{0.000124(10^{0.103\text{LDN}})}{0.03\text{LDN} + 0.08\text{LDN}.2(10^{0.03\text{LDN}}) + 0.00143(10^{0.08\text{LDN}})}
\]

PARAMETERS:
INPUT:
LDN : THE LDN VALUE USED TO CALCULATE A WEIGHT.
OUTPUT:
NONE

CALLING ROUTINE:
SUMMARY

SUBPROGRAMS USED:
NONE

FUNCTIONS USED:
NONE

LOCAL VARIABLE DICTIONARY:
E1 : TEMPORARY FOR VALUE OF .103*LDN
E2 : TEMPORARY FOR VALUE OF .03*LDN
E3 : TEMPORARY FOR VALUE OF .08*LDN
PERCENT : THE FUNCTION VARIABLE USED TO RETURN THE COMPUTED WEIGHT.
APPENDIX B

SITE II DATA BASE INSTALLATION
Four tapes were delivered by CACI, Inc. containing the 1981 SITE II database. These tapes contain data reflecting the 1970 census with 1980 to 1981 updates. The population data was contained on tapes NASA13, NASA14, and NASA15. The zip code data was contained on tape NASA00. NASA13 contained 15 PRU's or blocks of data each block 4424 characters in length, NASA14 and NASA15 contained 18 blocks each 4424 characters in length. The setup programs SETUP15 and SETUP18 were used to read the tape files according to the above mentioned specifications and write an intermediate file TAPE7 to be processed by the conversion program CONMN. The data from these tapes were converted from IBM character format to CDC CYBER binary data and written to the NOS labelled tapes ND0775 and ND1062.

Figures 16, 17 and 18 contain the job stream of control statements used to convert the IBM formatted population data and generate the NOS tapes. Figures 19, 20 and 21 illustrate the source for SETUP15, SETUP18 and CONMN, respectively. Figure 22 illustrates the job stream used to read the tape NASA00, convert the data using program CONZP and generate the new labelled tape ND1226. Figure 23 contains the source code for program CONZP.
CONMN, T300, CM60000.
USER.
CHARGE.
FTN, I=SETUP15.
FILE (TAPE5, RT=F, BT=K, FL=4424, MBL=4424, RB=1)
VSN, TAPE5=NASA13.
REQUEST, TAPE5, F=S, NT, D=1600, CV=EB, PO=R, LB=KU, NONLRC=NASA13
LDSET, FILES=TAPE5.
LGO.
RETURN, TAPE5.
REWIND, TAPE7, LGO.
GET, CONMN.
FTN, I=CONMN.
LOAD(LGO)
NOCO(CONMS)
VSN, TAPE1=ND0775.
LABEL, TAPE1, FI=AL, SI=MN1, QN=1, W, PO=W.
CONMS.
REWIND, TAPE3.
COPYBF, TAPE8, TAPE1.
RETURN, TAPE8.
LABEL, TAPE1, FI=AK, SI=MN1, QN=9999.
CONMS.
REWIND, TAPE8.
COPYBF, TAPE8, TAPE1.
RETURN, TAPE8.
LABEL, TAPE1, FI=IN, SI=MN1, QN=9999.
CONMS.
REWIND, TAPE3.
COPYBF, TAPE3, TAPE1.
RETURN, TAPE3.

Figure 16.-Step 1 job for the SITE II database conversion.
CONM2, T300, CM60000.
USER.
CHARGE.
GET, SETUP18.
FTN, I=SETUP18.
FILE(TAPE5, RT=F, BT=K, FL=4424, MBL=4424, RB=1)
VSN, TAPE5=NASA14.
REQUEST(TAPE5, F=S, NT, D=1600, LB=KU, CV=EB, PO=R) NONLRC=NASA14
LDSET(FILES=TAPE5)
LGO.
RETURN, TAPE5.
REWIND, TAPE7, LGO.
GET, CONMN.
FTN, I=CONMN.
LOAD(LGO)
NOGO(CONMS)
VSN(TAPE1=ND0775)
LABEL(TAPE1, FI=IA, SI=MN1, QN=9999, PO=W)
CONMS.
REWIND, TAPE8.
COPYBF, TAPE8, TAPE1.
RETURN, TAPE8.
LABEL(TAPE1, FI=KS, SI=MN1, QN=9999)
CONMS.
REWIND, TAPE8.
COPYBF, TAPE8, TAPE1.
RETURN, TAPE8.

Figure 17.-Step 2 job for the SITE II database conversion.
Figure 18.—Step 3 job for the SITE II database conversion.
PROGRAM SETUP15 (OUTPUT, TAPE5=/4424, TAPE7=/4424)
DIMENSION IN(443)
IREC=1
INREC=0
IFICT=1
INP=0
IOT=0

5 READ(5,100) IN
    IF(EOF(5)) 11, 7
7    INP=INP+1
    WRITE(7,100) IN
    IOT=IOT+1
    IF(INP.GE.2) INREC=INREC+6
    GO TO 5
11   ENDFILE 7
    IF(IFICT.EQ.15) GO TO 10
    PRINT 130, IREC, IOT, INREC, INP
    IREC=1
    INREC=0
    IFICT=IFICT+1
    INP=0
    GO TO 5
10   PRINT 120, IOT
    STOP
100  FORMAT (150A10, 150A10, 142A10, A4)
120  FORMAT (33H1NORMAL END OF PROCESSING, COUNT=, I5)
110  FORMAT (8X, 9X, I9)
130  FORMAT (26HOENDFILE WRITTEN. COUNTS=, 4I5)
END

Figure 19.-Setup program for 15 records.
PROGRAM SETUP18(OUTPUT, TAPE5=/4424, TAPE7=/4424)
DIMENSION IN(443)
IREC=1
INREC=0
IFICT=1
INP=0
IOT=0

5 READ(5,100) IN
  IF(EOF(5)) 11,7
  INP=INP+1
  WRITE(7,100) IN
  IOT=IOT+1
  IF(INP.GE.2) INREC=INREC+6
  GO TO 5
11 ENDFILE 7
  IF(IFICT.EQ.18) GO TO 10
  PRINT 130, IREC, IOT, INREC, INP
  IREC=1
  INREC=0
  IFICT=IFICT+1
  INP=0
  GO TO 5
10 PRINT 120, IOT
STOP
100 FORMAT(150A10,150A10,142A10,A4)
120 FORMAT(33H1NORMAL END OF PROCESSING, COUNT=,I5)
110 FORMAT(8X,9X,I9)
130 FORMAT(25H0ENDFILE WRITTEN. COUNTS=,4I5)
END

Figure 20.-Setup program to process 18 records.
PROGRAM CONMN(OUTPUT, TAPE6=OUTPUT, TAPE7=/4424, TAPE8)

C
C PROGRAM TO CONVERT MAIN DATABASE FROM CHAR TO BINARY NASA/LRC.
C DATABASE CREATED ON IBM 370 WITH FORMAT
C  BLKSIZE=4424, LRECL=4424, DEN=3, STANDARD EBCDIC
C
C PARAMS
C
C INPUT REC = 736 CHARACTERS
C INPUT BLOCK = 4416 CHARACTERS
C INPUT BLOCK FACTOR = 6
C OUTPUT REC = 89 BINARY WORDS 3 DBL PRECISION=92E
C OUTPUT BLOCK = 534 BINARY WORDS 3 DBL PRECISION=552E
C OUTPUT BLOCK FACTOR = 6

C DIMENSION INDEX(800), II(552), FOREC1(92,6), FOREC2(46,6)
C COMMON IOREC1(92,6)
C DOUBLE PRECISION FOREC2
C EQUIVALENCE(II(1), IOREC1(1,1))
C EQUIVALENCE(IOREC1(1,1),FOREC1(1,1))
C EQUIVALENCE(IOREC1(1,1),FOREC2(1,1))
C IREC=1
C INREC=0
C IFILE=7
C INP=0
C IOT=1
C
C CALL OPENMS(8, INDEX, 800, 0)
C 5 READ(IFILE,100) ((IOREC1(I,J), I=1, 2), (FOREC1(I,J), I=3, 4),
C 1(FOREC2(I,J), I=3, 5), (IOREC1(I,J), I=11, 92), J=1, 6)
C IF(EOF(IFILE)) 10, 7
C 7 INP=INP+1
C IF(INP.GE.2) IREC=IOREC1(2,1)
C CALL WRITMS(8, II, 552, IOT)
C IOT=IOT+1
C IF(INP.GE.2) INREC=INREC+6
C GO TO 5
C 11 IOT=IOT-1
C PRINT 120, INP, IOT, INREC, IREC
C CALL CLOSMS(8)
C STOP
C 10 PRINT 110, INP
C STOP
C 100 FORMAT(8X, 6(2I9, 2F7.4, 3F16.0, 82I8))
C 110 FORMAT(3HONORMAL END OF PROCESSING, COUNT=, I5)
C 120 FORMAT(140FILE WRITTEN., 12HINPUT BLOCKS, I5, 13HOUTPUT BLOCKS, I5, 13HINPUT RECORDS, I5, 7HCOUNTER, I5)
C END

Figure 21.-Database conversion program.
CONZIP,T300,CM60000.
USER.
CHARGE.
GET,CONZP.
FTN,I=CONZP.
FILE(TAPE7,RT=F,BT=K,FL=3208,MBL=3208,RB=1)
VSN,TAPE7=NASA00.
REQUEST(TAPE7,F=S,NT,D=1600,LB=KU,CV=EB,PO=R)
LDSET(FILES=TAPE7)
LGO.
RETURN,TAPE7.
VSN,TAPE2=ND1226.
LABEL(TAPE2,SI=ZIPCDS,FI=ZIPALL,QN=1,W,PO=W)
COPYBF,TAPE8,TAPE2.
RETURN,TAPE2.

Figure 22.—Job to process the zip code portion of SITE II.
PROGRAM CONZIP(OUTPUT,TAPE6=OUTPUT,TAPE7=/3208,TAPE8)

C PROGRAM TO CONVERT ZIP DATABASE FROM CHARACTER TO
C BINARY FOR NASA/LRC.
C DATABASE CREATED ON IBM 370 WITH FORMAT:
C LRECL=3204,BLKSIZE=3208,RECFM=VS,DEN=3
C PARAMS:
C INPUT RECORD = 32 CHARACTERS
C INPUT BLOCK = 3200 CHARACTERS
C OUTPUT BLOCK FACTOR = 100
C OUTPUT RECORD = 5 BINARY WORDS
C OUTPUT BLOCK = 500 BINARY WORDS
C OUTPUT BLOCK FACTOR = 100
C
DIMENSION INDEX(4000),II(500)
DIMENSION IOREC(5,100),FOREC(5,100)
EQUIVALENCE(II(1),IOREC(1,1))
EQUIVALENCE(IOREC(1,1),FOREC(1,1))
IOT=1
INP=0
CALL OPENMS(8, INDEX, 4000, 0)
5 READ(7,100)((IOREC(I,J),I=1,3),FOREC(4,J),IOREC(5,J),J=1,100)
    IF(EOF(7)) 10,7
6 INP=INP+1
7 CALL WRITMS(8,II,500,IOT)
   IOT=IOT+1
   GO TO 5
10 PRINT 110,INP
STOP
100 FORMAT(8X,100(I5,2I9,F5.4,I4))
110 FORMAT(33HNORMAL END OF PROCESSING, COUNT=,I5)
END

Figure 23.-Conversion program to process zip code data.
7.0 REFERENCES


The purpose of this document is to describe the Airport-noise Levels and Annoyance Model (ALAMO) in terms of a detailed description of the constituent modules, to describe the execution of ALAMO procedure files, necessary for system execution, and to present the source code documentation associated with code development at Langley Research Center. The modules constituting ALAMO are presented both in flow graph form, and through a description of the subroutines and functions that comprise them.
End of Document