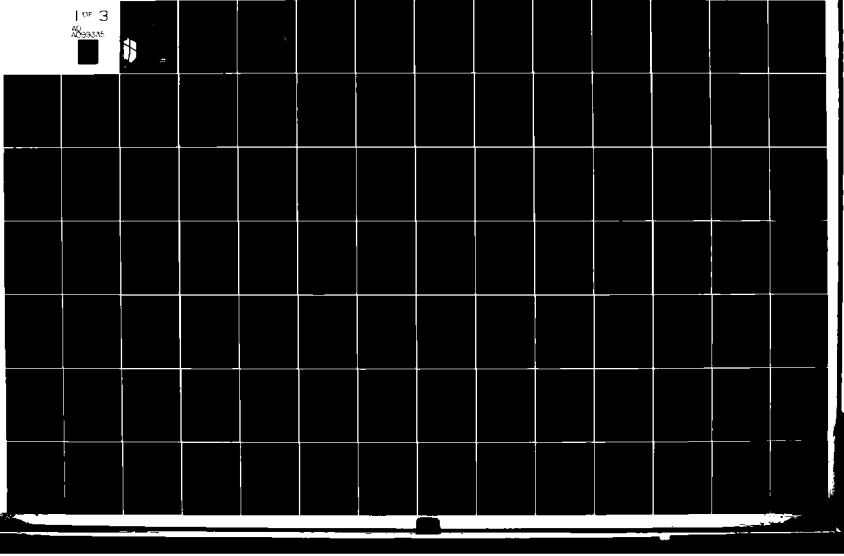
AD-A099 335 CONSTRUCTION ENGINEERING RESEARCH LAB (ARMY) CHAMPAIGN IL F/G 20/1
BLAST NOISE PREDICTION, VOLUME II. BNOISE 3.2 COMPUTER PROGRAM --ETC(U)
MAR 81 L L LITTLE, V I PAWLOWSKA, D L EFFLAND
UNCLASSIFIED CERL-TR-N-98-VOL-2 NL

1 OF 3
AD
A099335



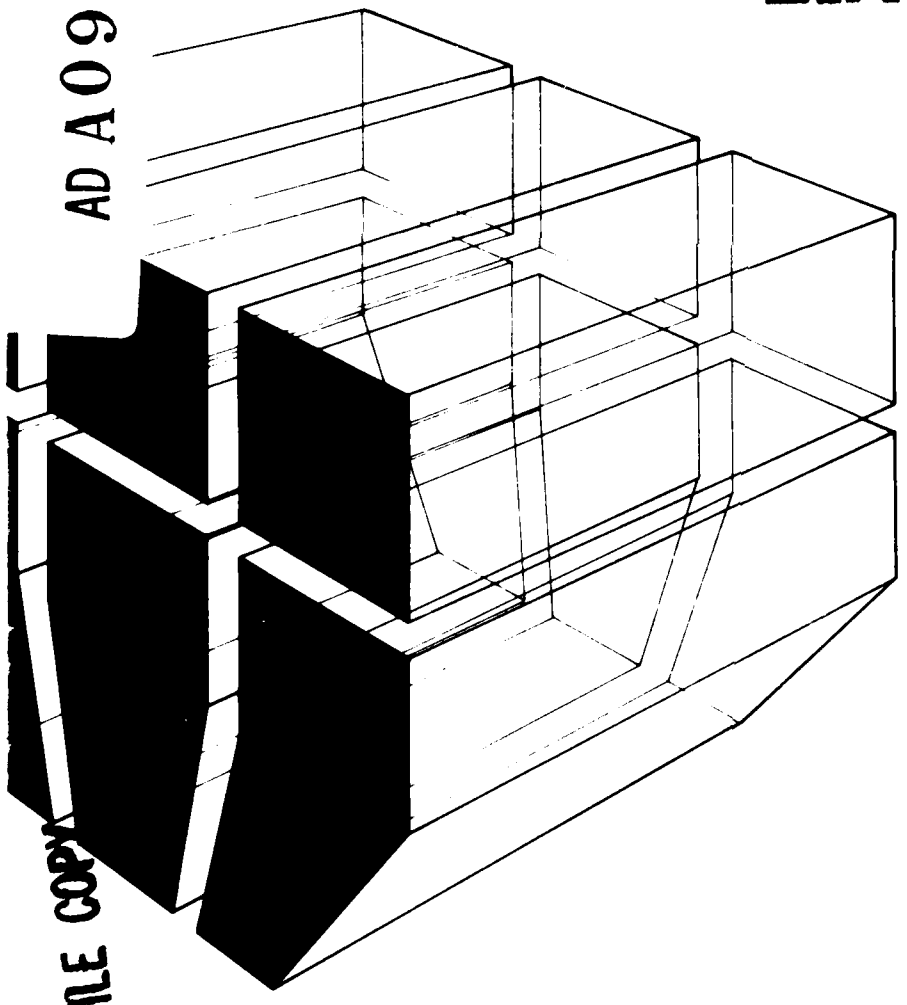
construction
engineering
research
laboratory

United States Army
Corps of Engineers
Serving the Army
Serving the Nation

TECHNICAL REPORT N-98
March 1981
Refined and Validated Noise Contour System

BLAST NOISE PREDICTION VOLUME II: BNOISE 3.2 COMPUTER PROGRAM DESCRIPTION AND PROGRAM LISTING

LEVEL II



by
Lincoln L. Little
Violet I. Pawlowska
David L. Effland

CERL

Approved for public release; distribution unlimited.

The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official indorsement or approval of the use of such commercial products The findings of this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

DESTROY THIS REPORT WHEN IT IS NO LONGER NEEDED
DO NOT RETURN IT TO THE ORIGINATOR

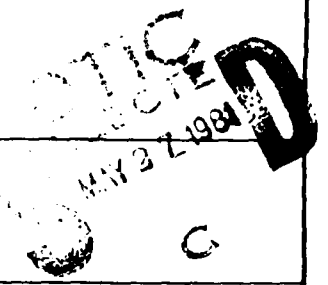
UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: 14 CERL-TR-N-98-VOL-2 / 2. GOVT ACCESSION NO.: AD-A099 335	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): BLAST NOISE PREDICTION. VOLUME II. BNOISE 3.2 COMPUTER PROGRAM DESCRIPTION AND PROGRAM LISTING.	5. TYPE OF REPORT & PERIOD COVERED: 9 FINAL 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): 10 Lincoln L. Little, Violet I. Pawlowska, David L. Effland	8. CONTRACT OR GRANT NUMBER(s): 1 [illegible]
9. PERFORMING ORGANIZATION NAME AND ADDRESS: U.S. ARMY CONSTRUCTION ENGINEERING RESEARCH LAB, P.O. Box 4005, Champaign, IL 61820 17	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 4A762720A896-A-012
11. CONTROLLING OFFICE NAME AND ADDRESS: (16) -A7 2,72 ØA 896	12. REPORT DATE: 11 March 1981 13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.



17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Copies are obtainable from the National Technical Information Service
Springfield, VA 22151

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

BNOISE 3.2 (computer program)
noise pollution
combat noise

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Over the past several years, the Construction Engineering Research Laboratory has gathered data from various sources dealing with blast noise generation and propagation and has performed several sets of field measurements designed to enhance this data base. These studies have been performed as a part of efforts aimed at improving methods of predicting the noise impact of military installations. Specifically included are measurements of the directivity pattern of major Army weapons and the statistical propagation of blast noise in the atmosphere. Volume I of this report develops and explains the relations

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Block 20 continued.

among the various data bases which have been developed to predict blast noise impact of Army facilities. From these studies, data bases and computational procedures which are used within the Blast Noise Prediction computer program (BNOISE 3.2) are developed.

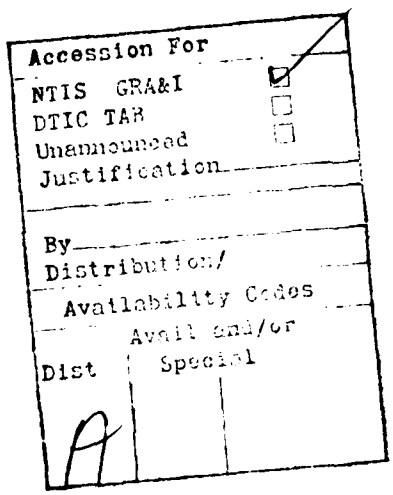
User instructions and a system description for the Blast Noise Prediction computer program, BNOISE 3.2, are given in Volume II. Also included is the procedure for using the program to obtain a noise contour for a specific set of data; how subroutines are invoked in a modular fashion; a description of module functions; module calling procedure; algorithms used by the program; and a summary of error messages.

FOREWORD

This research was conducted for the Directorate of Military Programs, Office of the Chief of Engineers (OCE), under Project 4A76270A896, "Environmental Quality for Construction and Operation of Military Facilities"; Task A, "Environmental Impact Monitoring Management Assessment and Planning"; Work Unit 012, "Refined and Validated Noise Contour System." The QCR number is 3.01.006. Mr. F. P. Beck, DAEN-MPE-I, is the OCE Technical Monitor.

The work was performed by the Environmental Division (EN), U.S. Army Construction Engineering Research Laboratory (CERL). Dr. R. K. Jain is Chief of EN.

COL Louis J. Circeo is Commander and Director of CERL, and Dr. L. R. Shaffer is Technical Director.



CONTENTS

CONTENTS (Cont'd)

TABLES

FIGURES

FIGURES (Cont'd)

BLAST NOISE PREDICTION
VOLUME II: BNOISE 3.2 COMPUTER PROGRAM DESCRIPTION AND PROGRAM LISTING

1 INTRODUCTION

Background

Over the past several years, the U.S. Army Construction Engineering Research Laboratory (CERL) has gathered data from various sources dealing with blast noise generation and propagation and has performed several sets of field exercises designed to further develop information regarding blast noise sources and the propagation of blast noise in the atmosphere. These CERL studies include measurements of the propagation of 735 five-pound charges set off at a central location at Fort Leonard Wood, MO,[1] measurements at Fort Sill on the directivity pattern of all of the Army's major weapons,[2] and small-scale studies at Fort Sill, OK and Fort Leonard Wood (Appendices A and B) designed to examine the weight relation between blast charge size and blast amplitude and duration. These studies were performed as a part of efforts aimed at developing approved methods of predicting the impact of blast noise at military installations. Data generated by these studies are used in the CERL-developed Blast Noise Prediction computer program (BNOISE 3.2), designed to predict the noise impact of military installations resulting from the blast-producing operations of armor, artillery, and demolition.

Purpose

The purpose of this report is to develop and explain the relations between and among various data developed to predict the blast noise impact of Army facilities, and from these data bases, to develop the computational procedures used within the Blast Noise Prediction computer program.

This volume contains the BNOISE 3.2 program description and program listing.

Mode of Technology Transfer

The program and documentation for the Blast Noise Prediction computer system will be available from the Department of the Army Assigned Responsible Agency (1981).

[1] P. D. Schomer, R. E. Goff, and L. M. Little, *The Statistics of Amplitude and Spectrum of Blasts Propagated in the Atmosphere*, Volumes I and II, Technical Report (TR) N-13, ADA033361 and ADA033646 (U.S. Army Construction Engineering Research Laboratory [CERL], November 1976).

[2] P. D. Schomer, L. M. Little, and A. B. Hunt, *Acoustic Directivity Patterns for Army Weapons*, Interim Report (IR) N-60, ADA066223 (CERL, October 1978).

2 PROGRAM DESCRIPTION

The Blast Noise Prediction computer program (BNOISE 3.2) was developed to allow faster computation of noise values in terms of C-weighted day/night average sound level (CDNL).* It also provides generalized and uniform noise-impact predictions when given a specific set of input data.[1] The output of the Blast Noise Prediction computer program is a set of CDNL contours which can be overlaid on a land-use map of military facility and its environs to allow rapid identification of noise-sensitive land areas.

This volume is written for those persons familiar with computer programming who require documentation and background for the Blast Noise Prediction computer program. It can also be used as a program reference manual.

System Overview

The Blast Noise Prediction program is a digital computer program which can produce CDNL contours for military facilities with impulsive noise sources (e.g., artillery, explosions or demolitions, and weapon blasts).

The program is written in Fortran IV and used on a Control Data Corporation (CDC) 6600 computer. It consists of a main controlling program, called CDNL, which invokes a series of subroutines when given a set of modules chosen by the user. There are also several supporting programs which aid CDNL.

Each subroutine contributes either directly or indirectly to the goal of producing CDNL contours for a facility. The data base for the program is a description of the facility and its noise sources; i.e., the location of firing points, targets, blast sites, and materials involved in producing the noise. Program modules which are to be used require instructions in the form of card input, headed by the module call card and followed by pertinent information chosen or provided by the user. In addition, input files such as the data base file or a file containing intermediate results from another module are necessary. Each module produces a printed report consisting of:

1. Module name
2. Input values supplied by the user, and/or default values supplied by the program
3. Module-specific output
4. Module execution time.

Some modules also produce output files on auxiliary storage (Figure 1 is a system flowchart). Table 1 summarizes the input/output (I/O) files used by the contouring program. This table lists the I/O files by number, a description of the files, the routines which use them for input, and the routines which produce them as output. If the programmer wishes to modify the program, he** should avoid changing these file identifiers (TAPE NUMBERS) as they already appear elsewhere in the program. TAPE5 is the input file used by all routines having card input. TAPE6 is the output file used by all routines which produce a printout (see Figure 1).

[1] P. D. Schomer, R. E. Goff, and L. M. Little, *The Statistics of Amplitude and Spectrum of Blast Propagation in the Atmosphere, Volume I and II*, TR N-13, ADA033361 and ADA035475 (CERL, 1976).

* The CDNL is the noise measurement criterion in Army Technical Manual (TM) 5-803-2, *Environmental Protection Planning in the Noise Environment* (Department of the Army [DA], 15 June 1978).

**The male pronoun is used throughout this report to refer to both genders.

USER'S RUN

When the user has a set of data for which he wishes to receive output from the Blast Noise Prediction program, he must put together a deck of cards called the USER'S RUN. This deck is composed of:

1. Job Control Language (JCL) cards
2. Input data cards
3. Module cards

Figure 2 presents the order of these sections within the deck. The user's main interest lies in providing the Blast Noise Prediction program with data in the proper form, knowing what the modules do, and understanding what type of output can be expected from them. The programmer should familiarize himself with the connection between the USER'S RUN deck and the corresponding code in the subroutines.

The programmer should know what input must be provided to the Blast Noise Prediction program by the user so he can relate items in the deck to the internal variables and subroutines within the program; it is also helpful to know how these data are acquired. Figure 3 shows the general composition of input data of the user's deck.

JCL Cards

The user must provide a set of system-specific JCL cards at the beginning of the USER'S RUN deck. These cards inform the computer that someone wants to use the stored Blast Noise Prediction program. The JCL cards do not change from run to run for the average user, and he should not concern himself with them beyond knowing that they must precede the cards of his actual data items and module calls.

Input Data

The Blast Noise Prediction program considers noise from two types of sources: firing points and target points. The firing point is the spot from which a projectile is being launched or the site of a demolition charge; the target point is the site of a projectile's impact; if data are to be useful to the Blast Noise Prediction program, a projectile's launching weapon must produce a sharp blast and not a drawnout, "whoosh-type" sound such as that emitted by most rockets.*

The data required on input cards are (1) the X and Y coordinates of the firing range for both the target and firing points, and (2) the amount of propellant and projectile charge, in TNT equivalents, producing the noise at both the firing or target point. In the case of sound occurring only at a firing point, the user must determine whether the sound is caused by an explosion at that point (e.g., a demolition charge) or if it is an instance of a weapon propelling a projectile which makes no noise upon impact at its target point (e.g., an illumination round). In the first case, only the firing point coordinate is needed, while in the second, both firing and target point coordinates are required, even though there is no sound at the target. This is because noise produced by a weapon has a directivity pattern associated with it that varies the amount of sound pressure around the weapon. The degree of variance depends, in great part, on the physical characteristics of the weapon itself. A projectile or demolition, however, produces noise omnidirectionally from its source.[4]

It is up to the user to differentiate between the amount of propelling charge at the firing point and the amount of projectile charge exploding at the target point. These values vary for different weapons and ammunition. Table 2 gives the weapon codes used for a certain set of standard weapons. The weight of propellant and projectile charges for these weapons, in TNT equivalents, and directivity information are given in Tables 3 and 4.

It can be seen from Table 3 that various amounts of propellant can be used to fire one size of projectile. The user is not restricted to the weapon types listed in Tables 2, 3, and 4. He may create data to suit his needs as long as they are put into proper format and labeled with a new gun-type code.

[4] P. D. Schomer, L. M. Little, and A. B. Hunt, [illegible] *Directivity Patterns for Army Weapons*, TR N-60/ADA066223 (CERL, January 1979).

* A [illegible] small charge is used to simulate the sound of rockets.

("New" means that the code number differs from all of the codes already provided.) This prevents the computer and/or other users from confusing it with an already listed code. At no time can two different weapons have the same code within one deck. The user must also specify the number of rounds fired by each weapon at each point during the day (0700 to 2200 hours) and during the night (2200 to 0700 hours).

The input data portion of the USER'S RUN deck is divided into:

1. Gun-type cards
2. Target point definition cards
3. Firing point definition cards.

The gun-type cards describe the weapons by specifying (1) the weight of explosive in the projectile, (2) the weight of the different propellant charges, (3) the name of the weapon, (4) the weight parameters A and B, and (5) the directivity pattern. Table 5 outlines the required format. The target point definition cards list the X and Y coordinates of the target area. The firing point definition cards, in addition to listing the X and Y coordinates of the firing point, specify (1) what types of weapons are the noise-producing sources, (2) how often each specified weapon is fired during the day and at night, (3) if the projectile impacts at the target, and (4) the height of impact. Tables 6 and 7 give the card format for target and firing point data, respectively.

Modules

A sequence of individual sections, called modules, informs the Blast Noise Prediction program what the user wants done to the input data he has provided. These modules correspond to major subroutines in the Blast Noise Prediction program. There are, however, more subroutines which are necessary for program operation than there are modules available to the user.

To identify and run a Blast Noise Prediction program module, the user must (1) input a card identifying the module to be run followed by (2) a card (or cards) specifying the parameters which that module requires to function correctly. The user varies the parameters to account for the input data and to produce the desired output. Output can be either a set of printed tables of CDNL values or an actual paper plot of CDNL contours, depending on which modules are invoked. There is some freedom in the ordering of the modules, however, certain modules use the output generated by other Blast Noise Prediction program modules. If the user chooses one of these "dependent" modules, he must be sure it is preceded by the modules which will provide the data it needs. Figure 1, a flowchart for Blast Noise Prediction program modules, shows which inputs they require and the output they produce.

The following modules are currently available from the Blast Noise Prediction program.

NEF-1 specifies whether the data base coordinates are in meters or feet. There is only one of these cards per USER'S RUN and it is always the first card in the module section. It is also used for specific options which might be available.

BASE generates line segments which can be drawn by the PLOT routine on the paper plot output. It can be used to create an outline of the facility for which the user is providing data. It can also be used to draw lines on the PLOT output to be used as a visual reference for lining up the paper plot with the actual map. BASE causes PLOT to draw line segments from coordinate to coordinate, as specified.

BOUNDS uses map coordinates to set the limits of the total area encompassed by CDNL calculations and to be enclosed by the PLOT drawing. It is used to define the area in which the user is interested.

FORMA tabulates target and firing point information from the input data. It compresses data into the form required by PUDDLE GRID.

LOCATOR labels target and firing points on the plot.

MAP does elementary error checking and is a preprocessor for the input data. It also produces a listing of input data and generates cross-reference tables.

The procedures for collecting and coding data are given in T. McBryan, *Compilation of Operational Noise Data*, TR N-82/AD A080429 (CERL, January 1980).

PLOT combines output from appropriate modules for use by the NASAPLOT* program to create actual plots of contours of CDNL levels.

POINT calculates CDNL for specific locations.

PUDDLE GRID creates a rectangular grid of CDNL values in the area specified by the user in the BOUNDS module.

SCATTER allows the PLOT routine to create [illegible] (dots) of noise sources; the number of dots is proportional to the blasting activity at that point.

STOP signals the Blast Noise Prediction program that the module section of the USER'S RUN deck has been terminated.

A multipunch 6789 card must be the last card in the USER'S RUN; the multipunch informs the computer that all input for the USER'S RUN has been submitted and that the system does not need to look for more cards in order to execute the run.

Figure 4 represents a possible sequence in which the modules of the Blast Noise Prediction program could be ordered by the user in the USER'S RUN. Tables 8 through 17 summarize the input required by each module of the Blast Noise Prediction program. These tables describe the instruction cards for the modules and list, in tabular format, the specifications for all the parameters required by each module. Tables 8 through 17 have the following column headings:

CARD ID: gives each card of a module a name as a reference; e.g., the first card in MAP is called MAP-1.

COLUMNS: the card columns in which data must appear.

VARIABLE NAME: the name given to these data within the program.

FORMAT: Fortran description of how that data item must look. There are three basic formats used by the Blast Noise Prediction program; each corresponds to a particular type of data.

1. A -- alphanumeric data such as module names; i.e., items combining the letters A through Z and the numbers 0 through 9. For example, A10 or a field of up to 10 characters (letters and numbers) must be left justified.

2. I -- integer number, e.g., I2 or a 2-column (digit) number with no decimal point (must be right justified in the columns in which it appears).

3. F -- all other data and normal decimal numbers for real arithmetic operations; e.g., F10.1 or a 10-digit real number at maximum (can be smaller as long as a decimal point is put in, e.g., 32.5 or 56.3). The 10 means the number is a total of 10 columns; one column is a decimal point and there is one digit after the decimal point. A shorter number simply means that the leading digits are zeros. *A letter character cannot appear in a field specified as an "I" or "F" format. The user must make sure data items are in proper format on the cards in order for the Blast Noise Prediction program to interpret the input correctly.*

DESCRIPTION: tells what parameter is being referred to and includes any special codes and/or notes about required data and their units, if any. *Any characters between quote marks ("") are the actual items written in the specified columns of that module card.*

COMMENTS/DEFAULTS: default values are the quantities the Blast Noise Prediction program will use if certain items are left blank by the user. If there is no default value given for it, that data item must be specified by the user. Special instructions regarding that card are included under this heading.

PREVIOUS CALLS REQUIRED: a module whose output is required as input to the current module and therefore must be called *before* the current module.

*See Volume I

Additional Module Input

Most of the data required by Blast Noise Prediction program modules are self-explanatory and are listed, by module, in Tables 8 through 17. However, some parameters for PUDDLE GRID, POINT, FORMA, and MAP require more detailed explanations.

PUDDLE GRID

The "grid size" specification in the PUDDLE GRID module specifies how often, in terms of X and Y coordinates, the CDNI values are to be computed (e.g., every 1000 m or 2000 ft are reasonable values if the overall land area included in the contour is fairly large). The smaller the values used, the finer the grid size, and the smoother the contour that is produced. This is because the program has a greater number of actual points to plot from and fewer to approximate. Cutting the grid size in half will cause four times as many points to be computed and will cost about four times as much to run, since run cost is proportional to the number of points computed. The user must consider cost when choosing grid size, because if the program has to compute a greater number of values it will run longer, and therefore will cost more.

PUDDLE GRID and POINT

The "inversion factors" specification in the PUDDLE GRID and POINT modules provides a set of meteorological data to the program. Weather conditions, especially temperature, have an effect on the way sound propagates through the atmosphere (see Volume I). Currently, the Blast Noise Prediction program does not take into consideration the effects of wind. However, it does take into account temperature inversions. Therefore, the user must provide appropriate inversion data for the location of his noise study. Inversion data tables available from the National Weather Service summarize radiosonde observations made at selected weather stations. A summary of this temperature data is presented in Table 18. This table is used to determine the temperature inversion factors required by PUDDLE GRID and POINT. The user should locate the city closest geographically and meteorologically. The number in that row under the column labeled "SURFACE" is the value used for inversion factor 1. The second factor is obtained under the "1-500 m" column; the third factor is found under the column labeled "1-3000 m."

FORMA

The "charge averaging technique" specification in the FORMA module allows the user to choose the range of charge sizes the program will use from among those given for the propelling charges for a specific weapon in the gun-type cards. This range is chosen in the firing point definition cards. For example, if the user states that weapon 1 normally uses charges from zones 1 through 5 (the range), the options yield the following results:

1. If MAX is chosen, the program will use the largest charge size in the given fields for its calculations.

2. If CAVE is chosen, the program will average the actual TNT equivalents of all the fields and use that average for computations.

3. If IAVE is chosen, the program will integer average the zone numbers and use the value of charge size found in that zone, i.e., 1 and 5 yield charge 3, 1 and 4 also yield charge 3. If upper and lower boundaries of the charge zones are the same number, then all three options will arrive at the same charge size. For example, if 3 and 3 are specified on the firing-point definition card as the charge range, then the program will use whatever charge value is in zone 3 of the propelling charges for that weapon, no matter which averaging technique is chosen.

MAP

If a user wishes to determine how many points will be generated by PUDDLE GRID for a specific grid size, he may request this information from the MAP module instead of calculating it himself. The user must specify the number of various grid sizes he wishes to try on the MAP-4 card, and then list these with as many MAP-5 cards as necessary. The output from MAP will list the number of points that will be generated by the chosen grid sizes. The MAP module can be used by the user to indicate the cost of producing a grid (see PUDDLE GRID, above). These MAP module cards must be

included even if the user does not wish to use this feature. The values commonly used are 01 for MAP 4 and 280 for MAP 5. If, for example, the area of interest is 10 000 x 10 000 m, a grid size of 1000 m would produce 10 times 10 or 100 grid points.

Module Output

All Blast Noise Prediction program modules produce printed reports which are tables of intermediate or final results and, if necessary, error messages. In addition, some of the information produced by particular modules is stored temporarily, to be used as input to other modules at a later point in the program. The printouts provide a hard copy record of information supplied to the Blast Noise Prediction program via module cards and allow easy verification of the accuracy of module facts and data items input to the main program.

Obtaining Results from Module Output

Module outputs of primary interest to the user are those produced by the PUDDLE GRID and PLOT modules. The paper contour produced by PLOT is often the end result sought by the person creating a USER'S RUN. Since the user can control the amount of area enclosed by the plot, he can specify the plot's scale to agree with the maps he is working with, or he can scale down the plot to obtain a smaller version of his results for future or intermediate reference.

If the user does not wish to incur the cost of an actual plot, then PUDDLE GRID output can be used as a source of intermediate results. PUDDLE GRID generates a list of CDNL values by coordinates in increments specified by the GRID SIZE. The user can approximate the noise levels in any given area by using this table; it can also be compared against the expected CDNL levels in a given area, thus checking the data. For example, if PUDDLE GRID output indicates that the region around a firing point has the lowest values of the entire table, the user should go back and recheck his data for errors.

If the user wants only the PUDDLE GRID values and not the actual plot, he must modify the JCL section of his USER'S RUN by excluding from the deck all cards which are used for plotting. (And, of course, the PLOT module is left out of the module input cards.)

All Blast Noise Prediction program modules also produce statistics giving the amount of system time spent by the computer in a given module. This information can be used to approximate run cost. This is done by multiplying system time by cost per time unit. For example, if a module ran for 300 ms and the cost is one third of a penny/ms, then that module's cost is approximately $1.00.

Module Error Messages

Because the Blast Noise Prediction program has no way of knowing whether the data it uses are correct, it will use incorrect data that are reasonable and in proper format. The program can only indicate errors in format and point out unreasonable data items (e.g., negative charge sizes); if there are enough errors and/or unreasonable items, the program will halt execution. Therefore, it is important that the user check the statistics produced by the modules to make sure the Blast Noise Prediction program is performing operations on the correct data.

Error messages tell the user that there is something wrong with the data provided to the Blast Noise Prediction program. Warning messages indicate that there is something out of the ordinary in the data, but not necessarily wrong. For example, a charge size input by the user that is appreciably larger than other charge sizes specified by the user would trigger a warning message, alerting the user to the probability of a misplaced decimal point.

Table 1
I/O File Summary

Fortran File Name	Description	Characteristics	Device	Output By Module Call: (Subroutine/Entry)	Input To Module Call: (Subroutine/Entry Name)
TAPE5 (Input)	Module instructions	Card image	Card reader		
TAPE6 (Output)	Module reports	Print line	Printer		
TAPE7	Data base	Card image	Auxiliary storage tape/disk		LOCATOR (LOCATR) MAP (MAP) FORM-A (FORMA) SCATTER (SCATTR)
TAPE8	Intermediate results noise source tabulation	Internal format	Auxiliary storage	FORM-A (FORMA)	PUDDLE GRID (PGRID)
TAPE1	PUDDLE GRID output	Card image (PHS2 NASAPLOT)	Auxiliary storage	PUDDLE GRID (PGRID)	PLOT (PLOT)
TAPE2	Base outline	Card image (PHS4 NASAPLOT)	Auxiliary storage	BASE (BASE)	PLOT (PLOT)
TAPE3	Target and firing point locations	Card image (PHS3 NASAPLOT)	Auxiliary storage	LOCATOR (LOCATR)	PLOT (PLOT)
TAPE4	Scattergram	Card image (PHS4 NASAPLOT)	Auxiliary storage	SCATTER (SCATTR)	PLOT (PLOT)
TAPE20	Acoustical data base	Internal format	Auxiliary storage	TABGEN	READ TABLE (READTB)
TAPE55	NASAPLOT input	Card image	Auxiliary storage	PLOT (PLOT)	NASAPLOT

OTHER TAPES: TAPE51, TAPE70, TAPE72, TAPE75, TAPE90, TAPE91, TAPE99

Table 2
Weapon Codes

Weapon	Code
105-mm howitzer (M102)	1
155-mm howitzer (M109)	2
8-in. howitzer (M110)	3
175-mm gun	4
155-mm howitzer (M109A1)	5
155-mm howitzer (M114)	6
8-in. howitzer (M110A1)	7
155-mm howitzer (M198)	8
Small charge TNT (0.25–90 lb)	10
Large charge TNT (110–500 lb)	11
60-mm mortar	20
81-mm mortar	22
107-mm mortar (4.2 in.)	23
57-mm recoilless rifle	30
90-mm recoilless rifle (M67)	31
106-mm recoilless rifle (M40A1)	32
20-mm gun	40
40-mm gun	41
57-mm gun	42
90-mm gun	43
2.75-in. rocket	50
3.5-in. rocket	51
66-mm rocket	52
LAW missile (M72)	53
TOW missile	54
Dragon missile	55
Shillelagh missile (from 152-mm gun)	56
40-mm grenade launcher (M203)	60
Rifle grenade (M79)	61
Hand grenade (M67)	62
M60 tank (105-mm) regular shell	90
M60 tank (105-mm) high velocity shell	91
152-mm tank gun (Sheridan) (M551) regular shell	92
152-mm tank gun (Sheridan) (M551) HEAT-T shell	93
165-mm cannon (M135)	94

Table 3
Projectile and Propellant Weights for Table 2 Weapon Codes
(TNT equivalent in pounds)

Weapon Code	Projectile Weight	Propellant Weight Charge Zones 1	2	3	4	5	6	7	8	9	10
1	4.6	0.5175	0.605	0.7731	1.004	1.3275	1.8656	2.7456			
2	15.4	1.7687	2.2875	3.0875	4.025	7.05	9.8375	13.275	38.0		
3	36.3	5.3188	6.2688	7.5125	9.5188	16.85	22.0125	28.1375		43.6	
4	31.3	23.56	39.7	57.24							
5	15.4	1.7687	2.2875	3.0875	4.025	7.05	9.8375	13.275			
6	15.4	1.7687	2.2875	3.08757	4.025	7.05	9.8375	13.275			
7	36.3	5.3188	6.2688	7.5125	9.5188	16.85	22.0125	28.1375	38.0	43.6	
8	15.4	2.0	2.8	3.3	5.6	8.5	14.2	17.3	26.0		
10		0.25	1.0	5.0	10.0	15.0	25.0	35.0	50.0	70.0	90.0
11		110.0	140.0	170.0	200.0	240.0	290.0	340.0	380.0	440.0	500.0
20	0.42	0.021	0.042	0.063	0.084						
21	0.6	0.22	0.25	0.28	0.31						
22	2.25	0.042	0.0653	0.0886	0.1119	1.352	0.1585	0.1818	0.1941	0.2284	
23	8.5	0.0803	0.1132	0.1642	0.3369	0.6717					
30	0.55	1.0									
31	1.72	1.31									
32	2.79	8.0									
40	0.05	0.2									
41	0.14	0.718									
42	0.44	2.28									
43	1.9	7.31									
50	5.0	0.01									
51	1.88	0.01									
52	0.13	0.85									
53	0.66	0.01									
54	5.3	0.01									
55	3.5	0.01									
56	8.0	14.0									
60	0.6	0.01									
61	0.6	0.0001									
62	0.6	0.01									
90	6.6	5.9									
91	2.14	11.5									
92	9.5	6.0									
93	6.3	6.0									
94	20.0	2.12									

Table 4
Directivity Information for Table 2 Weapon Codes

Weapon Code	A	B	0°	30°	60°	90°	120°	150°	180°	210°	240°	270°	300°	330°	Avg.
1	83.78	13.91	17.80	13.91	10.02	6.46	2.97	0.53	0.0	0.53	2.97	6.46	10.02	13.91	10.84
2	75.74	18.51	0.63	0.46	0.29	1.00	1.45	0.39	0.0	0.39	1.45	1.00	0.29	0.46	0.67
3	83.64	14.13	13.77	10.29	6.82	4.01	1.42	-0.64	0.0	-0.64	1.42	4.01	6.82	10.29	7.36
4	73.29	17.50	16.33	13.08	9.84	5.41	3.03	0.50	0.0	0.50	3.03	5.41	9.84	13.08	9.90
5	72.08	18.11	-1.88	-1.71	-1.53	-0.49	0.94	-0.52	0.0	-0.52	0.94	-0.49	-1.53	-1.71	-0.60
6	80.86	15.59	14.93	12.55	10.18	6.87	3.20	1.56	0.0	1.56	3.20	6.87	10.18	12.55	9.45
7	76.99	15.87	16.33	13.08	9.84	5.41	3.03	0.50	0.0	0.50	3.03	5.41	9.84	13.08	9.90
8	72.08	18.11	-1.88	-1.71	-1.53	-0.49	0.94	-0.52	0.0	-0.52	0.94	-0.49	-1.53	-1.71	-0.60
10															
11															
20	95.00	20.00	8.29	6.39	4.48	5.75	2.20	2.88	0.0	2.88	2.20	5.75	4.48	6.39	4.86
22	90.27	19.57	8.29	6.39	4.48	5.75	2.20	2.88	0.0	2.88	2.20	5.75	4.48	6.39	4.86
23	85.17	18.85	8.45	6.61	4.78	5.17	1.89	1.58	0.0	1.58	1.89	5.17	4.78	6.61	4.74
30	106.00	0.0	2.28	-0.87	-4.02	-4.77	-2.40	-0.07	0.0	-0.07	-2.40	-4.77	-4.02	-0.87	-1.29
31	107.10	0.0	-9.51	-9.75	-9.98	-7.18	-3.17	-0.49	0.0	-0.49	-3.17	-7.18	-9.98	-9.75	-4.11
32	111.80	0.0	2.28	-0.87	-4.02	-4.77	-2.40	-0.07	0.0	-0.07	-2.40	-4.77	-4.02	-0.87	-1.29
40	90.00	15.00	13.77	10.29	6.82	4.01	1.42	-0.64	0.0	-0.64	1.42	4.01	6.82	10.29	7.36
41	85.00	15.00	13.77	10.29	6.82	4.01	1.42	-0.64	0.0	-0.64	1.42	4.01	6.82	10.29	7.36
42	85.00	15.00	13.77	10.29	6.82	4.01	1.42	-0.64	0.0	-0.64	1.42	4.01	6.82	10.29	7.36
43	85.00	15.00	13.77	10.29	6.82	4.01	1.42	-0.64	0.0	-0.64	1.42	4.01	6.82	10.29	7.36
50	88.75	13.85													
51	88.75	13.85													
52	88.75	13.85													
53	88.75	13.85													
54	88.75	13.85													
55	88.75	13.85													
56	88.75	13.85													
60	85.00	0.0													
61	65.00	0.0													
62	65.00	0.0													
90	111.40	0.0	15.35	14.04	12.73	7.61	4.81	1.09	0.0	1.09	4.81	7.61	12.73	14.04	10.78
91	115.80	0.0	15.35	14.04	12.73	7.61	4.81	1.09	0.0	1.09	4.81	7.61	12.73	14.04	10.78
92	115.80	0.0	7.56	7.38	7.19	2.73	1.05	-0.39	0.0	-0.39	1.05	2.73	7.19	7.38	4.80
93	115.80	0.0	7.56	7.38	7.19	2.73	1.05	-0.39	0.0	-0.39	1.05	2.73	7.19	7.38	4.80
94	60.91	14.49	13.77	10.29	6.82	4.01	1.42	-0.64	0.0	-0.64	1.42	4.01	6.82	10.29	7.36

Table 5
Input Data Format Description -Gun-Type Cards

Card ID	Columns	Format	Description	Comments/Defaults
GUN-1	2-3	A2	Weapon code	2-digit number giving the code for a specific weapon name
	4-10	F7.0	Projectile weight	In TNT equivalent pounds of explosive
	11-17	F7.0	Propellant weight (zone 1)	In TNT equivalent pounds of explosive
	18-24	F7.0	Propellant weight (zone 2)	10 charge zones allowed (values as specified in Table 4)
	25-31	F7.0	Propellant weight (zone 3)	
	32-38	F7.0	Propellant weight (zone 4)	
	39-45	F7.0	Propellant weight (zone 5)	
	46-52	F7.0	Propellant weight (zone 6)	
	53-59	F7.0	Propellant weight (zone 7)	
	60-66	F7.0	Propellant weight (zone 8)	
	67-73	F7.0	Propellant weight (zone 9)	
	74-80	F7.0	Propellant weight (zone 10)	
GUN-2	4-23	2A10	Name of weapon	
GUN-3	1	A1	"*" to flag end of gun-type cards	
	4-9	F6.2	Parameter A	
	10-14	F5.2	Parameter B	
	15-19	F5.2	Decibel difference from rear of gun at 0° (0° is front of gun)	Values as specified in Table 4
	20-24	F5.2	30° decibel difference	Values as specified in Table 4
	25-29	F5.2	60° decibel difference	Values as specified in Table 4
	30-34	F5.2	90° decibel difference	Values as specified in Table 4
	35-39	F5.2	120° decibel difference	Values as specified in Table 4
	40-44	F5.2	150° decibel difference	Values as specified in Table 4
	45-49	F5.2	180° "0.0"	Rear of gun
	50-54	F5.2	210° decibel difference	Values as specified in Table 4
	55-59	F5.2	240° decibel difference	Values as specified in Table 4
	60-64	F5.2	270° decibel difference	Values as specified in Table 4
	65-69	F5.2	300° decibel difference	Values as specified in Table 4
	70-74	F5.2	330° decibel difference	Values as specified in Table 4
	75-79	F5.2	Average difference from rear of gun	Values as specified in Table 4

Table 6
Input Data Format Description–Target Cards

Card ID	Columns	Format	Description	Comments/Defaults
*One TRG-1 Card Per Target Point**				
TRG-1	1	A1	"*" if last target card	
	3-5	A3	1-3 character target ID	
	7-12	F6.0	Location, x-coordinate	
	13-18	F6.0	Location, y-coordinate	
	19-24	F6.0	Ground correction factor (dB)	1.5 dB

*Maximum 50 targets.

Table 7
Input Data Format Description –Firing Point Cards

Card ID	Columns	Format	Description	Comments/Defaults
One FP-1 card and associated FP-2 cards per firing point				
FP-1	3-5	A3	Firing point ID	
	7-12	F6.0	Location, x-coordinate	
	13-18	F6.0	Location, y-coordinate	
	19-24	F6.0	Ground correction factor (dB)	1.5 dB
FP-2	1	A1	"*" if last definition card for a particular firing point	
	19-20	A2	Gun type (code)	See Tables 3 and 4
	21-24	F4.0	Number of day firings	0.0
	25-28	F4.0	Number of night firings	0.0
	29-30	I2	Min charge zone	See Tables 3 and 4
	31-32	I2	Max charge zone	See Tables 3 and 4
	33-35	A3	Corresponding target ID	Must be blank if firing point sound is omnidirectional (i.e., demolition or explosion)
	36	I1	"1" = no noise at target	If target is blank, this must be set to "1"
	37-41	F5.0	Height, in feet (if applicable)	"-" below ground "+" above ground

Table 8
BASE Input Summary

CARD INPUT:

Card ID	Columns	Variable Name	Format	Description	Comments/Defaults
NEF-1	1-6	IMETER	A10	Distance Units "METERS"/"FEET"	
. . . . . CALLS TO OTHER MODULES					
BASE-1	1-4	HDR	A10	"BASE"	
BASE-2 CARD REPEATED AS NECESSARY TO DESCRIBE BASE OUTLINE					
BASE-2	1-10	XCOORD	F10.0	Coordinates of point in set describing one line segment	
	11-20	YCOORD	F10.0		
	21	STAR	A 1	"*" indicates end of continuous line, following point starts new line	
BASE-3[+]	21	STAR	A 1	"*" indicates end of data for "BASE"	

PREVIOUS CALLS REQUIRED: BOUNDS

REQUIRED FILES: None

OUTPUT: Printed report
TAPE2 PHS4 input to NASAPLOT, input to PLOT routine

[+]There must be one BASE-3 card. It will have an "*" in column 21.

Table 9
BOUNDS Input Summary

CARD INPUT:

Card ID	Columns	Variable Name	Format	Description	Comments/Defaults
NEF-1	1-6	IMETER	A10	Distance Units "METERS"/"FEET"	
. . . . . CALLS TO OTHER MODULES					
BDS-1	1-6	HDR	A10	"BOUNDS"	
BDS-2	1-10	XMIN	F10.0	Minimum X coordinate	Boundary values used by PUDDLE GRID, PLOT, SCATTER and LOCATOR must be set before calls to these routines.
	11-20	YMIN	F10.0	Minimum Y coordinate	
BDS-3	1-10	XMAX	F10.0	Maximum X coordinate	
	11-20	YMAX	F10.0	Maximum Y coordinate	

PREVIOUS CALLS REQUIRED: None

REQUIRED FILES: None

OUTPUT: Printed report

Table 10
FORMA Input Summary

CARD INPUT:

Card ID	Columns	Variable Name	Format	Description	Comments/Defaults
NEF-1	1-6	IMETER	A10	Distance Units "METERS"/"FEET"	
. . . . . CALLS TO OTHER MODULES					
FRMA-1	1-6	HDR	A10	"FORMA"	
FRMA-2	1-4	IFUNC	A10	Charge averaging technique ("MAX"/ "IAVE"/"CAVE")	"MAX" - Use maximum charge zone "IAVE" - Use average of charge zones "CAVE" - Use average of actual TNT equivalents
	11-20	GHCORR	F10.0	Ground correction	Default = 1.5 dB
	21	NOWIND	I1	When nonzero, prevents rewinding TAPE7	Default is rewind
FRMA-3	1-10	DAYS	F10.0	Number of days of information in data base	Default = 1 Day

PREVIOUS CALLS REQUIRED: None

REQUIRED FILES: TAPE7 data base

OUTPUT: Printed report
TAPE8 intermediate results

Table 11
LOCATOR Input Summary

CARD INPUT:

Card ID	Columns	Variable Name	Format	Description	Comments/Defaults
NEE-1	1-6	IMETER	A10	Distance Units "METERS"/"FEET"	
..... CALLS TO OTHER MODULES					
LOC-1	1-7	HDR	A10	"LOCATOR"	
LOC-2	1-10	CHOICE	A10	"ALL", "TARGET", or "FIRING"	Type of points to be marked on plot
	11-20	NAME	A10	"NAME" prints ID on plot	
	21-30	LOCATE	A10	"LOCATION" prints coordinates	
	31-40	SIZE	F10.0	Size of letters	Default = 0.14; best results are obtained if multiple of 0.035
	41-50	ANGLE	F10.0	Rotation of letters	Default = 0°

PREVIOUS CALLS REQUIRED: BOUNDS

REQUIRED FILES: TAPE7 data base

OUTPUT: Printed report
TAPE3 PHS3 input to NASAPLOT, input to PLOT routine.

Table 12
MAP Input Summary

CARD INPUT:

Card ID	Columns	Variable Name	Format	Description	Comments/Defaults
NEE-1	1-6	IMETER	A10	Distance Units "METERS"/"FEET"	
..... CALLS TO OTHER MODULES					
MAP-1	1-3	HDR	A10	"MAP"	
MAP-2	1	IFLAG	I1	Print data base information if ≠ 0	
	2	IFT	I1	Print target vs firing point table (if ≠ 0)	
	3	IFG	I1	Print target vs gun type table (if ≠ 0)	
	4	IGT	I1	Print gun type vs target table (if ≠ 0)	
	5	IGF	I1	Print gun type vs firing point table (if ≠ 0)	
	6	I01	I1	Do not print "Extraneous Data" message if ≠ 0	
MAP-3	1-10	DAYS	F10.0	Number of days of information in data base	
MAP-4*	1-2	N	I2	Number of grid sizes to be tested	Card should be repeated N times as specified by MAP-4
MAP-5	1-10	GRDSZ	F10.0	Grid size	

PREVIOUS CALLS REQUIRED: None

REQUIRED FILES: TAPE7 data base

OUTPUT: Printed report

*There must be one MAP-4 card and at least one MAP-5 card.

Table 13
PLOT Input Summary

CARD INPUT:

Card ID	Columns	Variable Name	Format	Description	Comments, Defaults
NEF-1	1-6	IMETER	A10	Distance Units "METERS"/"FEET"	
		. . . CALLS TO OTHER MODULES			
PLT-1	1-4	BDR	A10	"PLOT"	
PLT-2	1	PUDG	I1	"1" if PUDDLE GRID output (TAPE1) to be used	"0"
	2	LOC	I1	"1" if LOCATOR output (TAPE3) to be used	"0"
	3	SCAT	I1	"1" if SCATTER output (TAPE4) to be used	"0"
	4	BAS	I1	"1" if BASE output (TAPE2) to be used	"0"
PLT-3	1-7	SCALE	F7.0	Scale	5000 ($1000 \leq x \leq 99999$)
	8-11	PERCX	F4.0	%X	1.0 ($0.01 \leq x \leq 9.0$)
	12-15	PERCY	F4.0	%Y	1.0 ($0.01 \leq x \leq 9.0$)
	16-19	MAG	F4.0	Magnification	1.0 ($0.01 \leq x \leq 9.0$)
	20-23	PERCSM	F4.0	% smoothing	0.333 ($0.01 \leq x \leq 9.0$)
	24-26	START	I3	First contour level to be plotted	55 ($1 \leq x \leq 999$)
	27-29	STOP	I3	Last contour level to be plotted	75 ($1 \leq x \leq 999$)
	30-32	LSTART	I3	First contour level to be labeled	55 ($1 \leq x \leq 999$)
	33-35	LSTOP	I3	Last contour level to be labeled	75 ($1 \leq x \leq 999$)
	36-37	LABEL	I2	Label	1 ("1" = Labels, "-1" = No Labels)
	38-39	INC	I2	Contour increment	5 ($0 \leq x \leq 99$)
	40-41	LINC	I2	Label increment	5 ($0 \leq x \leq 99$)
Repeat PLT-4 and PLT-5 cards as much as needed					
PLT-4	1-10	X	F10.0	X-coordinate starting location of text	
	11-20	Y	F10.0	Y-coordinate starting location of text	
	21-30	HT	F10.0	Height	
	31-40	ANGLE	F10.0	Angle	0°
	41	IC	I1	0-plotter coordinates (inches) 1-MAP coordinates	Default = 0
	42-79	TEXT(I) (I=1,38)	38A1	Text (terminated by $)	
	80	STAR	A1	"*" indicates last text card	
PLT-5	1-10	HT1	F10.0	Height (if different from preceding)	
	11-79	TEXT(I)	69A1	Text continued (terminated by $)	Additional text for labeling
	80	STAR	A1	"*" if last card	
PLT-6	80	STAR	A1	"*" if last card, follows PLT-3 if PLT-4 and PLT-5 are not included	This card must be included

PREVIOUS CALLS REQUIRED: BOUNDS, PUDDLE GRID, BASE, LOCATOR, SCATTER (if they are used)

REQUIRED FILE: TAPE1, TAPE2, TAPE3, TAPE4 (if they are used)

OUTPUT: Printed report
TAPE55 NASAPLOT input file

Table 14
POINT Input Summary

CARD INPUT:

Card ID	Columns	Variable Name	Format	Description	Comments/Defaults
NEE-1	1-6	IMETER	A10	Distance Units "METERS"/"FEET"	
		CALLS TO OTHER MODULES			
PNT-1	1-10	HDR	A10	"POINT"	
PNT-2	1-10	RINV1	F10.0	Inversion factor 1	SURFACE
	11-20	RINV2	F10.0	Inversion factor 2	1-500 m
	21-30	RINV3	F10.0	Inversion factor 3	1-3000 m
	31-40	IBOTH	A10	"DAY", day noise only; "NIGHT," night only; "BOTH," both.	
		PNT-3 is to be repeated as much as needed			
PNT-3	1	IFLAG	A1	"*" indicates last PNT-3 card	
	2-10	Name	A9	Name of location X, Y	
	11-20	X	F10.0	X-coordinates	
	21-30	Y	F10.0	Y-coordinates	

PREVIOUS CALLS REQUIRED: BOUNDS

REQUIRED FILES: TAPE8 intermediate results from FORMA

OUTPUT: Printed report

Table 15
PUDDLE GRID Input Summary

CARD INPUT:

Card ID	Columns	Variable Name	Format	Description	Comments/Defaults
NEE-1	1-6	IMETER	A10	Distance Units "METERS"/"FEET"	
		CALLS TO OTHER MODULES			
PGRD-1	1-10	HDR	A10	"PUDDLE GRID"	
PGRD-2	1-10	RINV1	F10.0	Inversion factor 1	SURFACE
	11-20	RINV2	F10.0	Inversion factor 2	1-500 m
	21-30	RINV3	F10.0	Inversion factor 3	1-3000 m
	31-40	GRDSZ	F10.0	Grid size in meters or feet	
	41-50	IBOTH	A10	"DAY," day noise only; "NIGHT," night only, "BOTH," both	
	51-60	GRDNAME	A10	Name to be assigned to TAPE1 PUDDLE GRID	

PREVIOUS CALLS REQUIRED: BOUNDS

REQUIRED FILES: TAPE8 intermediate results from FORMA

OUTPUT: Printed report

TAPE1 PHS2 input to NASAPLOT to be processed by PLOT routine

Table 16
SCATTER Input Summary

CARD INPUT:

Card ID	Columns	Variable Name	Format	Description	Comments/Defaults
NET-1	1-6	IMETER	A10	Distance Units "METERS"/"FEET"	
	CALLS TO OTHER MODULES				
SCT-1	1-7	IHDR	A10	"SCATTER"	
SCT-2	1-4	IK(X) (X=1,4)	4A1	Combination (in any order) of the letters "T" = SCATTER information collected for *targets* "F" = SCATTER information collected for *firing points* "B", "D", "N" = *both day and night data, day only, night only*, respectively "G" = only specified gun types (from columns 5-60)	Default = "TFB"
	5-6	IDGUN(1)	A2	If "G" selected, these are the specified gun types	There should be no blank fields interspersed among IDs as scanning stops at first blank field encountered
	7-8	IDGUN(2)	A2		
	9-10	IDGUN(3)	A2		
	59-60	IDGUN(28)	A2		
	61-70	FACT	F10.0	Multiplier = SCATTER points will be multiplied by the factor before dividing by days	1
	71-80	SD	F10.0	Standard deviation (meters/feet)	(300.)
SCT-3	1-10	DAYS	F10.0	Number of days in data base	1

PREVIOUS CALLS REQUIRED: BOUNDS

OUTPUT: Printed report
TAPE4: PHS4 input to NASAPLOT, input to PLOT routine

Table 17
STOP Input Summary

CARD INPUT:

Card ID	Columns	Variable Name	Format	Description	Comments/Defaults
NET-1	1-6	IMETER	A10	Distance Units "METERS"/"FEET"	
	CALLS TO OTHER MODULES				
STP-1	1-4	IHDR	A10	"STOP"	Closes NASAPLOT input file

PREVIOUS CALLS REQUIRED: All modules for given USER'S RUN

REQUIRED FILES: None

OUTPUT: Printed report
TAPE55 = input to NASAPLOT

Table 18
Temperature Inversion Factors

Location	Surface	1-500 m	1-3000 m	Location	Surface	1-500 m	1-3000 m
Albany, NY	45.1	20.1	45.0	Little Rock, AR	64.0	13.4	29.8
Albuquerque, NM	71.9	6.0	11.4	Medford, OR	76.7	5.1	13.9
Amarillo, TX	73.2	14.2	21.9	Miami, FL	60.6	6.7	24.3
Anchorage, AK	55.2	13.3	25.1	Midland, TX	65.8	15.5	27.9
Annetle, AK	30.9	3.0	26.2	Montgomery, AL	66.5	12.1	27.0
Athens, GA	70.1	13.3	23.5	Nantucket, MA	46.6	18.3	46.5
Barter Island, AK	48.1	34.3	48.7	Nashville, TN	66.0	10.5	27.8
Bismark, ND	64.3	18.0	32.0	New York, NY	27.8	22.0	56.8
Boise, ID	79.7	4.5	8.5	Nome, AK	65.7	8.4	26.2
Brownsville, TX	61.0	9.5	30.4	North Platte, NE	65.7	16.5	29.9
Buffalo, NY	44.9	10.5	39.4	Oakland, CA	43.4	21.3	49.0
Burwood, LA	17.1	5.6	28.9	Oklahoma City, OK	63.4	17.0	30.9
Cape Hatheras, NC	44.7	9.8	36.6	Omaha, NB	64.1	19.6	32.6
Caribou, ME	44.2	20.6	45.6	Peoria, IL	68.2	14.4	27.6
Charleston, SC	69.7	14.2	23.0	Pittsburgh, PA	58.2	10.5	32.9
Columbus, MO	65.2	14.5	29.2	Point Barrow, AK	46.7	34.3	49.9
Dayton, OH	60.7	11.6	29.6	Portland, ME	55.0	17.6	36.6
Denver, CO	82.8	4.3	12.3	Rapid City, SD	74.8	7.7	19.3
Dodge City, KS	72.6	15.4	24.1	St. Cloud, MN	55.3	21.4	39.9
El Paso, TX	65.6	4.7	14.1	Salem, OR	63.6	7.3	22.3
Ely, NV	91.6	.6	2.8	Salt Lake City, UT	83.4	3.7	6.4
Fairbanks, AK	71.5	6.8	17.6	San Antonio, TX	34.6	14.0	51.1
Flint, MI	53.2	15.2	36.5	San Diego, CA	47.3	26.8	50.0
Fort Worth, TX	45.8	25.0	48.0	San Juan, PR	44.7	1.3	24.5
Glasgow, MT	73.9	10.9	20.1	Santa Monica, CA	42.2	26.6	53.0
Grand Junction, CO	84.0	1.3	3.7	Sault Sainte Marie, MI	53.9	15.4	36.5
Great Falls, WI	59.0	13.7	33.6	Seattle, WA	52.2	5.2	25.0
Green Bay, WI	59.0	13.7	33.6	Shreveport, LA	55.9	17.5	36.0
Greensboro, NC	65.9	12.4	25.1	Spokane, WA	70.0	14.1	19.0
Hilo, HI	85.1	.3	8.1	Tampa, FL	67.7	7.6	18.0
Huntington, WV	60.0	7.3	30.6	Tatoosh Island, WA	23.1	12.1	43.7
International Falls, MN	59.6	14.4	33.7	Topeka, KS	53.5	23.6	40.9
Jackson, MS	64.2	15.8	29.8	Tucson, AR	89.6	1.4	4.0
Jacksonville, FL	63.7	11.2	22.8	Wallops Island, VA	57.9	13.8	34.2
Lake Charles, LA	79.2	9.1	18.5	Washington, DC	67.0	7.7	26.9
Lander, WY	84.6	1.2	6.1	Winnemicca, NV	88.3	1.5	3.7
Las Vegas, NV	84.2	1.1	5.8	Winslow, AR	88.0	2.1	4.9
Lihue, HI	24.0	.1	62.6	Yukatat, AK	57.1	2.1	14.8

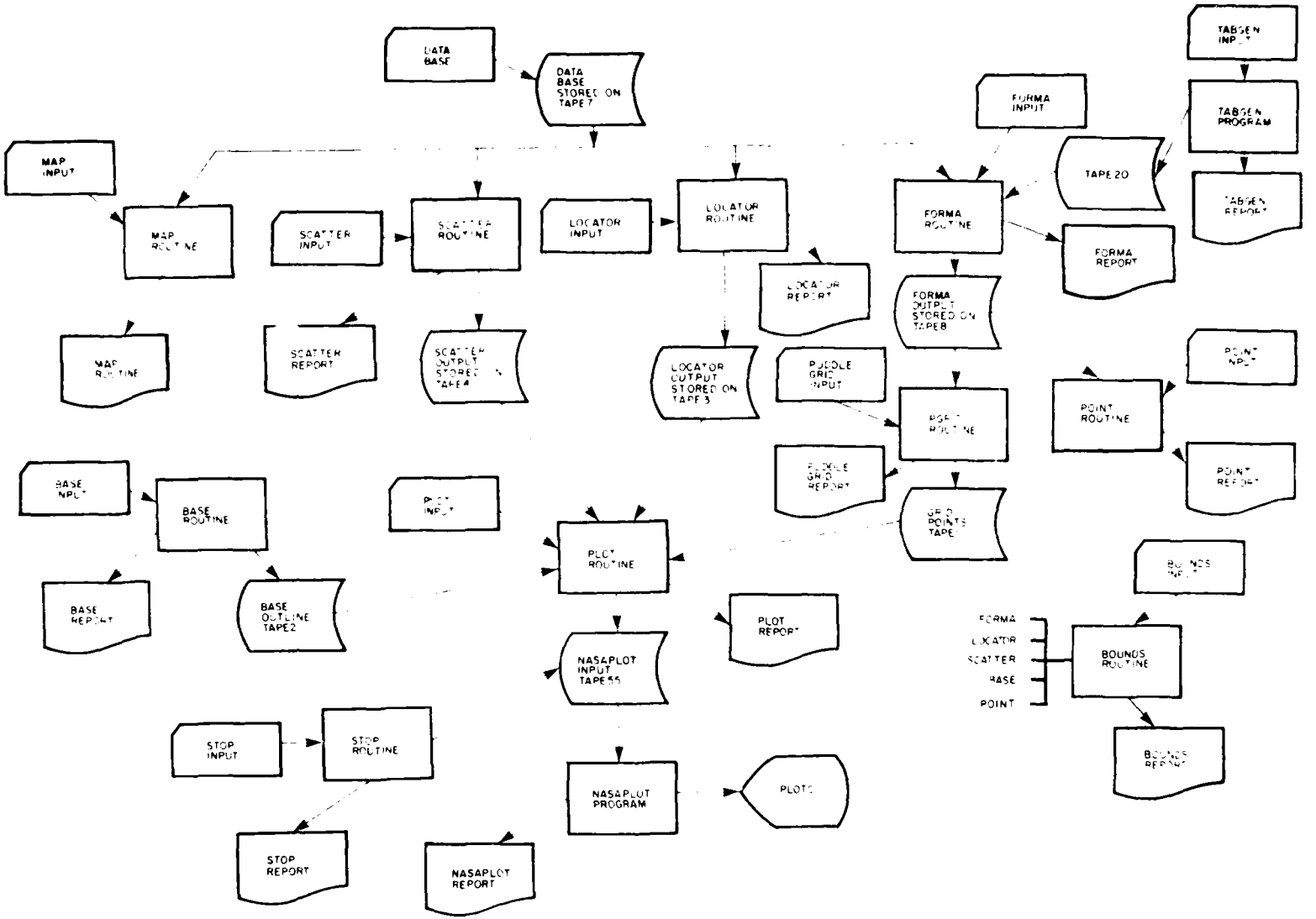


Figure 1. System flowchart.

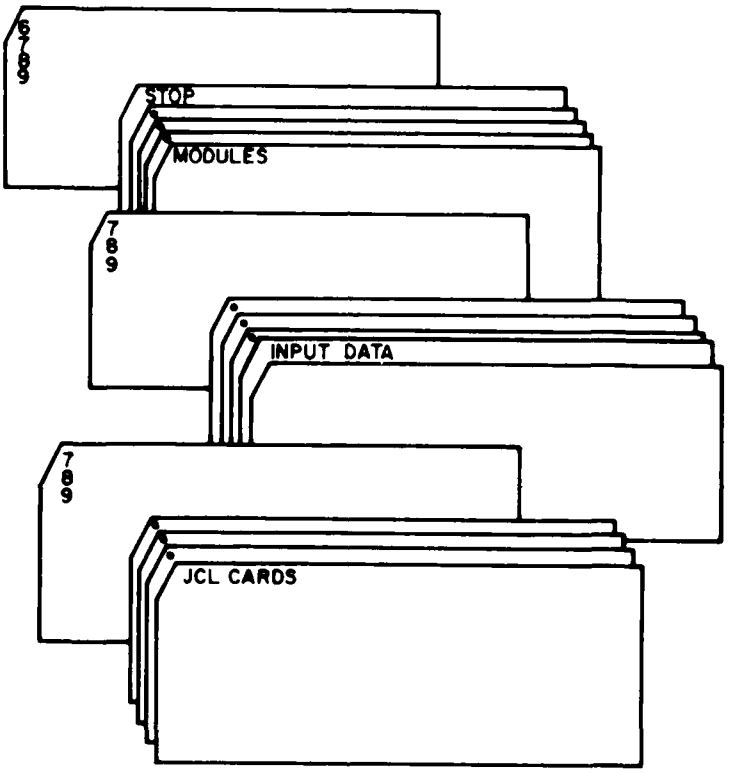


Figure 2 Order of sections within the deck

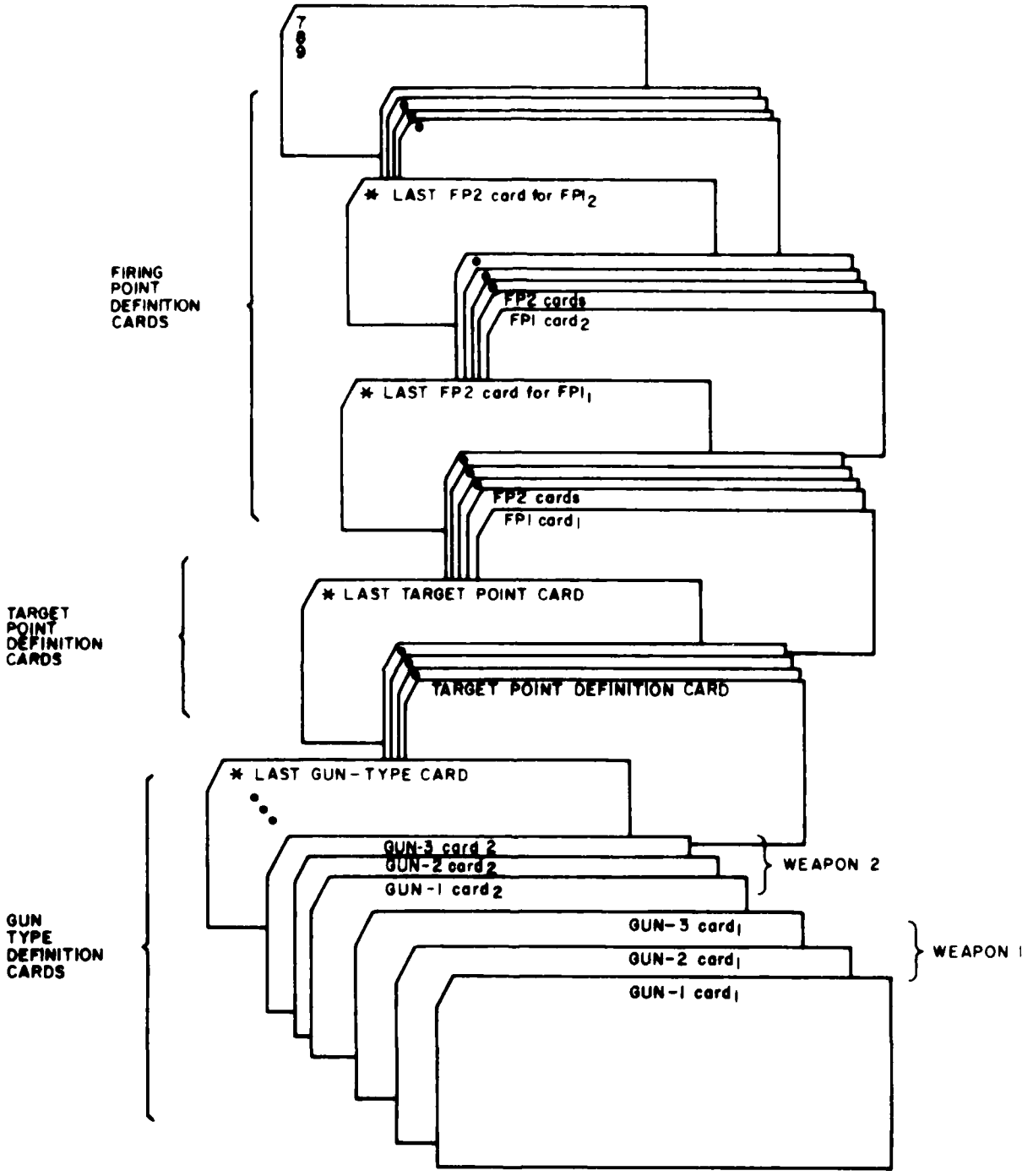


Figure 3. General composition of input data of the user's deck.

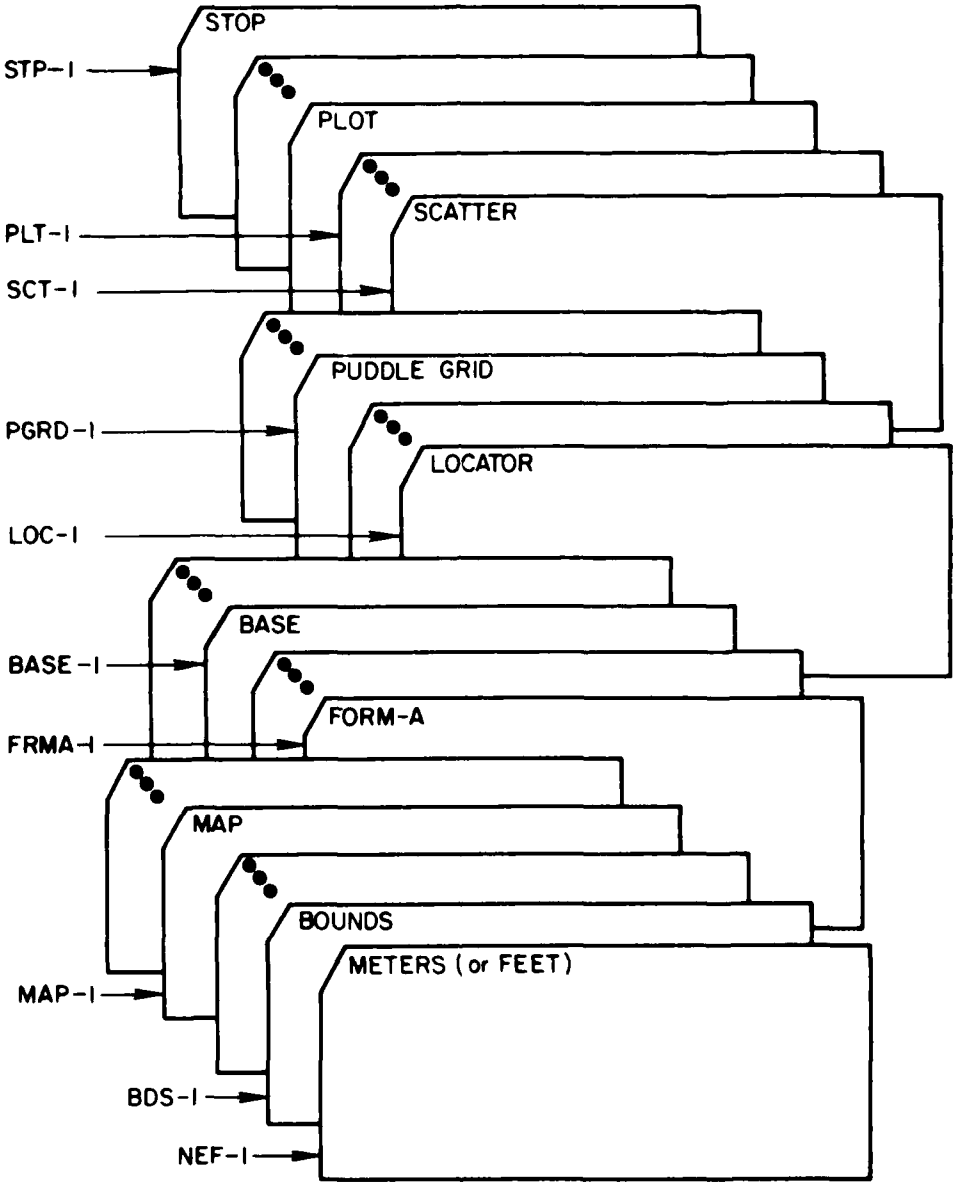


Figure 4. Sample composition of module.

3 PROGRAM AND SUBPROGRAM DESCRIPTIONS

This chapter describes the Fortran routines used by the Blast Noise Prediction program. The control program (CDNL) is described first, followed by descriptions of the subroutines and functions in alphabetical order. Flowcharts are included for some of the routines. Table 19 lists the common blocks used by the Blast Noise Prediction program.

The Blast Noise Prediction program uses a number of standard Fortran functions as well as the CDC routines:

ENCODE - memory-to-memory I/O

SECOND -- CPU time

EOF -- end of file test

EXIT -- program termination

DATE -- get date

TIME -- time of day

GETJN -- get job name

Any subprogram references are noted in the "subprograms/routines called" entries in the following sections.

Program CDNL

CDNL is the control program for the Blast Noise Prediction program. It calls the appropriate subroutines to perform the function(s) requested by the user.

Input: "NEF-1" card; module instruction cards.

Function/method: CDNL does program initialization, establishes certain program-wide default values, and sets the distance units (meters or feet) to be used for the execution according to the user's first input card (NEF-1). This should be followed by the first module call card. If this contains a valid module call -- MAP, FORMA, BASE, and so on -- CDNL branches to the corresponding calling sequence. Following execution of the requested module, control returns to CDNL which then processes the next module call card.

Subprograms/routines called: BASE, BOUNDS, FORMA, LOCATR, MAP, PGRID, PLOT, POINT, SCATTR, STOPP; CDC routine EOF.

Output: printed report.

Error diagnostic: "INVALID MODULE NAME". Read a module card and could not find it as an acceptable module name. Program halts.

Flowchart: Figure 5.

Subroutine BASE

Input: module instruction cards.

Function/method: creates the outline of the installation or region the user is working with on the PLOT output by drawing line segments from (X,Y) coordinate to (X,Y) coordinate of the specified region. The coordinates are read one at a time and checked for validity to ensure that they are within the boundaries specified in the BOUNDS module. The first (X,Y) coordinate is read and temporarily saved until the second (X,Y) coordinate is read and saved, at which time both coordinates are output to the NASAPLOT file as a pair of endpoints for a line segment. This pair of endpoints is also listed in the printed report to help the user verify that his input data are producing the intended output. The

next (X,Y) coordinate is then read and saved, and the second pair of endpoints is output to the NASAPLOT file and the printed report. This continues as long as the user requests a continuous line. When a break occurs (indicated by a "*" in column 21 -- see Table 8), two new coordinates are read, and the pairing starts again.

Subprograms/routines called: Fortran function SIGN, CDC routines EOF, SECOND.

Output: TAPE2 (NASAPLOT, PHS4, line cards), printed report.

Error diagnostics:

1. "***NEXT CARD NOT WITHIN BOUNDARIES***" indicates an X or Y coordinate of a line segment instruction in BASE extends beyond the minimum or maximum values specified in BOUNDS.

2. "ERROR - NO EOF CARD (* IN CC 21)
JOB ABORTED" indicates that the user forgot to include a BASE-3 card with a "*" in column 21 to indicate that the input for the BASE module has been completed.

3. "***DUE TO BOUNDARY ERRORS NO OUTPUT***
TAPE WAS CREATED THIS RUN" indicates that a type 1 or 2 error has occurred. The BASE module will have no output that can be used by the rest of the program.

Subroutine BOUNDS

Input: module instruction cards.

Function/method: sets the limits of the system in terms of map coordinates ("system" refers to the total area encompassed for CDNL calculations and for creation of the PLOT. It is used to define the overall area of interest), tests for valid coordinates and terminates execution if they are not found.

Subprogram/routines called: CDC routines EOF, SECOND.

Output: printed report.

Error diagnostics:

1. "*INCORRECT BOUNDARIES*
NO BOUNDS SET" indicates a mistake in specifying the boundaries -- the maximum values are less than the minimum ones.

2. "*ERROR -- MISSING INPUT CARD, JOB ABORTED*" indicates that a BOUNDS card is missing, the module has not been supplied all of its data.

Subroutine BDSET

This routine is used to set the BOUNDS. It calls Fortran routines ABS, INT. There is no error diagnostic.

Subroutine CALCNR (X,Y)

CALCNR calculates the CDNL value at a grid point (X,Y). It is used by the PGRID routine. For theoretical background, see CERL TR E-17 *Predicting Community Response to Blast Noise* and Volume I.

Input: TAPE8, via COMMON.

Function/method: the calculations proceed in the manner outlined in Chapter 5 involving the equations presented there. The CDNL value has the name CNR in the program. The height and weight factors are calculated in FORMA and found as SDBWH from TAPE8.

Subprograms/routines called: Fortran functions ALOG10, SQRT, ATAN2, IFIX.

Output: CDNL value for point specified.

Error diagnostics:

1. If both day and night cumulative noise sum = 0, CNR set to -99.0.
2. If (X,Y) is closer than 100 m to a noise source, it is limited to 100 m.
3. If (X,Y) is further away than 100 000 m, it is limited to 100 000 m.

Flowchart: Figure 6

Subroutine CFOUR

Calculates the C-weighted noise level for an explosion in the air. Calls Fortran function ALOG10. There is no error diagnostic.

Subroutine FORMA

FORMA tabulates the target and firing point information from the data base (TAPE7) in the form required by PUDDLE GRID.

Input: module instruction cards, TAPE7 (data base).

Function/method: the data base consists of an ID number and location for each target, and location coordinates for each firing point followed by the definition cards for all the noise sources at that site. These cards give (1) the gun type, (2) number of day and night firings, (3) maximum and minimum charge zones, (4) target ID or an indication that the source is omnidirectional, (5) an indication of whether the noise is produced at the target, and (6) how high or low the explosion is at the target.

For each data entry, FORMA calculates the sine (ANGSIN) and cosine (ANGCOS) of the noise direction, using value 999.0 to flag omnidirectional sources (including targets). It also calculates the decibel charge correction factors for TNT weight (DBWT) and height (DBHT). The last two are combined in array SDBWH. From this information, FORMA compiles a table with an entry for each given noise source, i.e., each given firing-point-target/height charge or target-charge combination (a target also being a source of omnidirectional noise when hit). For each entry, FORMA accumulates the number of day and night firings with that charge from (firing point) or toward (target point) that location. Charge is expressed in equivalent weight of TNT derived from the gun type and maximum and minimum charge zones, using the averaging techniques specified by the user.

The array CHARGE contains the equivalent TNT weights, one column is allowed per gun type, with rows corresponding to charge zones. Depending on the technique specified, the charge figure used in calculations may be the weight corresponding to the maximum charge zone (MAX), the weight corresponding to the average of maximum and minimum zones rounded to the next highest integer (AVE), or may be the average of the two weights corresponding to maximum and minimum charge zones (CAVE).

A header record of basic information (number of unique noise sources, number of days of data in the data base, coordinates of grid origin) is written to TAPE8, followed by the arrays:

XLOC	= X coordinates of sources
YLOC	= Y coordinates
SDBWH	= weight and height correction factors
ANGSIN	= sines of noise direction
ANGCOS	= cosines of noise direction
DAYNO	= number of day firings
DARKNO	= number of night firings

Subprograms/routines called: READTB, CFOUR; Fortran functions ALOG10, SQRT, ABS, MAXO, IABS, AMINI; CDC routines EXIT, SECOND, EOF.

Output: printed report, TAPE8.

Error diagnostics:

1. "...WARNING GUN IS POINTING AT SELF. LOCATION _____" indicates that the firing and target points have been defined to be the same spot at the specified location. Not the proper way to code a demolition.

2. "...ERROR... ALL SOURCES ARE TARGETS NUMBER OF SOURCES COUNTED ARE ____" indicates that no firing points have been specified. The user/programmer can have all firing points if all rounds are demolitions without targets, but not the other way around. If there are no target point cards, a blank card with an "*" is still required to indicate the end of the target data.

3. "...ERROR...NUMBER OF SOURCES WILL EXCEED SPACE NUMBER OF SOURCES NOW IS ____" indicates that too much data have been provided. Current limit is 2000 unique noise types.

4. "...ERROR... UNDEFINED TARGET ID FOR FIRING PT ____" indicates that a firing point is shown to have a target point which has not been defined in the target-point definition cards. Often caused by typing error or table overflow.

5. "...ERROR...UNDEFINED GUN ID, FOR FIRING PT ____" indicates that a firing point is shown to have a gun code which has not been defined in the gun-type cards. Could be caused by a typing error.

6. "...ERROR...EOF ENCOUNTERED WHILE PROCESSING GUN TYPE TABLE DATA" indicates that a "789", signaling end of input, has been found in the middle of the gun-type cards.

7. "...ERROR...GUN TYPES EXCEED TABLE LIMIT" indicates that too much data have been input. Current limit is 50.

8. "...ERROR...BLANK TARGET ID, OMNI FLAG NOT SET FOR SOURCE AT FIRING POINT ____" indicates that the demolition flag for a firing point has not been set and that no target has been specified for it. One of the preceding **must** be included.

Flowchart: Figure 7.

Subroutine LOCATR

Input: module instruction cards, TAPE7.

Function/method: labels target and firing points on the plot.

Subprograms/routines called: CDC routines EOF, SECOND.

Output: printed report, TAPE3 (PHS3 text cards).

Error diagnostic:

"*PREMATURE EOF ON DATA BASE FILE* *TAPE 3 NOT CREATED*" indicates that a "789" has been encountered in the wrong place.

Subroutine MAP

This routine is a checker for the data base.

Input: module instruction cards, TAPE7 (data base).

Function/method:

1. Reads and optionally prints target card images from TAPE7, checking for maximum and minimum coordinate values.

2 Reads and optionally prints firing point location and definition card images, continuing search for maximum and minimum coordinate values, and checking that target IDs are valid (i.e., previously encountered in No. 1 above)

3 Builds cross-reference tables and prints basic statistics.

4 If user inputs trial grid sizes, calculates size in grid units

Subprograms/routines called PUTXR, Fortran functions, ABS, MINO, CDC routines EOF, SECOND, SHIFT

Output printed report. Note: MAP will print a series of stars ("*****") if a number exceeds output format, but is within the limit of the input format.

Error diagnostics

1 "ERROR--EOF ENCOUNTERED WHILE READING TARGET CARDS" indicates that a "789", signaling end of input data, has been found in the middle of the target-point cards.

2 "ERROR--SOURCE DEFINITION CARDS ENDS IMPROPERLY (WITH A EOF)..." indicates that a "789" was encountered at the end of a set of definition cards instead of a star (*) to indicate the last of such cards

3 "ERROR--UNDEFINED TARGET ID" indicates that a target-point name entered on a firing-point card had no equivalent defining data in the target cards.

4 "ERROR--GUN TYPES EXCEED TABLE LIMIT IN MAP; ONLY FIRST x TYPES USED FOR SUBSEQUENT CROSS-CHECKING" indicates that not enough space is allowed for all cross-reference checking and that there are too many gun-type cards.

5 "ERROR--UNDEFINED GUN ID" indicates that a gun code entered on a firing-point card had no equivalent defining data among the gun-type cards. Often caused by typing errors or table overflow

6 "ERROR--BLANK TARGET ID, HIT FLAG NOT SET. FIRING PT. __" indicates that the demolition flag for a firing point has not been set and that no target has been specified for it. One of the preceding **must** be included

7 "ERROR"--EOF ENCOUNTERED WHILE READING GUN TYPE DEFINITION CARDS" indicates that an error was encountered while reading the gun-type cards instead of at the end of the data set

8 "ERROR--NO POSITIVE CHARGE FOR GUN TYPE x" indicates that a gun with code x has been defined with all zero or negative charge sizes. Improper definition.

9 "ERROR--x NEG. CHARGES ENCOUNTERED" indicates that x number of negative charge sizes were found among the gun-type cards

10 "WARNING--x CHARGES LARGER THAN ___ LBS" indicates that x number of charge sizes were greater than a given size. Currently the message is for 50 lb (23 kg).

11 "ERROR--DUPLICATE ID; FIRST OCCURRENCE USED FOR TABLE" indicates that the same firing point ID occurs in two (or more) places in the input data. Information from only the first appearance is used by MAP in the cross-reference tables. In this case, a new ID (such as "jj") will be generated and used in the cross-reference table for this firing point.

12 "WARNING--EXTRANEOUS DATA STARTING IN CARD COL., x,, CHECK ALL FIELDS" indicates that characters for this definition card are found in columns where they are not supposed to occur. This error is sometimes caused by an overturned card or a missing FP1 card. This message can be suppressed

13 "ERROR--HEIGHT CORRECTION DATA OUT OF RANGE" indicates that the specified height correction factor is too big. Sometimes caused by typing the height correction factor in the wrong column, causing a shift in the decimal point.

14 "WARNING--DUP-POINT IDENTICAL COORDINATES" indicates that the same point has occurred a second time in the input data. Caused by different IDs for the same coordinates.

15. "ERROR--TARGETS EXCEED TABLE LIMIT IN MAP, ONLY FIRST x USED FOR SUBSEQUENT CROSS-CHECKING" indicates too much input data; only the first x is allowed in the table.

16. "ERROR--FIRINGS DATA NEGATIVE OR BOTH ZERO ON DEF CARD FOR FIRING POINT x" indicates that no rounds or negative number of rounds have been specified for the amount fired per day and night for firing point x. Caused by blanks in both the day and night columns on a FP2 card, or entry of a negative number.

17. "ERROR--INVALID CHARGE NO., NONPOSITIVE OR NO GUN TABLE ENTRY, DEF CARD FOR FIRING PT "x indicates that a gun for firing point x has been improperly defined with a negative or zero charge size for the specified range, or that the charge range is wrong for the specified gun type.

18. "WARNING--LARGE HEIGHT IN DEF CARD FOR FIRING POINT x" indicates that the specified height correction factor is very big. User should check accuracy of input data.

19. "ERROR--FIRING POINTS EXCEED TABLE LIMIT IN MAP; ONLY FIRST x USED FOR SUBSEQUENT CROSS-CHECKING" indicates that too much input data are given. Current limit is x.

20. "WARNING--DUE TO PREVIOUS TABLE OVERFLOW, THE FOLLOWING CROSS-REFERENCE TABLE IS INCOMPLETE" indicates that one set of definition cards was too big, so MAP used only part of it in making the given table.

21. "ERROR--DUPLICATE ID, COORDINATES; DEF CARDS CHECKED FOR ERRORS, BUT OTHERWISE IGNORED" indicates that the same firing point has been specified more than once. MAP ignores data for uses in cross-reference tables, but checks data for validity.

22. "ERROR--DUPLICATE ID, DIFFERENT COORDINATES; TREATED AS SEPARATE ENTRY" indicates that the same firing-point ID is defined with different coordinates and is considered as two points. A new firing point ID (such as "j*") has been created and will be used in the cross-reference tables.

23. "ERROR--NONPOSITIVE CHARGE NOS. ENCOUNTERED FOR FIRING PT.__" indicates that the specified charge ranges are negative.

24. "ERROR--NONPOSITIVE TARGET CHARGE IN TABLE FOR GUN x; DEF CARD FOR FIRING POINT y" indicates that the projectile charge size for a given gun x specified for a firing point y is negative or zero.

25. "ERROR--MISSING DATA BASE FILE; EXECUTION ABORTED" indicates that a USER'S RUN program was created without an input data section being provided.

Subroutine PGRID

This routine outputs a rectangular printed grid of CDNL values as specified by the user in BOUNDS.

Input - module instruction cards, TAPE8 (read in by READIN), TAPE20 (read in by READTB).

Function/method

1. Reads from the module instruction cards the grid size, percent inversion values, choice of day or night calculations, or sets the defaults. Outputs the values read in and the limits of the grid area as set by BOUNDS.

2. Superimposes a grid on the specified area by dividing it into squares according to the size specified by GRIDSIZE.

3. CALCNR is called for each integral grid point within the specified area. The CDNL values are printed out in matrix form and output to TAPE1 for use by PLOT.

Subprograms/routines called - READIN, CALCNR, BDSET, READTB; Fortran functions AMOD, MOD; CDC routines EOF, SECOND.

Output - printed report, TAPE1

Error diagnostics

1. "***WARNING-SPECIFIED BOUNDS x,y, x,y DO NOT CORRESPOND TO INTEGRAL GRID BOUNDS. MODIFIED BOUNDS WILL BE USED TO PRODUCE THE GRID AND TO DEFINE ANY PLOT UTILIZING THIS GRID" indicates that the grid size specified by the user does not divide the bounds evenly. PUDDLE GRID extends the bounds, so grid size will divide evenly. (See PLOT errors.)

2. "***WARNING--GRID SIZE NOT MULTIPLE OF " indicates that a possible inappropriate grid size was chosen by the user.

3. "***ERROR--MISSING INPUT DIRECTIVE JOB ABORTED" indicates that one of the module cards is missing.

Subroutine PLOT

This routine combines all of the output from other modules into one tape for use by NASAPLOT.

Input - module instruction cards, TAPE1 (from PUDDLE GRID), TAPE2 (from BASE), TAPE3 (from LOCATOR), TAPE4 (from SCATTER).

Function/method

1. Reads which module output to use from instruction cards.
2. Reads plot parameters for use by NASAPLOT.
3. Reads appropriate files and outputs all to TAPE55 for use by NASAPLOT.

Subprograms/routines called - Fortran functions, ATAN, COS, SIN, SQRT; CDC routines EOF, DATE, SECOND, TIME, GETJN.

Output - printed report, plot.

Error diagnostics

1. "THE FOLLOWING FILES WERE REQUESTED BUT NOT AVAILABLE. JOB ABORTED." indicates that output from BASE, SCATTER, LOCATOR, or PUDDLE GRID was requested in the PLOT module without having been produced previously by one of these modules.

2. "WARNING - PUDDLE GRID BOUNDS DO NOT MATCH SPECIFIED BOUNDS.

PGRID VALUES USED	SPECIFIED BOUNDS
A	A2
B	B2
C	C2
D	D2

This message indicates that the user specified one set of bounds while PUDDLE GRID used some other values, usually as a result of the specified grid size. For example, if the x values of bounds were 20,000 and 43,000 and if the grid size was 2000, then the new bounds are 20,000 and 44,000 since the difference between the two must be a multiple of 2000. This error can also occur if PUDDLE GRID was saved on disk and the program later run again.

Subroutine POINT

This routine will output CDNL values at individual points as specified by the module instruction cards.

Input - Module instruction cards, Tape8 (read in by READIN), Tape20 (read in by READTB).

Function/method

1. Reads input outputs from the module instruction cards the percent inversion values, choice of day, or night calculations.

2. Reads the coordinates and names of the desired points. Reads each point as it calculates the CDNL and outputs it.

Subprograms/routines called: CALCNR, READIN, READTB; CDC routines SECOND, EOF.

Error diagnostic: none.

Subroutine READIN

This routine reads the data on TAPE8 into arrays in COMMON for use by other modules.

Input: TAPE8.

Subprograms/routines called: none.

Error diagnostic: none.

Subroutine READTB

Input: TAPE20.

Function/method: reads TAPE20 (TABGEN) tables; corrects it to new inversion factors.

Subprograms/routines called: Fortran function ALOG; CDC routine EOF.

Error diagnostic: none.

Subroutine SCATTR

Input: module instruction cards, TAPE7 (data base).

Function/method: this routine will write text cards out to tape so that the PLOT routine can plot them out as a scattergram. There will be one dot for each noise source that occurs per day for the data which are allowed by the specification cards. The dots of the scattergram will have a normal distribution with a standard deviation equal to the distribution value (found on specification card). The center of the normal distribution will be at the target or firing point location. No dots will be allowed outside of the limits established by the routine BOUNDS.

Subprograms/routines called: SCATPL; Fortran function SQRT, FLOAT; CDC routines EOF, SECOND.

Output: printed report, TAPE4.

Error diagnostics:

1. "***ERROR--TARGET TABLE OVERFLOW. EXECUTION ABORTED" indicates too much input data. Current limit is.

2. "***ERROR--ERROR IN DATA BASE. UNDEFINED TARGET ID--, EXECUTION ABORTED" indicates that a target point ID is given with no following defining information.

3. "***WARNING--EITHER TARGETS OR FIRING PTS MUST BE REQUESTED; EXECUTION CONTINUES WITH DEFAULT = BOTH"

4. "***ERROR--PREMATURE EOF ON TAPE 7, EXECUTION ABORTED" indicates that a "789" specification is encountered in the wrong spot.

5. "***ERROR-GUN OPTION SELECTED BUT NO GUN TYPES SPECIFIED. EXECUTION ABORTED"

Subroutine SCATPL

Generates the scattergram dots.

Subprogram/routines called: CDC routine RAND.

Output: TAPE4.

Error diagnostic

1 "***WARNING--GENERATED SCATTER POINT FOR LOCATION X, Y OUT OF BOUNDS AFTER N TRIES PT IGNORED." The computer could not generate a scatter point for location X, Y without it going outside of the limits set by BOUNDS. The point has not been plotted. A target or firing point may be too close to the limits set by BOUNDS.

Subroutine STOPP

Stops the program.

Subprograms/routines called none

Output Tape55, printed report

Error diagnostic none

Program TABGEN

TABGEN creates the tables of decibel values at given distances which are used in the calculation of CDNL by CALCNR. The tables are stored on TAPE20 and listed in Table 20.

Program NASAPLOT*

This section presents the control cards involved in the NASAPLOT contour program used by the Blast Noise Prediction program. It describes some of the aspects involved in the creation of the plot by computer methods. The NASAPLOT contour program operates in four distinct phases which can occur in or out of sequence. Program control can be initialized, transferred from one phase to another, and terminated by means of certain control cards. A description of these cards and their formats follow.

Job Card

cc 1

JOB Format (A4)

The JOB card must precede all data cards of a job. It is the initialization card for each job.

PHS1 Card

cc 1

PHS1 Format (A4)

The PHS1 card transfers control to Phase 1 of the program. Phase 1 introduces dimensions of the plot size, requests for a new plot page, angles of skew and rotation, map size, scale and position of origin, map scale factor, and parallax parameters. Phase 1 control cards are as follows:

CSYS Card

cc 1	7 14	15 22	
CSYS	SKEW	ROTATE	Format (A4,2X,2G8.3)

where

SKEW (col 7-14) is the rotation of the Y-axis in degrees; default angle is 0^{0}

* Source: National Aeronautics and Space Administration (NASA) Langley Research Center, Hampton, VA.

ROTATE (col 15-22) is the rotation of the X-axis in degrees; default angle is 0^{O}.

The CSYS card allows the user to use and specify a skewed and/or rotated coordinate system for the data control point. SKEW is the angle of skew, measured clockwise, and ROTATE is the angle of rotation measured counterclockwise, as illustrated in Figure 8. The shaded area in Figure 8 corresponds to values of X between XMIN and XMAX, and values of Y between YMIN and YMAX. The point (X,Y) will be plotted as plotter position (XP,YP).

where

```
XP = XOR + ((X - XMIN)/SCALE + SIN(SKEW) * (Y - YMIN)/
     YSCALE) * COS(ROTATE) - ((COS(SKEW) * (Y - YMIN)/
     YSCALE) * SIN(ROTATE)

YP = YOR + ((X - XMIN)/XSCALE + SIN(SKEW) * (Y - YMIN)/
     YSCALE) * SIN(ROTATE) - ((COS(SKEW) * (Y - YMIN)/
     YSCALE * COS(ROTATE)
```

PRLX Card.

```
cc 1    4    7    14   15   22    23  30
 PRLX    XPRLX  YPRLX  ZREF
```

Format (A4,2X,3G8.3)

where:

XPRLX (col 7-14) is the offset in X direction

YPRLX (col 15-22) is the offset in Y direction

ZREF (col 23-30) is the altitude to be used as the reference plane; default value is 0^{O}.

The PRLX card allows the user to introduce angles of parallax. This feature can be used to produce pairs of contour diagrams of the same data for stereo viewing. A stereo pair is produced by using the following procedure.

1. Use the MAPS card and the PRLX card for the first stereo view
2. Plot the first view by means of the PLOT card
3. If the two views are not to overlap, but are to be plotted side by side, use the MAPS card again to move the second view to a different position on the plotter page
4. Use the PRLX card for the second view
5. Use the PLOT card to plot the second view.

MAPS Card.

Format (A4,2X,8G8.3)

```
cc 1      7       15      23      31      39      47      55      63
 MAPS   XSCALE  YSCALE  XMIN    XMAX    YMIN    YMAX    XOR     YOR
```

where

XSCALE (col 7-14) is the scale factor in the X direction; default value is 1.0 units per inch of plot

YSCALE (col 15-22) is the scale factor in the Y direction, default value is 1.0 units per inch of plot

XMIN (col 23-30) is the minimum X value to be plotted

XMAX (col 31-38) the maximum X value to be plotted

YMIN (col 39-46) the minimum Y value to be plotted

YMAX (col 47-54) is the maximum Y value to be plotted

XOR (col 55-62) is the position on the plotter of the X origin, default value is 0.0 in.

YOR (col 63-70) is the position on the plotter of the Y origin; default value is 0.0

The MAPS card allows the user to specify the map size, map scale factor, and map origin parameters. The map will be determined for values of X between XMIN and XMAX and values of Y between YMIN and YMAX. The user may select the position on the plotter, (XOR,YOR), to which the point (XMIN,YMIN) will correspond.

SCLE Card

cc 1 7

SCLE PSCALE Format (A4,2X,G8.3)

where:

PSCALE (col 7-14) is the plotter scale factor; default value is 1.0.

The SCLE card allows the user to specify a scale factor, PSCALE, by which the map scale is multiplied before plotting. This scale factor is often used (in addition to XSCALE and YSCALE) to produce intermediate smaller size versions of a map, for consideration before a final full-scale version is attempted. The introduction of this scale factor does not change the position (XOR,YOR) on the plotter to which the point (XMIN,YMIN) will correspond.

DIMT Card

cc 1 7 15

DIMT XTABLE YTABLE Format (A4,2X,2G8.3)

where:

XTABLE (col 7-14) is the plot length in X direction in inches

YTABLE (col 15-22) is the plot length in Y direction in inches.

The DIMT card introduces plot size parameters, and can be used only once per job.

PAGE Card

cc 1

PAGE Format (A4)

The PAGE card allows the user to split the map into matching sections and to plot each section on a separate plotter page.

PHS2 Card

cc 1

PHS2 Format (A4)

The PHS2 card transfers control to Phase 2 of the program. Phase 2 accepts control points and computes the grid. Input unit, format, order for control points to be read, and grid mesh point insertion values are introduced. Phase 2 control cards are as follows:

CNPT Card. Format (A4, 2X, I3, 4I2, 5A10)

For control points being read from tape:

cc 1	7	10	12	14
CNPT	IPT	IX	IY	IZ

For control points being read from cards:

cc 1	7	10	12	14	16	18	67
CNPT	IPT	IX	IY	IZ	IL	KFMT	

where:

IPT	(col 7-9)	is the logical input device number (right justified)
IX	(col 10-11)	is the position of X values; normally 1 (right justified)
IY	(col 12-13)	is the position of Y values; normally 2 (right justified)
IZ	(col 14-15)	is the position of Z values; normally 3 (right justified)
IL	(col 16-17)	is the position of the end file flag; normally 4 (right justified); a value greater than or equal to .0001 terminates input
KFMT	(col 18-67)	is the format for the data to be read; it must include the parentheses.

The CNPT card allows the user to specify the format and the order of the X, Y, and Z coordinates of control points, as well as the input device used. NOTE: If control points are to be read from a RECIN tape, the data must be requested as TAPE9 on the tape request card. Input is terminated by an EOF on TAPE9.

GRID Card.

cc 1	7	15	23	
GRID	NROWS	MCOLS	NGH	Format (A4,2X,2I8,I3)

where

NROWS	(col 7-14)	is the number of rows (maximum 99, right justified)

MCOLS (col 15-22) is the number of columns (maximum 99, right justified)

NGH (col 23-25) is the number of neighboring points to use in determining the values at the grid points (maximum 10, right justified)

The GRID card causes a rectangular grid (parallelograms, if the coordinates are skewed) to be defined. NROWS and MCOLS are specified by the user -- there are NROWS times MCOLS grid cells, and (NROWS + 1) times (MCOLS + 1) mesh points in the grid. The grid mesh point values can be determined from control point input data, in which case NGH (on the GRID card) is used to specify the number of nearest control point neighbors to be used in calculating each grid mesh point value.

GRDI Card

cc 1	7	15	23	
GRDI	KROW	KCOL	VAL	Format (A4,2X,2I8,G8.3)

where

KROW (col 7-14) is the row requested (right justified)

KCOL (col 15-22) is the column requested (right justified)

VAL (col 23-30) is the value to be inserted.

The GRDI card is used to insert a value at a grid mesh point. The GRDI card must be used first to define the grid (the value of NGH is irrelevant in this case, and should be left blank), and then the GRDI cards may follow. NOTE: A combination of computed and inserted values at the grid mesh points is also allowed, in which case inserted values take precedence over computed values.

CPST Card

cc 1	7	15	23	25	
CPST	HCEN	HNUM	ND	IC	Format (A4,2X,2G8.3,2I2)

where

HCEN (col 7-14) is the height of the center mark

HNUM (col 15-22) is the height of the numbers

ND (col 23-24) is the number of digits (right justified)

IC (col 25-26) = 0 for center marks only
1 for center marks and numbers (right justified)

The CPST card allows the user to cause control points and altitudes to be posted on the map. Either the center marks (+) may be posted alone, or both the center marks and the values may be posted. The size of the center marks and the height of the numbers can both be selected by the user. NOTE: The CPST card must follow the GRID card.

GPST Card

cc 1	7	15	23	25	
GPST	HCEN	HNUM	ND	IC	Format (A4,2X,2G8.3,2I2)

where:

HCEN (col 7-14)	is the height of the center mark
HNUM (col 15-22)	is the height of the numbers
ND (col 23-24)	is the number of digits (right justified)
IC (col 25-26)	= 0 for the center marks only = 1 for center marks and numbers (right justified).

The GPST card allows the user to cause the grid mesh point values to be posted on the map. Either the center marks (+) may be posted alone, or both the center marks and the values may be posted. The size of the center marks and the height of the numbers can both be selected by the user. NOTE: The GPST card must follow the GRID card.

PHS3 Card

cc 1

PHS3 Format (A4)

The PHS3 card transfers control to Phase 3 of the program. Phase 3 plots contour lines and performs masking. Phase 3 control cards are as follows:

MASK Card

cc 1 7

MASK NPTS Format (A4,2X,I8)

where:

NPTS (col 7-14) is the number of points used to define the mask boundary.

The MASK card, used in conjunction with the MSKB card, allows the user to mask out areas within the rectangular boundary of the plot where contours are not to be plotted. This feature can be used to blank out portions of the map where the data are inaccurate or incomplete or to save certain areas for later contouring with different data. The boundary can define several independent regions, with or without holes, by choosing a boundary path appropriately. The boundary path can consist of line segments alone which the boundary path traverses in both directions. Figure 9 is an example of a boundary defining two independent regions, one of which contains a hole. The contours will be plotted in the area defined by grid cells whose centers are within the masking boundary (Figure 10). The MASK card specifies the number of boundary points contained by the MSKB cards.

MSKB Card Format (A4,2X,8G8.3)

cc 1	7	15	23	31	39	47	55	63
MSKB	XVAL(1)	YVAL(1)	XVAL(2)	YVAL(2)	XVAL(3)	YVAL(3)	XVAL(4)	YVAL(4)

where:

XVAL are the X coordinates of the masked area.

YVAL are the Y coordinates of the masked area.

The MSKB cards specify the coordinates of the boundary points. The boundary points must be introduced sequentially, such that the boundary is defined by the line starting from the first point and passing through each successive boundary point, and ending with a line between the last boundary point and the first. The boundary will thus form a closed curve within which contours will be plotted.

PLOT Card. Format (A4,2X,2G7.0,2G6.0,I5,I4,3I2,5G4.3,I1,G4.3)

cc	1	7	14	21	27	33	38	42	44	46
	PLOT	ZMIN	ZLMIN	DZ	DZL	NLEVS	KLEV	L1	L2	ID
		48	52	56	60	64	68	69		
		DISL	HGIL	DIST	TLNG	TLER	I	SKIP		

where:

ZMIN (col 7-13) is the minimum contour level requested

ZLMIN (col 14-20) is the minimum contour level to be labeled

DZ (col 21-26) is the increment to be used for plotting contour levels

DZL (col 27-32) is the increment to be used for plotting labeled contour lines

NLEVS (col 33-37) is the number of contour levels to be plotted (right justified)

KLEV (col 38-41) is the number of contour levels to be skipped before plotting a line of type L2 (right justified); KLEV-1 levels are skipped

L1 (col 42-43) is the line format for every contour line other than the KLEV contour line, values and line format are as follows (right justified):

± 1 = a solid light line
± 2 = a solid light line with tick marks on one side
± 3 = a solid light line with tick marks across the line

± 4 = a solid bold line
± 5 = a solid bold line with tick marks on one side
± 6 = a solid bold line with tick marks across the line
± 7 = a dashed line

Positive values include a label; negative values do not

L2 (col 44-45) is the line format for the KLEW contour line; values and line formats are the same as for L1

ID	(col 46-47)	is the number of digits after the decimal point in the label. If ID = -1, the decimal point will be omitted, values are right justified
DIST	(col 48-51)	is the distance (in inches) along a contour line from its beginning to its label
HGHT	(col 52-55)	is the height (in inches) of the characters in the label
DIST	(col 56-59)	is the distance (in inches) between dashes, for dashed lines, or between tick marks, for lines with tick marks
TLNG	(col 60-63)	is the length (in inches) of tick marks, if TLNG is positive, the tick marks will be on the uphill side of the line, if it is negative, the tick marks will be on the downhill side
TLER	(col 64-67)	is the tolerance distance (in inches) used in straight line or circular arc smoothing, one-third grid size recommended
I	(col 68)	0 for no smoothing, 1 for straight line smoothing, 2 for circular arc smoothing
SKIP	(col 69-72)	is the tolerance for skipping where contour lines are so close as to be indistinguishable, a positive value for SKIP will cause light line or dashed lines to be skipped if the distance between them is not greater than the SKIP value (in inches)

The PLOT card defines the type of contour line to be plotted, as well as the levels (altitudes) for which it will be plotted. A numerical label can be plotted for those contour lines which are sufficiently long. Each successive contour line plotted will be at level ZMIN + I times DZ, and each corresponding numerical label will be ZI MIN + I times DZI, where I assumes integer values from 1 to NLEV - 1. Line formats and the distance between dashes, for dashed lines, can be selected by the user, and is specified by the PLOT card. The distance between tick marks, the length of the tick marks, and whether they are on the uphill or downhill side of the line (for lines with tick marks), can be selected by the user, and are specified by the PLOT card. The PLOT card also permits the user to request that light contours be skipped whenever the distance between them does not exceed a distance specified by the user. This feature is useful in improving the intelligibility of the map.

The PLOT card also allows the user to choose a smoothing option. A contour line originally is composed of a sequence of straight line segments. It will be plotted as such, unless the smoothing option is used. Smoothing will effect either a rounding or cutting of the corner between successive straight line segments. Smoothing is accomplished such that the deviation of the smoothed contour line from the original contour line is always less than or equal to a tolerance distance specified by the user on the PLOT card. Either of two types of smoothing can be specified by the PLOT card: (1) straight line smoothing, which effects a cutting of the corner between contour segments; or (2) circular arc smoothing, which effects a rounding of the corner between contour segments. (In Figure 11, a pair of contour segments, A and B, adjoin at point P.)

Straight line smoothing (Figure 11) replaces the corner at P with a straight line segment from point P1 (on A) to point P2 (on B) such that P1 and P2 are each a distance C from P. C is determined such that the distance from P to the line from P1 to P2 is less than or equal to the tolerance distance specified by the user. C must also be less than or equal to one third of A and less than or equal to one third of B (Figure 12).

Circular arc smoothing replaces the corner at P with a circular arc tangent to both A and B at points P1 and P2, respectively. The circular arc from point P1 to P2 is determined such that the distance from P to the arc is less than or equal to the tolerance distance specified by the user. Furthermore, the distance from P to P1 must be less than or equal to one-half A, and the distance from P to P2 must be less than or equal to one-half B (Figure 13).

PHS4 Card

cc 1

PHS4 Format (A4)

The PHS4 card transfers control to Phase 4 of the program. Phase 4 plots special lines and text on the map and draws a border on the map. Phase 4 control cards are as follows:

LINE Card Format (A4,2X,6G8.3,I2)

cc 1	7	15	23	31	39	47	55
LINE	X1	Y1	Z1	X2	Y2	Z2	IC

where

X1,Y1,Z1 are the coordinates of the starting point

X2,Y2,Z2 are the coordinates of the ending point

IC (col 55-56) = 0 use plotter coordinates
= 1 use map coordinates (right justified)

The LINE card is used to draw a line on the contour map. Z is needed only for parallax drawing.

TEXT Card Format (A4,2X,5G8.3,I2,2A10,A4)

cc 1	7	15	23	31	39	47	49
TEXT	X1	Y1	Z1	HEIGHT	ANGLE	IC	ITEXT

where

X1,Y1,Z1 the coordinates of the first character of the text string

HEIGHT (col 31-38) is the height of the characters

ANGLE (col 39-46) is the angle for text string to be plotted

IC (col 47-48) = 0 use plotter coordinates
= 1 use map coordinates (right justified)

ITEXT (col 49-72) the character text string (left justified)

The TEXT card is used to draw annotation on the contour map; orientation and character size can be selected by the user.

CTEX Card

cc 1

CTEX TTEXT Format (A4,2X,6A10,A8)

where

TTEXT (cols. 7-72) is the character text string to be continued (left justified)

The CTEX card is used whenever the text string overflows either a TTEXT card or another CTEX card.

BRDR Card

cc 1

BRDR Format (A4)

The BRDR card is used to draw a border around the rectangular (parallelogram if the coordinates are skewed) containing the map.

END Card

cc 1

END Format (A4)

The end card ends phase control of a job.

STOP Card

cc 1

STOP Format (A4)

The STOP card stops all processing.

The NASAPLOT contour program allows certain options, such as map splitting and exploded views, and has the capability to process multiple jobs and to produce multiple outputs and plots within a single job.

If the user wants a map size that is too large to fit on one plotter page, then he may specify that the map into matching sections and plot each section on a separate plotter page. The following procedure is required to split a map.

1. The DIMT card is used to specify the plotter table size and to initiate the first page.
2. The MAPS card is used to specify the map section.
3. The GRID card is used to grid the map section.
4. The PLOT card is used to plot the map section.
5. Control is transferred to Phase 1 to start the next section, and the PAGE card is used to request a new plotter page.
6. Steps 2, 3, and 4 are repeated.
7. If an additional section is required, steps 5 and 6 are repeated.

Note that for a perfect match between sections, it is necessary to grid the sections such that the X distance and the Y distance between grid mesh points is unchanged in each successive section.

The user may produce, in the form of a separate map, an exploded view of any rectangular (parallelogram, if the coordinates are skewed) area of the map. The following procedure is required to produce an exploded view.

1. If a border is to be drawn around the area of interest on the original map, the LINE card is used to do so.

2. After the original map has been plotted, the MAPS card is used to specify the area of interest and the new (XOR, YOR) for plotting the exploded view.

3. The SCLE card is used to specify the factor by which the area will be exploded.

4. The PLOT card is used to plot the exploded view.

5. If a border is to be drawn around the exploded view, the BRDR card is used to do so.

The NASAPLOT contour program can process several jobs in one computer run. The program control cards corresponding to each job are stacked, with a JOB card at the beginning and an END card at the end of each job. A STOP card is required at the end of the deck (after the last END card). This program has been implemented such that the user can effect any number of manipulations and obtain any number of outputs and plots within a single job. The stereo, map splitting, and exploded view capabilities are examples of ways to obtain more than one plot within a single job. The program accepts any number of its control cards in any sequence within the logical constraints described in the PROGRAM CONTROL CARD section of this chapter. Control can be transferred from any phase to any other phase, and the map, grid, and control point parameters can be manipulated and plotted as many times as is desired within a single job.

Table 19
Common Blocks

Subprograms Routines	Block Names
LCDN	ANGLE BOUND CALC DEBUG FT GRID IO METRIC PARM PLOTCM SRCS
BASE	BOUND IO
BOUNDS	BOUND IO
FORMA	DEBUG FT GRID GUN IO SRCS TABLE CONTR
LOCATR	IO BOUND
MAP	DEBUG FT GRID IO
PGRID	BOUND CALC DEBUG FACTL FT IO METRIC SRCS
PLOT	BOUND IO METRIC PLOTCM
SCATTR	BOUND IO
STOPP	IO
CALCNR	CALC DEBUG FACTL FT GRID GUN IO METRIC PARM SRCS TABLE CONTR BDSET
JBRET	JOBBLK
PUTNR	IO
READIN	DEBUG FT GRID IO SRCS
READTB	DEBUG FACTL IO PARM TABLE
SCATPT	BOUND IO
POINT	BOUND CALC DEBUG FACTL FT IO METRIC SRCS

Table 20
TABGEN Output

TABLE GENERATION PROGRAM -- 74.20 8.60 18.67 PERCENT INVERSION

DAY FOCUS MAX 1

9	301 100.00 -199.00	DAY FOCUS MAX	
141.10	100.00METER	100.00	1.00
132.50	1000.00FEET	304.80	49.40
126.60	2000.00FEET	609.60	79.50
117.30	1.00MILE	1609.34	121.66
107.50	2.00MILE	3218.69	151.77
99.10	5.00MILE	8046.72	191.56
92.50	10.00MILE	16093.44	221.66
89.70	15.00MILE	24140.16	239.27
80.60	100000.00METER	100000.00	301.00

141.1	140.8	140.7	140.5	140.3	140.1	139.9	139.8	139.6	139.4	139.2	139.1	138.9	138.7	138.5	138.3	138.2	138.0	137.8	137.6
137.5	137.3	137.1	136.9	136.7	136.6	136.4	136.2	136.0	135.9	135.7	135.5	135.3	135.1	135.0	134.8	134.6	134.4	134.3	134.1
133.9	133.7	133.5	133.4	133.2	133.0	132.8	132.7	132.5	132.3	132.1	131.9	131.7	131.5	131.3	131.1	130.9	130.7	130.5	130.3
130.1	129.9	129.7	129.5	129.3	129.1	129.0	128.8	128.6	128.4	128.2	128.0	127.8	127.6	127.4	127.2	127.0	126.8	126.6	126.4
126.2	125.9	125.7	125.5	125.3	125.1	124.8	124.6	124.4	124.2	124.0	123.7	123.5	123.3	123.1	122.9	122.6	122.4	122.2	122.0
121.7	121.5	121.3	121.1	120.9	120.6	120.4	120.2	120.0	119.8	119.5	119.3	119.1	118.9	118.7	118.4	118.2	118.0	117.8	117.6
117.3	117.0	116.7	116.4	116.1	115.7	115.4	115.1	114.7	114.4	114.1	113.8	113.4	113.1	112.8	112.5	112.1	111.8	111.5	111.2
110.8	110.5	110.2	109.9	109.5	109.2	108.9	108.6	108.2	107.9	107.6	107.3	107.1	106.9	106.7	106.5	106.3	106.1	105.9	105.7
105.4	105.2	105.0	104.8	104.6	104.4	104.2	104.0	103.8	103.5	103.3	103.1	102.9	102.7	102.5	102.3	102.1	101.9	101.6	101.4
101.2	101.0	100.8	100.6	100.4	100.2	100.0	99.7	99.5	99.3	99.1	98.9	98.7	98.5	98.2	98.0	97.8	97.6	97.4	97.1
96.9	96.7	96.5	96.3	96.0	95.8	95.6	95.4	95.2	94.9	94.7	94.5	94.3	94.1	93.9	93.6	93.4	93.2	93.0	92.8
92.5	92.4	92.2	92.0	91.9	91.7	91.6	91.4	91.3	91.1	90.9	90.8	90.6	90.5	90.3	90.1	90.0	89.8	89.7	89.5
89.4	89.2	89.1	88.9	88.8	88.6	88.5	88.3	88.2	88.0	87.9	87.8	87.6	87.5	87.3	87.2	87.0	86.9	86.7	86.6
86.4	86.3	86.1	86.0	85.8	85.7	85.5	85.4	85.2	85.1	84.9	84.8	84.7	84.5	84.4	84.2	84.1	83.9	83.8	83.6
83.5	83.3	83.2	83.0	82.9	82.7	82.6	82.4	82.3	82.1	82.0	81.9	81.7	81.6	81.4	81.3	81.1	81.0	80.8	80.7
80.6																			

Table 20 (Cont'd)

DAY FOCUS MEAN					2	
9	301	100.00	-199.00	DAY FOCUS MEAN		
140.60	100.00METER			100.00	1.00	
132.10	1000.00FEET			304.80	49.40	
124.00	2000.00FEET			609.60	79.50	
113.30	1.00MILE			1609.34	121.66	
102.50	2.00MILE			3218.69	151.77	
94.10	5.00MILE			8046.72	191.56	
87.50	10.00MILE			16093.44	221.66	
84.70	15.00MILE			24140.16	239.27	
75.60	100000.00METER			100000.00	301.00	

140.6	140.3	140.2	140.0	139.8	139.6	139.5	139.3	139.1	138.9	138.8	138.6	138.4	138.2	138.1	137.9	137.7	137.5	137.4	137.2
137.0	136.8	136.6	136.5	136.3	136.1	135.9	135.8	135.6	135.4	135.2	135.1	134.9	134.7	134.5	134.4	134.2	134.0	133.8	133.7
133.5	133.3	133.1	133.0	132.8	132.6	132.4	132.3	132.1	131.8	131.5	131.3	131.0	130.7	130.5	130.2	129.9	129.7	129.4	129.1
128.8	128.6	128.3	128.0	127.8	127.5	127.2	127.0	126.7	126.4	126.2	125.9	125.6	125.3	125.1	124.8	124.5	124.3	124.0	123.7
123.5	123.2	123.0	122.7	122.5	122.2	122.0	121.7	121.5	121.2	121.0	120.7	120.4	120.2	119.9	119.7	119.4	119.2	118.9	118.7
118.4	118.2	117.9	117.7	117.4	117.1	116.9	116.6	116.4	116.1	115.9	115.6	115.4	115.1	114.9	114.6	114.4	114.1	113.8	113.6
113.3	113.0	112.6	112.3	111.9	111.6	111.2	110.8	110.5	110.1	109.8	109.4	109.1	108.7	108.3	108.0	107.6	107.3	106.9	106.5
106.2	105.8	105.5	105.1	104.7	104.4	104.0	103.7	103.3	103.0	102.6	102.3	102.1	101.9	101.7	101.5	101.3	101.1	100.9	100.7
100.4	100.2	100.0	99.8	99.6	99.4	99.2	99.0	98.8	98.5	98.3	98.1	97.9	97.7	97.5	97.3	97.1	96.9	96.6	96.4
96.2	96.0	95.8	95.6	95.4	95.2	95.0	94.7	94.5	94.3	94.1	93.9	93.7	93.5	93.2	93.0	92.8	92.6	92.4	92.1
91.9	91.7	91.5	91.3	91.0	90.8	90.6	90.4	90.2	89.9	89.7	89.5	89.3	89.1	88.9	88.6	88.4	88.2	88.0	87.8
87.5	87.4	87.2	87.0	86.9	86.7	86.6	86.4	86.3	86.1	85.9	85.8	85.6	85.5	85.3	85.1	85.0	84.8	84.7	84.5
84.4	84.2	84.1	83.9	83.8	83.6	83.5	83.3	83.2	83.0	82.9	82.8	82.6	82.5	82.3	82.2	82.0	81.9	81.7	81.6
81.4	81.3	81.1	81.0	80.8	80.7	80.5	80.4	80.2	80.1	79.9	79.8	79.7	79.5	79.4	79.2	79.1	78.9	78.8	78.6
78.5	78.3	78.2	78.0	77.9	77.7	77.6	77.4	77.3	77.1	77.0	76.9	76.7	76.6	76.4	76.3	76.1	76.0	75.8	75.7
75.6																			

Table 20 (Cont'd)

DAY BASE MAX

9 301 100.00 -199.00

140.10	100.00METER
131.60	1000.00FEET
123.60	2000.00FEET
111.60	1.00MILE
99.30	2.00MILE
89.00	5.00MILE
84.00	10.00MILE
80.80	15.00MILE
68.60	100000.00METER

3

DAY BASE MAX

100.00	1.00
304.80	49.40
609.60	79.50
1609.34	121.66
3218.69	151.77
8046.72	191.56
16093.44	221.66
24140.16	239.27
100000.00	301.00

140.1	139.8	139.7	139.5	139.3	139.1	139.0	138.8	138.6	138.4	138.3	138.1	137.9	137.7	137.6	137.4	137.2	137.0	136.9	136.7	
136.5	136.3	136.1	136.0	135.8	135.6	135.4	135.3	135.1	134.9	134.7	134.6	134.4	134.2	134.0	133.9	133.7	133.5	133.3	133.2	
133.0	132.8	132.6	132.5	132.3	132.1	131.9	131.8	131.6	131.3	131.0	130.8	130.5	130.2	130.0	129.7	129.4	129.2	128.9	128.7	
128.4	128.1	127.9	127.6	127.3	127.1	126.8	126.5	126.3	126.0	125.7	125.5	125.2	124.9	124.7	124.4	124.1	123.9	123.6	123.3	
123.0	122.7	122.5	122.2	121.9	121.6	121.3	121.0	120.8	120.5	120.2	119.9	119.6	119.3	119.0	118.8	118.5	118.2	117.9	117.6	
117.3	117.1	116.8	116.5	116.2	115.9	115.6	115.3	115.1	114.8	114.5	114.2	113.9	113.6	113.4	113.1	112.8	112.5	112.2	111.9	
111.6	111.3	110.9	110.4	110.0	109.6	109.2	108.8	108.4	108.0	107.6	107.2	106.8	106.4	105.9	105.5	105.1	104.7	104.3	103.9	
103.5	103.1	102.7	102.3	101.9	101.5	101.0	100.6	100.2	99.8	99.4	99.1	98.9	98.6	98.3	98.1	97.8	97.6	97.3	97.0	
96.8	96.5	96.3	96.0	95.7	95.5	95.2	95.0	94.7	94.5	94.2	93.9	93.7	93.4	93.2	92.9	92.6	92.4	92.1	91.9	
91.6	91.3	91.1	90.8	90.6	90.3	90.1	89.8	89.5	89.3	89.0	88.8	88.7	88.5	88.3	88.2	88.0	87.8	87.7	87.5	
87.3	87.2	87.0	86.9	86.7	86.5	86.4	86.2	86.0	85.9	85.7	85.5	85.4	85.2	85.0	84.9	84.7	84.5	84.4	84.2	
84.0	83.8	83.7	83.5	83.3	83.1	82.9	82.8	82.6	82.4	82.2	82.0	81.8	81.7	81.5	81.3	81.1	80.9	80.8	80.6	
80.4	80.2	80.0	79.8	79.6	79.4	79.2	79.0	78.8	78.6	78.4	78.2	78.0	77.8	77.6	77.4	77.2	77.0	76.8	76.6	
76.4	76.2	76.0	75.8	75.6	75.4	75.2	75.0	74.8	74.6	74.4	74.2	74.0	73.8	73.6	73.4	73.2	73.0	72.8	72.7	
72.5	72.3	72.1	71.9	71.7	71.5	71.3	71.1	70.9	70.7	70.5	70.3	70.1	69.9	69.7	69.5	69.3	69.1	68.9	68.7	
68.6																				

Table 20 (Cont'd)

DAY BASE MEAN						DAY BASE MEAN	
9	301	100.00	-199.00		4		
137.60	100.00METER				100.00	1.00	
125.00	1000.00FEET				304.80	49.40	
116.70	2000.00FEET				609.60	79.50	
105.80	1.00MILE				1609.34	121.66	
94.40	2.00MILE				3218.69	151.77	
85.10	5.00MILE				8046.72	191.56	
79.70	10.00MILE				16093.44	221.66	
75.80	15.00MILE				24140.16	239.27	
60.60	100000.00METER				100000.00	301.00	

137.6	137.2	136.9	136.7	136.4	136.2	135.9	135.6	135.4	135.1	134.9	134.6	134.3	134.1	133.8	133.6	133.3	133.0	132.8	132.5
132.3	132.0	131.7	131.5	131.2	131.0	130.7	130.4	130.2	129.9	129.7	129.4	129.1	128.9	128.6	128.4	128.1	127.8	127.6	127.3
127.1	126.8	126.5	126.3	126.0	125.8	125.5	125.2	125.0	124.7	124.4	124.1	123.9	123.6	123.3	123.0	122.8	122.5	122.2	121.9
121.7	121.4	121.1	120.8	120.6	120.3	120.0	119.7	119.5	119.2	118.9	118.6	118.4	118.1	117.8	117.5	117.3	117.0	116.7	116.4
116.2	115.9	115.7	115.4	115.1	114.9	114.6	114.4	114.1	113.9	113.6	113.3	113.1	112.8	112.6	112.3	112.0	111.8	111.5	111.3
111.0	110.8	110.5	110.2	110.0	109.7	109.5	109.2	108.9	108.7	108.4	108.2	107.9	107.7	107.4	107.1	106.9	106.6	106.4	106.1
105.8	105.5	105.1	104.7	104.3	104.0	103.6	103.2	102.8	102.5	102.1	101.7	101.3	100.9	100.6	100.2	99.8	99.4	99.0	98.7
98.3	97.9	97.5	97.2	96.8	96.4	96.0	95.6	95.3	94.9	94.5	94.2	94.0	93.8	93.5	93.3	93.1	92.6	92.6	92.4
92.1	91.9	91.7	91.4	91.2	91.0	90.7	90.5	90.3	90.0	89.8	89.6	89.3	89.1	88.9	88.6	88.4	88.2	87.9	87.7
87.5	87.2	87.0	86.8	86.5	86.3	86.0	85.8	85.6	85.3	85.1	84.9	84.8	84.6	84.4	84.2	84.0	83.9	83.7	83.5
83.3	83.1	83.0	82.8	82.6	82.4	82.2	82.1	81.9	81.7	81.5	81.3	81.2	81.0	80.8	80.6	80.4	80.3	80.1	79.9
79.7	79.5	79.3	79.1	78.9	78.6	78.4	78.2	78.0	77.7	77.5	77.3	77.1	76.9	76.6	76.4	76.2	76.0	75.7	75.5
75.3	75.0	74.8	74.5	74.3	74.0	73.8	73.5	73.3	73.0	72.8	72.5	72.3	72.1	71.8	71.6	71.3	71.1	70.8	70.6
70.3	70.1	69.8	69.6	69.3	69.1	68.8	68.6	68.4	68.1	67.9	67.6	67.4	67.1	66.9	66.6	66.4	66.1	65.9	65.6
65.4	65.2	64.9	64.7	64.4	64.2	63.9	63.7	63.4	63.2	62.9	62.7	62.4	62.2	62.0	61.7	61.5	61.2	61.0	60.7
60.6																			

Table 20 (Cont'd)

DAY NEG MAX					DAY NEG MAX	
9	301	100.00	-199.00		5	
135.10	100.00METER				100.00	1.00
121.60	1000.00FEET				304.80	49.40
112.60	2000.00FEET				609.60	79.50
101.60	1.00MILE				1609.34	121.66
89.60	2.00MILE				3218.69	151.77
79.00	5.00MILE				8046.72	191.56
73.00	10.00MILE				16093.44	221.66
68.60	15.00MILE				24140.16	239.27
55.60	100000.00METER				100000.00	301.00

135.1	134.7	134.4	134.1	133.8	133.6	133.3	133.0	132.7	132.5	132.2	131.9	131.6	131.3	131.1	130.8	130.5	130.2	129.9	129.7
129.4	129.1	128.8	128.5	128.3	128.0	127.7	127.4	127.2	126.9	126.6	126.3	126.0	125.8	125.5	125.2	124.9	124.6	124.4	124.1
123.8	123.5	123.2	123.0	122.7	122.4	122.1	121.9	121.6	121.3	121.0	120.7	120.4	120.1	119.8	119.5	119.2	118.9	118.6	118.3
118.0	117.7	117.4	117.1	116.8	116.5	116.2	115.9	115.6	115.3	115.0	114.7	114.4	114.1	113.8	113.5	113.2	112.9	112.6	112.3
112.1	111.8	111.6	111.3	111.0	110.8	110.5	110.3	110.0	109.7	109.5	109.2	108.9	108.7	108.4	108.2	107.9	107.6	107.4	107.1
106.9	106.6	106.3	106.1	105.8	105.6	105.3	105.0	104.8	104.5	104.3	104.0	103.7	103.5	103.2	102.9	102.7	102.4	102.2	101.9
101.6	101.3	100.9	100.5	100.1	99.7	99.3	98.9	98.5	98.1	97.7	97.3	96.9	96.5	96.1	95.7	95.3	94.9	94.5	94.1
93.7	93.3	92.9	92.5	92.1	91.7	91.3	90.9	90.5	90.1	89.7	89.4	89.1	88.9	88.6	88.3	88.1	87.8	87.5	87.3
87.0	86.7	86.5	86.2	85.9	85.7	85.4	85.1	84.9	84.6	84.3	84.1	83.8	83.5	83.3	83.0	82.7	82.5	82.2	81.9
81.7	81.4	81.1	80.9	80.6	80.3	80.1	79.8	79.5	79.3	79.0	78.8	78.6	78.4	78.2	78.0	77.8	77.6	77.4	77.2
77.0	76.8	76.6	76.4	76.2	76.0	75.8	75.6	75.4	75.2	75.0	74.8	74.6	74.4	74.2	74.0	73.8	73.6	73.4	73.2
73.0	72.8	72.5	72.3	72.0	71.8	71.5	71.3	71.0	70.8	70.5	70.3	70.0	69.8	69.5	69.3	69.0	68.8	68.6	68.3
68.1	67.9	67.7	67.5	67.3	67.1	66.9	66.7	66.4	66.2	66.0	65.8	65.6	65.4	65.2	65.0	64.8	64.6	64.3	64.1
63.9	63.7	63.5	63.3	63.1	62.9	62.7	62.4	62.2	62.0	61.8	61.6	61.4	61.2	61.0	60.8	60.5	60.3	60.1	59.9
59.7	59.5	59.3	59.1	58.9	58.7	58.4	58.2	58.0	57.8	57.6	57.4	57.2	57.0	56.8	56.5	56.3	56.1	55.9	55.7
55.6																			

Table 20 (Cont'd)

DAY NEG MEAN					DAY NEG MEAN	
9	301	100.00	-199.00	6		
131.10	100.00METER				100.00	1.00
117.20	1000.00FEET				304.80	49.40
106.40	2000.00FEET				609.60	79.50
97.30	1.00MILE				1609.34	121.66
85.90	2.00MILE				3218.69	151.77
73.80	5.00MILE				8046.72	191.56
69.40	10.00MILE				16093.44	221.66
65.80	15.00MILE				24140.16	239.27
52.60	100000.00METER				100000.00	301.00

131.1	130.7	130.4	130.1	129.8	129.5	129.2	128.9	128.7	128.4	128.1	127.8	127.5	127.2	126.9	126.6	126.4	126.1	125.8	125.5
125.2	124.9	124.6	124.4	124.1	123.8	123.5	123.2	122.9	122.6	122.3	122.1	121.8	121.5	121.2	120.9	120.6	120.3	120.0	119.8
119.5	119.2	118.9	118.6	118.3	118.0	117.7	117.5	117.2	116.8	116.4	116.1	115.7	115.4	115.0	114.7	114.3	113.9	113.6	113.2
112.9	112.5	112.1	111.8	111.4	111.1	110.7	110.3	110.0	109.6	109.3	108.9	108.6	108.2	107.8	107.5	107.1	106.8	106.4	106.2
106.0	105.8	105.5	105.3	105.1	104.9	104.7	104.5	104.2	104.0	103.8	103.6	103.4	103.2	102.9	102.7	102.5	102.3	102.1	101.9
101.7	101.4	101.2	101.0	100.8	100.6	100.4	100.1	99.9	99.7	99.5	99.3	99.1	98.8	98.6	98.4	98.2	98.0	97.8	97.6
97.3	97.0	96.8	96.2	95.8	95.5	95.1	94.7	94.3	94.0	93.6	93.2	92.8	92.4	92.1	91.7	91.3	90.9	90.5	90.2
89.8	89.4	89.0	88.7	88.3	87.9	87.5	87.1	86.8	86.4	86.0	85.7	85.4	85.1	84.8	84.5	84.2	83.9	83.5	83.2
82.9	82.6	82.3	82.0	81.7	81.4	81.1	80.8	80.5	80.2	79.9	79.6	79.3	79.0	78.7	78.4	78.1	77.8	77.5	77.2
76.9	76.6	76.3	75.9	75.6	75.3	75.0	74.7	74.4	74.1	73.8	73.7	73.5	73.4	73.2	73.1	72.9	72.8	72.6	72.5
72.3	72.2	72.1	71.9	71.8	71.6	71.5	71.3	71.2	71.0	70.9	70.7	70.6	70.4	70.3	70.2	70.0	69.9	69.7	69.6
69.4	69.2	69.0	68.8	68.6	68.4	68.2	68.0	67.8	67.6	67.4	67.2	67.0	66.8	66.6	66.4	66.2	66.0	65.8	65.5
65.3	65.1	64.9	64.7	64.5	64.3	64.0	63.8	63.6	63.4	63.2	63.0	62.8	62.5	62.3	62.1	61.9	61.7	61.5	61.3
61.0	60.8	60.6	60.4	60.2	60.0	59.8	59.6	59.3	59.1	58.9	58.7	58.5	58.3	58.1	57.8	57.6	57.4	57.2	57.0
56.8	56.6	56.3	56.1	55.9	55.7	55.5	55.3	55.1	54.8	54.6	54.4	54.2	54.0	53.8	53.6	53.3	53.1	52.9	52.7
52.6																			

Table 20 (Cont'd)

DAY EX NEG MAX						DAY EX NEG MAX	
9	301	100.00	-199.00			7	
120.10	100.00METER					100.00	1.00
107.60	1000.00FEET					304.80	49.40
99.60	2000.00FEET					609.60	79.50
90.60	1.00MILE					1609.34	121.66
81.20	2.00MILE					3218.69	151.77
64.70	5.00MILE					8046.72	191.56
59.60	10.00MILE					16093.44	221.66
57.00	15.00MILE					24140.16	239.27
46.60	100000.00METER					100000.00	301.00

120.1	119.7	119.5	119.2	118.9	118.7	118.4	118.2	117.9	117.6	117.4	117.1	116.9	116.6	116.4	116.1	115.8	115.6	115.3	115.1
114.8	114.5	114.3	114.0	113.8	113.5	113.3	113.0	112.7	112.5	112.2	112.0	111.7	111.4	111.2	110.9	110.7	110.4	110.2	109.9
109.6	109.4	109.1	108.9	108.6	108.3	108.1	107.8	107.6	107.3	107.0	106.8	106.5	106.2	106.0	105.7	105.4	105.2	104.9	104.7
104.4	104.1	103.9	103.6	103.3	103.1	102.8	102.5	102.3	102.0	101.7	101.5	101.2	100.9	100.7	100.4	100.1	99.9	99.6	99.4
99.2	99.0	98.7	98.5	98.3	98.1	97.9	97.7	97.5	97.3	97.0	96.8	96.6	96.4	96.2	96.0	95.8	95.5	95.3	95.1
94.9	94.7	94.5	94.3	94.1	93.8	93.6	93.4	93.2	93.0	92.8	92.6	92.3	92.1	91.9	91.7	91.5	91.3	91.1	90.8
90.6	90.3	90.0	89.7	89.4	89.1	88.8	88.5	88.2	87.8	87.5	87.2	86.9	86.6	86.3	86.0	85.7	85.3	85.0	84.7
84.4	84.1	83.8	83.5	83.2	82.8	82.5	82.2	81.9	81.6	81.3	80.9	80.5	80.1	79.7	79.2	78.8	78.4	78.0	77.6
77.2	76.8	76.3	75.9	75.5	75.1	74.7	74.3	73.8	73.4	73.0	72.6	72.2	71.8	71.4	70.9	70.5	70.1	69.7	69.3
68.9	68.5	68.0	67.6	67.2	66.8	66.4	66.0	65.6	65.1	64.7	64.5	64.4	64.2	64.0	63.9	63.7	63.5	63.4	63.2
63.0	62.8	62.7	62.5	62.3	62.2	62.0	61.8	61.7	61.5	61.3	61.2	61.0	60.8	60.6	60.5	60.3	60.1	60.0	59.8
59.6	59.5	59.3	59.2	59.0	58.9	58.7	58.6	58.4	58.3	58.1	58.0	57.9	57.7	57.6	57.4	57.3	57.1	57.0	56.8
56.6	56.5	56.3	56.1	56.0	55.8	55.6	55.4	55.3	55.1	54.9	54.8	54.6	54.4	54.3	54.1	53.9	53.8	53.6	53.4
53.3	53.1	52.9	52.7	52.6	52.4	52.2	52.1	51.9	51.7	51.6	51.4	51.2	51.1	50.9	50.7	50.6	50.4	50.2	50.1
49.9	49.7	49.5	49.4	49.2	49.0	48.9	48.7	48.5	48.4	48.2	48.0	47.9	47.7	47.5	47.4	47.2	47.0	46.9	46.7
46.6																			

Table 20 (Cont'd)

DAY EX NEG MEAN					DAY EX NEG MEAN	
9	301	100.00	-199.00		100.00	1.00
118.10	100.00METER				304.80	49.40
106.10	1000.00FEET				609.60	79.50
97.00	2000.00FEET				1609.34	121.66
86.90	1.00MILE				3218.69	151.77
77.10	2.00MILE				8046.72	191.56
60.50	5.00MILE				16093.44	221.66
54.60	10.00MILE				24140.16	239.27
49.10	15.00MILE				100000.00	301.00
39.10	100000.00METER					

118.1 117.7 117.5 117.2 117.0 116.7 116.5 116.2 116.0 115.7 115.5 115.2 115.0 114.8 114.5 114.3 114.0 113.8 113.5 113.3
113.0 112.8 112.5 112.3 112.0 111.8 111.5 111.3 111.0 110.8 110.5 110.3 110.0 109.8 109.5 109.3 109.1 108.8 108.6 108.3
108.1 107.8 107.6 107.3 107.1 106.8 106.6 106.3 106.1 105.8 105.5 105.2 104.9 104.6 104.3 104.0 103.7 103.3 103.0 102.7
102.4 102.1 101.8 101.5 101.2 100.9 100.6 100.3 100.0 99.7 99.4 99.1 98.8 98.5 98.2 97.9 97.6 97.3 97.0 96.8
96.5 96.3 96.0 95.8 95.6 95.3 95.1 94.8 94.6 94.4 94.1 93.9 93.6 93.4 93.2 92.9 92.7 92.4 92.2 92.0
91.7 91.5 91.3 91.0 90.8 90.5 90.3 90.1 89.8 89.6 89.3 89.1 88.9 88.6 88.4 88.1 87.9 87.7 87.4 87.2
86.9 86.6 86.3 86.0 85.7 85.3 85.0 84.7 84.3 84.0 83.7 83.4 83.0 82.7 82.4 82.1 81.7 81.4 81.1 80.8
80.4 80.1 79.8 79.5 79.1 78.8 78.5 78.2 77.8 77.5 77.2 76.8 76.4 76.0 75.5 75.1 74.7 74.3 73.9 73.5
73.0 72.6 72.2 71.8 71.4 71.0 70.5 70.1 69.7 69.3 68.9 68.5 68.0 67.6 67.2 66.8 66.4 65.9 65.5 65.1
64.7 64.3 63.9 63.4 63.0 62.6 62.2 61.8 61.4 60.9 60.5 60.3 60.1 59.9 59.7 59.5 59.3 59.1 58.9 58.7
58.6 58.4 58.2 58.0 57.8 57.6 57.4 57.2 57.0 56.8 56.6 56.4 56.2 56.0 55.8 55.6 55.4 55.2 55.0 54.8
54.6 54.3 54.0 53.7 53.4 53.1 52.8 52.5 52.2 51.8 51.5 51.2 50.9 50.6 50.3 50.0 49.7 49.3 49.1 48.9
48.7 48.6 48.4 48.3 48.1 47.9 47.8 47.6 47.4 47.3 47.1 47.0 46.8 46.6 46.5 46.3 46.1 46.0 45.8 45.7
45.5 45.3 45.2 45.0 44.9 44.7 44.5 44.4 44.2 44.0 43.9 43.7 43.6 43.4 43.2 43.1 42.9 42.7 42.6 42.4
42.3 42.1 41.9 41.8 41.6 41.4 41.3 41.1 41.0 40.8 40.6 40.5 40.3 40.2 40.0 39.8 39.7 39.5 39.3 39.2
39.1

Table 20 (Cont'd)

DAY EX NEG MIN

9	301	1[illegible].[illegible]	-199.00	DAY EX NEG MIN		
111.10	100.00METER				100.00	1.00
97.10	1000.00FEET				304.80	42.40
88.10	2000.00FEET				609.60	79.50
78.60	1.00MILE				1609.34	121.66
68.10	2.00MILE				3218.69	151.77
52.60	5.00MILE				8046.72	191.56
44.10	10.00MILE				16093.44	221.66
38.10	15.00MILE				24140.16	239.27
27.10	100000.00METER				100000.00	301.00

111.1	110.7	110.4	110.1	109.8	109.5	109.2	108.9	108.6	108.4	108.1	107.8	107.5	107.2	106.9	106.6	106.3	106.0	105.7	105.5
105.2	104.9	104.6	104.3	104.0	103.7	103.4	103.1	102.9	102.6	102.3	102.0	101.7	101.4	101.1	100.8	100.5	100.3	100.0	99.7
99.4	99.1	98.8	98.5	98.2	97.9	97.7	97.4	97.1	96.8	96.5	96.2	95.9	95.6	95.3	95.0	94.7	94.4	94.1	93.8
93.5	93.2	92.9	92.6	92.3	92.0	91.7	91.4	91.1	90.8	90.5	90.2	89.9	89.6	89.3	89.0	88.7	88.4	88.1	87.[illegible]
87.7	87.4	87.2	87.0	86.7	86.5	86.3	86.1	85.8	85.6	85.4	85.2	84.9	84.7	84.5	84.3	84.0	83.8	83.6	83.[illegible]
83.1	82.9	82.7	82.5	82.2	82.0	81.8	81.6	81.3	81.1	80.9	80.7	80.4	80.2	80.0	79.8	79.5	79.3	79.1	78.9
78.6	78.3	78.0	77.6	77.3	76.9	76.6	76.2	75.9	75.5	75.2	74.8	74.5	74.1	73.8	73.4	73.1	72.7	72.4	72.[illegible]
71.7	71.3	71.0	70.6	70.3	69.9	69.6	69.2	68.9	68.5	68.2	67.8	67.3	66.9	66.5	66.0	65.6	65.1	64.7	64.3
63.8	63.4	62.9	62.5	62.1	61.6	61.2	60.7	60.3	59.9	59.4	59.0	58.5	58.1	57.7	57.2	56.8	56.3	55.9	55.5
55.0	54.6	54.1	53.7	53.3	52.8	52.4	51.9	51.5	51.1	50.6	50.4	50.2	50.0	49.7	49.5	49.3	49.1	48.9	48.7
48.5	48.2	48.0	47.8	47.6	47.4	47.2	46.9	46.7	46.5	46.3	46.1	45.9	45.6	45.4	45.2	45.0	44.8	44.6	44.4
44.1	43.8	43.5	43.1	42.8	42.5	42.1	41.8	41.4	41.1	40.7	40.4	40.1	39.7	39.4	39.0	38.7	38.4	38.1	37.9
37.7	37.5	37.3	37.2	37.0	36.8	36.6	36.5	36.3	36.1	35.9	35.7	35.6	35.4	35.2	35.0	34.9	34.7	34.5	34.3
34.1	34.0	33.8	33.6	33.4	33.2	33.1	32.9	32.7	32.5	32.4	32.2	32.0	31.8	31.6	31.5	31.3	31.1	30.9	30.8
30.6	30.4	30.2	30.0	29.9	29.7	29.5	29.3	29.1	29.0	28.8	28.6	28.4	28.3	28.1	27.9	27.7	27.5	27.4	27.2
27.1																			

Table 20 (Cont'd)

NIGHT FOCUS MAX

9	301	100.00	-199.00
140.90	100.00METER		
132.10	1000.00FEET		
127.10	2000.00FEET		
115.90	1.00MILE		
106.40	2.00MILE		
96.70	5.00MILE		
93.00	10.00MILE		
91.60	15.00MILE		
86.60	100000.00METER		

10

NIGHT FOCUS MAX	
100.00	1.00
304.80	49.40
609.60	79.50
1609.34	121.66
3218.69	151.77
8046.72	191.56
16093.44	221.66
24140.16	239.27
100000.00	301.00

140.9	140.6	140.4	140.3	140.1	139.9	139.7	139.5	139.4	139.2	139.0	138.8	138.6	138.4	138.3	138.1	137.9	137.7	137.5	137.4
137.2	137.0	136.8	136.6	136.4	136.3	136.1	135.9	135.7	135.5	135.4	135.2	135.0	134.8	134.6	134.4	134.3	134.1	133.9	133.7
133.5	133.4	133.2	133.0	132.8	132.6	132.4	132.3	132.1	131.9	131.8	131.6	131.4	131.3	131.1	130.9	130.8	130.6	130.4	130.3
130.1	129.9	129.8	129.6	129.4	129.3	129.1	128.9	128.8	128.6	128.4	128.3	128.1	127.9	127.8	127.6	127.4	127.3	127.1	126.8
126.6	126.3	126.0	125.8	125.5	125.2	125.0	124.7	124.4	124.2	123.9	123.6	123.4	123.1	122.9	122.6	122.3	122.1	121.8	121.5
121.3	121.0	120.7	120.5	120.2	119.9	119.7	119.4	119.1	118.9	118.6	118.3	118.1	117.8	117.5	117.3	117.0	116.7	116.5	116.2
115.9	115.6	115.3	115.0	114.7	114.4	114.1	113.7	113.4	113.1	112.8	112.5	112.2	111.8	111.5	111.2	110.9	110.6	110.3	110.0
109.6	109.3	109.0	108.7	108.4	108.1	107.7	107.4	107.1	106.8	106.5	106.2	106.0	105.7	105.5	105.2	105.0	104.8	104.5	104.3
104.0	103.8	103.5	103.3	103.1	102.8	102.6	102.3	102.1	101.8	101.6	101.3	101.1	100.9	100.6	100.4	100.1	99.9	99.6	99.4
99.2	98.9	98.7	98.4	98.2	97.9	97.7	97.4	97.2	97.0	96.7	96.6	96.5	96.3	96.2	96.1	96.0	95.8	95.7	95.6
95.5	95.4	95.2	95.1	95.0	94.9	94.7	94.6	94.5	94.4	94.2	94.1	94.0	93.9	93.8	93.6	93.5	93.4	93.3	93.1
93.0	92.9	92.9	92.8	92.7	92.6	92.5	92.5	92.4	92.3	92.2	92.1	92.1	92.0	91.9	91.8	91.7	91.7	91.6	91.5
91.4	91.3	91.3	91.2	91.1	91.0	90.9	90.9	90.8	90.7	90.6	90.5	90.4	90.4	90.3	90.2	90.1	90.0	90.0	89.9
89.8	89.7	89.6	89.6	89.5	89.4	89.3	89.2	89.2	89.1	89.0	88.9	88.8	88.7	88.7	88.6	88.5	88.4	88.3	88.3
88.2	88.1	88.0	87.9	87.9	87.8	87.7	87.6	87.5	87.5	87.4	87.3	87.2	87.1	87.0	87.0	86.9	86.8	86.7	86.6
86.6																			

Table 20 (Cont'd)

NIGHT FOCUS MEAN					NIGHT FOCUS MEAN	
9	301	100.00	-199.00		11	
139.30	100.00METER				100.00	1.00
130.40	1000.00FEET				304.80	49.40
124.30	2000.00FEET				609.60	79.50
110.90	1.00MILE				1609.34	121.66
101.40	2.00MILE				3218.69	151.77
91.70	5.00MILE				8046.72	191.56
88.00	10.00MILE				16093.44	221.66
86.60	15.00MILE				24140.16	239.27
82.10	100000.00METER				100000.00	301.00

139.3	139.0	138.8	138.7	138.5	138.3	138.1	137.9	137.7	137.6	137.4	137.2	137.0	136.8	136.6	136.4	136.3	136.1	135.9	135.7
135.5	135.3	135.2	135.0	134.8	134.6	134.4	134.2	134.1	133.9	133.7	133.5	133.3	133.1	133.0	132.8	132.6	132.4	132.2	132.0
131.9	131.7	131.5	131.3	131.1	130.9	130.7	130.6	130.4	130.2	130.0	129.8	129.6	129.4	129.2	129.0	128.8	128.6	128.4	128.2
127.9	127.7	127.5	127.3	127.1	126.9	126.7	126.5	126.3	126.1	125.9	125.7	125.5	125.3	125.1	124.9	124.7	124.5	124.3	124.1
123.7	123.5	123.0	122.7	122.4	122.1	121.8	121.4	121.1	120.8	120.5	120.2	119.9	119.5	119.2	118.9	118.6	118.3	117.9	117.6
117.3	117.0	116.7	116.4	116.0	115.7	115.4	115.1	114.8	114.4	114.1	113.8	113.5	113.2	112.9	112.5	112.2	111.9	111.6	111.3
111.0	110.6	110.3	110.0	109.7	109.4	109.1	108.7	108.4	108.1	107.8	107.5	107.2	106.8	106.5	106.2	105.9	105.6	105.3	105.0
104.6	104.3	104.0	103.7	103.4	103.1	102.7	102.4	102.1	101.8	101.5	101.2	101.0	100.7	100.5	100.2	100.0	99.6	99.5	99.3
99.0	98.8	98.5	98.3	98.1	97.8	97.6	97.3	97.1	96.8	96.6	96.3	96.1	95.9	95.6	95.4	95.1	94.9	94.6	94.4
94.2	93.9	93.7	93.4	93.2	92.9	92.7	92.4	92.2	92.0	91.7	91.6	91.5	91.3	91.2	91.1	91.0	90.8	90.7	90.6
90.5	90.4	90.2	90.1	90.0	89.9	89.7	89.6	89.5	89.4	89.2	89.1	89.0	88.9	88.8	88.6	88.5	88.4	88.3	88.1
88.0	87.9	87.9	87.8	87.7	87.6	87.5	87.5	87.4	87.3	87.2	87.1	87.1	87.0	86.9	86.8	86.7	86.7	86.6	86.5
86.4	86.4	86.3	86.2	86.1	86.1	86.0	85.9	85.9	85.8	85.7	85.6	85.6	85.5	85.4	85.3	85.3	85.2	85.1	85.1
85.0	84.9	84.8	84.8	84.7	84.6	84.5	84.5	84.4	84.3	84.3	84.2	84.1	84.0	84.0	83.9	83.8	83.7	83.7	83.6
83.5	83.4	83.4	83.3	83.2	83.2	83.1	83.0	82.9	82.9	82.8	82.7	82.6	82.6	82.5	82.4	82.4	82.3	82.2	82.1
82.1																			

Table 20 (Cont'd)

NIGHT BASE MAX

9	301	100.00	-199.00
138.10	100.00METER		
128.60	1000.00FEET		
122.60	2000.00FEET		
109.60	1.00MILE		
99.60	2.00MILE		
89.20	5.00MILE		
83.50	10.00MILE		
79.90	15.00MILE		
68.10	100000.00METER		

12

NIGHT BASE MAX	
100.00	1.00
304.80	49.40
609.60	79.50
1609.34	121.66
3218.69	151.77
8046.72	191.56
16093.44	221.66
24140.16	239.27
100000.00	301.00

138.1	137.8	137.6	137.4	137.2	137.0	136.8	136.6	136.4	136.2	136.0	135.8	135.6	135.5	135.3	135.1	134.9	134.7	134.5	134.3
134.1	133.9	133.7	133.5	133.3	133.1	132.9	132.7	132.5	132.3	132.1	131.9	131.7	131.5	131.3	131.1	130.9	130.7	130.5	130.3
130.2	130.0	129.8	129.6	129.4	129.2	129.0	128.8	128.6	128.4	128.2	128.0	127.8	127.6	127.4	127.2	127.0	126.8	126.6	126.4
126.2	126.0	125.8	125.6	125.4	125.2	125.0	124.8	124.6	124.4	124.2	124.0	123.8	123.6	123.4	123.2	123.0	122.8	122.6	122.3
122.0	121.7	121.4	121.1	120.8	120.4	120.1	119.8	119.5	119.2	118.9	118.6	118.3	118.0	117.7	117.4	117.1	116.7	116.4	116.1
115.8	115.5	115.2	114.9	114.6	114.3	114.0	113.7	113.4	113.0	112.7	112.4	112.1	111.8	111.5	111.2	110.9	110.6	110.3	110.0
109.7	109.3	109.0	108.7	108.3	108.0	107.7	107.3	107.0	106.7	106.3	106.0	105.7	105.3	105.0	104.7	104.3	104.0	103.7	103.3
103.0	102.7	102.3	102.0	101.7	101.3	101.0	100.7	100.4	100.0	99.7	99.4	99.1	98.9	98.6	98.4	98.1	97.8	97.6	97.3
97.1	96.8	96.5	96.3	96.0	95.7	95.5	95.2	95.0	94.7	94.4	94.2	93.9	93.7	93.4	93.1	92.9	92.6	92.4	92.1
91.8	91.6	91.3	91.0	90.8	90.5	90.3	90.0	89.7	89.5	89.2	89.0	88.8	88.6	88.5	88.3	88.1	87.9	87.7	87.5
87.3	87.1	86.9	86.8	86.6	86.4	86.2	86.0	85.8	85.6	85.4	85.2	85.0	84.9	84.7	84.5	84.3	84.1	83.9	83.7
83.5	83.3	83.1	82.9	82.7	82.5	82.3	82.1	81.9	81.7	81.5	81.3	81.1	80.9	80.7	80.5	80.3	80.1	79.9	79.7
79.5	79.3	79.1	78.9	78.7	78.5	78.3	78.1	77.9	77.8	77.6	77.4	77.2	77.0	76.8	76.6	76.4	76.2	76.0	75.8
75.7	75.5	75.3	75.1	74.9	74.7	74.5	74.3	74.1	73.9	73.7	73.5	73.4	73.2	73.0	72.8	72.6	72.4	72.2	72.0
71.8	71.6	71.4	71.3	71.1	70.9	70.7	70.5	70.3	70.1	69.9	69.7	69.5	69.3	69.2	69.0	68.8	68.6	68.4	68.2
68.1																			

Table 20 (Cont'd)

NIGHT BASE MEAN						NIGHT BASE MEAN	
9	501	100.00	-199.00			13	
135.19	100.00METER					100.00	1.00
124.38	1000.00FEET					304.80	92.96
117.20	2000.00FEET					609.60	79.53
104.60	1.00MILE					1609.34	121.88
95.30	2.00MILE					3218.69	151.77
84.70	5.00MILE					8046.72	191.56
78.80	10.00MILE					16093.44	221.88
76.10	15.00MILE					24140.16	239.27
65.10	100000.00METER					100000.00	301.00

135.1	134.8	134.5	134.3	134.1	133.9	133.6	133.4	133.2	133.0	132.8	132.5	132.3	132.1	131.9	131.6	131.4	131.2	131.0	130.7
130.5	130.3	130.1	129.9	129.6	129.4	129.2	129.0	128.7	128.5	128.3	128.1	127.8	127.6	127.4	127.2	126.9	126.7	126.5	126.3
126.1	125.8	125.6	125.4	125.2	124.9	124.7	124.5	124.3	124.0	123.8	123.6	123.3	123.1	122.9	122.6	122.4	122.2	122.0	121.7
121.4	121.2	121.0	120.7	120.5	120.3	120.0	119.8	119.6	119.3	119.1	118.9	118.6	118.4	118.1	117.9	117.7	117.4	117.2	116.9
116.6	116.3	116.0	115.7	115.4	115.1	114.8	114.5	114.2	113.9	113.6	113.3	113.0	112.7	112.4	112.1	111.8	111.5	111.2	110.9
110.6	110.3	110.0	109.7	109.4	109.1	108.8	108.5	108.2	107.9	107.6	107.3	107.	106.7	106.4	106.1	105.8	105.5	105.2	104.9
104.6	104.3	104.0	103.7	103.4	103.1	102.8	102.5	102.2	101.9	101.6	101.3	101.0	100.7	100.3	100.	99.7	99.4	99.	98.7
98.5	98.2	97.9	97.5	97.2	96.9	96.6	96.3	96.0	95.7	95.4	95.1	94.8	94.5	94.3	94.	93.8	93.5	93.	92.7
92.7	92.4	92.2	91.9	91.6	91.4	91.1	90.8	90.6	90.3	90.0	89.8	89.5	89.2	89.0	88.7	88.4	88.2	87.9	87.6
87.4	87.1	86.8	86.6	86.3	86.0	85.8	85.5	85.2	85.0	84.7	84.5	84.3	84.1	83.9	83.7	83.5	83.3	83.1	82.9
82.8	82.6	82.4	82.2	82.0	81.8	81.6	81.4	81.2	81.0	80.8	80.6	80.4	80.2	80.0	79.8	79.6	79.4	79.2	79.
78.8	78.7	78.5	78.4	78.2	78.1	77.9	77.8	77.6	77.4	77.3	77.1	77.0	76.8	76.7	76.5	76.4	76.2	76.1	75.9
75.7	75.5	75.3	75.2	75.0	74.8	74.6	74.5	74.3	74.1	73.9	73.7	73.6	73.4	73.2	73.1	72.9	72.7	72.5	72.4
72.1	72.0	71.8	71.6	71.4	71.2	71.1	70.9	70.7	70.5	70.4	70.2	70.0	69.8	69.6	69.5	69.3	69.1	68.9	68.7
68.6	68.4	68.2	68.0	67.9	67.7	67.5	67.3	67.1	67.0	66.8	66.6	66.4	66.3	66.1	65.9	65.7	65.5	65.4	65.2
65.1																			

Table 20 (Cont'd)

```
NIGHT NEG MAX                                              14
        9        301      100.00     -199.00    NIGHT NEG MAX
   133.60     100.00METER                              100.00       1.00
   121.60    1000.00FEET                               304.80      49.40
   111.60    2000.00FEET                               609.60      79.50
    98.60       1.00MILE                              1609.34     121.66
    90.80       2.00MILE                              3218.69     151.77
    77.40       5.00MILE                              8046.72     191.56
    72.70      10.00MILE                             16093.44     221.66
    67.40      15.00MILE                             24140.16     239.27
    52.60  100000.00METER                           100000.00     301.00
```

133.6	133.2	133.0	132.7	132.5	132.2	132.0	131.7	131.5	131.2	131.0	130.7	130.5	130.3	130.0	129.8	129.5	129.3	129.0	128.8
128.5	128.3	128.0	127.8	127.5	127.3	127.0	126.8	126.5	126.3	126.0	125.8	125.5	125.3	125.0	124.8	124.6	124.3	124.1	123.8
123.6	123.3	123.1	122.8	122.6	122.3	122.1	121.8	121.6	121.2	120.9	120.6	120.2	119.9	119.6	119.2	118.9	118.6	118.2	117.9
117.6	117.2	116.9	116.6	116.3	115.9	115.6	115.3	114.9	114.6	114.3	113.9	113.6	113.3	112.9	112.6	112.3	111.9	111.6	111.3
111.0	110.7	110.4	110.1	109.8	109.4	109.1	108.8	108.5	108.2	107.9	107.6	107.3	107.0	106.7	106.4	106.1	105.7	105.4	105.1
104.8	104.5	104.2	103.9	103.6	103.3	103.0	102.7	102.4	102.0	101.7	101.4	101.1	100.8	100.5	100.2	99.9	99.6	99.3	99.0
98.7	98.4	98.1	97.9	97.6	97.3	97.1	96.8	96.6	96.3	96.1	95.8	95.5	95.3	95.0	94.8	94.5	94.2	94.0	93.7
93.5	93.2	92.9	92.7	92.4	92.2	91.9	91.6	91.4	91.1	90.9	90.6	90.2	89.9	89.5	89.2	88.9	88.5	88.2	87.9
87.5	87.2	86.8	86.5	86.2	85.8	85.5	85.2	84.8	84.5	84.2	83.8	83.5	83.1	82.8	82.5	82.1	81.8	81.5	81.1
80.8	80.5	80.1	79.8	79.4	79.1	78.8	78.4	78.1	77.8	77.4	77.3	77.1	76.9	76.8	76.6	76.5	76.3	76.2	76.0
75.8	75.7	75.5	75.4	75.2	75.1	74.9	74.8	74.6	74.4	74.3	74.1	74.0	73.8	73.7	73.5	73.4	73.2	73.0	72.9
72.7	72.4	72.1	71.8	71.5	71.2	70.9	70.6	70.3	70.0	69.7	69.4	69.1	68.8	68.5	68.2	67.9	67.6	67.3	67.1
66.9	66.6	66.4	66.1	65.9	65.7	65.4	65.2	64.9	64.7	64.5	64.2	64.0	63.7	63.5	63.3	63.0	62.8	62.6	62.3
62.1	61.8	61.6	61.4	61.1	60.9	60.6	60.4	60.2	59.9	59.7	59.4	59.2	59.0	58.7	58.5	58.2	58.0	57.8	57.5
57.3	57.0	56.8	56.6	56.3	56.1	55.8	55.6	55.4	55.1	54.9	54.6	54.4	54.2	53.9	53.7	53.4	53.2	53.0	52.7
52.6																			

Table 20 (Cont'd)

```
NIGHT NEG MEAN                                         15
      9        301      100.00    -199.00     NIGHT NEG MEAN
   132.10     100.00METER                        100.00        1.00
   118.40    1000.00FEET                         304.80       49.40
   107.30    2000.00FEET                         609.60       79.50
    93.90       1.00MILE                        1609.34      121.66
    86.90       2.00MILE                        3218.69      151.77
    73.40       5.00MILE                        8046.72      191.56
    68.40      10.00MILE                       16093.44      221.66
    64.90      15.00MILE                       24140.16      239.27
    51.60  100000.00METER                     100000.00      301.00
```

132.1	131.7	131.4	131.1	130.8	130.5	130.3	130.0	129.7	129.4	129.1	128.8	128.6	128.3	128.0	127.7	127.4	127.1	126.9	[illegible]
126.3	126.0	125.7	125.4	125.2	124.9	124.6	124.3	124.0	123.8	123.5	123.2	122.9	122.6	122.3	122.1	121.8	121.5	121.2	[illegible]
120.6	120.4	120.1	119.8	119.5	119.2	118.9	118.7	118.4	118.0	117.6	117.3	116.9	116.5	116.2	115.8	115.4	[illegible]	114.7	114.3
113.9	113.6	113.2	112.8	112.5	112.1	111.7	111.4	111.0	110.6	110.3	109.9	109.5	109.1	108.8	108.4	[illegible]	107.7	107.3	[illegible]
106.7	106.3	106.0	105.7	105.4	105.1	104.8	104.4	104.1	103.8	103.5	103.2	102.9	102.5	102.2	101.9	101.6	101.3	[illegible]	[illegible]
100.3	[illegible]	99.7	99.4	99.0	98.7	98.4	98.1	97.8	97.4	97.1	96.8	96.5	96.2	95.9	95.5	95.2	94.9	94.6	94.3
94.0	93.7	93.5	93.2	93.0	92.8	92.5	92.3	92.1	91.8	91.6	91.4	91.1	90.9	90.7	90.5	90.2	[illegible]	[illegible]	89.5
89.3	89.1	88.8	88.6	88.4	88.1	87.9	87.7	87.4	87.2	87.0	86.7	86.5	[illegible]	85.6	85.3	[illegible]	[illegible]	[illegible]	[illegible]
83.6	83.3	82.9	82.6	82.2	81.9	81.6	81.2	80.9	80.5	80.2	79.9	79.5	79.2	78.8	78.5	78.2	77.8	77.5	[illegible]
76.8	76.5	76.1	75.8	75.5	75.1	74.8	74.4	74.1	73.8	73.4	73.2	73.1	72.9	72.7	72.6	72.4	[illegible]	72.1	[illegible]
71.7	71.6	71.4	71.3	71.1	70.9	70.8	70.6	70.4	70.3	70.1	69.9	69.8	69.6	69.4	69.3	69.1	68.9	[illegible]	[illegible]
68.4	68.2	68.0	67.8	67.6	67.4	67.2	67.0	66.8	66.6	66.4	66.2	66.0	65.8	65.7	65.5	65.3	65.1	64.9	[illegible]
[illegible]	64.2	64.0	63.8	63.6	63.3	63.1	62.9	62.7	62.5	62.3	62.1	61.8	61.6	61.4	61.2	61.0	60.8	60.5	60.3
60.1	59.9	59.7	59.5	59.2	59.0	58.8	58.6	58.4	58.2	58.0	57.7	57.5	57.3	57.1	56.9	56.7	56.4	56.2	[illegible]
55.8	55.6	55.4	55.2	54.9	54.7	54.5	54.3	54.1	53.9	53.6	53.4	53.2	53.0	52.8	52.6	52.4	52.1	51.9	51.7
51.6																			

Table 20 (Cont'd)

NIGHT EX NEG MAX

9 301 100.00 -199.00

122.50	100.00METER
108.80	1000.00FEET
99.60	2000.00FEET
86.60	1.00MILE
78.10	2.00MILE
66.40	5.00MILE
62.00	10.00MILE
58.40	15.00MILE
45.10	100000.00METER

16

NIGHT EX NEG MAX

100.00	1.00
304.80	49.40
609.60	79.50
1609.34	121.66
3218.69	151.77
8046.72	191.56
16093.44	221.66
24140.16	239.27
100000.00	301.00

122.5	122.1	121.8	121.5	121.2	120.9	120.6	120.3	120.1	119.8	119.5	119.2	118.9	118.6	118.3	118.0	117.8	117.5	117.2	116.9
116.6	116.3	116.0	115.8	115.5	115.2	114.9	114.6	114.3	114.0	113.7	113.5	113.2	112.9	112.6	112.3	112.0	111.7	111.4	111.2
110.9	110.6	110.3	110.0	109.7	109.4	109.1	108.9	108.6	108.3	108.0	107.7	107.4	107.1	106.8	106.5	106.2	105.9	105.6	105.3
105.0	104.7	104.4	104.1	103.8	103.5	103.2	102.9	102.6	102.3	102.0	101.7	101.4	101.1	100.8	100.5	100.2	99.9	99.6	99.3
99.0	98.7	98.4	98.1	97.8	97.4	97.1	96.8	96.5	96.2	95.9	95.6	95.3	95.0	94.7	94.4	94.1	93.7	93.4	93.1
92.8	92.5	92.2	91.9	91.6	91.3	91.0	90.7	90.4	90.0	89.7	89.4	89.1	88.8	88.5	88.2	87.9	87.6	87.3	87.0
86.7	86.4	86.1	85.8	85.6	85.3	85.0	84.8	84.5	84.2	84.0	83.7	83.4	83.1	82.9	82.6	82.3	82.1	81.8	81.5
81.3	81.0	80.7	80.5	80.2	79.9	79.6	79.4	79.1	78.8	78.6	78.3	77.9	77.6	77.3	76.9	76.6	76.3	76.0	75.7
75.3	75.0	74.6	74.3	74.0	73.7	73.3	73.0	72.7	72.3	72.0	71.7	71.3	71.0	70.7	70.4	70.1	69.7	69.4	69.1
68.7	68.4	68.1	67.7	67.4	67.1	66.7	66.4	66.1	65.7	65.4	65.3	65.2	65.1	65.0	64.8	64.7	64.6	64.5	64.4
64.3	64.2	64.1	63.9	63.8	63.7	63.6	63.5	63.4	63.3	63.1	63.0	62.9	62.8	62.7	62.6	62.5	62.4	62.2	62.1
62.0	61.8	61.6	61.4	61.2	61.0	60.8	60.6	60.4	60.2	60.0	59.8	59.6	59.4	59.2	59.0	58.8	58.6	58.4	58.2
57.9	57.7	57.5	57.3	57.1	56.8	56.6	56.4	56.2	56.0	55.8	55.6	55.3	55.1	54.9	54.7	54.5	54.3	54.0	53.8
53.6	53.4	53.2	53.0	52.7	52.5	52.3	52.1	51.9	51.7	51.5	51.2	51.0	50.8	50.6	50.4	50.2	50.0	49.7	49.5
49.3	49.1	48.9	48.7	48.4	48.2	48.0	47.8	47.6	47.4	47.1	46.9	46.7	46.5	46.3	46.1	45.9	45.6	45.4	45.2
45.1																			

Table 20 (Cont'd)

```
NIGHT EX NEG MEAN                                   17
      9        301     100.00    -199.00    NIGHT EX NEG MEAN
   121.30    100.00METER                          100.00       1.00
   106.50   1000.00FEET                           304.80      49.40
    97.60   2000.00FEET                           609.60      79.50
    82.60      1.00MILE                          1609.34     121.66
    73.20      2.00MILE                          3218.69     151.77
    60.50      5.00MILE                          8046.72     191.56
    55.10     10.00MILE                         16093.44     221.66
    51.60     15.00MILE                         24140.16     239.27
    39.60 100000.00METER                       100000.00     301.00
```

121.3	120.8	120.5	120.2	119.9	119.6	119.3	119.0	118.7	118.4	118.1	117.8	117.5	117.2	116.9	116.6	116.3	115.9	115.6	115.3
115.0	114.7	114.4	114.1	113.8	113.5	113.2	112.9	112.6	112.3	112.0	111.7	111.4	111.1	110.8	110.4	110.1	109.8	109.5	109.2
108.9	108.6	108.3	108.0	107.7	107.4	107.1	106.8	106.5	106.2	105.9	105.6	105.3	105.0	104.7	104.4	104.1	103.8	103.5	103.2
102.9	102.6	102.3	102.0	101.7	101.4	101.1	100.9	100.6	100.3	100.0	99.7	99.4	99.1	98.8	98.5	98.2	97.9	97.6	[illegible]
96.9	96.5	96.2	95.8	95.5	95.1	94.8	94.4	94.0	93.7	93.3	93.0	92.6	92.3	91.9	91.6	91.2	90.8	90.5	90.1
89.8	89.4	89.1	88.7	88.4	88.0	87.6	87.3	86.9	86.6	86.2	85.9	85.5	85.1	84.8	84.4	84.1	83.7	83.4	83.0
82.7	82.3	82.0	81.7	81.4	81.1	80.8	80.5	80.2	79.8	79.5	79.2	78.9	78.6	78.3	78.0	77.7	77.3	77.0	76.7
76.4	76.1	75.8	75.5	75.2	74.8	74.5	74.2	73.9	73.6	73.3	73.0	72.6	72.3	72.0	71.7	71.4	71.1	70.7	70.4
70.1	69.8	69.5	69.1	68.8	68.5	68.2	67.9	67.5	67.2	66.9	66.6	66.3	65.9	65.6	65.3	65.0	64.7	64.3	[illegible]
63.7	63.4	63.1	62.8	62.4	62.1	61.8	61.5	61.2	60.8	60.5	60.3	60.2	60.0	59.8	59.6	59.4	59.3	59.1	58.9
58.7	58.5	58.4	58.2	58.0	57.8	57.6	57.5	57.3	57.1	56.9	56.7	56.6	56.4	56.2	56.0	55.8	55.7	55.5	55.3
55.1	54.9	54.7	54.5	54.3	54.1	53.9	53.7	53.5	53.3	53.1	52.9	52.7	52.5	52.4	52.2	52.0	51.8	51.6	51.4
51.2	51.0	50.8	50.6	50.4	50.2	50.0	49.8	49.6	49.4	49.2	49.0	48.8	48.6	48.4	48.3	48.1	47.9	47.7	47.5
47.3	47.1	46.9	46.7	46.5	46.3	46.1	45.9	45.7	45.5	45.3	45.1	44.9	44.8	44.6	44.4	44.2	44.0	43.8	43.6
43.4	43.2	43.0	42.8	42.6	42.4	42.2	42.0	41.8	41.6	41.4	41.3	41.1	40.9	40.7	40.5	40.3	40.1	39.9	39.7
39.6																			

Table 20 (Cont'd)

NIGHT EX NEG MIN					18	
9	301	100.00	-199.00	NIGHT EX NEG MIN		
117.10	100.00METER				100.00	1.00
100.80	1000.00FEET				304.80	49.40
90.70	2000.00FEET				609.60	79.50
75.10	1.00MILE				1609.34	121.66
64.10	2.00MILE				3218.69	151.77
50.60	5.00MILE				8046.72	191.56
44.60	10.00MILE				16093.44	221.66
40.60	15.00MILE				24140.16	239.27
29.00	100000.00METER				100000.00	301.00

117.1	116.6	116.3	115.9	115.6	115.2	114.9	114.6	114.2	113.9	113.6	113.2	112.9	112.6	112.2	111.9	111.5	111.2	110.9	110.5
110.2	109.9	109.5	109.2	108.8	108.5	108.2	107.8	107.5	107.2	106.8	106.5	106.2	105.8	105.5	105.1	104.8	104.5	104.1	103.8
103.5	103.1	102.8	102.5	102.1	101.8	101.4	101.1	100.8	100.4	100.1	99.8	99.4	99.1	98.8	98.4	98.1	97.7	97.4	97.1
96.7	96.4	96.1	95.7	95.4	95.1	94.7	94.4	94.1	93.7	93.4	93.1	92.7	92.4	92.0	91.7	91.4	91.0	90.7	90.3
90.0	89.6	89.2	88.9	88.5	88.1	87.7	87.4	87.0	86.6	86.3	85.9	85.5	85.2	84.8	84.4	84.0	83.7	83.3	82.9
82.6	82.2	81.8	81.5	81.1	80.7	80.3	80.0	79.6	79.2	78.9	78.5	78.1	77.8	77.4	77.0	76.6	76.3	75.9	75.5
75.2	74.8	74.4	74.1	73.7	73.3	73.0	72.6	72.2	71.9	71.5	71.1	70.8	70.4	70.0	69.7	69.3	68.9	68.6	68.2
67.9	67.5	67.1	66.8	66.4	66.0	65.7	65.3	64.9	64.6	64.2	63.9	63.5	63.2	62.8	62.5	62.2	61.8	61.5	61.1
60.8	60.5	60.1	59.8	59.4	59.1	58.8	58.4	58.1	57.7	57.4	57.1	56.7	56.4	56.0	55.7	55.4	55.0	54.7	54.4
54.0	53.7	53.3	53.0	52.7	52.3	52.0	51.6	51.3	51.0	50.6	50.4	50.2	50.0	49.8	49.6	49.4	49.2	49.0	48.8
48.6	48.4	48.2	48.0	47.8	47.6	47.4	47.2	47.0	46.8	46.6	46.4	46.2	46.0	45.8	45.6	45.4	45.2	45.0	44.8
44.6	44.4	44.2	44.0	43.7	43.5	43.3	43.0	42.8	42.6	42.4	42.1	41.9	41.7	41.5	41.2	41.0	40.8	40.6	40.4
40.2	40.0	39.8	39.6	39.4	39.2	39.1	38.9	38.7	38.5	38.3	38.1	37.9	37.7	37.6	37.4	37.2	37.1	36.8	36.7
36.4	36.2	36.0	35.9	35.7	35.5	35.3	35.1	34.9	34.7	34.5	34.4	34.2	34.0	33.8	33.6	33.4	33.2	33.0	32.9
32.7	32.5	32.3	32.1	31.9	31.7	31.5	31.3	31.2	31.0	30.8	30.6	30.4	30.2	30.0	29.8	29.7	29.5	29.3	29.1
29.0																			

Table 20 (Cont'd)

```
DAY FOCUS PERCENT                                        19
     7         301      100.00     -199.00     DAY FOCUS PERCENT
   0.00    1000.00FEET                                 304.80      49.40
   0.00    2000.00FEET                                 609.60      79.50
    .46       1.00MILE                                1609.34     121.66
   2.30       2.00MILE                                3218.69     151.77
   9.60       5.00MILE                                8046.72     191.56
   9.70      10.00MILE                               16093.44     221.66
   3.10      15.00MILE                               24140.16     239.27
```

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
.0	.0	.0	.1	.1	.1	.1	.1	.1	.1	.1	.1	.2	.2	.2	.2	.2	.2	.2	.2
.2	.3	.3	.3	.3	.3	.3	.3	.3	.3	.3	.4	.4	.4	.4	.4	.4	.4	.4	.4
.5	.5	.6	.6	.7	.8	.8	.9	.9	1.0	1.1	1.1	1.2	1.2	1.3	1.4	1.4	1.5	1.6	1.6
1.7	1.7	1.8	1.9	1.9	2.0	2.0	2.1	2.2	2.2	2.3	2.4	2.6	2.8	3.0	3.2	3.4	3.5	3.7	3.9
4.1	4.3	4.5	4.6	4.8	5.0	5.2	5.4	5.6	5.7	5.9	6.1	6.3	6.5	6.7	6.8	7.0	7.2	7.4	7.6
7.8	7.9	8.1	8.3	8.5	8.7	8.9	9.0	9.2	9.4	9.6	9.6	9.6	9.6	9.6	9.6	9.6	9.6	9.6	9.6
9.6	9.6	9.6	9.6	9.6	9.6	9.7	9.7	9.7	9.7	9.7	9.7	9.7	9.7	9.7	9.7	9.7	9.7	9.7	9.7
9.7	9.4	9.0	8.6	8.3	7.9	7.5	7.1	6.8	6.4	6.0	5.6	5.3	4.9	4.5	4.1	3.8	3.4	3.1	3.1
3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1
3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1
3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1
3.1																			

Table 20 (Cont'd)

```
DAY BASE PERCENT                                               20
       7        301     100.00     -199.00     DAY BASE PERCENT
   25.07   1000.00FEET                               304.80      49.40
   32.44   2000.00FEET                               609.60      79.50
   18.53      1.00MILE                              1609.34     121.66
   20.60      2.00MILE                              3218.69     151.77
   25.90      5.00MILE                              8046.72     191.56
   31.80     10.00MILE                             16093.44     221.66
   30.00     15.00MILE                             24140.16     239.27
```

25.1	25.1	25.1	25.1	25.1	25.1	25.1	25.1	25.1	25.1	25.1	25.1	25.1	25.1	25.1	25.1	25.1	25.1	25.1	25.1
25.1	25.1	25.1	25.1	25.1	25.1	25.1	25.1	25.1	25.1	25.1	25.1	25.1	25.1	25.1	25.1	25.1	25.1	25.1	25.1
25.1	25.1	25.1	25.1	25.1	25.1	25.1	25.1	25.1	25.3	25.6	25.8	26.1	26.3	26.6	26.8	27.1	27.3	27.5	27.8
28.0	28.3	28.5	28.8	29.0	29.3	29.5	29.7	30.0	30.2	30.5	30.7	31.0	31.2	31.5	31.7	31.9	32.2	32.4	32.1
31.8	31.5	31.1	30.8	30.5	30.1	29.8	29.5	29.1	28.8	28.5	28.2	27.8	27.5	27.2	26.8	26.5	26.2	25.8	25.5
25.2	24.9	24.5	24.2	23.9	23.5	23.2	22.9	22.5	22.2	21.9	21.6	21.2	20.9	20.6	20.2	19.9	19.6	19.2	18.9
18.6	18.6	18.7	18.7	18.8	18.9	18.9	19.0	19.1	19.1	19.2	19.3	19.3	19.4	19.5	19.6	19.6	19.7	19.8	19.8
19.9	20.0	20.0	20.1	20.2	20.2	20.3	20.4	20.4	20.5	20.6	20.7	20.8	21.0	21.1	21.2	21.4	21.5	21.6	21.6
21.9	22.0	22.2	22.3	22.4	22.6	22.7	22.8	23.0	23.1	23.2	23.4	23.5	23.6	23.8	23.9	24.0	24.2	24.3	24.4
24.6	24.7	24.8	25.0	25.1	25.2	25.4	25.5	25.6	25.8	25.9	26.1	26.3	26.5	26.7	26.9	27.1	27.3	27.5	27.7
27.8	28.0	28.2	28.4	28.6	28.8	29.0	29.2	29.4	29.6	29.8	30.0	30.2	30.4	30.6	30.8	31.0	31.2	31.4	31.6
31.8	31.7	31.6	31.5	31.4	31.3	31.2	31.1	31.0	30.9	30.8	30.7	30.6	30.5	30.4	30.3	30.2	30.1	30.0	30.0
30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
30.0																			

Table 20 (Cont'd)

DAY NEG PERCENT					
7	301	100.00	-199.00	21 DAY NEG PERCENT	
72.69	1000.00FEET			304.80	49.40
59.22	2000.00FEET			609.60	79.50
64.88	1.00MILE			1609.34	121.66
39.60	2.00MILE			3218.69	151.77
35.00	5.00MILE			8046.72	191.56
32.60	10.00MILE			16093.44	221.66
32.10	15.00MILE			24140.16	239.27

72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7
72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7
72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.2	71.8	71.3	70.9	70.4	70.0	69.5	69.1	68.6	68.2	67.7
67.3	66.8	66.4	65.9	65.5	65.0	64.6	64.1	63.7	63.2	62.8	62.4	61.9	61.5	61.0	60.6	60.1	59.7	59.2	[illegible]
59.5	59.6	59.8	59.9	60.0	60.2	60.3	60.4	60.6	60.7	60.8	61.0	61.1	61.2	61.4	61.5	61.6	61.8	61.9	[illegible]
62.2	62.3	62.4	62.6	62.7	62.8	63.0	63.1	63.2	63.4	63.5	63.6	63.8	63.9	64.1	64.2	64.3	64.5	64.6	[illegible]
64.9	64.2	63.3	62.5	61.7	60.8	60.0	59.1	58.3	57.5	56.6	55.8	54.9	54.1	53.3	52.4	51.6	50.7	49.9	[illegible]
48.2	47.4	46.5	45.7	44.9	44.0	43.2	42.3	41.5	40.7	39.8	39.5	39.4	39.3	39.2	39.1	38.9	38.8	38.7	38.6
38.5	38.4	38.2	38.1	38.0	37.9	37.8	37.7	37.6	37.4	37.3	37.2	37.1	37.0	36.9	36.7	36.6	36.5	36.4	36.3
36.2	36.0	35.9	35.8	35.7	35.6	35.5	35.4	35.2	35.1	35.0	34.9	34.8	34.8	34.7	34.6	34.5	34.4	34.4	34.3
34.2	34.1	34.0	34.0	33.9	33.8	33.7	33.6	33.6	33.5	33.4	33.3	33.3	33.2	33.1	33.0	32.9	32.9	32.8	32.7
32.6	32.6	32.5	32.5	32.5	32.5	32.4	32.4	32.4	32.3	32.3	32.3	32.3	32.2	32.2	32.2	32.2	32.1	32.1	[illegible]
32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1
32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1
32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1	32.1
32.1																			

Table 20 (Cont'd)

DAY EX NEG PERCENT

7	301	100.00	-199.00
2.24	1000.00FEET		
8.34	2000.00FEET		
16.13	1.00MILE		
37.50	2.00MILE		
29.50	5.00MILE		
25.90	10.00MILE		
34.80	15.00MILE		

22

DAY EX NEG PERCENT

304.80	49.40
609.60	79.50
1609.34	121.66
3218.69	151.77
8046.72	191.56
16093.44	221.66
24140.16	239.27

2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2
2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2
2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.5	2.7	2.9	3.1	3.3	3.5	3.7	3.9	4.1	4.3	4.5
4.7	4.9	5.1	5.3	5.5	5.7	5.9	6.1	6.3	6.5	6.7	6.9	7.1	7.3	7.5	7.7	7.9	8.1	8.3	8.5
8.7	8.9	9.1	9.3	9.4	9.6	9.8	10.0	10.2	10.4	10.6	10.7	10.9	11.1	11.3	11.5	11.7	11.8	12.0	12.2
12.4	12.6	12.8	13.0	13.1	13.3	13.5	13.7	13.9	14.1	14.3	14.4	14.6	14.8	15.0	15.2	15.4	15.5	15.7	15.9
16.1	16.7	17.4	18.1	18.9	19.6	20.3	21.0	21.7	22.4	23.1	23.8	24.5	25.2	26.0	26.7	27.4	28.1	28.8	29.5
30.2	30.9	31.6	32.3	33.1	33.8	34.5	35.2	35.9	36.6	37.3	37.4	37.2	37.0	36.7	36.5	36.3	36.1	35.9	35.7
35.5	35.3	35.1	34.9	34.7	34.5	34.3	34.1	33.9	33.7	33.5	33.3	33.1	32.9	32.7	32.5	32.3	32.1	31.9	31.7
31.5	31.3	31.1	30.9	30.7	30.5	30.3	30.1	29.9	29.7	29.5	29.4	29.3	29.1	29.0	28.9	28.8	28.7	28.6	28.4
28.3	28.2	28.1	28.0	27.8	27.7	27.6	27.5	27.4	27.2	27.1	27.0	26.9	26.8	26.6	26.5	26.4	26.3	26.2	26.
25.9	26.3	26.8	27.3	27.8	28.3	28.8	29.4	29.9	30.4	30.9	31.4	31.9	32.4	32.9	33.4	33.9	34.4	34.8	34.8
34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8
34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8
34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8
34.8																			

Table 20 (Cont'd)

NIGHT FOCUS PERCENT

7	301	100.00	-199.00
0.00	1000.00FEET		
0.00	2000.00FEET		
2.64	1.00MILE		
6.10	2.00MILE		
7.60	5.00MILE		
7.10	10.00MILE		
4.40	15.00MILE		

23

NIGHT FOCUS PERCENT

304.80	49.40
609.60	79.50
1609.34	121.66
3218.69	151.77
8046.72	191.56
16093.44	221.66
24140.16	239.27

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	.1
.1	.2	.3	.3	.4	.4	.5	.6	.6	.7	.8	.8	.9	.9	1.0	1.1	1.1	1.2	1.3	1.3
1.4	1.4	1.5	1.6	1.6	1.7	1.8	1.8	1.9	1.9	2.0	2.1	2.1	2.2	2.3	2.3	2.4	2.4	2.5	2.6
2.6	2.7	2.9	3.0	3.1	3.2	3.3	3.4	3.5	3.7	3.8	3.9	4.0	4.1	4.2	4.3	4.5	4.6	4.7	4.8
4.9	5.0	5.1	5.3	5.4	5.5	5.6	5.7	5.8	6.0	6.1	6.1	6.2	6.2	6.2	6.3	6.3	6.4	6.4	6.4
6.5	6.5	6.5	6.6	6.6	6.7	6.7	6.7	6.8	6.8	6.8	6.9	6.9	7.0	7.0	7.0	7.1	7.1	7.1	7.2
7.2	7.3	7.3	7.3	7.4	7.4	7.4	7.5	7.5	7.6	7.6	7.6	7.6	7.6	7.5	7.5	7.5	7.5	7.5	7.5
7.4	7.4	7.4	7.4	7.4	7.4	7.3	7.3	7.3	7.3	7.3	7.3	7.2	7.2	7.2	7.2	7.2	7.2	7.1	7.1
7.1	7.0	6.8	6.7	6.5	6.4	6.2	6.1	5.9	5.7	5.6	5.4	5.3	5.1	5.0	4.8	4.7	4.5	4.4	4.4
4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4
4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4
4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4
4.4																			

Table 20 (Cont'd)

```
NIGHT BASE PERCENT                                   24
      7          301      100.00     -199.00     NIGHT BASE PERCENT
    27.34    1000.00FEET                               304.80      49.40
    47.67    2000.00FEET                               609.60      79.50
    65.86       1.00MILE                              1609.34     121.66
    39.00       2.00MILE                              3218.69     151.77
    27.30       5.00MILE                              8046.72     191.56
    20.00      10.00MILE                             16093.44     221.66
    16.70      15.00MILE                             24140.16     239.27
```

27.3	27.3	27.3	27.3	27.3	27.3	27.3	27.3	27.3	27.3	27.3	27.3	27.3	27.3	27.3	27.3	27.3	27.3	27.3	27.3
27.3	27.3	27.3	27.3	27.3	27.3	27.3	27.3	27.3	27.3	27.3	27.3	27.3	27.3	27.3	27.3	27.3	27.3	27.3	27.3
27.3	27.3	27.3	27.3	27.3	27.3	27.3	27.3	27.3	28.1	28.8	29.4	30.1	30.8	31.5	32.1	32.8	33.5	34.2	34.8
35.5	36.2	36.9	37.5	38.2	38.9	39.6	40.2	40.9	41.6	42.3	42.9	43.6	44.3	45.0	45.6	46.3	47.0	47.7	48.1
48.5	49.0	49.4	49.8	50.3	50.7	51.1	51.6	52.0	52.4	52.8	53.3	53.7	54.1	54.6	55.0	55.4	55.9	56.3	56.7
57.2	57.6	58.0	58.5	58.9	59.3	59.7	60.2	60.6	61.0	61.5	61.9	62.3	62.8	63.2	63.6	64.1	64.5	64.9	65.4
65.8	65.1	64.2	63.3	62.4	61.5	60.7	59.8	58.9	58.0	57.1	56.2	55.3	54.4	53.5	52.6	51.7	50.8	49.9	49.1
48.2	47.3	46.4	45.5	44.6	43.7	42.8	41.9	41.0	40.1	39.2	38.8	38.5	38.2	37.9	37.6	37.3	37.0	36.7	36.4
36.1	35.8	35.6	35.3	35.0	34.7	34.4	34.1	33.8	33.5	33.2	32.9	32.6	32.3	32.0	31.7	31.4	31.1	30.8	30.6
30.3	30.0	29.7	29.4	29.1	28.8	28.5	28.2	27.9	27.6	27.3	27.1	26.8	26.6	26.3	26.1	25.9	25.6	25.4	25.1
24.9	24.6	24.4	24.2	23.9	23.7	23.4	23.2	22.9	22.7	22.5	22.2	22.0	21.7	21.5	21.3	21.0	20.8	20.5	20.3
20.0	19.8	19.7	19.5	19.3	19.1	18.9	18.7	18.5	18.3	18.2	18.0	17.8	17.6	17.4	17.2	17.0	16.8	16.7	16.7
16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7
16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7
16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7	16.7
16.7																			

Table 20 (Cont'd)

```
NIGHT NEG PERCENT                                   25
       7          301     100.00      -199.00     NIGHT NEG PERCENT
    72.66    1000.00FEET                                  304.80      49.40
    44.21    2000.00FEET                                  609.60      79.50
    24.35       1.00MILE                                 1609.34     121.66
    29.50       2.00MILE                                 3218.69     151.77
    31.30       5.00MILE                                 8046.72     191.56
    25.00      10.00MILE                                16093.44     221.66
    33.70      15.00MILE                                24140.16     239.27
```

72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7
72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7
72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	72.7	71.6	70.7	69.7	68.8	67.8	66.9	66.0	65.0	64.1	63.1	62.2
61.2	60.3	59.3	58.4	57.4	56.5	55.6	54.6	53.7	52.7	51.8	50.8	49.9	48.9	48.0	47.0	46.1	45.2	44.2	43.7
43.3	42.8	42.3	41.9	41.4	40.9	40.4	40.0	39.5	39.0	38.6	38.1	37.6	37.1	36.7	36.2	35.7	35.3	34.8	34.3
33.8	33.4	32.9	32.4	32.0	31.5	31.0	30.6	30.1	29.6	29.1	28.7	28.2	27.7	27.3	26.8	26.3	25.8	25.4	24.9
24.4	24.5	24.7	24.8	25.0	25.2	25.3	25.5	25.7	25.9	26.0	26.2	26.4	26.5	26.7	26.9	27.1	27.2	27.4	27.6
27.7	27.9	28.1	28.3	28.4	28.6	28.8	28.9	29.1	29.3	29.5	29.5	29.6	29.6	29.7	29.7	29.8	29.8	29.8	29.9
29.9	30.0	30.0	30.1	30.1	30.2	30.2	30.3	30.3	30.3	30.4	30.4	30.5	30.5	30.6	30.6	30.7	30.7	30.8	30.8
30.8	30.9	30.9	31.0	31.0	31.1	31.1	31.2	31.2	31.3	31.3	31.1	30.9	30.7	30.5	30.3	30.1	29.8	29.6	29.4
29.2	29.0	28.8	28.6	28.4	28.2	28.0	27.8	27.5	27.3	27.1	26.9	26.7	26.5	26.3	26.1	25.9	25.7	25.5	25.2
25.0	25.4	25.9	26.4	26.9	27.4	27.9	28.4	28.9	29.4	29.9	30.4	30.8	31.3	31.8	32.3	32.8	33.3	33.7	33.7
33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7
33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7
33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7	33.7
33.7																			

Table 20 (Cont'd)

NIGHT EX NEG PERCENT

7	301	100.00	-199.00
0.00	1000.00FEET		
8.12	2000.00FEET		
7.15	1.00MILE		
25.40	2.00MILE		
33.80	5.00MILE		
47.90	10.00MILE		
45.20	15.00MILE		

26
NIGHT EX NEG PERCENT

304.80	49.40
609.60	79.50
1609.34	121.66
3218.69	151.77
8046.72	191.56
16093.44	221.66
24140.16	239.27

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	.3	.6	.8	1.1	1.4	1.6	1.9	2.2	2.5	2.7	3.0
3.3	3.5	3.8	4.1	4.3	4.6	4.9	5.2	5.4	5.7	6.0	6.2	6.5	6.8	7.0	7.3	7.6	7.8	8.1	8.1
8.1	8.1	8.0	8.0	8.0	8.0	7.9	7.9	7.9	7.9	7.8	7.8	7.8	7.8	7.8	7.7	7.7	7.7	7.7	7.6
7.6	7.6	7.6	7.5	7.5	7.5	7.5	7.5	7.4	7.4	7.4	7.4	7.3	7.3	7.3	7.3	7.2	7.2	7.2	7.2
7.2	7.7	8.3	8.9	9.5	10.1	10.7	11.3	11.9	12.5	13.1	13.7	14.3	14.9	15.5	16.1	16.8	17.4	18.0	18.6
19.2	19.8	20.4	21.0	21.6	22.2	22.8	23.4	24.0	24.6	25.2	25.6	25.8	26.0	26.2	26.4	26.6	26.8	27.0	27.2
27.5	27.7	27.9	28.1	28.3	28.5	28.7	28.9	29.1	29.4	29.6	29.8	30.0	30.2	30.4	30.6	30.8	31.0	31.3	31.5
31.7	31.9	32.1	32.3	32.5	32.7	32.9	33.2	33.4	33.6	33.8	34.2	34.7	35.2	35.6	36.1	36.6	37.0	37.5	38.0
38.5	38.9	39.4	39.9	40.3	40.8	41.3	41.7	42.2	42.7	43.1	43.6	44.1	44.5	45.0	45.5	45.9	46.4	46.9	47.4
47.8	47.8	47.6	47.5	47.3	47.2	47.0	46.9	46.7	46.5	46.4	46.2	46.1	45.9	45.8	45.6	45.5	45.3	45.2	45.2
45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2
45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2
45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2	45.2
45.2																			

Table 20 (Cont'd)

CHARGE	SIZE	CORRECTI			CHARGE SIZE	CORRECTI
	21	601	100.00	201.00	27	
-29.80	.01				.01	5.14
-26.10	.02				.02	28.88
-22.50	.04				.04	61.21
-18.90	.08				.08	90.21
-15.40	.16				.16	120.38
-12.00	.31				.31	150.49
-8.80	.63				.63	180.59
-5.6	1.25				1.25	210.69
-2.70	2.50				2.50	240.79
0.00	5.00				5.00	270.90
2.40	10.00				10.00	301.00
4.60	20.00				20.00	331.10
6.60	40.00				40.00	361.21
8.50	80.00				80.00	391.31
10.20	160.00				160.00	421.41
11.80	320.00				320.00	451.51
13.40	640.00				640.00	481.62
14.90	1280.00				1280.00	511.72
16.40	2560.00				2560.00	541.82
17.60	5120.00				5120.00	571.93
18.20	9999.00				9999.00	601.00

-29.8	-29.8	-29.8	-29.8	-29.8	-29.6	-29.4	-29.3	-29.1	-29.0	-28.8	-28.7	-28.5	-28.3	-28.2	-28.0	-27.9	-27.7	-27.6	-27.4
-27.2	-27.1	-26.9	-26.8	-26.6	-26.5	-26.3	-26.2	-26.0	-25.9	-25.8	-25.7	-25.6	-25.5	-25.4	-25.3	-25.1	-25.0	-24.9	-24.8
-24.7	-24.6	-24.5	-24.4	-24.2	-24.1	-24.0	-23.9	-23.8	-23.7	-23.6	-23.5	-23.4	-23.2	-23.1	-23.0	-22.9	-22.8	-22.7	-22.6
-22.5	-22.3	-22.2	-22.1	-22.0	-21.8	-21.7	-21.6	-21.5	-21.3	-21.2	-21.1	-21.0	-20.8	-20.7	-20.6	-20.5	-20.4	-20.2	-20.1
-20.0	-19.9	-19.7	-19.6	-19.5	-19.4	-19.2	-19.1	-19.0	-18.9	-18.8	-18.6	-18.5	-18.4	-18.3	-18.2	-18.1	-17.9	-17.8	-17.7
-17.6	-17.5	-17.4	-17.2	-17.1	-17.0	-16.9	-16.8	-16.7	-16.5	-16.4	-16.3	-16.2	-16.1	-16.0	-15.9	-15.7	-15.6	-15.5	-15.4
-15.3	-15.2	-15.0	-14.9	-14.8	-14.7	-14.6	-14.5	-14.4	-14.3	-14.1	-14.0	-13.9	-13.8	-13.7	-13.6	-13.5	-13.4	-13.2	-13.1
-13.0	-12.9	-12.8	-12.7	-12.6	-12.5	-12.3	-12.2	-12.1	-12.0	-11.9	-11.8	-11.7	-11.6	-11.5	-11.4	-11.3	-11.1	-11.	-10.9
-10.8	-10.7	-10.6	-10.5	-10.4	-10.3	-10.2	-10.1	-10.0	-9.9	-9.8	-9.7	-9.6	-9.4	-9.3	-9.2	-9.1	-9.0	-8.9	-8.8
-8.7	-8.6	-8.5	-8.4	-8.3	-8.2	-8.1	-8.0	-7.9	-7.7	-7.6	-7.5	-7.4	-7.3	-7.2	-7.1	-7.0	-6.9	-6.8	-6.7
-6.6	-6.5	-6.4	-6.3	-6.2	-6.0	-5.9	-5.8	-5.7	-5.6	-5.5	-5.4	-5.3	-5.2	-5.1	-5.0	-4.9	-4.8	-4.8	-4.7
-4.6	-4.5	-4.4	-4.3	-4.2	-4.1	-4.0	-3.9	-3.8	-3.7	-3.6	-3.5	-3.4	-3.3	-3.2	-3.1	-3.0	-2.9	-2.8	-2.7
-2.6	-2.5	-2.5	-2.4	-2.3	-2.2	-2.1	-2.0	-1.9	-1.8	-1.7	-1.7	-1.6	-1.5	-1.4	-1.3	-1.2	-1.1	-1.0	-.9
-.8	-.8	-.7	-.6	-.5	-.4	-.3	-.2	-.1	-.0	.0	.1	.2	.3	.4	.4	.5	.6	.7	.8
.8	.9	1.0	1.1	1.2	1.2	1.3	1.4	1.5	1.6	1.6	1.7	1.8	1.9	2.0	2.0	2.1	2.2	2.3	2.4
2.4	2.5	2.6	2.7	2.7	2.8	2.9	2.9	3.0	3.1	3.2	3.2	3.3	3.4	3.5	3.5	3.6	3.7	3.8	3.8
3.9	4.0	4.0	4.1	4.2	4.3	4.3	4.4	4.5	4.6	4.6	4.7	4.8	4.8	4.9	5.0	5.0	5.1	5.2	5.2
5.3	5.4	5.4	5.5	5.6	5.6	5.7	5.8	5.8	5.9	6.0	6.0	6.1	6.2	6.2	6.3	6.4	6.4	6.5	6.6
6.6	6.7	6.7	6.8	6.9	6.9	7.0	7.1	7.1	7.2	7.2	7.3	7.4	7.4	7.5	7.6	7.6	7.7	7.8	7.8
7.9	7.9	8.0	8.1	8.1	8.2	8.3	8.3	8.4	8.4	8.5	8.6	8.6	8.7	8.7	8.8	8.8	8.9	9.0	9.0
9.1	9.1	9.2	9.2	9.3	9.4	9.4	9.5	9.5	9.6	9.6	9.7	9.8	9.8	9.9	9.9	10.0	10.0	10.1	10.1
10.2	10.3	10.3	10.4	10.4	10.5	10.5	10.6	10.6	10.7	10.7	10.8	10.8	10.9	10.9	11.0	11.1	11.1	11.2	11.2
11.3	11.3	11.4	11.4	11.5	11.5	11.6	11.6	11.7	11.7	11.8	11.9	11.9	12.0	12.0	12.1	12.1	12.2	12.2	12.3
12.3	12.4	12.4	12.5	12.5	12.6	12.6	12.7	12.8	12.8	12.9	12.9	13.0	13.	13.1	13.1	13.2	13.2	13.3	13.3
13.4	13.4	13.5	13.5	13.6	13.6	13.7	13.7	13.8	13.8	13.9	13.9	14.0	14.0	14.1	14.1	14.2	14.2	14.3	14.3
14.4	14.4	14.5	14.5	14.6	14.6	14.7	14.7	14.8	14.8	14.9	14.9	15.0	15.0	15.1	15.1	15.2	15.2	15.3	15.3
15.4	15.4	15.5	15.5	15.6	15.6	15.7	15.7	15.8	15.8	15.9	15.9	16.0	16.0	16.1	16.1	16.2	16.2	16.3	16.3
16.4	16.4	16.5	16.5	16.5	16.6	16.6	16.7	16.7	16.7	16.8	16.8	16.9	16.9	16.9	17.0	17.0	17.1	17.1	17.1
17.2	17.2	17.3	17.3	17.3	17.4	17.4	17.5	17.5	17.5	17.6	17.6	17.6	17.7	17.7	17.7	17.7	17.7	17.8	17.8
17.8	17.8	17.8	17.9	17.9	17.9	17.9	17.9	18.0	18.0	18.0	18.0	18.0	18.1	18.1	18.1	18.1	18.1	18.2	18.2
18.2																			

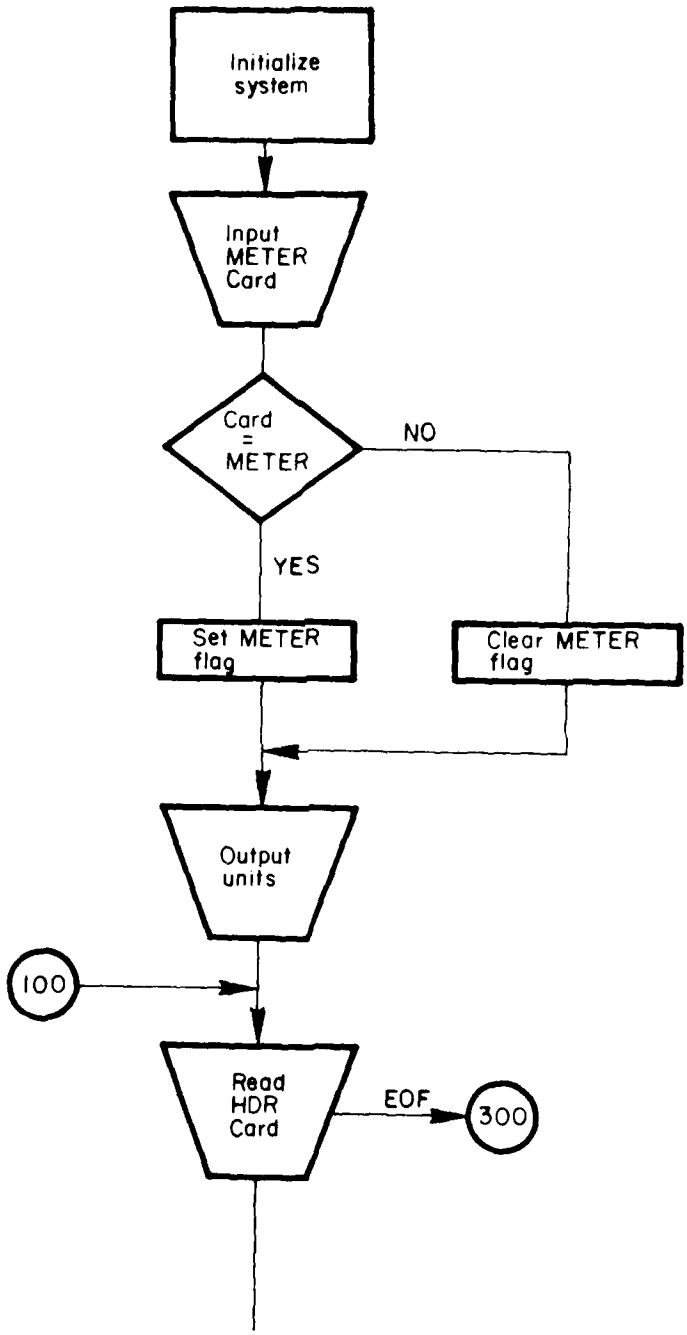


Figure 5 CDNL flowchart

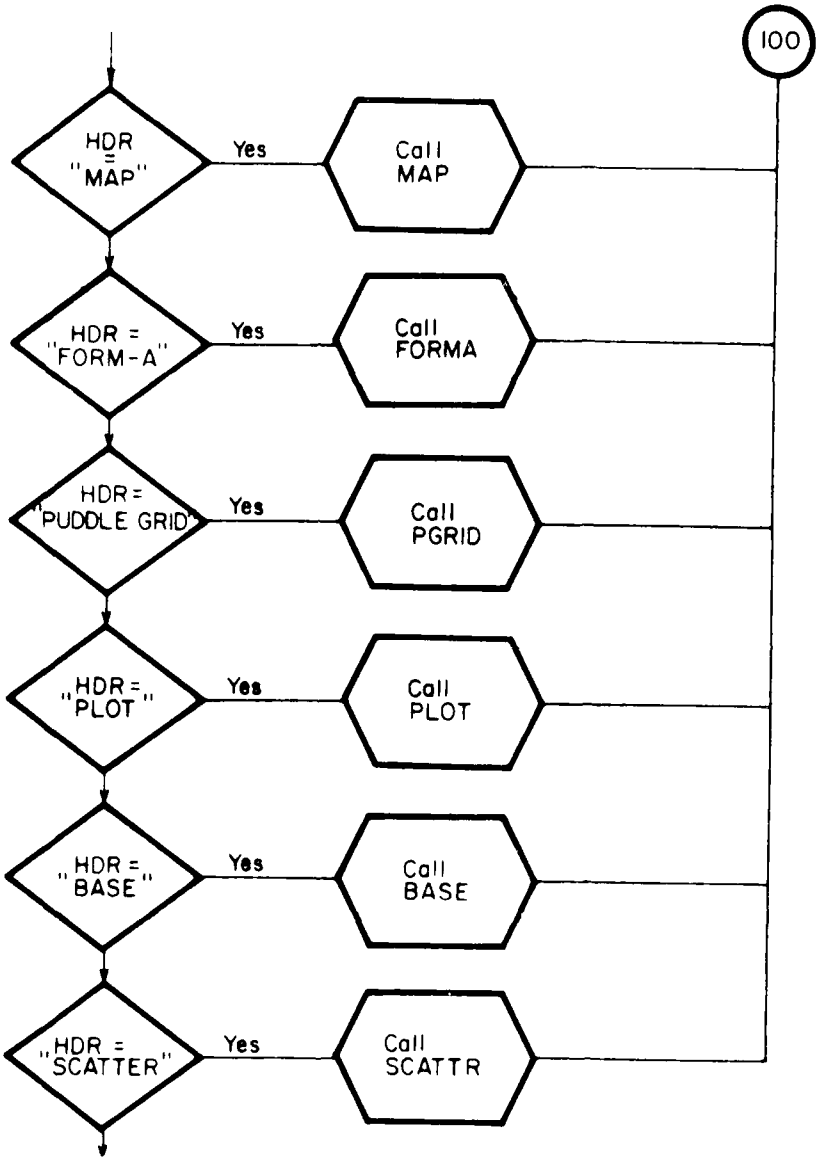


Figure 5. (Cont'd)

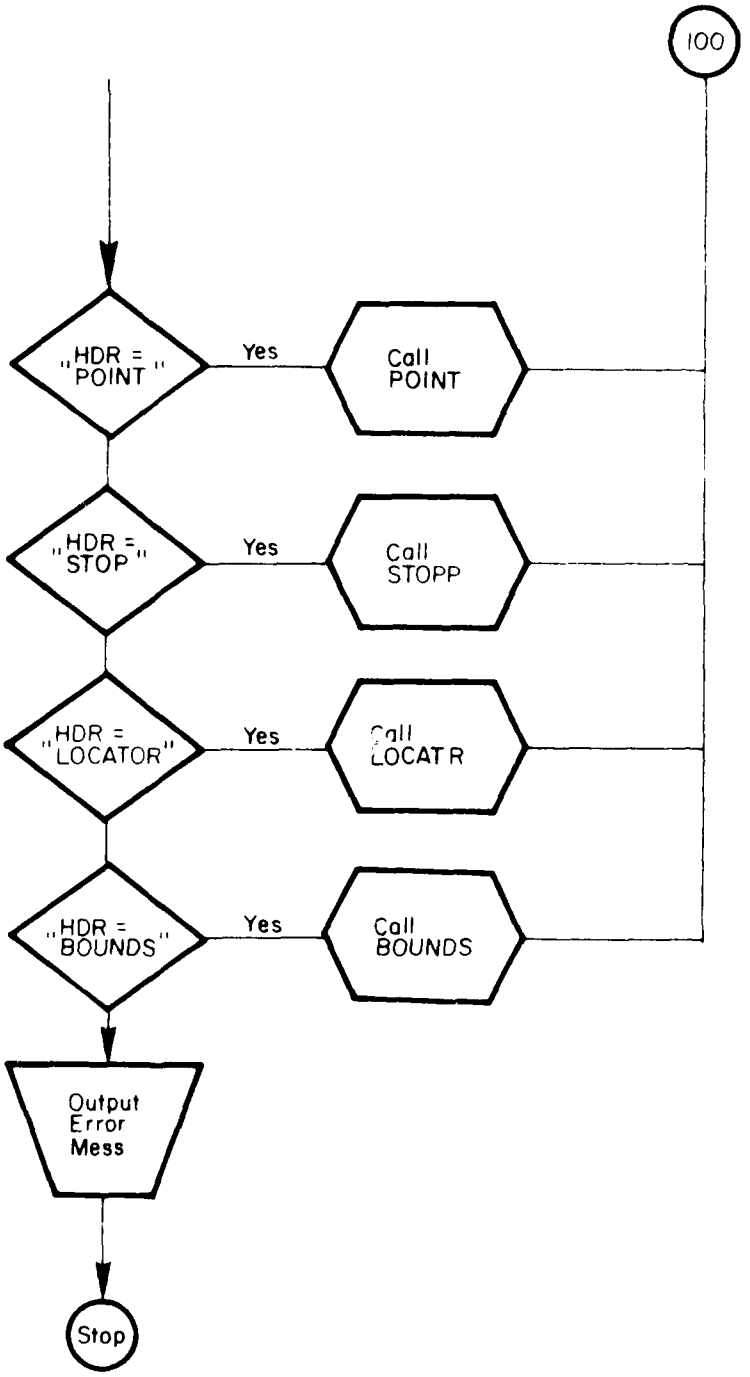


Figure 5. (Cont'd)

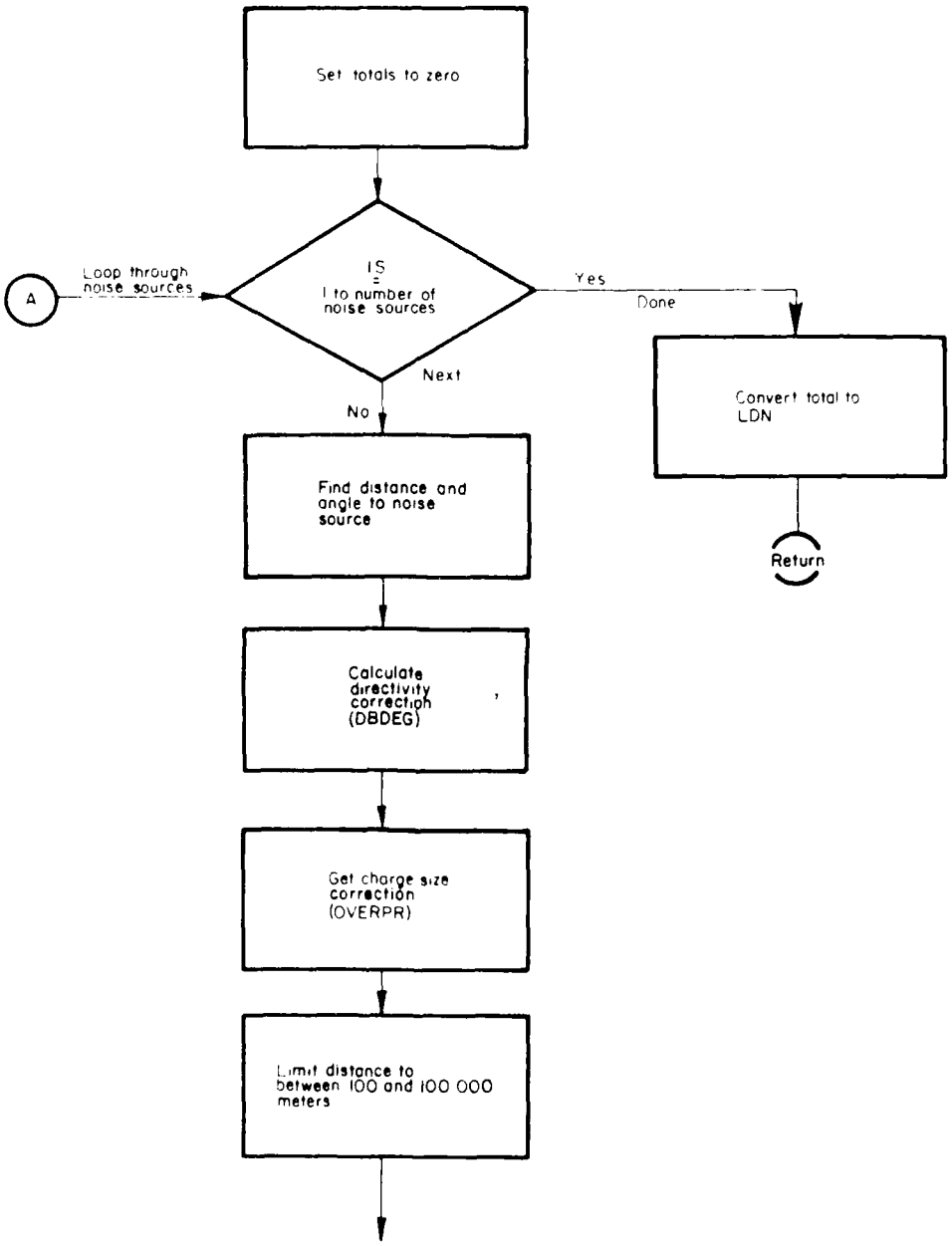


Figure 6 CALCNR flowchart

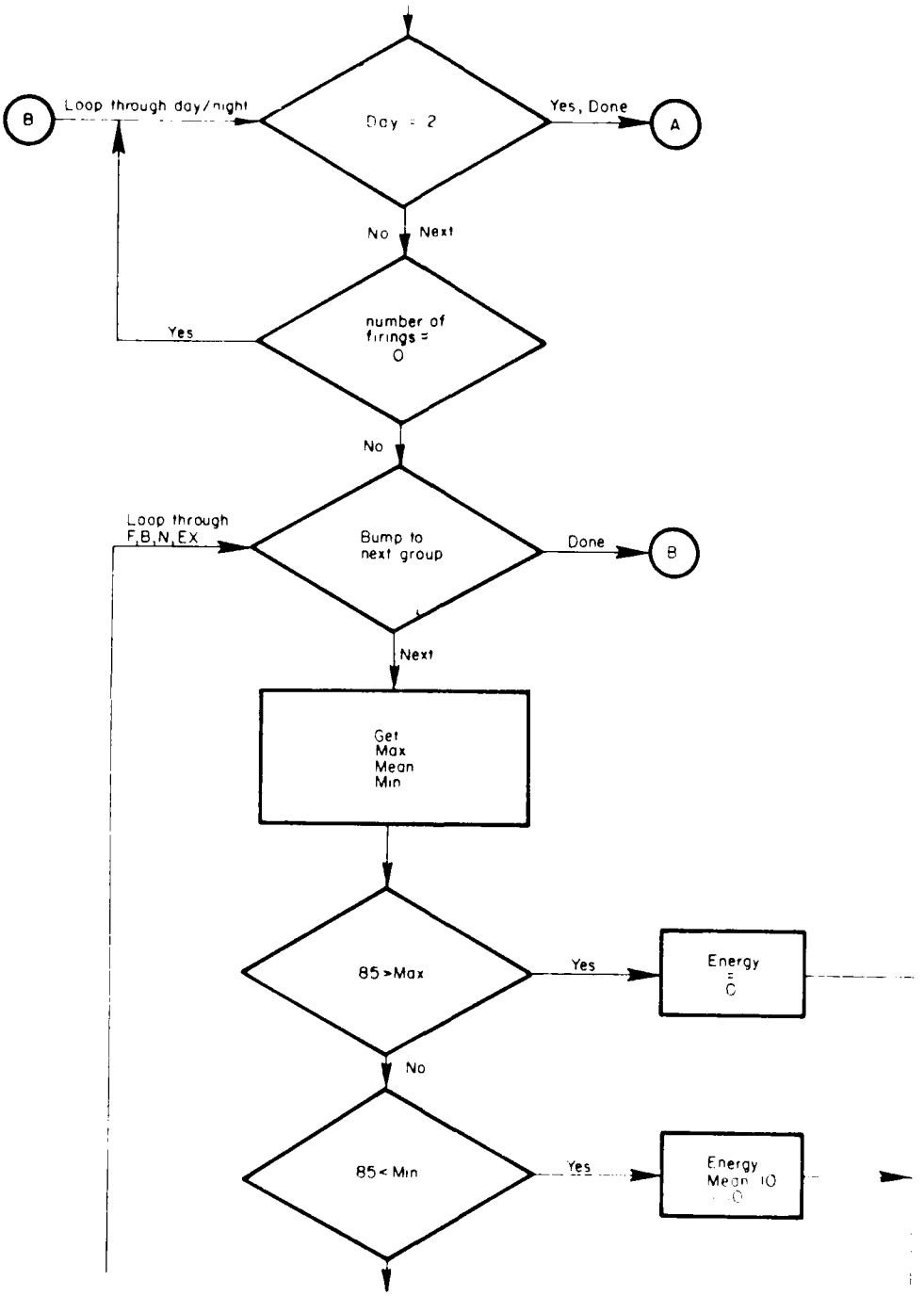


Figure 6. (Cont'd)

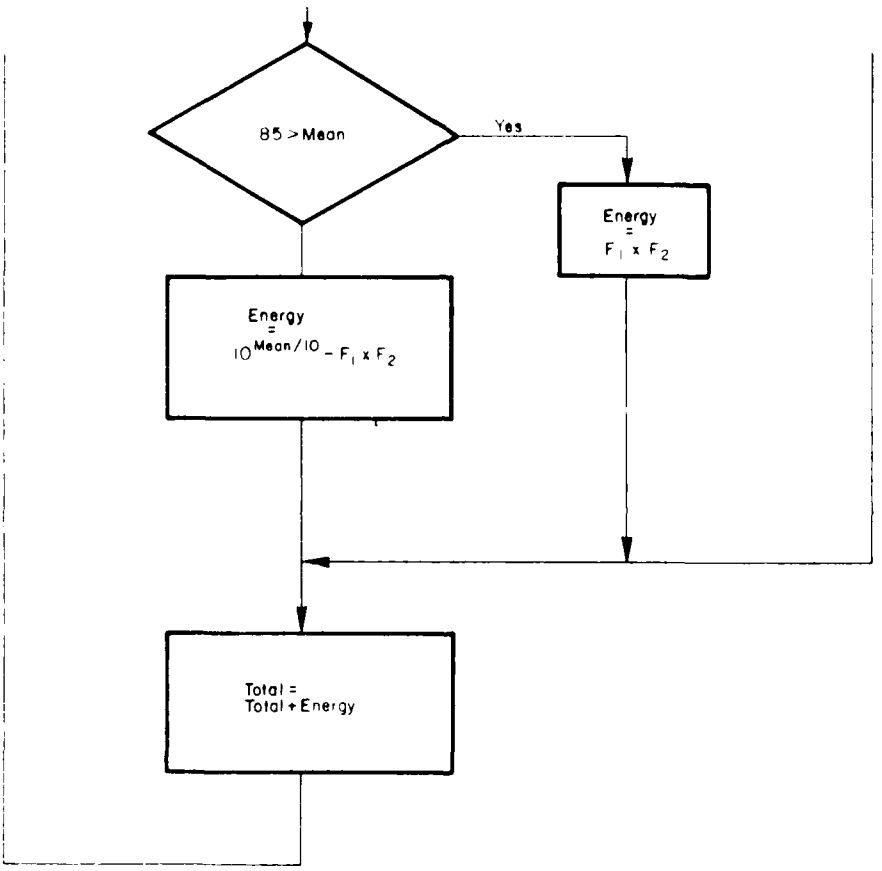


Figure 6. (Cont'd).

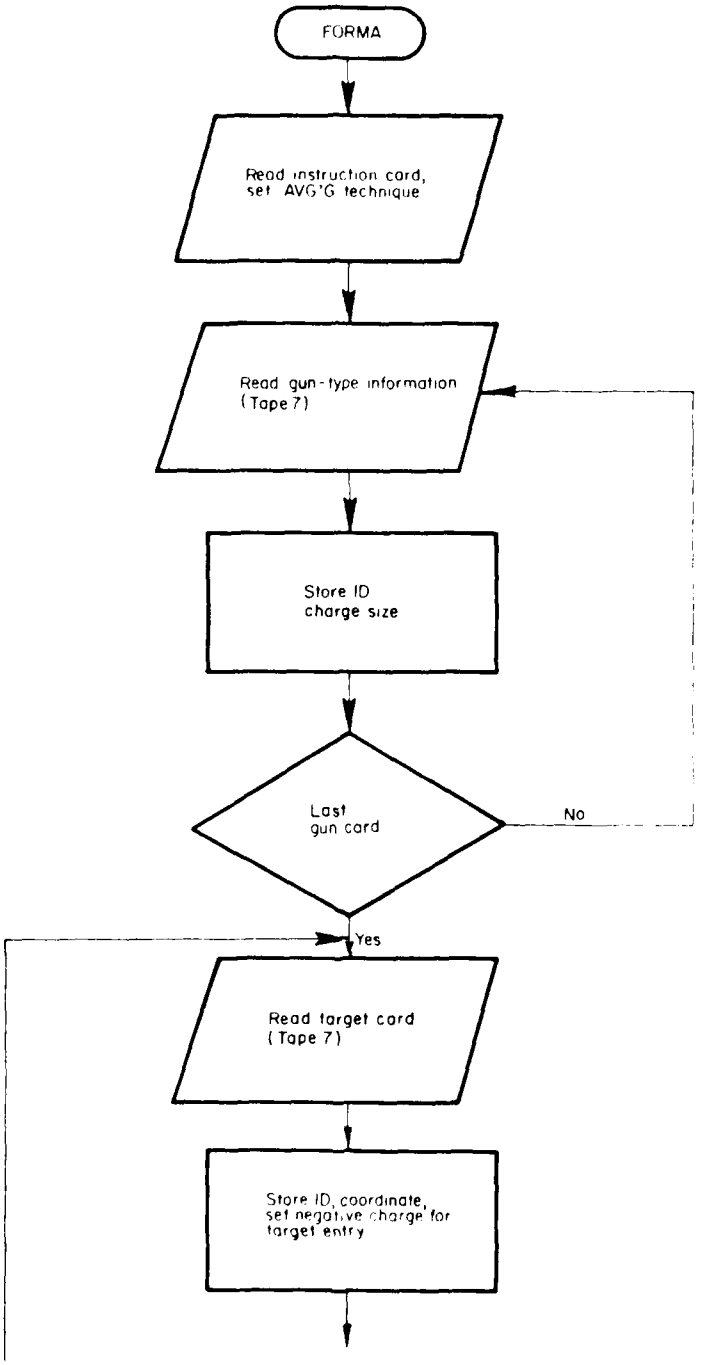


Figure 7 FORMA flowchart

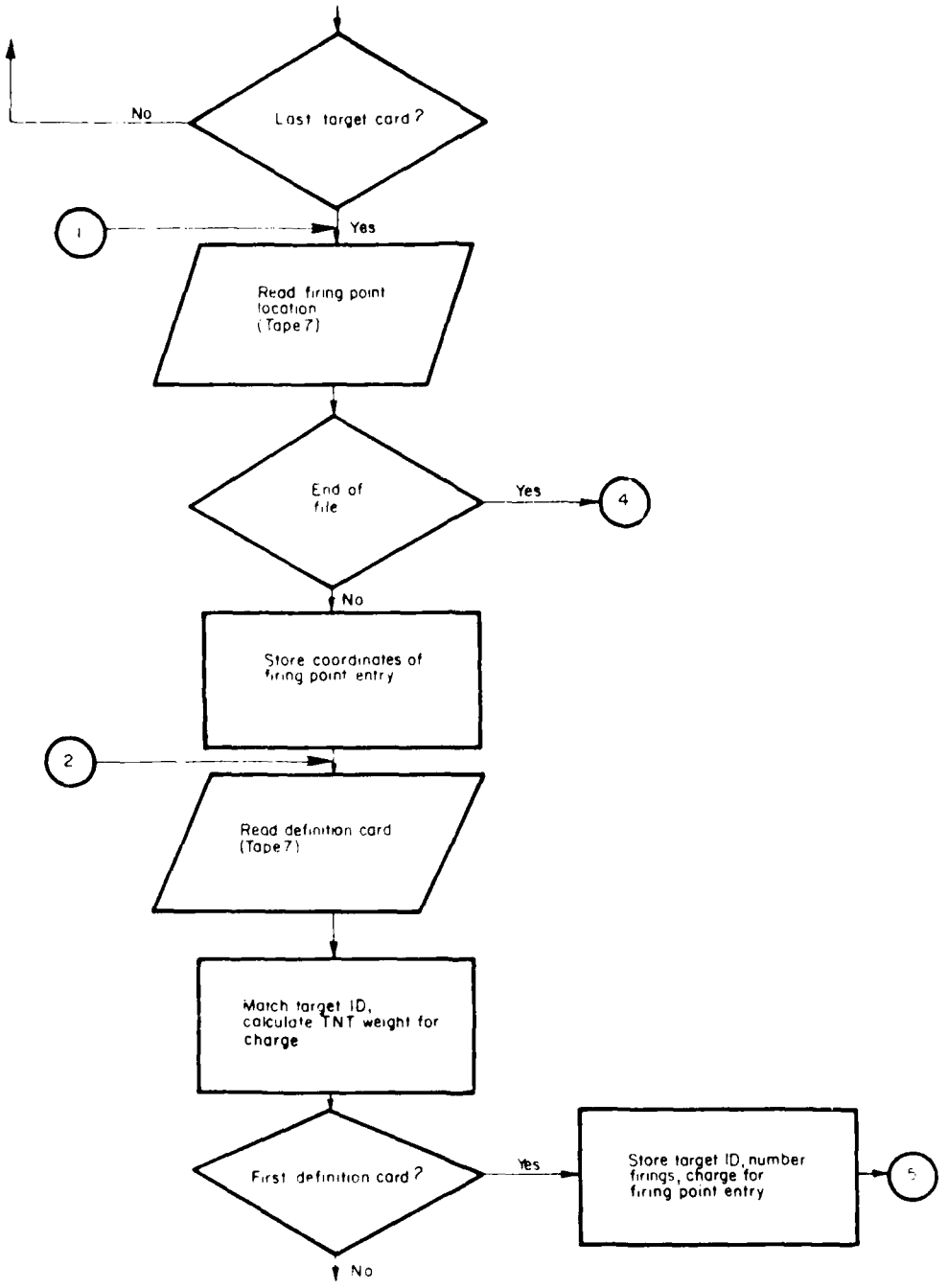


Figure 7. (Cont'd)

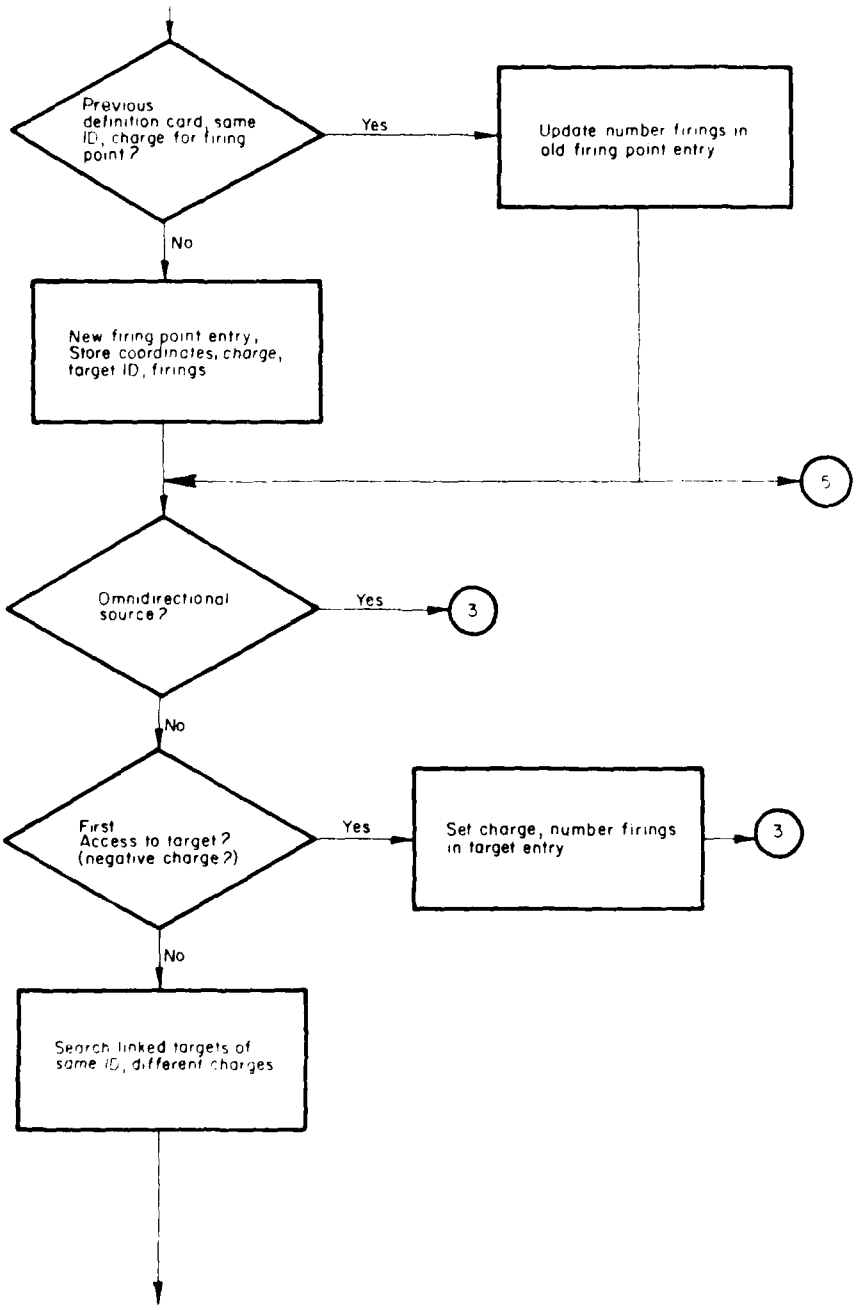


Figure 7. (Cont'd)

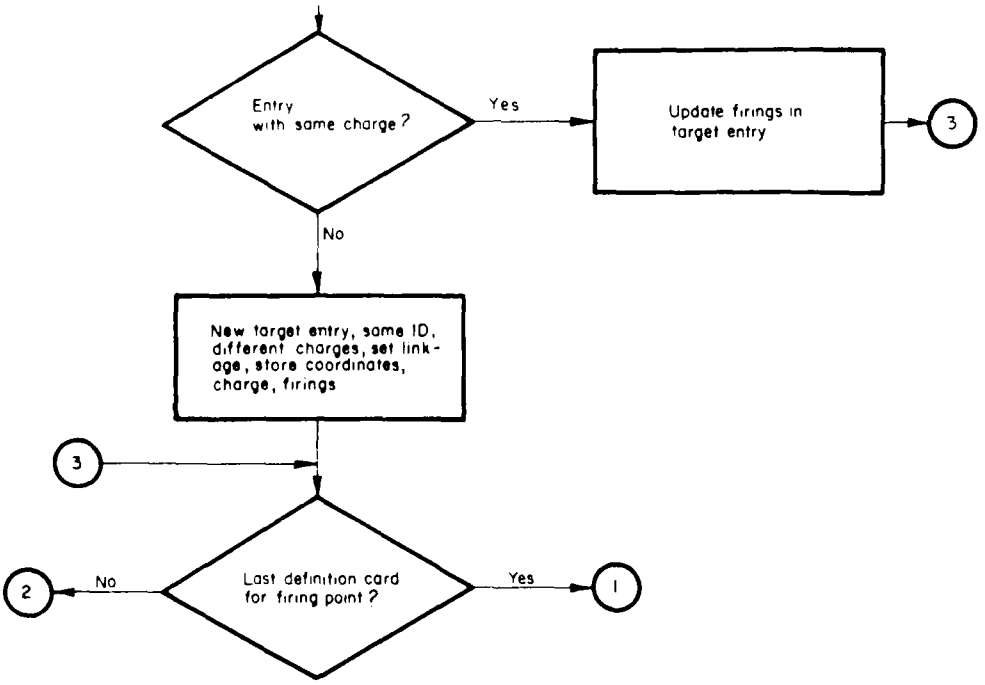


Figure 7. (Cont'd).

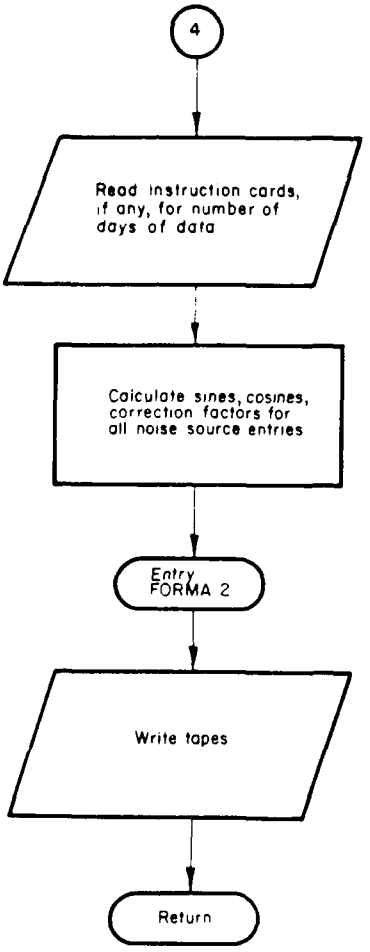


Figure 7. (Cont'd)

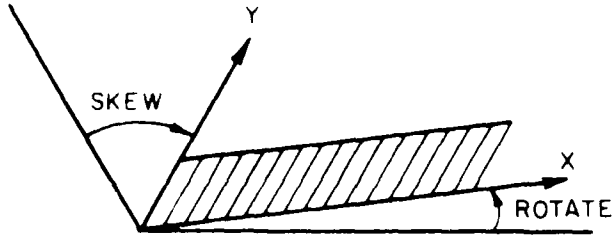


Figure 8 Determination of angles.

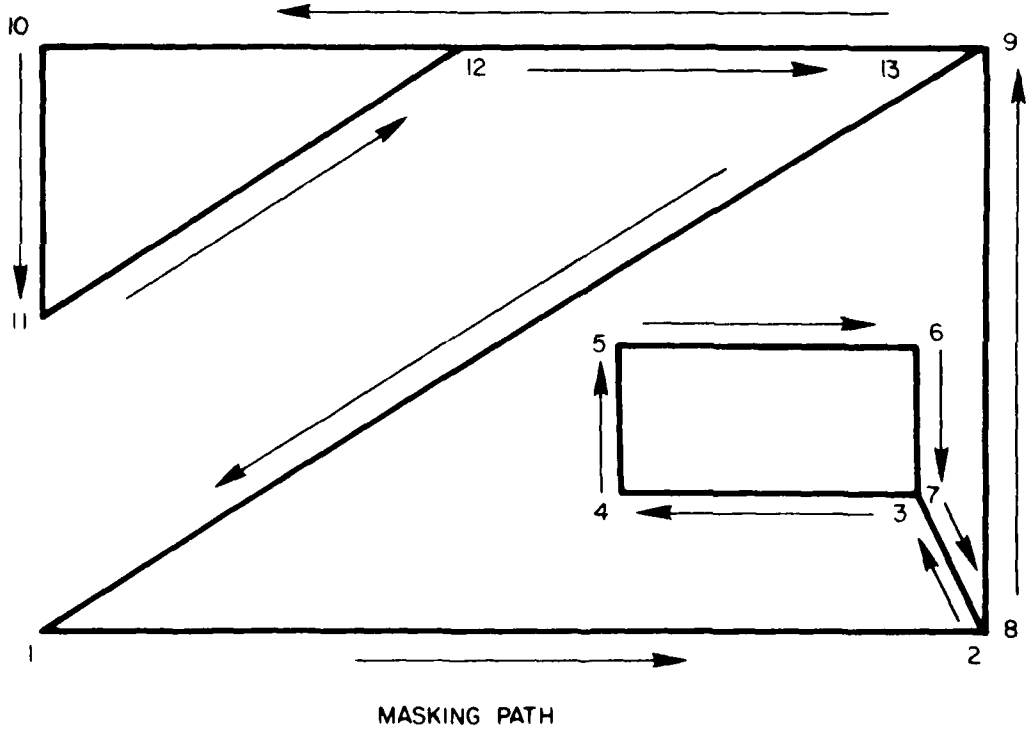


Figure 9 Masking path.

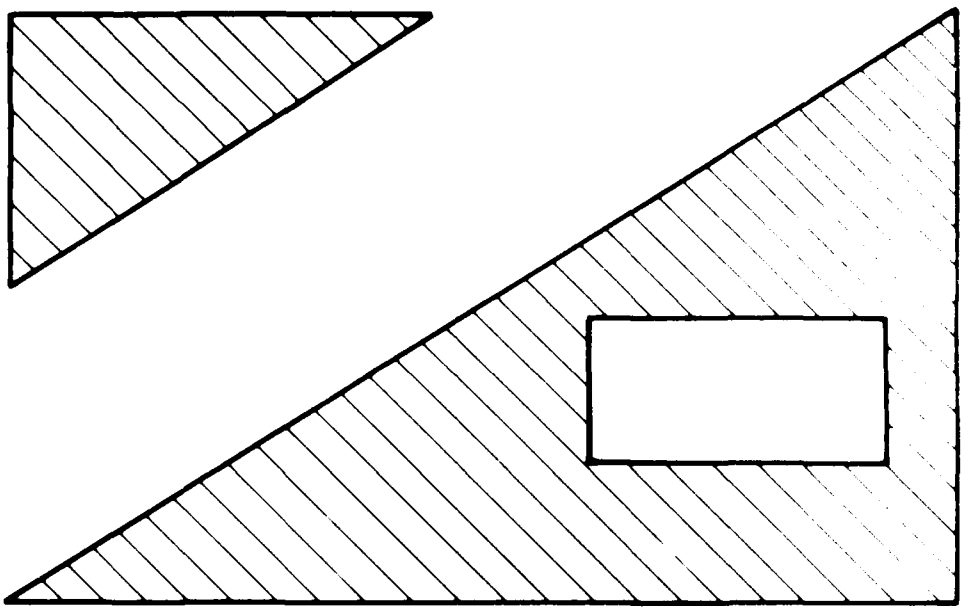


Figure 10 Contouring regions

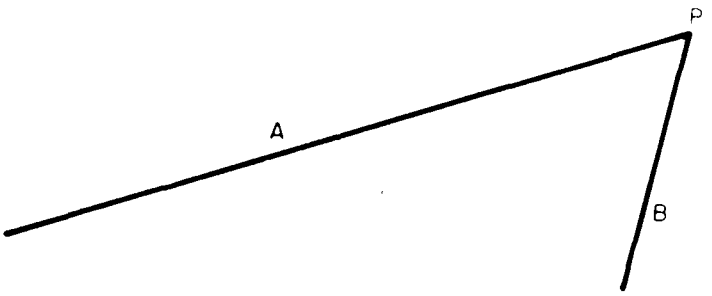


Figure 11 Pair of contour segments

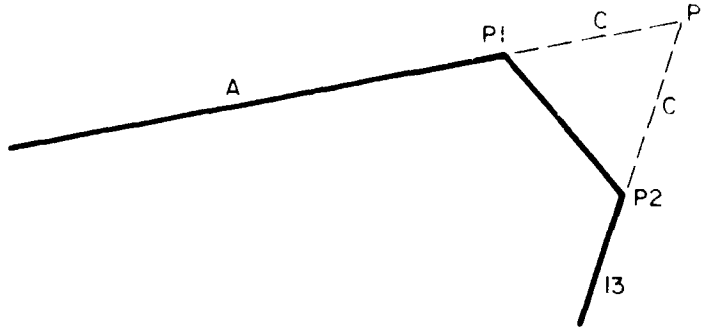


Figure 12 C as less than or equal to one-third of A.

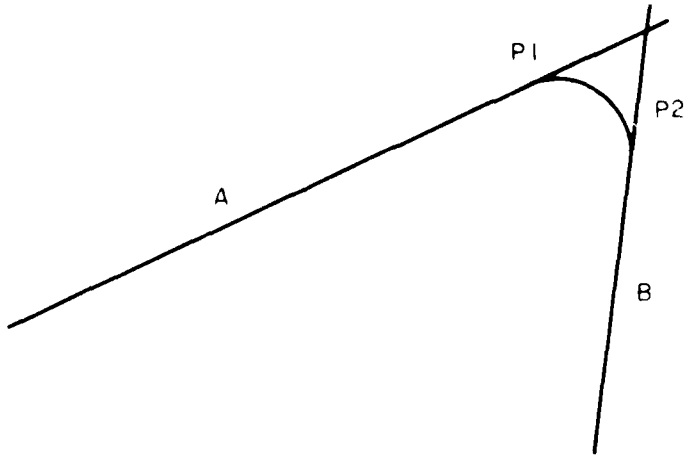


Figure 13 Circular arc smoothing.

4 EXAMPLE OF A BLAST NOISE PROGRAM RUN

This chapter presents a sample run of the Blast Noise Prediction program and includes [illegible] both input and output. This example does not include data collection, but does show what [illegible] should do with the operational data when he has received it. The example also describes what [illegible] the user wanted from the sample data and the output that the computer produced as a result [illegible] instructions.

Data Received

These data were obtained from the hypothetical installation "SHOW" for the month of [illegible] days). A map of the installation in metric coordinates (Figure 14) was received with the [illegible] types of activities occurred during this time, and five sites were involved in these activities. All [illegible] were fired from three firing points toward two target points.

The first activity used a 155-mm self-propelled howitzer (M109) to fire 360 rounds from firing point 2 (located at coordinates 38000, 200000) to target point 2 (located at coordinates 32000, [illegible]). One hundred and fifty of the rounds were fired during the day (0700 to 2200 hours), and 10 were fired at night (2200 to 0700 hours). Charge zones 4 and 5 were used for this weapon. The second activity used an 81-mm mortar which fired 325 rounds from firing point 1 (located at coordinates 27000, 230000) to target point 2. All 325 projectiles were nonexploding, i.e., smoke or illuminating. Three hundred of the rounds were fired during the day, and 25 were fired at night, using charge zones 5 to 7. The third activity involved demolitions at firing point 3, located at coordinates 32000, [illegible]. There were 100 daytime explosions, each of which set off 15 lb (6 kg) of ammunition. The last activity used a nonstandard weapon (not listed in Table 2). It fired 17 night rounds from firing point 1 to target point 1 (located at coordinates 27000, 250000), using charge zones 3 and 4. All rounds exploded 100 ft (30 m) above the target. The user learned that this weapon fires a projectile containing 10.5 lb (4.8 kg) of explosive, and has six charge zones containing 1.2, 3.4, 5.7, 7.3, 9.2, and 12.1 lb (0.54, 1.54, 3.31, 4.17, and 5.49 kg) of propellant in zones 1 through 6, respectively, and that it had a [illegible] pattern which is the same as the 105 mm self-propelled howitzer (M102).

The user wants to generate a set of CDNL contours for this set of data and have the [illegible] installation drawn on them for reference. In addition, he wants an indication of the actual [illegible] tudes and locations.

Creating the USER'S RUN

JCL Cards

The user obtains an account number for his run. He then types a set of JCL cards with [illegible] the rest of the cards in his deck.

Input Data Cards

The first section of the deck is composed of the input data cards the user has created [illegible] operational data he has received. For this example, there will be four sets of gun type cards, [illegible] point cards, and three sets of firing point cards.

Tables 3 to 5 provide information for the first three gun types, which correspond to the [illegible] weapons in the example. The last weapon is not on the list, since it is not a standard weapon. [illegible] fore, the user must create his own weapon code and gun type card for it from the information [illegible] received. The gun type cards are shown in Figure 15. A weapon code of 80 was chosen and [illegible] into columns 2 and 3 of that gun type card (Gun 1d). The rest of the information was [illegible] appropriate columns, as shown in Table 5. A "1" was placed in column 1 of card Gun 5, the [illegible] last gun type card. Following the gun type cards, target point cards (TRG 1a, 2a) were [illegible] Table 6. Short target point names were made up to fit into the allocated columns.

Two types of cards are associated with each firing point. Cards of type FP-1 (Table 7) give the name and location of the firing point. In the example, shortened firing point names were created to fit into the allocated columns. The location information which was [illegible] the shortened names were typed into the appropriate columns. Cards of type FP-2 (Table 8) are used to describe the activities occurring at each firing point. Each FP-1 card must have at least one but may have many FP-2 cards associated with it. Two activities which differ only in "no. of day rounds" or "no. of night rounds" but which are identical in all other descriptions may be specified on two separate cards or on one card in which case the "no. of rounds" will be the sum of the two activities. Since firing point 2 had two weapons firing from it, two FP-2 cards were required to describe activities there. The first (see Figure 18) described the special weapon (gun type "80"). It fires 15 night rounds (columns 28 to 30) and no day rounds (columns 21 through 24). The minimum charge zone (columns 29 and 30) and maximum charge zone (columns 31 and 32) are 3 and 4, respectively. The weapon was fired toward target point 1 (FP1) (columns 35 through 38). All rounds exploded 100 ft (30 m) above the ground (columns 39 through 42) at the target point.

Card FP-2b described the 81-mm mortar (gun type 22). The number of rounds, charge zones, and target point were typed into the appropriate columns. In addition, a 1 was typed in the "no. rounds at target flag" column, since the rounds were noiseless, i.e., smoke or illumination. The second firing point had only one activity, so only one FP-2 type card was required (FP-2c). This card described the activities of the 155-mm self-propelled howitzer (M109). The number of rounds it shoots, charge zones, and target point were typed into the appropriate columns.

The only activity at firing point 3 was demolition. A 15-lb (6-kg) explosive corresponds to charge zone 5 for the standard gun-type used which has been created for small demolition (gun type 100). Card FP-2d described the demolition. The number of rounds and the minimum and maximum charge zones were typed into the appropriate columns. The target point, which was left blank, and the "1" in column 36 signified an omnidirectional noise at the firing point. Cards FP-2b, FP-2c, and FP-2d all had a "1" in column 1, since they were the last cards associated with each firing point.

The input data for the rest of the deck consisted of the gun-type, target point, and firing point definition cards.

Module Cards

The second portion of the deck consisted of the module cards which tell the computer what it must do with the input data. In this example, the data were checked and several types of plots were generated. The first plot (Figure 16) had an outline of the installation with target and firing points marked, along with the CDNL contours for the area. The second plot (Figure 17) had the installation outline and a noise density plot (scattergram).

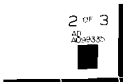
All grid coordinates are given in metric units, so the first card (NEED) must contain the specification "METERS". Next, a MAP-1 card invokes the MAP module, which checks the data and produces cross-reference tables. The MAP-2 card has a "1" in columns 1 through 5, requesting all of the cross-referenced tables. The number of days (300) is entered on the MAP-3 card. The MAP-4 card is set to "01", and MAP-5 is set to "280", since the cost is not of interest for this run. The BOUNDS module must be invoked before most of the other modules can be entered. This is accomplished by the BDS-1 card. The coordinates of the lower left-hand corner of the rectangular area under consideration are (-4000, 0), the upper right ones are (+9000, +9000). These values are entered on cards BDS-2 and BDS-3, respectively.

The BASE module is used to produce an outline of the installation. In this example, the user also wanted a registration mark (+) on the plot so that it could easily be overlaid on a map of the area. The BASE module is invoked with the BASE-1 card. The horizontal registration line is defined with the BASE-2a and BASE-2b cards. The star ("*") in column 21 of BASE-2b indicates the end of the segment. Cards BASE-2c and BASE-2d define the vertical registration line. BASE cards BASE-3 through BASE-3d define the installation outline. Since the last segment of the installation outline is connected to the first, only the last (BASE-3d) card needs a "1" in column 2 to indicate the end of the figure. The BASE-4 card indicates to the program that there is no more input for the BASE module.

AD-A099 335 CONSTRUCTION ENGINEERING RESEARCH LAB (ARMY) CHAMPAIGN IL F/G 20/1
BLAST NOISE PREDICTION. VOLUME II. BNOISE 3.2 COMPUTER PROGRAM --ETC(
MAR 81 L L LITTLE, V I PAWLOWSKA, D L EFFLAND

UNCLASSIFIED CERL-TR-N-98-VOL-2 NL

2 OF 3
AD
A099335



The FORMA and PUDDLE GRID modules must be included to produce a contour. FORMA must precede PUDDLE GRID, since PUDDLE GRID uses the output from FORMA. A FRMA-1 card is used to invoke the FORMA module. "MAX" is chosen as the parameter on the FRMA-2 card, since the user would like to investigate the worst case of his operations. The number of days (30) is typed on the FRMA-3 card. PUDDLE GRID may now be invoked with the PGRD-1 card. The closest city (both geographically and meteorologically) to the installation "SHOW" is Midland, TX. The inversion factors 65.8, 15.5, and 27.9 are found in Table 18. These values are entered into columns 1 through 10, 11 through 20, and 21 through 30 on card PGRD-2. The grid size chosen is 2000 m (columns 31 through 40, card PGRD-2). The SCATTER module, which is invoked with an SCT-1 card, generates a representation of the noise density by drawing dots in proportion to the number of operations. All day and night firing from both firing and target points will be considered. Thus, "T," "F," and "B" are typed in columns 1, 2, and 3, respectively, of card SCT-2. A multiplier of "2" is used to help represent the operations visually (columns 61 through 70, card SCT-2). The number of days (30) is typed on the SCT-3 card.

The LOCATOR module is invoked with LOC-1 card. It uses "0" and "X" to mark the locations of the firing and target points, respectively, on the plot. In the example, the user wanted both types of points marked, so "ALL" was entered on card LOC-2.

The first PLOT invoked with the PLT-1a card uses output from the PUDDLE GRID, LOCATOR, and BASE modules. This is specified by the "1" placed in columns 1, 2, and 4 of card PLT-2a. The user wanted the scale to be 1:50000, and contours to be drawn from 55 to 75 dB in 5-dB increments. Since those are the default values for these specifications on the PLT-3a card, the columns can be left blank. Columns 16 through 19 of card PLT-3a show that a 0.3 magnification will be used (this is done for reproduction purposes). No labeling is to be done, so cards PLT-4 and PLT-5 are omitted. The "*" in column 80 of card PLT-6a indicates that the first plot is completed.

The second plot, invoked with the PLT-1b card, uses output from the SCATTER and BASE modules, and is specified by the "1" typed in columns 3 and 4 of card PLT-2b. The user wanted the resultant drawing to be the same size as the first plot, so the specifications on the PLT-3b card were the same as those used on the PLT-3a card. Again, there were no labels to be printed, so the PLT-4 and PLT-5 cards were omitted. The PLT-6b card signified that the second plot was completed.

The STOPP module, which is invoked with the STP-1 card, indicates that the user has finished providing instructions for that run. The USER'S RUN consists of the JCL, input data, and module cards, respectively, separated by the "789" cards. A final card, which has a 6789 typed in the first column, signifies the end of information to the computer.

Output Received

The user received several pages of site-dependent information from the computer at the beginning and end of his output. These pages, which may include local Automated Data Processing (ADP) announcements and billing information, are not shown here, since this report is concerned only with the Blast Noise Prediction program. The user's output is shown as Figure 18.

The first page of output (page A) tells the user the units of his coordinates. Note that height/depth information is always in feet. This output was generated by the NEF-1 card.

The next module is MAP. Since the user in the example requested that the data be listed, they were printed out on pages B, C, and D. The gun-type cards were listed on page B. The warning (not necessarily an error) on page B was generated, since two of the charges were greater than 50 lb (20 kg). In this example, it is not an error, since the user did want 70 and 90 lb (28 and 36 kg) for charge zones 9 and 10 of gun type 10. Page C lists the target points; page D lists the firing-point information and gives the total number of errors and warnings found in the input data.

Page E summarizes the input data read, showing how many gun-type, target, and firing point cards were read. It provides statistics on the number of total firings and the number of firings per day for both day and night. Charge information about minimum and maximum charge zones and weight are also given. The average weight of firings is given in pounds, and the maximum height and depth are

given in feet. The maximum and minimum values for coordinates read are given, along with the coordinate pair in which each occurs. The final item on Page E is the number of points resulting from the grid size specified in MAP.

Pages F, G, H, and I are the cross-reference tables requested on the MAP-2 card. The first cross-reference table (targets by firing points) provides the average number of rounds fired by day for each target and firing point combination. The second table lists targets by gun type and provides the average total weight exploding at each target for each gun type. The third table provides the same information as the second table, but in a different order -- gun types by targets. The last table lists gun types by firing point, giving the average daily charge weight of each combination. Page I also lists the amount of time that the computer spent on the MAP module.

Page J gives the information for the BOUNDS module, including the maximum and minimum coordinates and the amount of time spent by the computer in this module.

The BASE module output on page K shows the start and end coordinates for the lines to be drawn. It informs the user when a new figure is started and how much time the computer spent in this module.

Page L lists the FORMA module output. First, it informs the user which charge averaging technique was used. It lists the number of days for the data base, as well as the number of unique noise sources, including the number of unique explosion types, i.e., unique combinations of height/depth, weight, location, and weapon types. All omnidirectional noise sources, i.e., demolitions and projectile explosions, are classified as the same type for the above statistic. the average number of rounds fired per day and night is given. In MAP, the number of explosions was given, which is always greater than or equal to the number of rounds. The amount of time spent in FORMA is the last piece of information provided.

Page M -- the first page of the PUDDLE GRID output -- tells the user what the start and stop coordinates used by this module are. In this example, the limits had to be changed, since they were not an integral multiple of the grid size chosen. The new units are listed, along with a warning specifying that they were changed. The weather factors are listed, and PUDDLE GRID specifies that both day and night values are used in calculations. The grid size is also provided. A table of values is listed by coordinates in increments of grid size. This table is pages N and O. The amount of time spent in PUDDLE GRID is the last item printed.

Page P lists the SCATTER module's output, followed by the condition under which the scattergram is produced, i.e., day/night/both target/firing point/gun type. The limits of the scattergram are given next. The multiplier, standard deviation, and number of days are listed next, followed by a summary of what will be plotted, which tells how many dots will be plotted for both the firing points and target points. The amount of time spent in SCATTER is the last item printed.

The LOCATOR module output provides the options requested first (page Q), i.e., target/firing points, name, location. Next, the size and angle of the characters to be plotted are given, followed by a listing of the targets and firing points and the amount of time spent in this module.

Pages R and S give the PLOT output for plots 1 and 2, respectively. The first plot (page R) lists the file name used by the system and the date and time that the file was produced. This information is also drawn on the plot itself. For this plot, contours were drawn with a base outline, and the firing point and target points were overlaid. A warning is given about the bounds since PUDDLE GRID modified them. Values used by PLOT for its various parameters are listed. The values having a "*" are equal to default values. If any text cards had been included, they would have been listed next (see Table 13, PLT-4 and PLT-5).

Finally, information concerning the physical plot is given, including the size, how many units (as specified by the NEF-1 card) 1 in. (25.4 mm) equals, and how many sections of plotter paper will be needed. The last piece of information is the amount of time spent in that module.

The second plot uses information from the BASE and SCATTER modules, and provides the same type of information given in the first plot.

The final page (page T) gives the STOPP module's output, which states that the internally stored information is ready to be plotted.

Figures 16 and 17 are the plots produced by this run.

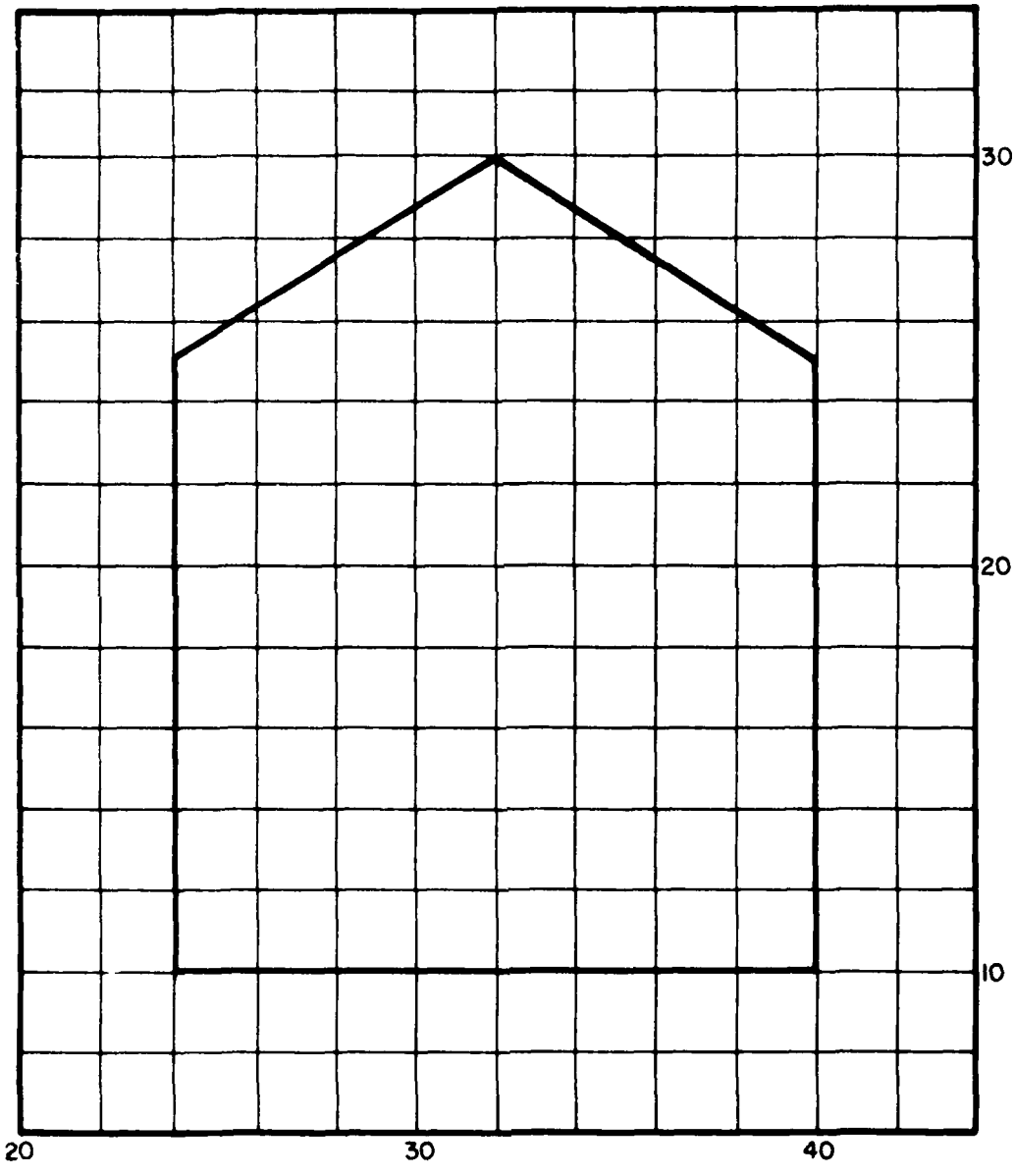


Figure 14. Military installation SHOW.

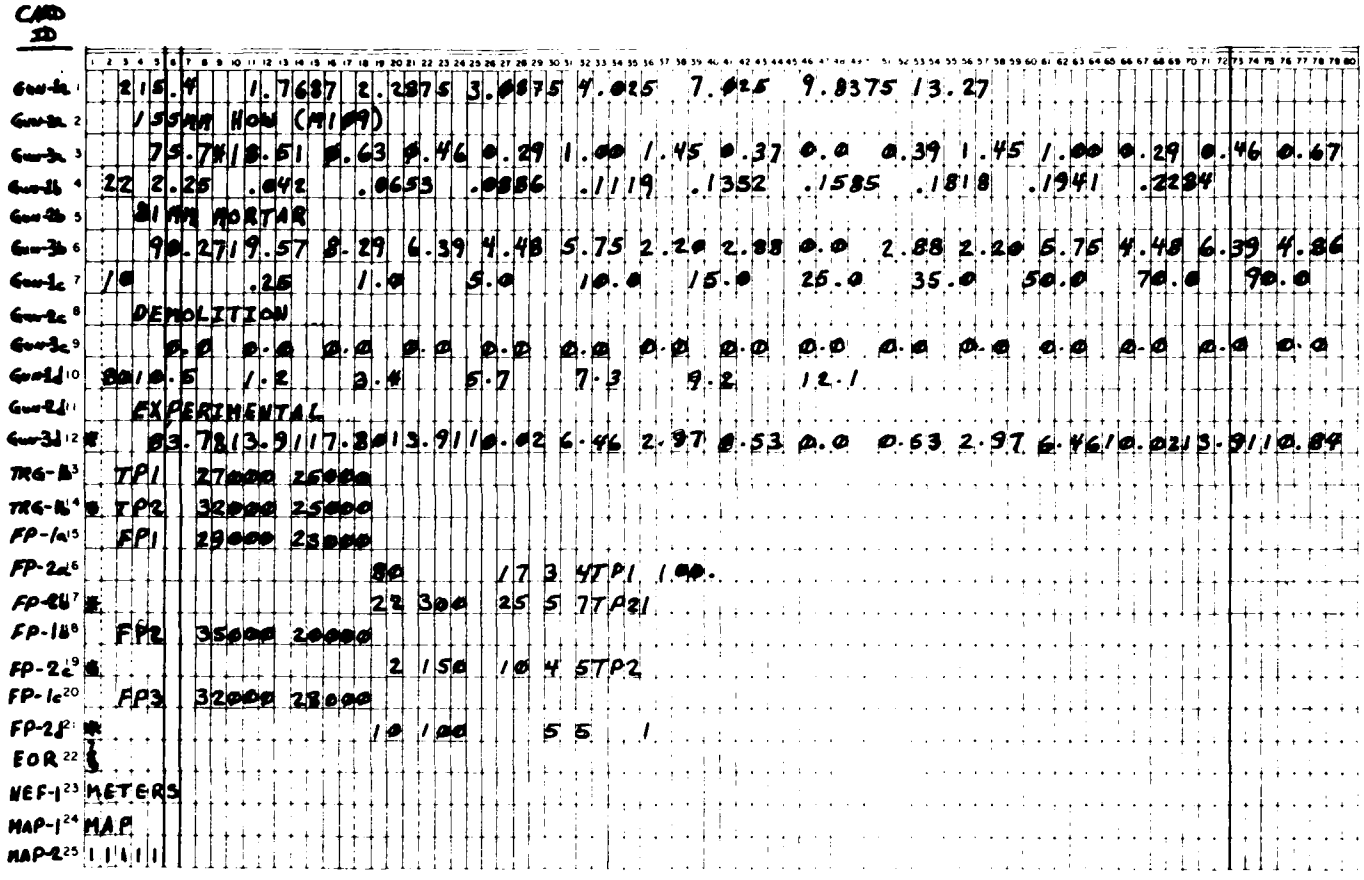


Figure 15 Input data

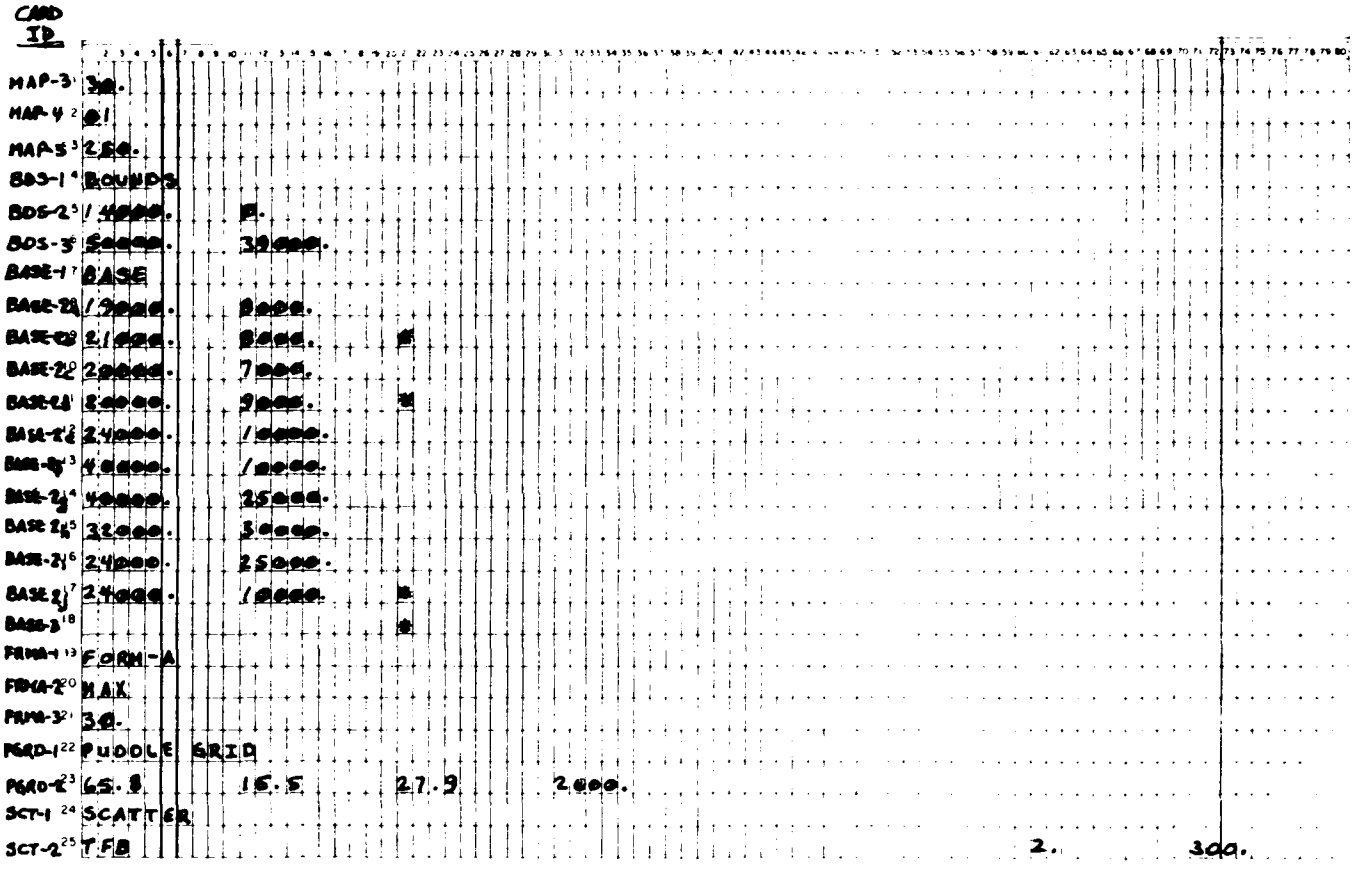


Figure 15. (Cont'd).

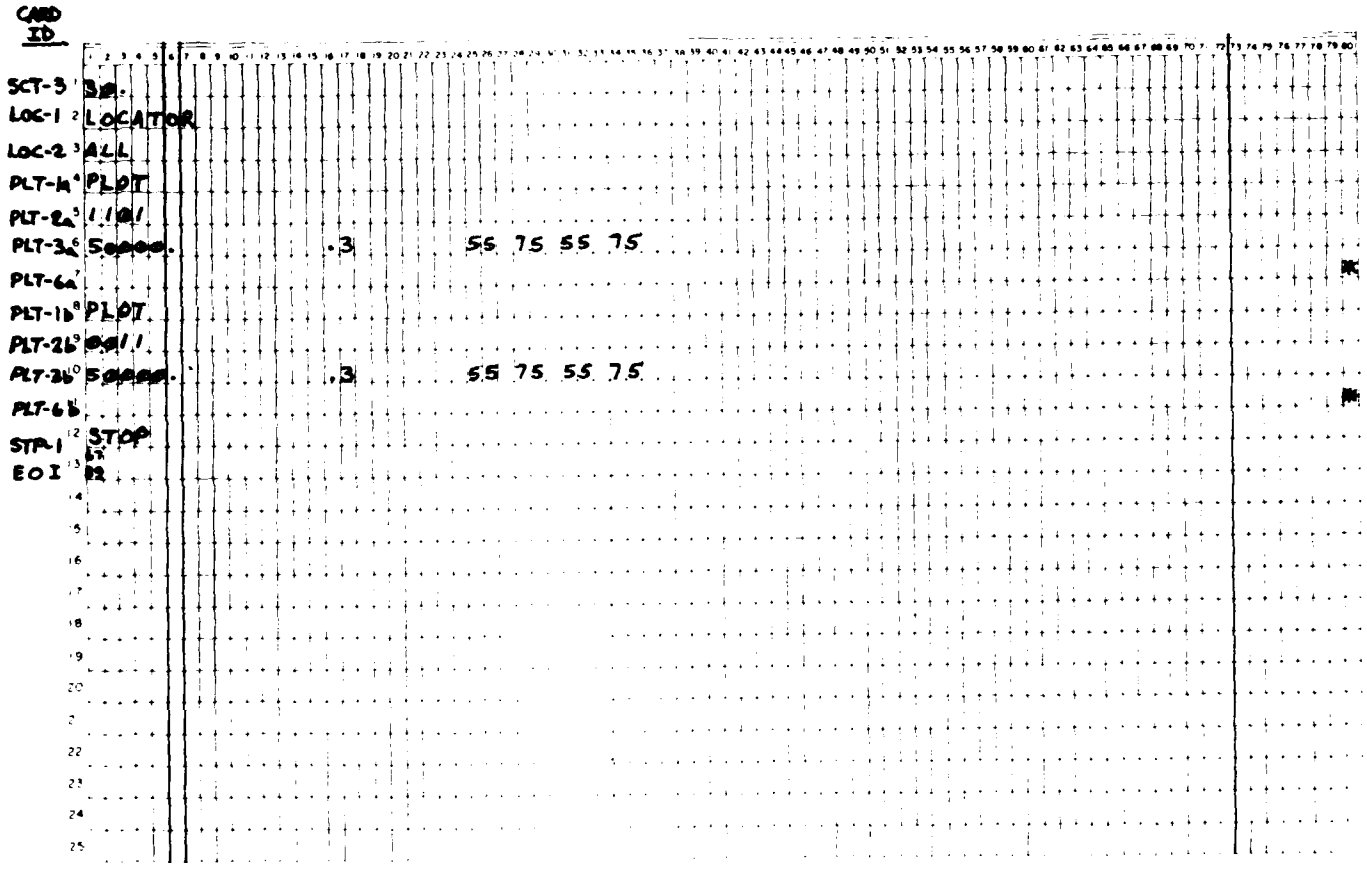


Figure 15. (Cont'd)

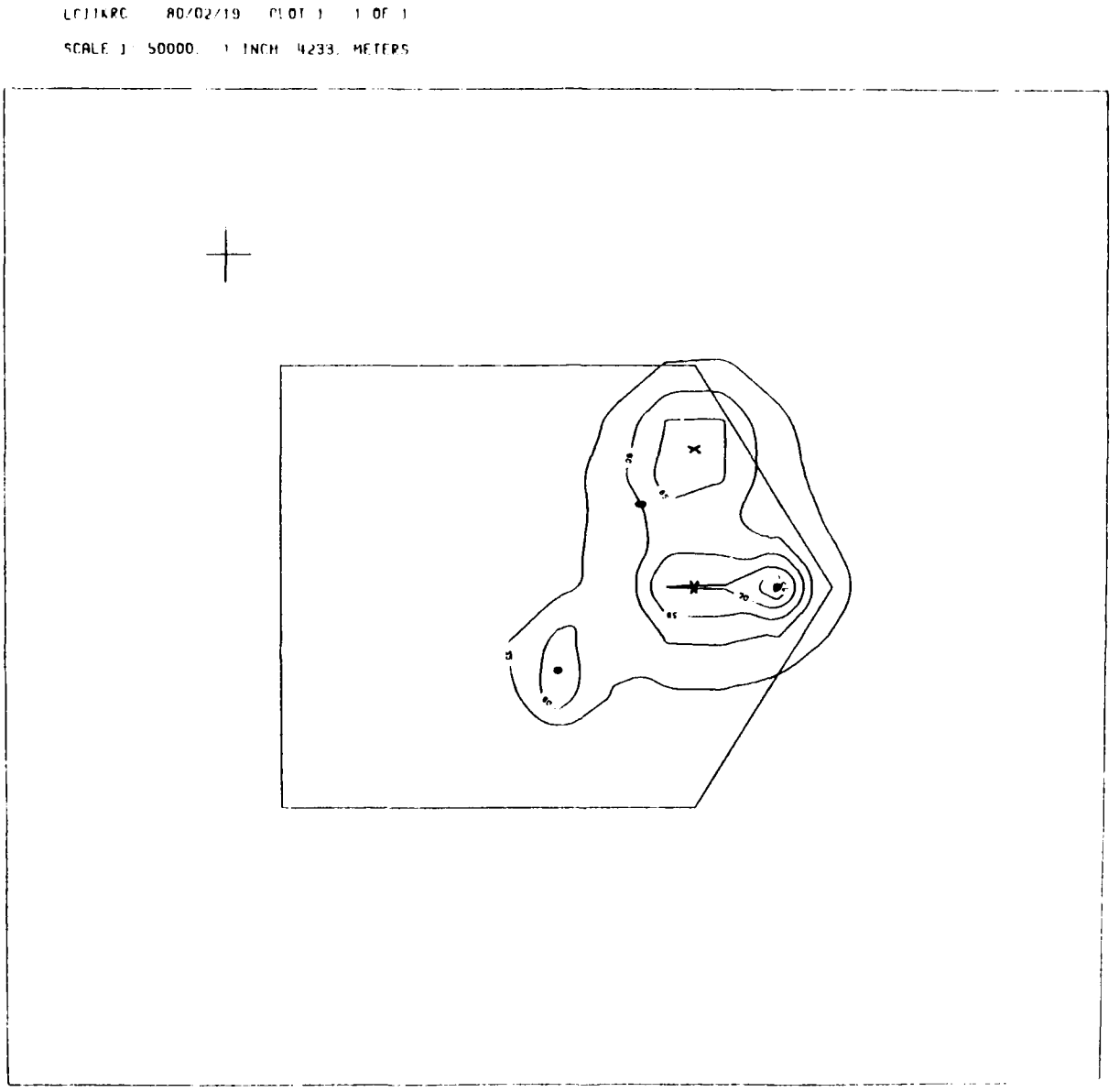


Figure 16. CLDN contours for installation SHOW.

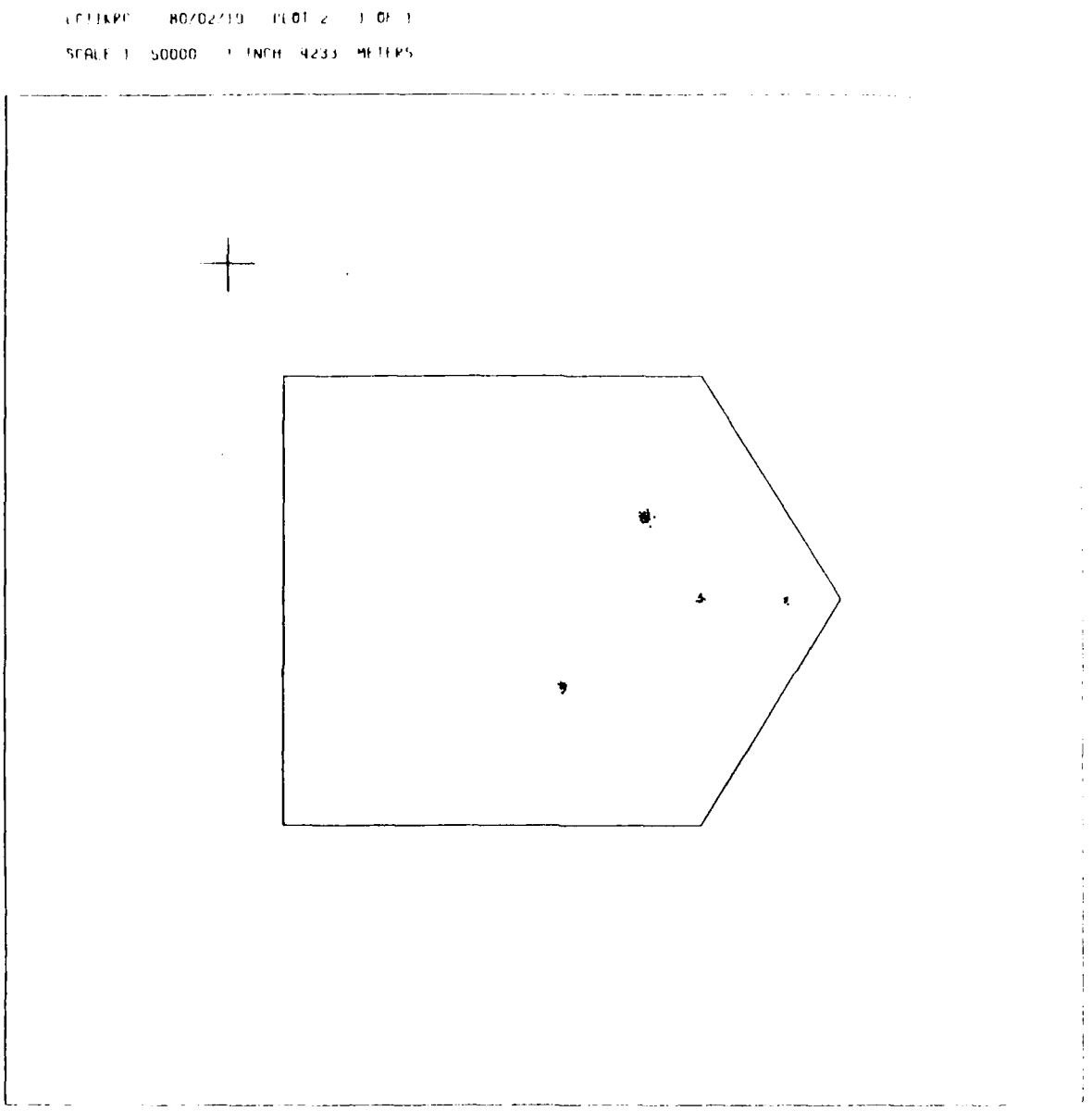


Figure 17. Scattergram for installation SHOW.

```
.....     BNOISE3.2          .....


                              DISTANCES EXPRESSED IN METERS
```

Page A

```
.....     MAP OF SOURCE POINTS     .....

.....     GUN TYPE CARDS     .....

FLAG  G TYPE  T CHARGE                                         PROPELLANT WEIGHTS

         2       15.40       1.77    2.29    3.09    4.03    7.03    9.84   13.27    0.00    0.00    0.00
        22        2.25        .04     .07     .09     .11     .14     .16     .18     .19     .23    0.00
        10        0.00        .25    1.00    5.00   10.00   15.00   25.00   35.00   50.00   70.00   90.00
.....  WARNING --  2 CHARGES LARGER THAN 50. LBS
     *  80       10.50       1.20    3.40    5.70    7.30    9.20   12.10    0.00    0.00    0.00    0.00


GTYPE     NAME              EQ PARA  EQ PARB                     DIRECTIVITY VALUES                              AVG

  2    155MM HOW (M109)      75.74    18.51     .63   .46   .29  1.00  1.45   .37  0.00   .39  1.45  1.00   .29   .46    .67
 22    81MM MORTAR           90.27    19.57    8.29  6.39  4.48  5.75  2.20  2.88  0.00  2.88  2.20  5.75  4.48  6.39   4.86
 10    DEMOLITION             0.00     0.00    0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00   0.00
 80    EXPERIMENTAL          83.78    13.91   17.80 13.91 10.02  6.46  2.97   .53  0.00   .53  2.97  6.46 10.02 13.91  10.84

.....  2 ERROR/WARNING CONDITIONS DETECTED FOR THIS CARD TYPE
```

Page B

Figure 18. Printed output from the Blast Noise Prediction computer program.

..... TARGET CARDS .....

FLAG	ID	X	Y	HT CORR
	TP1	27000.	25000.	0.
*	TP2	32000.	25000.	0.

..... 0 ERROR/WARNING CONDITIONS DETECTED FOR THIS CARD TYPE

Page C

..... FIRING PT. SOURCE AND DEFINITION CARDS .....

FLAG	ID	X	Y	HT CORR	G TYPE	DAYNO	NIGHTNO	MIN	MAX	T ID	FLAG	HGT
	FP1	29000.	23000.	0.								
					80	0.	17.	3	4	TP1	0	100.0
*					22	300.	25.	5	7	TP2	1	0.0
	FP2	35000.	20000.	0.								
*					2	150.	10.	4	5	TP2	0	0.0
	FP3	32000.	28000.	0.								
*					10	100.	0.	5	5		1	0.0

..... 0 ERROR/WARNING CONDITIONS DETECTED FOR THIS CARD TYPE

..... END OF INPUT PHASE: 2 ERROR/WARNING CONDITIONS DETECTED

Page D

Figure 18. (Cont'd)

```
NUMBER OF DATA BASE CARD IMAGES READ IS     21

NUMBER OF GUN TYPES READ IS      4

NUMBER OF TARGETS READ IS      2

NUMBER OF SOURCES READ IS      3


.....      DATA BASE TIME PERIOD:  30. DAY(S)

TOTAL DAY FIRINGS IS        550.00
TOTAL NIGHT FIRINGS IS       52.00

TOTAL PER DAY DAY FIRINGS IS         18.33
TOTAL PER DAY NIGHT FIRINGS IS        1.73

MINIMUM CHARGE NUMBER     3
MAXIMUM CHARGE NUMBER     7

MINIMUM CHARGE WEIGHT         .1 LBS
MAXIMUM CHARGE WEIGHT       15.0 LBS

TOTAL DAY CHARGE WEIGHT PER DAY        215.3 LBS
TOTAL NIGHT CHARGE WEIGHT PER DAY       17.7 LBS

MAXIMUM HEIGHT IS      100.00
MAXIMUM DEPTH IS         0.00


MAXIMUM X IS     35000.0 IN PAIR  (    35000.0 ,     20000.0 )

MAXIMUM Y IS     28000.0 IN PAIR  (    32000.0 ,     28000.0 )

MINIMUM X IS     27000.0 IN PAIR  (    27000.0 ,     25000.0 )

MINIMUM Y IS     20000.0 IN PAIR  (    35000.0 ,     20000.0 )

 FOR GRID SIZE    250.0, GRID DIMENSIONS =    32.0 X     32.0
```


Figure 18. (Cont'd).

CROSS-REFERENCE: TARGETS BY FIRING POINTS ; DAILY FIRINGS

TARGET ID	FPT ID		FPT ID		FPT ID	FPT ID	FPT ID	FPT ID	FPT ID
TP1	FP1	.6							
TP2	FP1	10.8	FP2	5.3					

Page F

CROSS-REFERENCE: TARGETS BY GUN TYPES; DAILY PROJECTILE CHARGE WEIGHT (LBS)

TARGET ID	GUN ID		GUN ID	GUN ID	GUN ID	GUN ID	GUN ID	GUN ID
TP1	80	6.0						
TP2	2	82.1						

Page G

CROSS-REFERENCE: GUN TYPES BY TARGETS; DAILY PROJECTILE CHARGE WEIGHT (LBS)

GUN ID	TAR ID		TAR ID	TAR ID	TAR ID	TAR ID	TAR ID	TAR ID
2	TP2	82.1						
22	***							
10	***							
80	TP1	6.0						

Page H

Figure 18. (Continued)

CROSS-REFERENCE: GUN TYPES BY FIRING POINTS; DAILY PROPELLENT CHARGE WEIGHT (LBS)

GUN ID	FPT ID		FPT ID	FPT ID	FPT ID	FPT ID	FPT ID	FPT ID
2	FP2	37.5						
22	FP1	2.0						
10	FP3	50.0						
80	FP1	4.1						

..... TIME FOR MAPPING SUBPROGRAM IS .079 SECONDS

Page I

..... BOUNDS .....

MINIMUM BOUNDARY = 14000. 0. MAXIMUM BOUNDARY = 50000. 39000.

BOUNDARY VALUES VERIFIED

..... TIME IN BOUNDS IS .002 .....

Page J

Figure 18. (Cont'd)

..... BASE .....

LINE CARDS

	X-Y START		X-Y END	
LINE	19000.	8000.	21000.	8000.
NEW LINE				
LINE	20000.	7000.	20000.	9000.
NEW LINE				
LINE	24000.	10000.	40000.	10000.
LINE	40000.	10000.	40000.	25000.
LINE	40000.	25000.	32000.	30000.
LINE	32000.	30000.	24000.	25000.
LINE	24000.	25000.	24000.	10000.

..... TIME IN BASE IS .018 .....

Page K

..... FORM-A CALCULATION .....

..... FOR TNT EQUIVALENT, FORM-A WILL USE MAXIMUM CHARGE ZONE

..... DATA BASE TIME PERIOD: 30. DAY(S)

NUMBER OF UNIQUE NOISE SOURCES COUNTED IS 6

TOTAL PER DAY DAY EXPLOSIONS IS 23.33
TOTAL PER DAY NIGHT EXPLOSIONS IS 2.63

..... TIME FOR FORM-A SUBPROGRAM IS .237 SECONDS

Page L

Figure 18. (Cont'd)

```
     .....     PUDDLE GRID     .....

*****  WARNING -- SPECIFIED BOUNDS (  14000.0,      0.0 ):(  50000.0,  39000.0 ) DO NOT CORRESPOND TO INTEGRAL GRID BOUNDS.
          MODIFIED BOUNDS WILL BE USED TO PRODUCE THE GRID AND TO DEFINE ANY PLOT UTILIZING THIS GRID

110

          START AT DATA BASE COORDINATES (  14000.0,      0.0 )

          STOP  AT DATA BASE COORDINATES (  50000.0,  40000.0 )

               INVERSION =   65.80   15.50   27.90

          CALCULATIONS FOR NEF WILL USE BOTH D + N OF DAY AND NIGHT CALCULATIONS


           GRID SIZE =  2000.0 ; DISTANCES IN METERS
```


Figure 18. (Cont'd).

	14000	16000	18000	20000	22000	24000	26000	28000	30000	32000	34000	36000	38000	40000	42000
40000	41.2	42.1	42.9	43.3	43.9	44.6	45.0	45.4	45.4	45.4	45.2	45.0	44.4	43.7	42.9
38000	42.0	43.0	43.8	44.7	45.1	45.7	46.1	46.6	46.6	46.7	46.5	46.2	45.5	44.8	44.1
36000	42.8	43.7	44.7	45.6	46.0	46.8	47.4	47.9	48.1	48.3	47.9	47.4	46.6	45.9	44.9
34000	43.4	44.5	45.5	46.4	47.3	47.8	48.7	49.5	49.8	49.9	49.6	48.9	47.9	46.8	45.8
32000	43.8	45.0	46.2	47.2	48.3	49.1	50.3	51.2	51.8	51.9	51.5	50.3	49.1	47.8	46.5
30000	43.8	45.1	46.7	47.9	49.3	51.1	52.3	53.4	54.7	57.6	54.2	51.9	50.3	48.7	47.1
28000	44.0	45.4	46.6	48.2	50.6	53.2	55.9	56.6	58.8	92.5	58.2	53.3	51.0	49.3	47.5
26000	44.1	45.6	46.8	48.5	51.1	55.5	65.4	65.5	61.3	70.3	60.4	54.2	51.7	49.6	47.9
24000	44.1	45.6	46.8	48.5	51.1	55.3	65.4	67.5	61.8	70.1	60.2	54.1	51.7	49.6	47.9
22000	43.7	45.1	46.3	47.9	50.0	52.7	55.7	57.6	56.8	56.7	56.3	54.7	51.5	49.5	47.8
20000	43.4	44.7	45.9	47.2	48.6	50.3	51.9	52.6	53.1	53.6	61.8	62.8	51.5	49.1	47.3
18000	42.9	44.1	45.3	46.4	47.3	48.4	49.6	50.2	50.9	51.4	54.1	52.8	49.6	47.9	46.5
16000	42.4	43.5	44.4	45.3	46.3	47.2	47.9	48.6	49.0	49.5	49.4	49.0	47.9	46.8	45.7
14000	41.7	42.6	43.6	44.5	45.3	46.0	46.6	47.1	47.5	47.7	47.5	47.2	46.4	45.6	44.7
12000	40.9	41.9	42.7	43.5	44.4	44.9	45.3	45.8	46.1	46.3	46.0	45.7	45.1	44.5	43.6
10000	40.1	40.9	41.8	42.5	43.1	43.8	44.2	44.6	44.8	44.8	44.6	44.4	43.8	43.2	42.3
8000	39.6	40.0	40.8	41.4	42.0	42.6	42.9	43.3	43.5	43.4	43.3	43.0	42.4	42.0	41.3
6000	39.1	39.5	39.9	40.3	41.0	41.5	41.7	41.9	42.2	42.0	41.9	41.6	41.2	40.8	40.2
4000	38.7	38.9	39.2	39.5	39.9	40.2	40.5	40.7	40.9	40.8	40.6	40.4	40.1	39.7	39.2
2000	38.3	38.5	38.7	38.9	39.1	39.2	39.4	39.6	39.7	39.6	39.5	39.3	39.1	38.9	38.7
0	38.0	38.1	38.3	38.6	38.6	38.8	38.9	39.0	38.9	38.8	38.9	38.8	38.6	38.5	38.3

Figure 18. (Cont'd)

	44000	46000	48000	50000
40000	42.1	41.1	40.2	39.3
38000	43.1	42.0	41.0	39.9
36000	44.0	42.9	41.7	40.6
34000	44.7	43.7	42.4	41.2
32000	45.4	44.3	42.9	41.5
30000	45.8	44.7	43.5	42.0
28000	46.3	44.9	43.8	42.2
26000	46.4	45.1	43.9	42.3
24000	46.5	45.1	43.9	42.3
22000	46.4	45.0	43.7	42.2
20000	46.0	44.7	43.4	42.0
18000	45.5	44.2	42.8	41.4
16000	44.5	43.3	42.1	40.9
14000	43.7	42.5	41.3	40.3
12000	42.6	41.6	40.7	39.6
10000	41.5	40.7	39.9	39.1
8000	40.4	39.8	39.2	38.7
6000	39.6	39.1	38.7	38.4
4000	38.9	38.7	38.4	38.1
2000	38.5	38.3	38.1	37.8
0	38.2	37.9	37.7	37.6

..... TIME FOR PUDDLING SUBPROGRAM IS .776 SECONDS

Page 0

Figure 18. (Cont'd)

```
.....     SCATTER     .....

THE SCATTER DIAGRAM WILL REPRESENT ..DAY+NIGHT.. DATA FOR:

        ...TARGETS
        ...FIRING PTS
        ...ALL GUN TYPES

BOUNDED BY (  14000.0,      0.0) - (  50000.0,  39000.0)

    ...MULTIPLIER =       2.0

    ...STANDARD DEVIATION =   300.000

    ... 30. DAYS DATA IN DATA BASE

OUTPUT FILE TAPE4 CONTAINS NASAPLOT PHS4 INPUT DATA REPRESENTING:

       2 TARGETS   :    12 SCATTER PTS.
       3 FIRING PTS:    41 SCATTER PTS.

   MULTIPLIER USED =       2.0

          .....     TIME IN SCATTER IS    .057 SECONDS
```


Figure 18. (Cont'd)

```
.....     LOCATOR      .....

   ..OPTIONS REQUESTED..

    TARGET
    FIRING POINT

     SIZE=  .14**     ANGLE=  0.00**
     STARS INDICATE DEFAULT VALUES

     ID CODE         X COORD.     Y COORD.     G CORR.     BOUNDS
..TARGET DATA..

     TP1             27000.       25000.          0.
     TP2             32000.       25000.          0.
..FIRING POINT DATA..

     FP1             29000.       23000.          0.
     FP2             35000.       20000.          0.
     FP3             32000.       28000.          0.

.....    .018 SECONDS IN LOCATOR  .....
```


Figure 18. (Cont'd).

```
                    .......      PLOT       .......

                           LOSITIH
21.26.55.   81/04/01.      PLOT  1

              FOLLOWING FILES WERE REQUESTED

                 PGRID
                 BASE
                 LOCATOR

WARNING...PUDDLE GRID BOUNDS DO NOT MATCH SPECIFIED BOUNDS...

                 PGRID VALUES USED          SPECIFIED BOUNDS
      XMIN         14000.00                   14000.00
      YMIN             0.00                       0.00
      XMAX         50000.00                   50000.00
      YMAX         40000.00                   39000.00

                  VALUES USED BY PLOT

        **SCALE     =50000.0

        **PERCENT X=1.00          **PERCENT Y=1.00

          MAG       = .30         **PERC SMTH= .33

        **START     = 55          **STOP      = 75

        **L START   = 55          **L STOP    = 75

        **LABEL     = 1           **INCREMENT= 5

        **L INCREMT= 5              GRID SIZE= 2000.

         STARS INDICATE DEFAULT VALUES

         FOLLOWING CARDS WERE USER TEXT INPUT

         NO USER TEXT CARDS INPUT

         THIS PLOT IS   9 INCHES BY  10 INCHES

         ONE INCH IS EQUAL TO  4233. METERS

         IT CONSISTS OF  2 PAGES IN THE X DIRECTION AND  1 SECTIONS IN THE Y DIRECTION

           ..... TIME IN PLOT IS    .219 .....
```

Page R

Figure 18. (Cont'd).

```
                 .......      PLOT      .......

                LDSITIH
21.26.56.   81/04/01.   PLOT  2

        FOLLOWING FILES WERE REQUESTED

            BASE
            SCATTER

             VALUES USED BY PLOT

     **SCALE     =50000.0

     **PERCENT X=1.00          **PERCENT Y=1.00

       MAG       = .30         **PERC SMTH= .33

     **START     = 55          **STOP      = 75

     **L START   = 55          **L STOP    = 75

     **LABEL     = 1           **INCREMENT= 5

     **L INCREMT= 5              GRID SIZE=     0.

      STARS INDICATE DEFAULT VALUES

      FOLLOWING CARDS WERE USER TEXT INPUT

      NO USER TEXT CARDS INPUT

      THIS PLOT IS   9 INCHES BY   9 INCHES

      ONE INCH IS EQUAL TO  4233. METERS

      IT CONSISTS OF  2 PAGES IN THE X DIRECTION AND  1 SECTIONS IN THE Y DIRECTION

        ..... TIME IN PLOT IS    .043 .....
```

Page S

```
                  .....      STOP      .....


              NASAPLOT INPUT FILE COMPLETED
```

Page T

Figure 18. (Cont'd).

5 CALCULATION OF CDNL

General

Noise travels some distance from its point of origin (source) before dissipating. Therefore, to accurately evaluate the noise produced at a point, it is necessary to determine the level of sound produced in the entire area surrounding its origin. The size of the area **around** a noise-producing source which must be evaluated depends on the initial energy of the noise, i.e., a low-energy noise will affect a smaller area than a high-energy noise.

The Blast Noise Prediction program can evaluate the noise level of sound produced in an area around a noise-source and uses the CDNL measure to quantify these noise level units in decibels. The capability to quantify noise produced at a source is of particular value to Army planners who must evaluate and predict community response to the noise-producing activities of Army installations. To create CDNL values, the Blast Noise Prediction program uses a series of calculations which rely on a specific set of formulas and tables. These formulas incorporate data which must be provided by the user; therefore, raw noise-source data must be converted into a form suitable for submission to the computer. Once the raw data have been properly compiled, they can be turned over to the computer for processing.

The formulas require several user-supplied specifications:

1. The number and times of rounds (blasts) occurring during the day and night. A penalty of 10 dB is applied to night operations (2200 and 0700 hours).

2. The locations of both the firing point and the target point.

3. Weather information for the region being surveyed.

In addition, the user must specify the boundary coordinates for his area of interest and tell the Blast Noise Prediction program into what size blocks he wants the area to be broken up into for computational purposes. The program uses this information to divide the user's area into a grid of points, i.e., a matrix of X and Y coordinates. Each of these points is a given distance away from the noise source being evaluated since the calculations for the noise level incorporate distance. The Blast Noise Prediction program then calculates the noise level of each source at each given point. This calculation is repeated for each point. The values are stored and printed out by coordinates, and are used (at user discretion) to produce a reference table of the noise level at each given point. The user can also request the program to generate a paper plot showing the contours of the noise levels in the area around the source(s) being evaluated. The program creates these contours by joining points having the same noise-level values. (The user, however, must first tell the program which levels he wants plotted, e.g., 55 to 75 dB every 5 dB. The user can also request an outline of the installation or other relevant region. This additional information on the plot can serve as a visual reference for the noise levels.) Finally, these contours are used to predict the noise impact of Army artillery, armor, and demolition activities.[7]

The information tables produced by the TABGEN portion of the Blast Noise Prediction program provide a listing of decibel values for distances away from a blast produced by a 5-lb charge. (See Table 20.)

The background on the information which Table 20 provides rests on theory for sound propagation in the atmosphere.[8] The speed of sound is a function of both wind and temperature; as these conditions change with altitude, sound waves are refracted or focused.

Figures 19 through 22 illustrate four simple cases of this phenomenon: (1) a negative sound velocity gradient, (2) a positive sound velocity gradient, (3) a positive sound velocity gradient which changes to a more sharply positive velocity gradient, and (4) a negative gradient followed by a positive gradient at a higher altitude.

[7] P. D. Schomer, et al., *Predicting Community Response to Blast Noise*, TR E-17/ADA773690 (CERL, December 1973).
[8] P. D. Schomer, et al., *The Statistics of Amplitude and Spectrum of Blasts Propagated in the Atmosphere*, Volumes I and II, TR N-13/ADA033473 and ADA033361 (CERL, November 1976).

In Case 1, sound is refracted upward, producing noise levels on the ground lower than those produced under uniform velocity or zero gradient conditions. For Case 2, sound rays are refracted downward, and the sound intensity on the ground is somewhat greater than that under uniform velocity gradient conditions. With combinations of these gradients, sound rays can travel over different paths and still arrive at an observation point simultaneously to produce a focus. In Case 3, separate groups of sound rays are created by two positive gradients -- the upper gradient is stronger than the lower. A weak focus, labeled F, is created at the points where both groups meet at the surface. In Case 4, sound is refracted upward in the lower negative gradient and downward in the upper positive gradient. The result is an increase of noise levels at the sharp focus in the region labeled F, and a reduction of noise levels in the silent zone between F and the blast site.

The probability of obtaining given amplitudes at various distances is a key statistic required for noise-impact prediction. Amplitude distributions have been created based on CERL blast data. Figure 23 shows a sample of such a distribution. Each such distribution is subdivided into four regions. The energy averages of the measured blasts within each region are calculated and plotted as a function of distance to produce the amplitude vs distance curves (Figures 24 and 25). The percentage of blasts lying in each range can be determined for each distribution and then related to distance, and an explanation developed to relate the statistics to environmental (e.g., atmospheric) conditions.

These regions, from highest to lowest, are called FOCUS, BASE, NEGATIVE, and EXCESS NEGATIVE. In addition, the lines separating the regions are called MEAN to distinguish the arithmetic mean from the line at the top of the region, which is labeled MAX, and the line at the bottom of the region, labeled MIN. It can be seen that FOCUS MIN and BASE MAX are the same line, i.e., the line between FOCUS MEAN and BASE MEAN. Thus, there are nine separate lines to be concerned with.

Statistics obtained from blast data are based on a 5-lb charge fired during the day with atmospheric conditions considered to be standard inversion factors.* Deviations from these basic conditions are accounted for by corrections added during the calculations.

Units

The amplitude is given in decibels and defined as

$$10 \log_{10}\left(\frac{P}{P_o}\right)^2 \qquad \text{[Eq 1]}$$

where

P_o = reference pressure of 20μPa

P = pressure in Pascals.

The distance between a noise source and the point at which its noise level is to be calculated using the X and Y coordinates read from a map is determined as follows:

$$distance = \left[(x_1 - x_2)^2 + (y_1 - y_2)^2\right]^{1/2} \qquad \text{[Eq 2]}$$

where

(x_1, y_1) = coordinates of the point

(x_2, y_2) = coordinates of the source.

If the map coordinates are in feet, they can be transformed to metric units by the conversion: 1 ft = 0.3048 m, 1 mile = 1609 m.

* Standard inversion factors are selected from the weather conditions at Fort Leonard Wood, MO.

1. Night correction -- if firings are at night, add a 10 dB penalty to compensate for increased human sensitivity to noise occurring at night.

2. Charge size correction -- The difference between a 5-lb open air explosion and the blast to be predicted must be found. If the blast is not created by a weapon, but is the result of an omnidirectional explosion, then Figure 26 is used. If it is a weapon, then the following equation is used:

$$Charge\ correction = A + B * \log_{10}(W * 16) - AVG - 119 \qquad \text{[Eq 3]}$$

where:

A, B, and AVG are found in Table 4.
W is the weight of the explosive in pounds.

3. Height correction -- the height or ground correction is used to compensate for the attenuating effects of the earth on sound wave propagation for explosions occurring above or below the ground.

a. If height > 0 then ground correction $= 0$ dB.

b. If $height < 0$, $depth = \frac{(height)}{(charge\ size)^{\frac{1}{3}}}$ [Eq 4]

where:

height is in feet.

charge size is in pounds.

If $2.24 \geq depth \geq 0$ then

$$Ground\ correction = \frac{depth}{-1.363} \qquad \text{[Eq 5]}$$

If $depth > 2.25$, then

$$Ground\ correction = \frac{(depth)(17.46)}{11.933} \qquad \text{[Eq 6]}$$

4. Gun angle correction -- The sound produced by a round fired by a weapon has a directivity pattern associated with it as a result of the physical characteristics of the weapon; noise produced by a demolition blast, however, is omnidirectional. Therefore, a correction factor must be added to account for directivity. This factor incorporates the angle between the direction in which the weapon muzzle is pointed and the point at which the calculation is being made (Figure 27). The correction factor can be found by looking it up in Table 4 for the appropriate gun type and closest angle, e.g., any angle between 15 and 45^{0} would be included under "30^{0}" in the table. There are still other variables to consider, but their effects are considered further on in the calculations.

V. Sound-Energy Exposure

At this stage, a table of all the corrected decibel values should be made. However, before the arithmetic operations (addition or subtraction) can be performed, the decibel values must be

I. Data Needed

Before proceeding with the calculations, the following set of data must be readily available:

1. Weapon type
2. Distance (in meters) from the noise source at which the decibel level is to be calculated
3. The time of firing, i.e., during the day (0700 to 2200 hours) or night (2200 to 0700 hours)
4. Weight of explosives in pounds (charge size or zone)
5. Height above or below ground (in feet) of the explosion
6. If a weapon, the angle of fire in relation to the point of calculation
7. Inversion factors for
   a. Surface
   b. 1 to 500 m
   c. 1 to 5000 m
8. Number of rounds fired
9. Number of days for which data were collected.

II. Blast Amplitudes

To find the appropriate C-weighted decibel values in the tables produced by TABGEN, the following calculations must be made: 100 $\log_{10}$ (distance in meters) - 199 = position in table counting across from left to right values for

1. Focus MAX
2. Focus MEAN
3. Base MAX
4. Base MEAN
5. Negative MAX
6. Negative MEAN
7. Excess Negative MAX
8. Excess Negative MEAN
9. Excess Negative MIN

The values listed above must be taken from Table 20. The user must be sure to choose either the day or night tables, depending on the data.

III. Percentages

The next set of values required are the percentages which are found in the same position in the tables as that calculated for the decibel values. There are four numbers, one for each of the regions FOCUS, BASE, NEGATIVE, and EXCESS NEGATIVE; these numbers are also differentiated as to day or night.

IV. Correction Factors

After the preceding initial values are obtained, a series of decibel correction factors are added to them, so that they correspond to the given data. These factors are a result of weapon type, charge size, right, height above or below ground, and angle of gun with respect to the receiver (point of calculation). After all the correction factors have been calculated, they are added together and then added to the number of decibels found initially. They are

transformed back to sound exposure in energy. Only levels above 85 dB are to be used in calculating the total SEL.* There are four separate cases which can be used for calculating the sound exposure, depending on the magnitude of the values in the table. Each region, FOCUS, BASE, NEGATIVE, and EXCESS NEGATIVE is considered separately. Before converting decibel values to energy, the user must first determine whether the corrected values for each region are greater than or equal to ≥ 85 dB.

Let

$$A = Mean - Min \quad \text{[Eq 7]}$$

$$C = Max - Mean \quad \text{[Eq 8]}$$

$$K = \frac{4.343(10^{C/10} - 1) - C}{A - 4.343(1 - 10^{-A/10})} \quad \text{[Eq 9]}$$

Case I. $Max \geq 85$? No. $Mean \geq 85$? No. $Min \geq 85$? No.

$$Sound\ Exposure = SE = 0 \quad \text{[Eq 10]}$$

Case II. $Max \geq 85$? Yes. $Mean \geq 85$? No. $Min \geq 85$? No.

$$F_1 = \frac{1.373 \times 10^9}{(KA + C)} \quad \text{[Eq 11]}$$

$$F_2 = 10^{\frac{Max - 85}{10}} - 1 \quad \text{[Eq 12]}$$

$$SE = F_1 * F_2 \quad \text{[Eq 13]}$$

Case III. $Max \geq 85$? Yes. $Mean \geq 85$? Yes. $Min \geq 85$? No.

$$F_1 = \frac{1.373 \times 10^9 K}{KA + C} \quad \text{[Eq 14]}$$

$$F_2 = 1 - 10^{\frac{Mean - 85}{10}} \quad \text{[Eq 15]}$$

$$M = 10^{\frac{Mean}{10}} \quad \text{[Eq 16]}$$

$$SE = M \cdot F_1 F_2 \quad \text{[Eq 17]}$$

*Based on the Committee on Hearing, Bioacoustics, and Biomechanics Assembly (CHABBA) C-weighted impulse [illegible] factor. *Environmental Protection Planning in the Noise Environment*, TM 5-803-2, Air Force Manual (AFM) 19-10, and Navy Publication (NAVFAC P-970) (Departments of the Air Force, Army, and the Navy, 15 June 1978); [illegible] CFR [illegible] January 197[illegible]; and *Guidelines for Preparing Environmental Impact Statements*, Report of Working Group [illegible] National Research Council, CHABBA of Behavioral and Social Sciences, 197[illegible].

Case IV: *Max* ≥ 85? *Yes. Mean* ≥ 85? *Yes. Min* ≥ 85? *Yes.*

$$SE = 10^{[illegible]} \qquad \text{[Eq 18]}$$

VI. Percentage Correction Factors

The tables created by TABGEN are under the conditions of standard percent/temperature inversion factors.* These are 74.2 percent at ground level; 8.6 percent at 0 to 500 m, and 18.67 percent at 0 to 3000 m. If these values differ for the area under consideration, the percentage values found in TABGEN are modified.

PC1 = ground or surface inversion factor

PC2 = 0 to 500 m

PC3 = 0 to 3000 m

If Day

$$Ratio = \frac{PC1 + PC2}{82.8} \qquad \text{[Eq 19]}$$

If night, and distance between source and point ≤ 2 miles:

$$Ratio\ R1 = \frac{PC1}{74.2} \qquad \text{[Eq 20]}$$

If night, and distance ≥ 10 miles

$$Ratio\ R2 = \frac{PC3}{2(18.67)} + 1/2 \qquad \text{[Eq 21]}$$

If night, and 2 ≤ distance ≤ 10

$$Ratio = \frac{R2 - R1}{0.7}\ Log10\ \frac{(distance)}{2}\ miles + R_1 \qquad \text{[Eq 22]}$$

where

distance is in miles

The percentages taken from the day or night percentage vs distance/curves are identified as follows

F = % FOCUS
N = % NEGATIVE
B = % BASE
EN = % EXCESS NEGATIVE

$$F1 = new\ FOCUS\ factor = F \times ratio \qquad \text{[Eq 23]}$$

$$B1 = new\ BASE\ factor = B \times ratio \qquad \text{[Eq 24]}$$

$$\Delta = (B - B1) + (F - F1) \qquad \text{[Eq 25]}$$

*Research is underway to refine this portion of the prediction by including both temperature and wind effects.

$$N1 = new\ NEGATIVE\ factor = \frac{N}{N+EN} + \Delta + N, \text{ if } < 0 = 0 \qquad \text{[Eq 26]}$$

$$EN1 = new\ EXCESS\ NEGATIVE\ factor = \frac{EN}{N+EN} \times \Delta + EN, \text{ if } < 0 = 0. \qquad \text{[Eq 27]}$$

VII. Sound-Exposure Total

A. Total

At this point, the sound-exposure values for each region FOCUS, BASE, NEGATIVE, and EXCESS NEGATIVE found in Section V are multiplied by the corresponding percentages found in Section VI. The four resulting numbers are then summed to determine the total sound exposure.

$$SE_T = SE_F + SE_B + SE_N + SE_{EN} \qquad \text{[Eq 28]}$$

where

SE = SOUND EXPOSURE * PERCENTAGE

T = TOTAL

F = FOCUS

B = BASE

N = NEGATIVE

EN = EXCESS NEGATIVE

B. Rounds

Calculation thus far has been one round per day or night. This is the stage at which calculations include more than one round, if necessary. Multiple rounds are accounted for by dividing the total number of rounds from a given source by the total length of time (in days) over which firing was averaged to obtain a single value for number of rounds per day. This value is then multiplied by the total sound exposure calculated from that one source to the point.

$$SE_{NEW} = \frac{\left[SE_T\right]\ (number\ of\ rounds)}{(day)} \qquad \text{[Eq 29]}$$

where

SE_{NEW} = sound exposure for specified conditions

SE_T = sound exposure total from Eq 28.

The sound exposures thus calculated for day and night values are added together. If there is only one source, then this is the final sound exposure (SE_{EN}) for the point and the calculations proceed to Section VIII. If there are more sources, this sound-exposure value is an intermediate result. Very seldom, however, is there only one source. If there is only one firing point and one target point, there are two sources. (Note: since the noise at the target point is omnidirectional, just as for a demolition or explosion, there should be no gun-angle correction for it during the calculation.) The final sound exposure at a given point is the sum of the energies due to all sources at that point. Thus, all the

calculations from Section II to VII are repeated to determine the sound exposure due to each of the sources at the point.)

VIII. CDNL

The value for the final sound exposure is transformed into decibel units by a logarithmic transformation to obtain the SEL

$$SEL = 10 \log_{10} SE_{FN} \qquad \text{[Eq 30]}$$

where

SE_{FN} is the final total sound exposure from all sources.

To obtain the CDNL values, a constant is subtracted from this value. This constant is the logarithm of the number of seconds in a day and arises from the way the equations for CDNL are handled by the program.

$$CDNL = SEL - 49.365 \qquad \text{[Eq 31]}$$

Example 1

I Firing Data*

(I1) Demolition

(I2) Distance from source = 4000 m

(I3) Daytime

(I4) Charge size = 5 lb

(I5) Above ground 200 ft

(I6) Omnidirection explosion (no target)

(I7) Standard inversion factors

(I7a) 74.2 percent

(I7b) 8.6 percent

(I7c) 18.67 percent

(I8) One round

(I9) 1 day

Blast amplitude (II) and percentage (III) use a distance of 4000 m. Find the decibel values for the regions FOCUS, BASE, NEGATIVE, and EXCESS in the TABGEN tables. Position in table = 100 log 4000 - 199 = 161 = first column in ninth row. Table 21 lists all values.

(IV) Correction factors

Question	Answer	Decibel Correction
(a) Night ?	No	0
(b) Charge size ?	5 lb	0
(c) Height ?	In air	0
(d) Gun angle ?	Not applicable	0
	TOTAL	0 dB

* Input Questions from which values in the following pages are derived are denoted by (I), (II), etc.

Since the correction factors total 0 dB, the corrected decibel values are the same as those found initially from TABGEN.

$$(V)\ BASE\ is\ Case\ IV\ SE = 10^{\frac{100.4}{10}} = 1.10 \times 10^{10} \qquad \text{[Eq 18]}$$

$$(V)\ FOCUS\ is\ Case\ IV\ SE = 10^{\frac{92.3}{10}} = 1.62 \times 10^{9} \qquad \text{[Eq 18]}$$

(V) NEGATIVE is Case II

$$A = 82.9 - 77.2 = 5.7 \qquad \text{[Eq 7]}$$

$$C = 87.0 - 82.9 = 4.1 \qquad \text{[Eq 8]}$$

$$K = \frac{4.343(10^{\frac{4.1}{10}} - 1) - 4.1}{5.7 - 4.343(1 - 10^{\frac{4.1}{10}})} = 1.077 \qquad \text{[Eq 9]}$$

$$F_1 = \frac{1.373 \times 10^{9}}{(1.077)(5.7) + 4.1} = 1.341 \times 10^{8} \qquad \text{[Eq 14]}$$

$$F_2 = 10^{\frac{87.88}{10}} - 1 = 0.585 \qquad \text{[Eq 15]}$$

$$SE = (1.341 \times 10^{8})(0.585) = 7.845 \times 10^{7} \qquad \text{[Eq 7]}$$

(V) EXCESS NEGATIVE is Case I

$$SE = 0 \qquad \text{[Eq 10]}$$

(VIIA)

$$SE_I = (1.10 \times 10^{10})(0.041) = 4.51 \times 10^{8}$$

$$SE_B = (1.62 \times 10^{9})(0.219) = 3.55 \times 10^{8}$$

$$SE_N = (7.845 \times 10^{7})(0.385) = 3.02 \times 10^{7}$$

$$SE_S = 0$$

$$SE_I = 8.372 \times 10^{8} \qquad \text{[Eq 28]}$$

(VIIB) No round correction since number of rounds and number of days both = 1

$$SE_{TN} = SE_T \qquad \text{[Eq 32]}$$

(VIII) $SEL = 10 \log_{10} (8.362 \times 10^8) = 89.2$ [Eq 30]

$$CDNL = 89.2 - 49.365 = 39.9 \qquad \text{[Eq 31]}$$

Example 2

(I) 105-mm howitzer (M102)

(I2) Distance from source = 1/2 mile

(I3) Night

(I4) Charge zone 6, A = 83.78; B = 13.91; AVG = 10.84

(I5) Not applicable

(I6) 90^0

(I7) Inversion factors

(I7a) 93 percent

(I7b) 2 percent

(I7c) 5 percent

(I8) 100 rounds

(I9) 2 days

(II and III) 1/2 mile = 804.5 m

Position in TABGEN table = $100 \log_{10} (804.5) - 199 = 91$ = eleventh column in fifth row. Table 21 lists the decibel values found in TABGEN.

(IV) Correction factors

		dB correction
(IV1) Nighttime ?	yes	10
(IV2) Charge size:	1.8656 lb	-25.5
(IV3) Height ?	Not applicable	0
(IV4) Gun angle ?	90^0	6.5
	TOTAL	-9.0

Add the correction factor to the initial values in TABGEN (Table 22).

(V) Sound exposure

$$F = 10^{\frac{111.5}{10}} = 1.41 \times 10^{11} \qquad \text{[Eq 18]}$$

$$B = 10^{\frac{104.6}{10}} = 2.88 \times 10^{10} \qquad \text{[Eq 18]}$$

$$N = 10^{\frac{94}{10}} = 2.82 \times 10^{9} \qquad \text{[Eq 18]}$$

EXCESS NEGATIVE is Case II

$$A = 84.3 - 77.3 = 7.0$$
$$C = 86.9 - 84.3 = 2.6$$

$$K = \frac{4.343(10^{\frac{2.6}{10}} - 1) - \frac{2.6}{10}}{7.0 - 4.393(1 - 10^{\frac{-10}{10}})} = 0.272 \qquad \text{[Eq 9]}$$

$$F_1 = \frac{1.373 \times 10^{9}}{(0.272)(7.0) + 2.6} = 3.05 \times 10^{8} \qquad \text{[Eq 14]}$$

$$F_2 = 10^{\frac{86.9 - 88}{10}} - 1 = 0.549 \qquad \text{[Eq 15]}$$

$$SE = (3.05 \times 10^{8})(0.549) = 1.67 \times 10^{8} \qquad \text{[Eq 17]}$$

(VI) PC1 = 93 PC2 = 2 PC3 = 5

$$\textit{Night, < 2 miles, } RI = \frac{93}{74.2} = 1.25 \qquad \text{[Eq 20]}$$

$$F1 = (6.1)(1.25) = 7.625 \textit{ percent} \qquad \text{[Eq 23]}$$

$$B1 = (39.0)(1.25) = 48.75 \textit{ percent} \qquad \text{[Eq 24]}$$

$$\Delta = (6.1 - 7.625 + (39.0 - 48.75) = -11.275 \qquad \text{[Eq 25]}$$

$$N1 = \frac{(39.7)}{54.9}(-11.275) + 39.7 = 31.55 \textit{ percent} \qquad \text{[Eq 27]}$$

$$EN1 = \frac{(15.2)}{54.9}(-11.275) + 15.2 = 12.08 \textit{ percent} \qquad \text{[Eq 27]}$$

(VIIA) $SE_F = (141 \times 10^{10})(0.0762) = 1.074 \times 10^{10}$ [Eq 28]

$SE_B = (2.88 \times 10^{10})(0.4875) = 1.404 \times 10^{10}$

$SE_N = (2.82 \times 10^{9})(0.3155) = 8.897 \times 10^{8}$

$SE_{FN} = (1.67 \times 10^{8})(0.1208) = 2.022 \times 10^{7}$

$SE_T = 2.569 \times 10^{10}$ [Eq 28]

(VIIB) $$\frac{100 \textit{ rounds}}{2 \textit{ day}} = 50 \textit{ rounds/day} \qquad \text{[Eq 30]}$$

$$SI = SI_{AN} - (2.569 \times 10^{-3})(500) - 1.285 \times 10^{-1} \quad \text{[Eq 30]}$$

(XIII)

$$SII = 10 \log_{10}(1.285 \times 10^{-1}) = 121.1\ dB \quad \text{[Eq 33]}$$

$$CDNI = 121.1 - 49.365 = 71.7\ dB \quad \text{[Eq 34]}$$

Table 21
Decibel Values from TABGEN for Example 1

		(II) Decibels	(III) Percent
I	MAX	105.4	4.1
	MEAN	100.4	
B	MIN MAX	96.8	21.9
	MEAN	92.1	
N	MIN MAX	87.0	38.7
	MEAN	82.9	
EN	MIN MAX	77.2	
	MEAN	73.0	35.5
	MIN	63.8	

Table 22
Corrected Decibel Values from TABGEN for Example 2

		(II) Decibels	(III) Percent	(IV) Corrected Decibel Values
	MAX	123.9	8	114.9
I	MEAN	120.5		111.5
	MIN MAX	118.9	52.8	109.9
B	MEAN	113.6		104.6
	MIN MAX	107.9	38.6	98.9
N	MEAN	103.5		94.5
	MIN MAX	95.9		86.9
EN	MEAN	93.3	7.8	84.3
	MIN	86.3		77.3

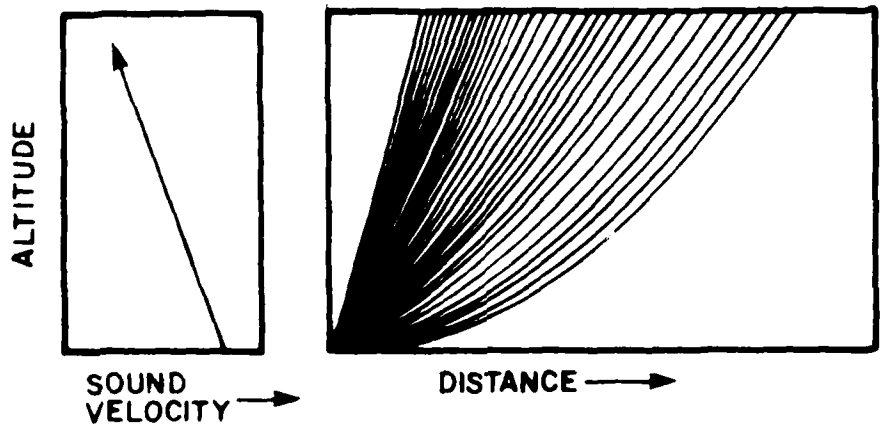


Figure 19 Negative sound velocity gradient

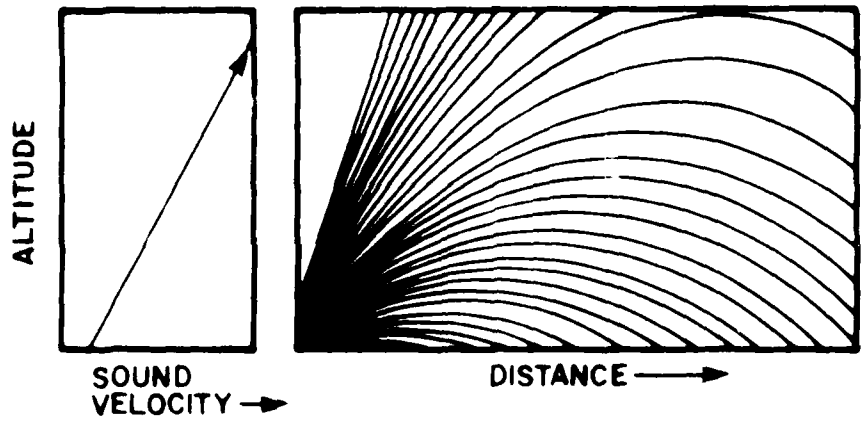


Figure 20 Positive sound velocity gradient

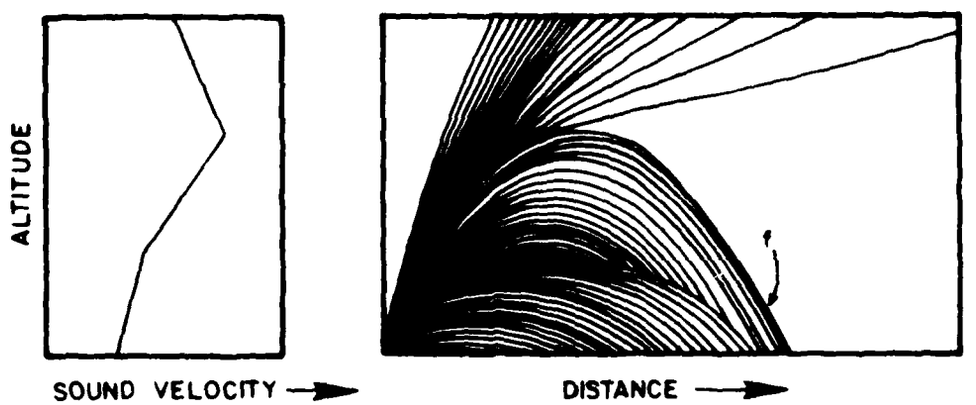


Figure 21 Sharply charged positive sound

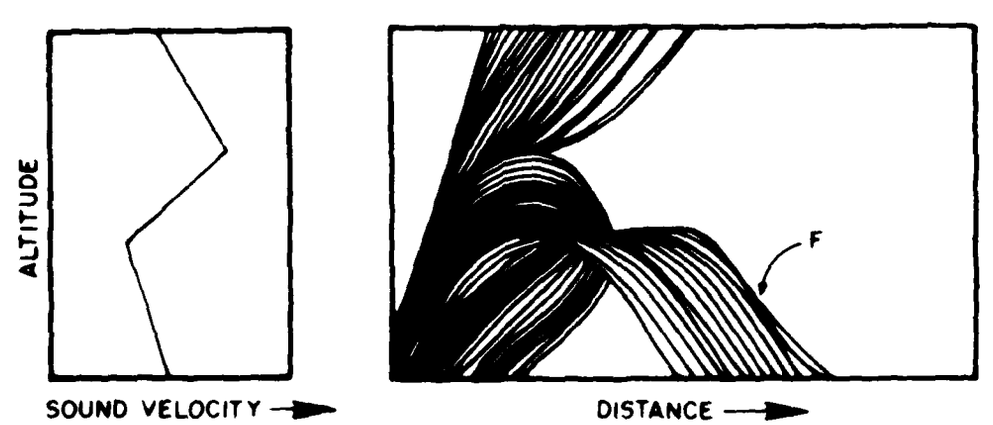


Figure 22

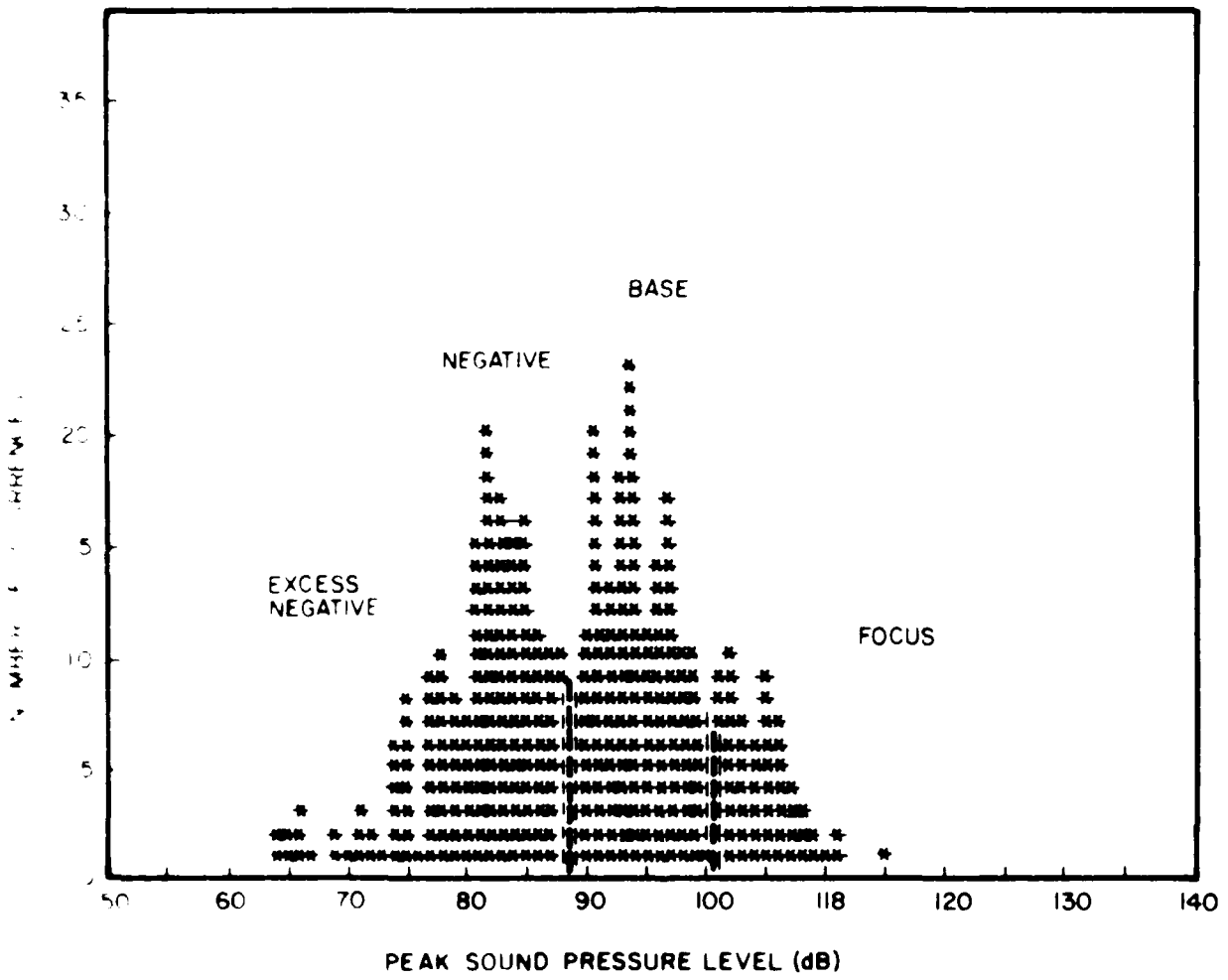


Figure 23 Sample amplitude description

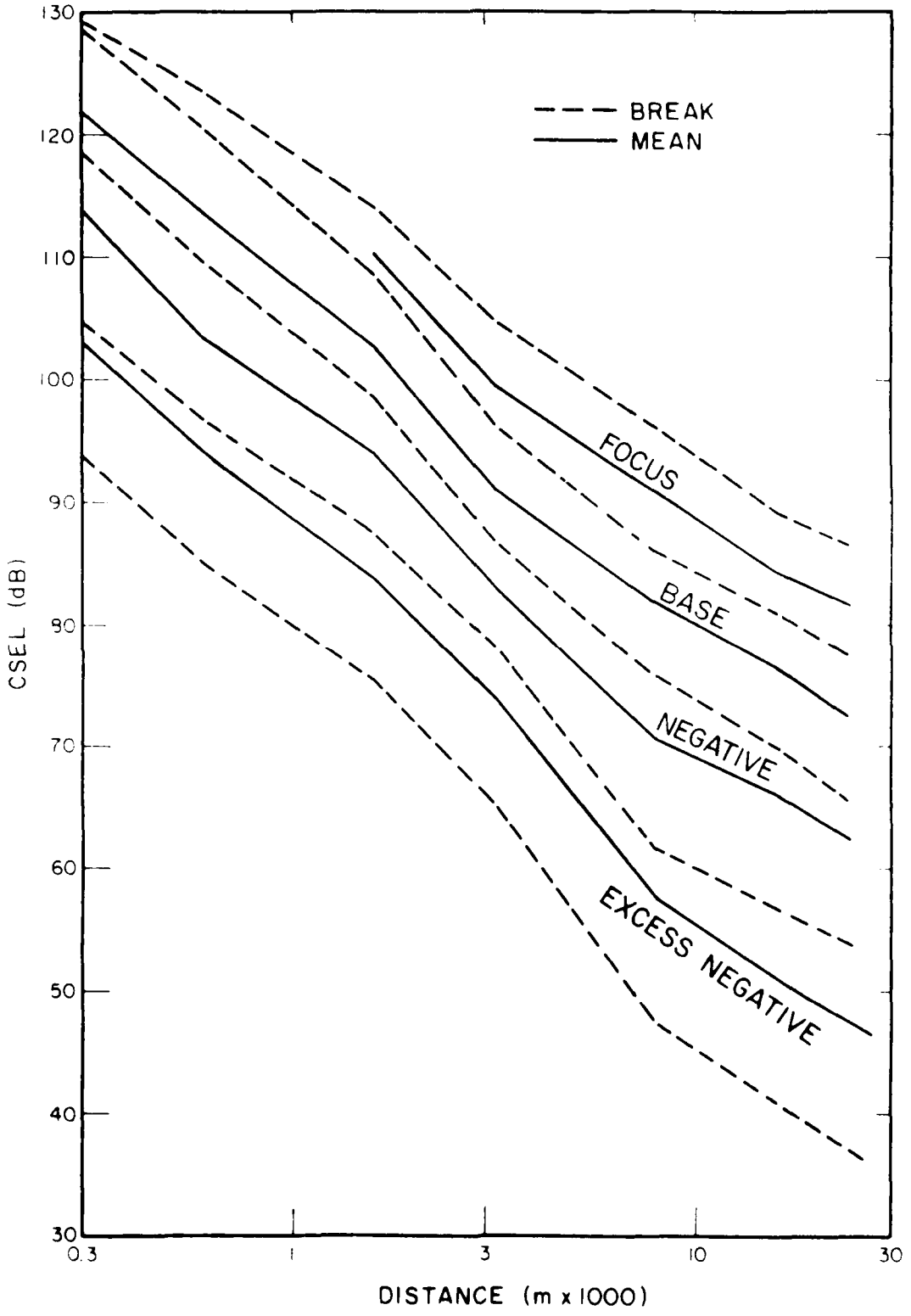


Figure 24 Means and break (day)

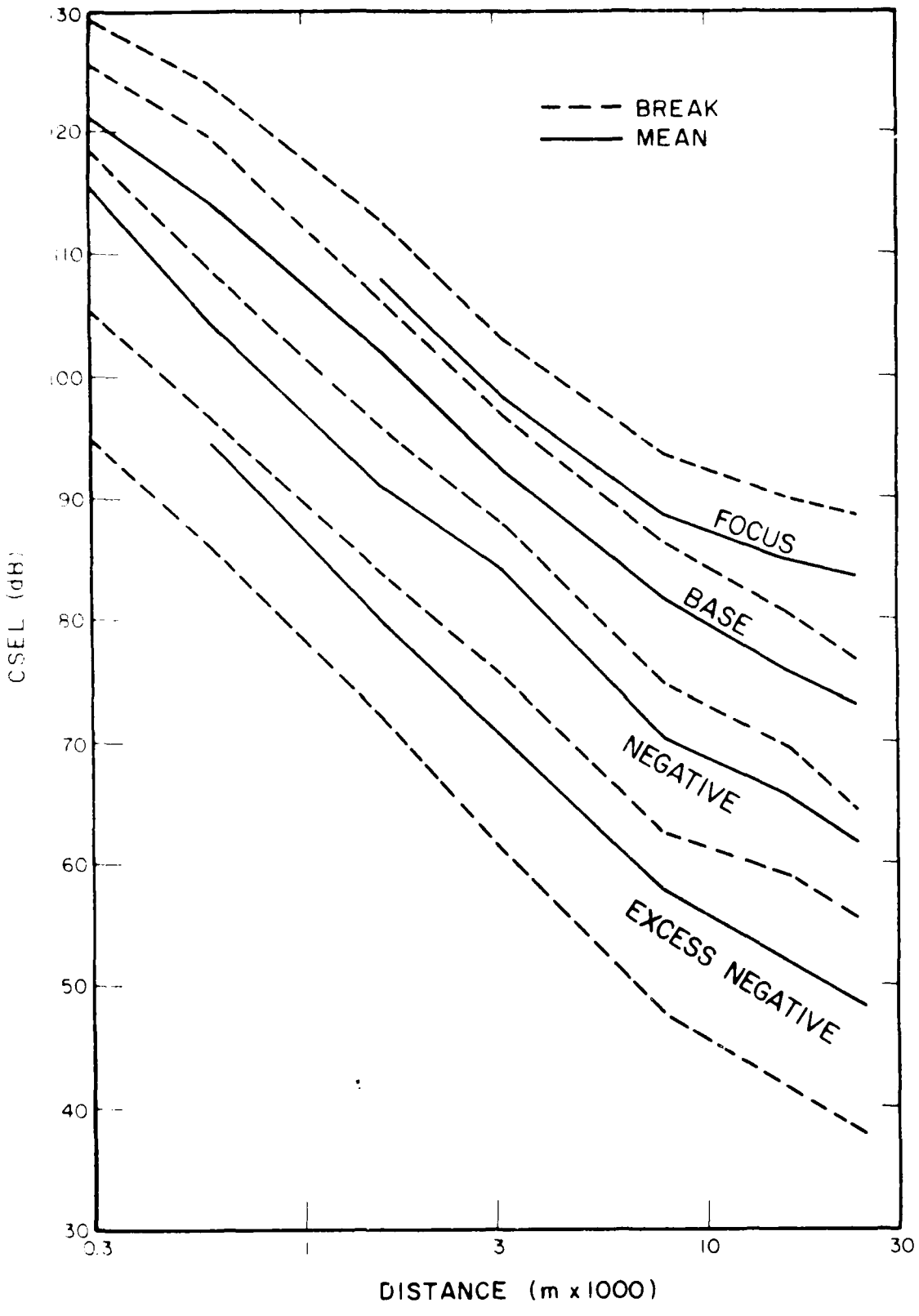


Figure 25 Means and break (night)

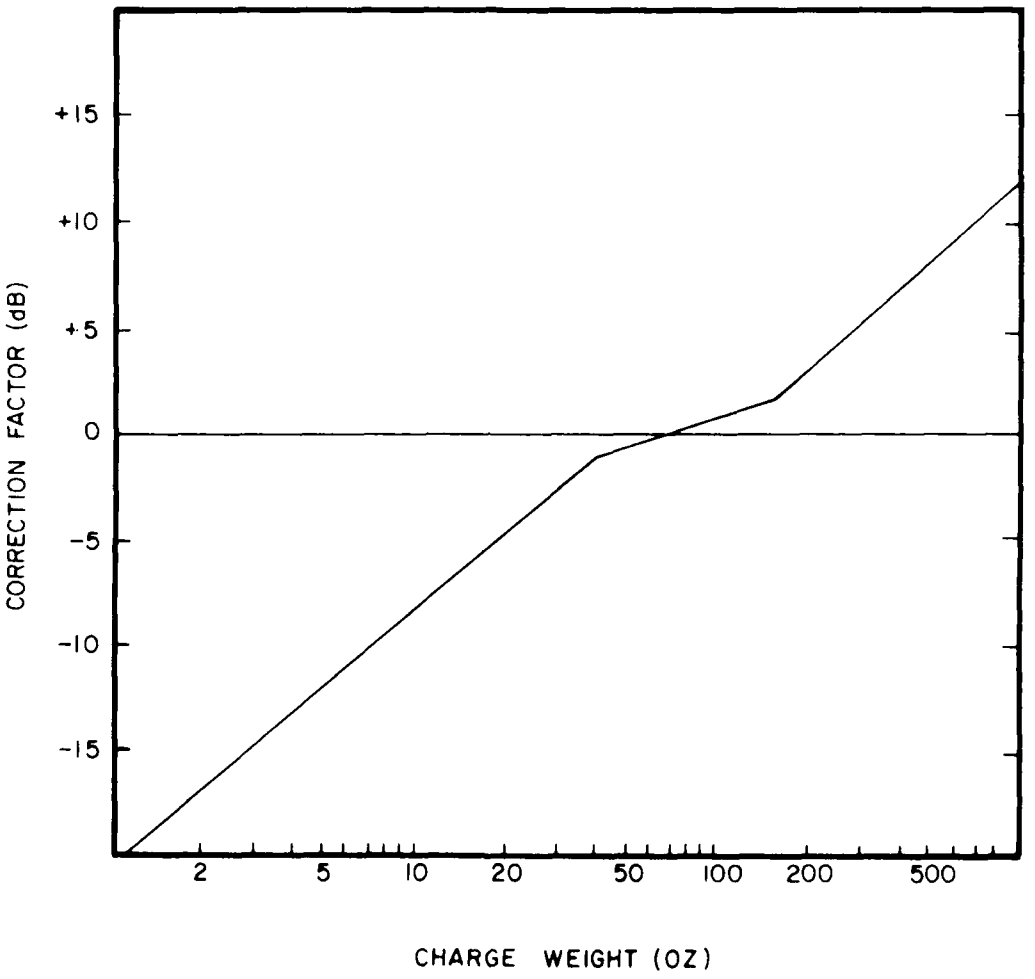


Figure 26 Charge size correction graph.

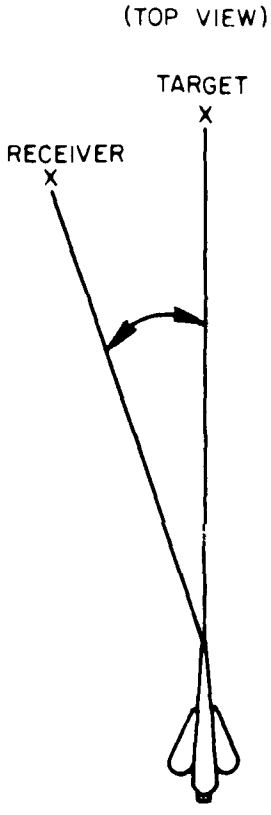


Figure 27 Angle illustrations.

APPENDIX:

BNOISE 3.2 SOURCE LISTING

```
      PROGRAM ZCON(INPUT=128,OUTPUT=128,TAPE5=INPUT,TAPE6=OUTPUT,
     1TAPE7=128,TAPE8=512,TAPE1=128,TAPE2=128,TAPE3=128,TAPE4=128,
     2 TAPE9=512,TAPE11=128,TAPE10=128,TAPE70=128,TAPE75=128,         ZNEF   4
     3 TAPE8 =512,TAPE99=128,TAPE44=128,TAPE51=128,TAPE55=128,        ZNEF   5
     4 TAPE1)


C     COMMON BLOCK CONTAINS VARIABLES KARD, WHICH REPRESENTS THE
C     CARD READER AND KPRINT, WHICH REPRESENTS THE LINE PRINTER
      COMMON/IO/KARD,KPRINT                                           ZNEF   6
C     ARRAY HEAD WILL CONTAIN THE NAMES OF THE DIFFERENT
C     MODULES CALLED BY THE MAIN PROGRAM.
      DIMENSION HEAD(10)                                              ZNEF   7
C     COMMON BLOCK BOUND CONTAINS THE MAXIMUM AND MINIMUM
C     BOUNDS FOR THE X AND Y GRID COORDINATES AND THE LOGICAL
C     FLAG BDS THAT CHECKS TO SEE IF THESE VALUES ARE ACCEPTABLE
      COMMON/BOUND/XMIN,YMIN,XMAX,YMAX,BDS                            ZNEF   8
      LOGICAL BDS,LARGE                                               ZNEF   9
C     BLOCK PLTCOM CONTAINS INFORMATION WHICH IS COMPUTED
C     IN SUBROUTINE PLOT AND USED BY THE NASAPLOT PROGRAM
C     FOR CONTOUR MAKING.
      COMMON/PLTCOM/PLTCT,SXTABLE,SYTABLE,LARGE                       ZNEF  10
C     PLTCT IS THE NUMBER OF PLOTS REQUESTED
      INTEGER PLTCT                                                   ZNEF  11
C     BLOCK SRCS CONTAINS THE NUMBER OF SOURCES.  IT IS
C     USED IN FORMA, CALCNR, AND PGRID.
      COMMON /SRCS/ NSRCS
C     COMMON BLOCK FT CONTAINS INFORMATION CONCERNING
C     THE LOCATIONS OF THE TARGETS AND FIRING POINTS,
C     TOTAL NUMBER OF DAY AND NIGHT FIRINGS,(ALL OF WHICH
C     ARE DECLARED ARRAYS), THE GRID SIZE GRDSZ, AND THE
C     CNR, WHICH IS THE DB VALUE AT A GIVEN POINT.
      COMMON/FT/GRDSZ,CNP,XLOC,YLOC,DAYNO,DARKNO,TEMPB                ZNEF  13
C     TEMPB IS NEEDED TO ALLOW FOR SPACE IN THE MAPS SUBROUTINE
      DIMENSION TEMPB(7000),TEMPA(5000)

      DIMENSION XLOC(2000),YLOC(2000),DARKNO(2000),DAYNO(2000)        ZNEF  14
C     IN COMMON BLOCK GRID, NRCNR,EAS,RCNR,TEMP,AND DIST
C     ARE NO LONGER USED IN THE PROGRAM OR THE SUBROUTINES
C     ARRAY SDBWH STORES CAHRGE INFORMATION; ARRAYS ANGSIN,
C     AND ANGCOS STORE SINE AND COSINE VALUES FOR THE ANGLE
C     OF FIRING.
      COMMON/GRID/NRCNR,EAS,RCNR,DIST,SDBWH,               ANGCOS,ANGSIN
     1  ,TEMP(4000),TEMPA
C    TMPA IS NEEDED FOR EXTRA SPACE IN THE MAPS SUBROUTINE
      DIMENSION RCNR(10),DIST(10),SDBWH(2000),
     1ANGCOS(2000),ANGSIN(2000)                                       ZNEF  17
      COMMON/DEBUG/CHECK,REED,TABRD                                   ZNEF  20
      COMMON/PARM/THRESH,PENITE,PPIP,IMP
C     LOGICAL FLAGS:  CHECK=FLAG FOR GROUND CORR FACTOR
C     ON NEF=1 CAL ; REED=FLAG FOR READI.;TABRD=FLAG FOR READTB.
      COMMON/METRIC/METERS,IMETER,IBOTH                               ZNEF  21
      LOGICAL METERS                                                  ZNEF  22
C     DAYCNR AND DARKCNR ARE LOGICAL FLAGS THAT DENOTE
C     IF WE ARE COMPUTING DAYTIME OR NIGHTTIME DATA.
      COMMON/CALC/DAYCNR,DARKCNR                                      ZNEF  23
```

```
      LOGICAL DAYCNR,DRKCNR                                              ZNEF  24
C     THIS COMMON BLOCK(ANGLE) IS NO LONGER USED.
      COMMON/ANGLE/COSC,SINC                                             ZNEF  25
      DIMENSION COSC(72),SINC(72)                                        ZNEF  26
      LOGICAL CHECK,REED,TAHRD                                           ZNEF  27
C     STP IS A LOGICAL FLAG THAT IS SOMETIMES USED TO INVOKE THE
C     STOPP ROUTINE.
      LOGICAL STP                                                        ZNEF  29
C     KARD = TAPE 5(INPUT) AND KPRINT = TAPE 6(OUTPUT)
      DATA KARD/5/,KPRINT/6/                                             ZNEF  30
      DATA GRDSZ,EAS/5000.0,0.500/
C     INITIALIZE ARRAY HEAD TO MODULE NAMES
      DATA HEAD/10HMAP       ,10HFORM=A     ,10HPOINT     ,10HPUDDLE GRI,
     110HPLOT      ,10HBASE       ,10HSCATTER   ,
     210HSTOP      ,10HLOCATOR    ,10HBOUNDS    /,                        ZNEF  36
     3 NP/10/

C     INITIALIZE FLAGS AND COUNTERS
      STP=.FALSE.                                                        ZNEF  39
      REED=.FALSE.                                                       ZNEF  40
      TAHRD=.FALSE.                                                      ZNEF  41
      CHECK=.FALSE.                                                      ZNEF  42
      LARGE=.FALSE.
C
      DRKCNR=DAYCNR=.TRUE.                                               ZNEF  43
      BDS=.FALSE.                                                        ZNEF  44
      XMIN=0.                                                            ZNEF  45
      YMIN=0.                                                            ZNEF  46
      XMAX=0.                                                            ZNEF  47
      YMAX=0.                                                            ZNEF  48
      PLTCT=0                                                            ZNEF  49
      SXTABLE=0.                                                         ZNEF  50
      SYTABLE=0.                                                         ZNEF  51

C THE PROGRAM READS THE FIRST CARD OF THE MODULE INPUT, WHICH CONTAINS THE UNITS
C USED IN COMPUTATION(METERS OR FEET).  THE FIRST PAGE OF OUTPUT STATES
C "DISTANCE EXPRESSED IN METERS"
      READ(KARD,700)HOR,ICHECK,LCHECK,TI,PI,PPI,IMP                      ZNEF  59
C     IMPULSE FACTOR INSERTED BY PAUL SCHOMEN  10 JAN 80
      PPIP=10**(PPI/10)
C     IF THE UNIT METERS IS REQUESTED FOR USE, SET APPROPRIATE FLAG
      METERS=(HOR .EQ. 10HMETERS     )                                   ZNEF  60
C      CHECK FOR THE AVAILABILITY OF LARGE PAPER
      LARGE=(LCHECK .GT. 0)
C   SET PARAMETERS IF REQUESTED.
      THRESH=85.0
      PERITE=10.0
      IF(TI .NE. 0) THRESH=TI
      IF(PI .NE. 0) PERITE=PI
      CHECK=(ICHECK .GT. 0)                                              ZNEF  61
C     UNITS COULD BE IN FEET
      IMETER=10HFEET                                                     ZNEF  62
C     IF THE METERS FLAG IS TRUE, THEN WE USE METERS.
      IF (METERS) IMETER=10HMETERS                                       ZNEF  63
```

```
      C     USING DAYTIME AND NIGHTTIME DATA
            IN THE ... THEN ...                                    ZNEF  64
            WRITE(KPRINT,M ...)ISTEM                               ZNEF  65

      C*    ECHO SPECIAL PARAMETERS FOR TESTING
            WRITE(KPRINT,A10) THRESH,PENITE,PRIP,IMP

      C THE ARRAY HEAD(15) CONTAINS THE NAMES OF THE MODULES.  THEREFORE, THE IF-GO-
      C TO STATEMENT RELATES TO THAT TEST FOR EACH MODULE NAME.
       100  READ(KARD,700) MOR                                     ZNEF  66
            IF(EOF(KARD).NE.0.) GO TO 300                          ZNEF  67
      C     THIS LOOP READS EACH MODULE NAME AND THE MULTIPLE GO TO SENDS
      C     THE PROGRAM TO THE APPROPRIATE CALL STATEMENT.
            DO 150 I=1,NP                                          ZNEF  68
            IF (MOR .EQ. HEAD(I)) GO TO 200                        ZNEF  69
       150  CONTINUE                                               ZNEF  70
      C     AN ILLEGAL CALL WAS ATTEMPTED
            WRITE(KPRINT,701) MOR                                  ZNEF  71
            STOP                                                   ZNEF  72

      C     JUMP TO APPROPRIATE CALL ACCORDING TO THE VALUE OF I
      200   GOTO (210,220,225,230,240,250,260,310,320,330) , I


        210 CALL MAP                                               ZNEF  75
            GO TO 100                                              ZNEF  76
        220 CALL FORMA                                             ZNEF  77
            GO TO 100                                              ZNEF  78
      225   CALL POINT
            GO TO 100                                              ZNEF  80
        230 CALL PGRID                                             ZNEF  81
            GO TO 100                                              ZNEF  83
        240 CALL PLOT                                              ZNEF  84
            GO TO 100                                              ZNEF  85
        250 CALL BASE                                              ZNEF  86
            GO TO 100                                              ZNEF  87
        260 CALL SCATTR                                            ZNEF  88
            GO TO 100                                              ZNEF  89

      C CLOSE NASAPLOT INPUT FILE                                  ZNEF 114
        310 CALL STOPP                                             ZNEF 115
            STP=.TRUE.                                             ZNEF 116
            GO TO 100                                              ZNEF 117
        320 CALL LOCATR                                            ZNEF 118
            GO TO 100                                              ZNEF 119
        330 CALL BOUNDS                                            ZNEF 120
            GO TO 100                                              ZNEF 121
       300  CONTINUE                                               ZNEF 122
      C CALL PLOT TERMINATION ROUTINE IF NOT PREVIOUSLY CALLED     ZNEF 123
            IF(.NOT.STP)CALL STOPP                                 ZNEF 124
            STOP                                                   ZNEF 125
       700  FORMAT(A10,2I1,8X,3F10.0,I2)                           ZNEF 126
       701  FORMAT(1H1,///,*.....ERROR..... NAME = 10H*,A10,* IS NOT A LEGAL CZNEF 127
           1ALL TO NEF MAIN PROGRAM.....*,//)                      ZNEF 128
        800 FORMAT(1H1///T50,*.....    HNOISE3.2       .....*/////  ZNEF 132
           151X,*DISTANCES EXPRESSED IN *,A10)                     ZNEF 133
```

```
C*   FORMAT FOR TEST ECHO
  810 FORMAT(///1H ,1X,*TIMES=*,5X,F5.1/1H ,10X,*PERMITE*,5X,F5.1/1H ,
175     1 1X,*PPI=*,E12.4/1H ,10X,*I=P*,10X,I2)
        END                                                    ZNEF 134
```

SYMBOLIC REFERENCE MAP (R=5)

ENTRY POINTS DEF LINE REFERENCES
11536 LCDR 1

VARIABLES	SN	TYPE		ALLOCATION								
3746	ANGCOS	REAL	ARRAY	GRID	REFS	44	47					
7666	ANGSIN	REAL	ARRAY	GRID	REFS	44	47					
4	BOS	LOGICAL		METRIC	REFS	18	19	DEFINED	85			
0	CHECK	LOGICAL		PARM	REFS	49	62	DEFINED	81	110		
1	CNR	REAL		FT	REFS	34						
0	COSC	REAL	ARRAY	ANGLE	REFS	60	61					
13562	DAMAND	REAL	ARRAY	FT	REFS	34	38					
0	DAYCNR	LOGICAL		CALC	REFS	57	58	DEFINED	84			
7642	DAYND	REAL	ARRAY	FT	REFS	34	38					
14	DIST	REAL	ARRAY	GRID	REFS	44	47					
1	DRKCNR	LOGICAL		CALC	REFS	57	58	DEFINED	84			
1	EAS	REAL		GRID	REFS	44	DEFINED	68				
0	GRDSZ	REAL		FT	REFS	34	DEFINED	68				
12015	HDR	REAL			REFS	102	130	133	DEFINED	98	125	
12024	HEAD	REAL	ARRAY		REFS	14	130	DEFINED	70			
12023	I	INTEGER			REFS	130	137	DEFINED	129			
2	IBUTH	INTEGER		METRIC	REFS	53	DEFINED	116				
12016	ICHECK	INTEGER			REFS	110	DEFINED	98				
1	IMETER	INTEGER		METRIC	REFS	53	117	DEFINED	112	114		
3	IMP	INTEGER		PARM	REFS	50	120	DEFINED	98			
0	KARD	INTEGER		IO	REFS	11	126	DEFINED	67	I/O REFS	98	125
1	KPRINT	INTEGER		IO	REFS	11	DEFINED	67	I/O REFS	117	120	133
3	LARGE	LOGICAL		PLOTCM	REFS	19	23	DEFINED	82	104		
12017	LCHECK	INTEGER			REFS	104	DEFINED	98				
0	METERS	LOGICAL		METRIC	REFS	53	54	114	DEFINED	102		
11702	NP	INTEGER			REFS	129	DEFINED	70				
0	NRCNR	INTEGER		GRID	REFS	44						
0	NSHCS	INTEGER		SHCS	REFS	28						
1	PERMITE	REAL		PARM	REFS	50	120	DEFINED	107	109		
12021	PI	REAL			REFS	2*109	DEFINED	98				
0	PLTCT	INTEGER		PLOTCM	REFS	23	25	DEFINED	90			
12022	PPI	REAL			REFS	100	DEFINED	98				
2	PPIP	REAL		PARM	REFS	50	120	DEFINED	100			
2	RCNR	REAL	ARRAY	GRID	REFS	44	47					
1	REED	LOGICAL		DEBUG	REFS	49	62	DEFINED	79			
26	SDMMM	REAL	ARRAY	GRID	REFS	44	47					
110	SINC	REAL	ARRAY	ANGLE	REFS	60	61					
12014	STP	LOGICAL			REFS	65	165	DEFINED	78	157		
1	SXTABLE	REAL		PLOTCM	REFS	23	DEFINED	91				
2	SYTABLE	REAL		PLOTCM	REFS	23	DEFINED	92				
2	TAPRD	LOGICAL		PARM	REFS	49	62	DEFINED	80			

VARIABLES	SN	TYPE									
13[illegible]	TEMP	REAL	ARRAY	[illegible]	REFS	44					
23446	TEMPA	REAL	ARRAY	[illegible]	REFS	36	44				
175 2	TEMPB	REAL	ARRAY	FT	REFS	34	36				
[illegible]	THRESH	REAL		[illegible]	REFS	5	120	DEFINED	106	108	
12[illegible]	TI	REAL			REFS	2*18	DEFINED	94			
2	XLOC	REAL	ARRAY	FT	REFS	34	38				
2	XMAX	REAL		[illegible]	REFS	16	DEFINED	82			
0	XMIN	REAL		[illegible]	REFS	16	DEFINED	86			
3722	YLOC	REAL	ARRAY	FT	REFS	34	38				
3	YMAX	REAL		[illegible]	REFS	18	DEFINED	89			
1	YMIN	REAL		[illegible]	REFS	18	DEFINED	87			

FILE NAMES	MODE
5	INPUT
254	OUTPUT
2060	TAPE1
7430	TAPE10
2334	TAPE2
3340	TAPE20
2610	TAPE3
3064	TAPE4
0	TAPE5
6700	TAPE51
7154	TAPE55
254	TAPE6
530	TAPE7
5144	TAPE70
4414	TAPE71
4670	TAPE72
5420	TAPE75
1004	TAPE8
5674	TAPE90
6150	TAPE91
6424	TAPE99

VARIABLES USED AS FILE NAMES, SEE ABOVE

EXTERNALS	TYPE	ARGS	REFERENCES								
BASE		0	150								
BOUNDS		0	161								
EOF	REAL	1	126								
FORMA		0	142								
LOCATR		0	159								
MAP		0	140								
PGRID		0	146								
PLOT		0	148								
POINT		0	144								
SCATTR		0	152								
STOPP		0	156	165							

STATEMENT LABELS		DEF LINE	REFERENCES							
11601	100	125	141	143	145	147	149	151	153	158	160
			162								
0	150	131	129								
11616	200	137	130								
11635	210	140	137								
11640	220	142	137								
11643	225	144	137								

STATEMENT LABELS			DEF LINE	REFERENCES
11646	230		148	137
11651	240		149	137
11654	250		150	137
11657	260		152	137
11674	300		163	126
11662	310		156	137
11666	320		159	137
11671	330		161	137
11742	700	FMT	167	98 125
11746	701	FMT	168	133
11761	800	FMT	170	117
11773	810	FMT	174	120

LOOPS	LABEL	INDEX	FROM-TO	LENGTH	PROPERTIES	EXITS
11611	150	I	129 131	2H	INSTACK	

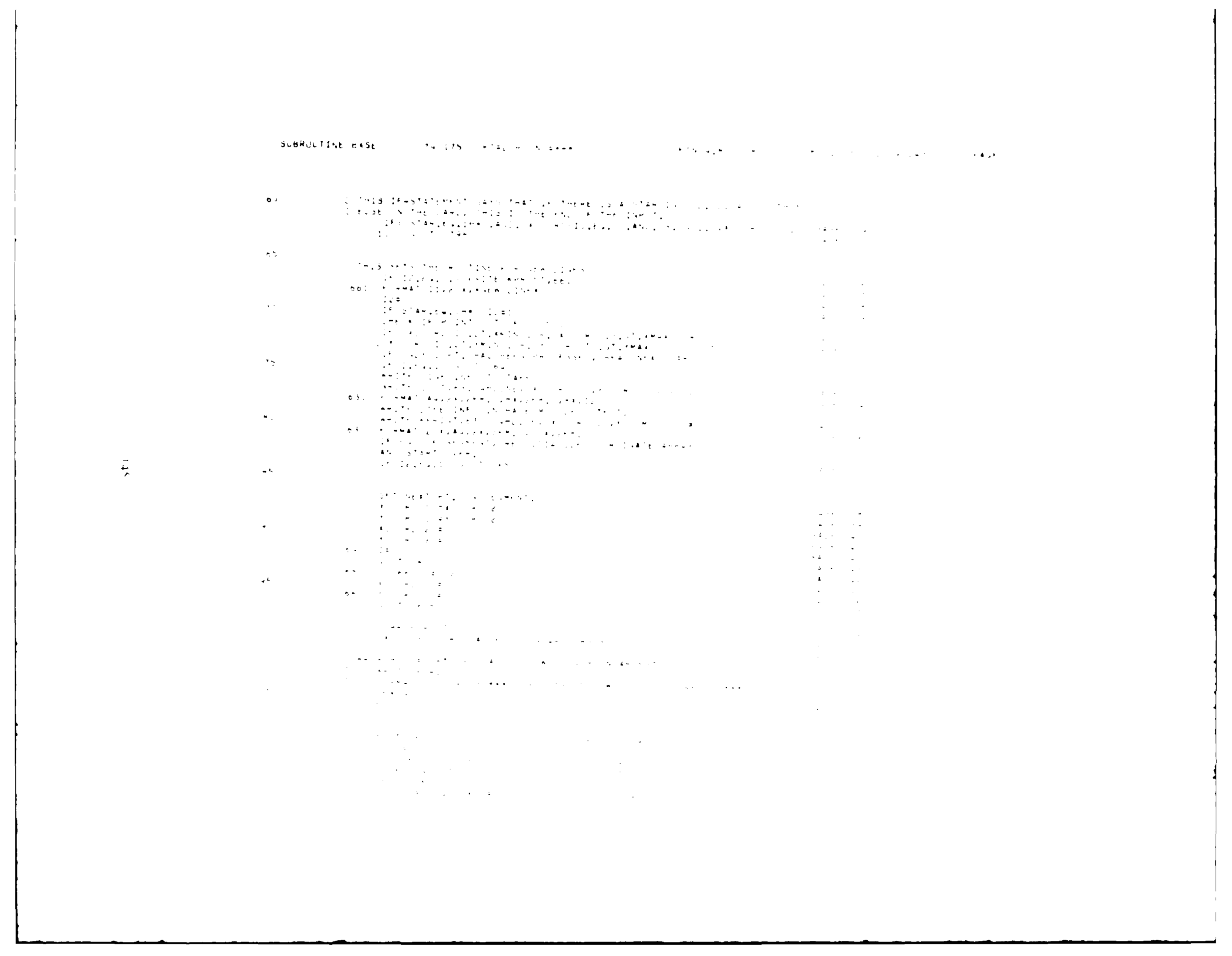
COMMON BLOCKS	LENGTH	MEMBERS - BIAS NAME(LENGTH)							
IO	2	0 KARD	(1)	1 KPRINT	(1)				
BOUND	5	0 XMIN	(1)	1 YMIN	(1)	2 XMAX	(1)		
		3 YMAX	(1)	4 BDS	(1)				
PLOTCM	4	0 PLTCT	(1)	1 SXTABLE	(1)	2 SYTABLE	(1)		
		3 LARGE	(1)						
SRCS	1	0 NSRCS	(1)						
FT	15002	0 GRDSZ	(1)	1 CNR	(1)	2 XLOC	(2000)		
		2002 YLOC	(2000)	4002 DAYNO	(2000)	6002 DARKNO	(2000)		
		8002 TEMPB	(7000)						
GRID	15022	0 NRCNR	(1)	1 EAS	(1)	2 RCNR	(10)		
		12 DIST	(10)	22 SOBHH	(2000)	2022 ANGCOS	(2000)		
		4022 ANGSIN	(2000)	6022 TEMP	(4000)	10022 TEMPA	(5000)		
DEBUG	3	0 CHECK	(1)	1 REED	(1)	2 TABRD	(1)		
PARM	4	0 THRESH	(1)	1 PENITE	(1)	2 PPIP	(1)		
		3 IMP	(1)						
METRIC	3	0 METERS	(1)	1 IMETER	(1)	2 IBOTH	(1)		
CALC	2	0 DAYCNR	(1)	1 DRKCNR	(1)				
ANGLE	144	0 COSC	(72)	72 SINC	(72)				

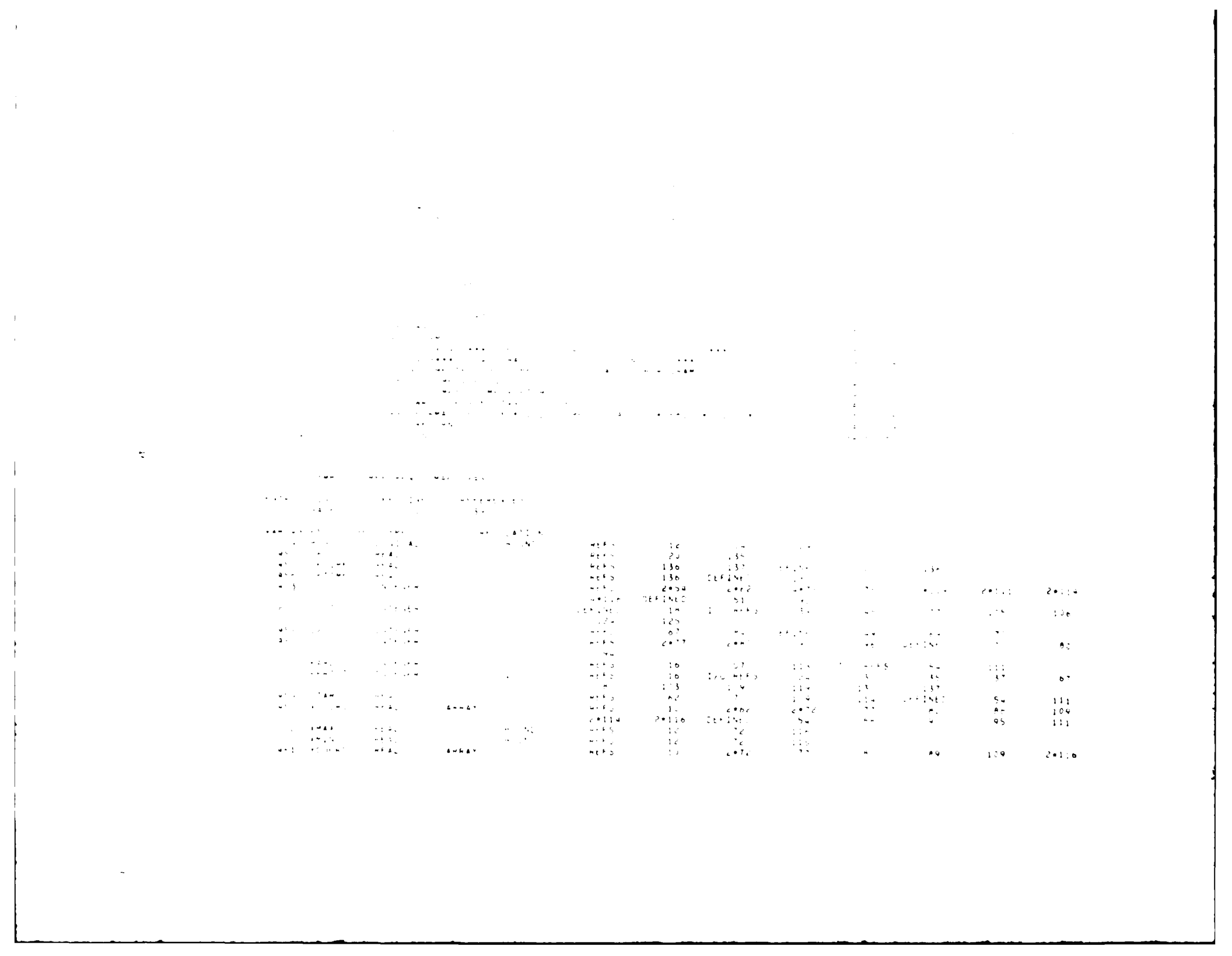
STATISTICS

PROGRAM LENGTH	2067B	1079
BUFFER LENGTH	7747B	4071
CM LABELED COMMON LENGTH	727608	30192

600008 CM USED

```
      SUBROUTINE BASE                                                 BASE
C * THIS SUBROUTINE CREATES THE OUTLINE OF THE BASE OR REGION THE USER IS
C WORKING WITH ... BY DRAWING LINE SEGMENTS FROM ... COORDINATE
C COORDINATE OF THE SPECIFIED REGION.


C     THESE TWO ARRAYS HOLD THE VALUES OF THE STARTING
C     AND ENDING POINTS FOR THE LINES DRAWN.
      DIMENSION XCOORD(2),YCOORD(2)                                   BASE    3
C     BLOCK HOLD CONTAINS BOUNDARY INFORMATION.
      COMMON/B  NE/XMIN,YMIN,XMAX,YMAX,BDS                            BASE    4
C     BOUNDS FLAG.
      LOGICAL BDS                                                     BASE    5
C     KARD=INPUT; KPRINT=OUTPUT
      COMMON /IO/KARD,KPRINT                                          BASE    6
C     IOUT=TAPE 2
      DATA IOUT/2/                                                    BASE    7
C     CDC FUNCTION TO COMPUTE TIME SPENT IN A PROGRAM
      CPTIME=SECOND(CP)                                               BASE    8


      WRITE(KPRINT,400)                                               BASE    9
  400 FORMAT(1H1///35X,*.....      BASE      .....*//)                BASE   10

C  IF THERE ARE NOBOUNDS IN THE INPUT MODULES, COMPLAIN.
      IF(BDS) GO TO 450                                               BASE   11
      WRITE(KPRINT,470)
  470 FORMAT(*  ...ERROR,  NO BOUNDS CALL BEFORE BASE....*)
      STOP

C THIS PRINTS OUT THE HEADING "LINE CARDS", "X-Y START", "X-Y END"
  450 WRITE(KPRINT,592) 0                                             BASE   12
  592 FORMAT(I1,42X,*LINE CARDS*)                                     BASE   13
      WRITE(KPRINT,593) 0                                             BASE   14
  593 FORMAT(I1,29X,*X-Y START*,35X,*X-Y END*)                        BASE   15
      REWIND IOUT                                                     BASE   16


C PHASE 4 PLOTS SPECIAL LINES AND TEXT ON THE MAP AND DRAWS A BORDER IN THE
C MAP.
C     WRITE HEADER FOR PHASE 4
      WRITE(IOUT,590)                                                 BASE   17
  590 FORMAT(4HPHS4)                                                  BASE   18
C     I2 ACTS AS A FLAG TO TELL IF A SEGMENT HAS BEEN COMPLETED
C     (I2=1)
      I2=0                                                            BASE   19
C     I SERVES AS COUNTER FOR STARTING AND ENDING PT. INFO.
  605 I=1                                                             BASE   20
C     READ BASE-2 CARD WITH COORDS OF POINT IN SET DESCRIBING
C     A LINE SEGMENT.
  610 READ(KARD,620) XCOORD(I),YCOORD(I),STAR                         BASE   21
  620 FORMAT(2F10.0,A1)                                               BASE   22
C ERROR MESSAGE="NO EOF CARD: "JOB ABORTED"
      IF(EOF(KARD).NE.0.) GO TO 7..                                   BASE   23
```





```
VARIABLES     SN  TYPE          RELOCATION
                                              DEFINED     54     84      8      88     111
    3  YMAX       REAL             /ROUND/       REFS     12     72    116
    1  YMIN       REAL             /ROUND/       REFS     12     72    116
       VARIABLES  USED AS FILE NAMES, SEE ABOVE

EXTERNALS          TYPE    ARGS       REFERENCES
       EOF         REAL      1             57      113
       SECOND      REAL      1              2      135

INLINE FUNCTIONS    TYPE    ARGS        DEF LINE   REFERENCES
       SIGN        REAL      2  INTRIN                 62      114

STATEMENT LABELS                   DEF LINE   REFERENCES
  225  400       FMT                  25         24
   12  450                            35         29
  236  470       FMT                  31          3
  271  590       FMT                  46         45
  25   592       FMT                  36         35
  26   593       FMT                  38         37
   27  605                            51         97
   37  610                            54         93
  3 2  620       FMT                  55         54      111
  341  630       FMT                  61         60
  325  631       FMT                  78         77
  125  640                            92         75
  127  650                            94         84
       660                            96         94
  311  661       FMT                  68         67
  133  710                           103         72      118
  351  715       FMT                 104        103
  147  717                           109        116
  367  720       FMT                 110        109
  173  790                           119         57      113
  4 4  791       FMT                 120        119
  2 2  796                           130        114
  427  797       FMT                 131        130
  214  798                           135         62      126
  437  799       FMT                 138        137

LOOPS   LABEL      INDEX      FROM-TO      LENGTH    PROPERTIES
   71                J        77 77         108                 EXT REFS
  1 6                J        80 80         104                 EXT REFS
  13   660           J        94 96          35      INSTACK

COMMON BLOCKS   LENGTH      MEMBERS - BIAS NAME(LENGTH)
     ROUND         5                 0 XMIN    (1)            1 YMIN    (1)            2 XMAX    (1)
                                     3 YMAX    (1)            4 BUS     (1)
     IO            2                 0 KARD    (1)            1 KPRINT  (1)

STATISTICS
  PROGRAM LENGTH                     4676     311
  UNLABELED COMMON LENGTH              76       ?
         60000B CM USED
```

```
      SUBROUTINE B SET(MIN,MAX,GRID,SET)
C
C     THIS SUBROUTINE IS USED TO TEST IF THE MIN AND MAX VALUES OF THE
C     AXIS SIZE, IF NOT THEY ARE MODIFIED UNTIL BOTH ARE MULTIPLES
C     OF THE GRID SIZE.  IT ALSO SETS A FLAG (SET) TO SAY IF ANY OF THE
C     MAX, MIN OR GRID SIZE HAS BEEN CHANGED.  THE NUMBER OF GRID POINTS BETWEEN MIN AND
C     MAX IS ALSO GIVEN IN N.  WHEN THE VALUES ARE CHANGED THE DIFFERENCE
C     SHOULD BE THE MAXIMUM POSSIBLE.
C
      REAL MIN,MAX,GRID,M,N
      LOGICAL SET
      DATA M /1.0/
C
C     GET THE DIFFERANCE AND SEE IF A MULTIPLE OF GRID
      N=(MAX-MIN)/GRID
      M=0
      IF (ABS(N-M).LT.TOL) RETURN
C
C     NOT A MULTIPLE OF GRID SO MODIFY
      M=INT(MIN/GRID)
C     MAKE MIN SMALLER IF IT IS NEGATIVE SINCE FUNCTION FIX DOESN'T
      IF (MIN.LT.0.0) M=M-1
      MIN=M*GRID
C
C     TAKE CARE OF MAX      MAKE IT BIGGER IF IT IS POSITIVE
      M= INT(MAX/GRID)
      IF (MAX.GT.0.0) M=M+1
      MAX=M*GRID
C
      N=(MAX-MIN)/GRID
C     SET FLAG
      SET=.TRUE.
      RETURN
      END
```

```
        SYMBOLIC REFERENCE MAP (R=3)

 ENTRY POINTS      DEF LINE      REFERENCES
     3  BDSET             1         17          34

 VARIABLES       SN   TYPE              RELOCATION
    37  D             REAL                                  REFS        16         17    DEFINED       15
     0  GRID          REAL                     F.P.         REFS        10         15         21       24         27         29         31
                                                         DEFINED         1
    40  M             INTEGER                               REFS        23         24         28       29    DEFINED         21         23
                                                              27        28
     0  MAX           REAL                     F.P.         REFS        10         15         27       28         31
                                                         DEFINED         1         29
     0  MIN           REAL                     F.P.         REFS        10         15         21       23         31
                                                         DEFINED         1         24
     0  [illegible]   INTEGER                  F.P.         REFS        17    DEFINED          1       16         31
     0  SET           LOGICAL                  F.P.         REFS        11    DEFINED          1       33


        SUBROUTINE BDSET        74/175  OPT=2 ROUND=+-*/            FTN 4.8  508          80/10/30. 12.45.43       PAGE      2

 VARIABLES       SN   TYPE              RELOCATION
    36  TOL           REAL                                  REFS        10         17    DEFINED       12

 INLINE FUNCTIONS     TYPE     ARGS          DEF LINE   REFERENCES
        ABS           REAL       1   INTRIN                  17
        INT           INTEGER    1   INTRIN                  21         27

 STATISTICS
   PROGRAM LENGTH                 41B        33
        60000B CM USED
```

```
      SUBROUTINE BOUNDS                                                 BOUNDS 2
C * THIS SUBROUTINE DEFINES THE AREA OF INTEREST FOR THE LOON CALCULATIONS AND
C SETS THE LIMITS OF THE AREA IN TERMS OF MAP COORDINATES USED FOR THE CREATION
C OF THE PLOT.


      COMMON/BOUND/XMIN,YMIN,XMAX,YMAX,BDS                              BOUNDS 3
      COMMON /IO/KARD,KPRINT                                            BOUNDS 4
      LOGICAL BDS                                                       BOUNDS 5
      CPTIME=SECOND(CP)                                                 BOUNDS 6


      WRITE(KPRINT,591)                                                 BOUNDS 7
  591 FORMAT(1H1///35X,*.....     BOUNDS     .....*//)                  BOUNDS 8
      BDS=.FALSE.                                                       BOUNDS 9

C THIS INFORMATION IS FOUND ON THE BDS-2 CARD
C XMIN= MINIMUM X COORDINATE
C YMIN= MINIMUM Y COORDINATE
      READ(KARD,600) XMIN,YMIN                                          BOUNDS10
C ERROR MESSAGE:  * MISSING INPUT CARD:  JOB ABORTED
      IF(EOF(KARD).NE.0.) GO TO 900                                     BOUNDS11

C THIS INFORMATION IS FOUND ON THE BDS-3 CARD
C XMAX= MAXIMUM X COORDINATE
C YMAX= MAXIMUM Y COORDINATE
      READ(KARD,600) XMAX,YMAX                                          BOUNDS12
      IF(EOF(KARD).NE.0.) GO TO 900                                     BOUNDS13
  600 FORMAT(2F10.0)                                                    BOUNDS14

C     WRITE OUT MIN AND MAX BOUNDS;OUTPUT FOR BOUNDS
      WRITE(KPRINT,599) XMIN,YMIN,XMAX,YMAX                             BOUNDS15
  599 FORMAT(20X,*MINIMUM BOUNDARY = *,2F8.0, 5X,*MAXIMUM BOUNDARY = *,2BOUNDS16
     1F8.0)                                                             BOUNDS17

C      MIN OR MAX ERROR TEST                                            BOUNDS18
      IF((XMAX.LE.XMIN).OR.(YMAX.LE.YMIN))GO TO 700                     BOUNDS19
C     BOUNDS VERIFIED--SET FLAG TRUE
      BDS=.TRUE.                                                        BOUNDS20

C VERIFICATION=" BOUNDARY VALUES VERIFIED
      WRITE(KPRINT,650)                                                 BOUNDS21
  650 FORMAT(1H0,19X,*BOUNDARY VALUES VERIFIED*)                        BOUNDS22
      GO TO 798                                                         BOUNDS23

C ERROR MESSAGE:"INCORRECT BOUNDARIES//NO BOUNDS SET"
  700 WRITE(KPRINT,705)                                                 BOUNDS24
  705 FORMAT(20X,*INCORRECT BOUNDARIES*//40X,*NO BOUNDS SET*)           BOUNDS25
      BDS=.FALSE.                                                       BOUNDS26
```

```
                C CALCULATION OF TIME SPENT IN BOUNDS SUBPROGRAM
      60          798  CPTIME1=SECOND(CP)                                          BOUNDS27
                       CPTIME=CPTIME1-CPTIME                                       BOUNDS28
                       WRITE(KPRINT,799) CPTIME                                    BOUNDS29
                  799  FORMAT(1HO,20X,*..... TIME IN BOUNDS IS*,F8.3,* .....*)    BOUNDS30
                       RETURN                                                      BOUNDS31
      65          C EOF EXIT                                                      BOUNDS32
                    900 WRITE(KPRINT,901)                                         BOUNDS33
                    901 FORMAT(//* ERROR -- MISSING INPUT CARD; JOB ABORTED*)     BOUNDS34
                        STOP                                                      BOUNDS35
                        END                                                       BOUNDS36
```

151

SYMBOLIC REFERENCE MAP (R=3)

ENTRY POINTS

		DEF LINE	REFERENCES
1	BOUNDS	1	64

VARIABLES

		SN	TYPE	RELOCATION								
9	ACS		LOGICAL	BOUND	REFS	8	10	DEFINED	17	44	56	
167	CP		REAL		REFS	11	60					
166	CPTIME		REAL		REFS	61	62	DEFINED	11	61		
170	CPTIME1		REAL		REFS	61	DEFINED	60				
0	KARD		INTEGER	IO	REFS	9	25	31	I/O REFS	23	30	
1	KPRINT		INTEGER	IO	REFS	9	I/O REFS	15	36	48	54	62
						66						
2	XMAX		REAL	BOUND	REFS	8	36	42	DEFINED	30		
0	XMIN		REAL	BOUND	REFS	8	36	42	DEFINED	23		
3	YMAX		REAL	BOUND	REFS	8	36	42	DEFINED	30		
1	YMIN		REAL	BOUND	REFS	8	36	42	DEFINED	23		

VARIABLES USED AS FILE NAMES, SEE ABOVE

```
EXTERNALS          TYPE    ARGS         REFERENCES
      EOF        REAL       1              2?          31
      SECOND     REAL       1              1?          6?

STATEMENT LABELS                    DEF LINE    REFERENCES
   55  591       FMT                   ?          1?
  11?  599       FMT                   ?          36
   77  60?       FMT                   3?         23          3?
  123  65        FMT                   ?          ?9
   31  ??                              ?          ?2
  133  7?5       FMT                   ?          54
   34  79?                             ?          5?
  146  799       FMT                   63         62
   43  90?                             ??         25          31
  157  901       FMT                   67         66

COMMON BLOCKS    LENGTH      MEMBERS - BIAS NAME(LENGTH)
      BOUND         5                  0 XMIN    (1)          1 YMIN   (1)          2 XMAX   (1)
                                       3 YMAX    (1)          4 ROS    (1)
      IC            2                  0 KARL    (1)          1 KPRINT (1)


     SUBROUTINE BOUNDS     74/175  OPT=2 ROUND=+-*/          FTN 4.8  508        80/10/30. 12.45.43        PAGE     3

STATISTICS
  PROGRAM LENGTH                  1719     121
  CM LABELED COMMON LENGTH          7B       7
        60000B CM USED
```

```
      SUBROUTINE CALCNR(X,Y)                                            CALCNR 2
C                                                                       CALCNR 3
C *CALCNR CALCULATES THE LDN VALUES FOR A GRID POINT X,Y.
C                                                                       CALCNR 5
C DATA USED:
C DISTANCE FROM NOISE SOURCE(METERS)
C DAY OR NIGHT FIRING
C CHARGE SIZE
C HEIGHT ABOVE OR BELOW GROUND
C ANGLE(IF IT IS A GUN)
C INVERSION FACTORS: SURFACE;0-500 METERS,0-3000 METERS
C NUMBER OF ROUNDS FIRED
C NUMBER OF DAYS DATA WAS COLLECTED


      INTEGER DAY
      COMMON/IO/KARD,KPRINT                                             CALCNR 6
C     BLOCK FACTI CONTAINS INVERSION FACTORS
      COMMON/FACTI/RINV1,RINV2,RINV3
      COMMON /SRCS/ NSRCS
      COMMON/FT/GRDSZ,CNR,XLOC,YLOC,DAYNO,DARKNO                        CALCNR 9
      DIMENSION XLOC(2000),YLOC(2000),DARKNO(2000),DAYNO(2000)          CALCNR10
C     BLOCK GRID CONTAINS UNUSED VARIABLES(SEE ZNEF)
      COMMON/GRID/NRCNR,EAS,RCNR,DIST,SDBWM,                ANGCOS,ANGSIN
C     BLOCK CONTR CONTAINS GUN TYPE TABLE, INFO FOR GUN CONTOURS
C     (IN INPUT) AND VALUE FOR THE LIMIT OF THE GUN TYPE TABLE.
      COMMON/CONTR/IDGUN,CONTOR
      DIMENSION IDGUN(50),CONTOR(50,15)
      DIMENSION RCNR(10),DIST(10),SDBWM(2000),
     1ANGCOS(2000),ANGSIN(2000)                                         CALCNR13
C     ARRAY GUNT(IN COMMON BLOCK GUN) CONTAINS ID'S FOR
C     WEAPONS ARRANGED BY SUBROUTINE FORMA.
      INTEGER GUNT(2000)                                                CALCNR14
      COMMON /GUN/GUNT                                                  CALCNR15
      COMMON/METWIC/METERS,IMETER,IBOTH                                 CALCNR18
      LOGICAL METERS                                                    CALCNR19
      COMMON/CALC/DAYCNR,DRKCNR                                         CALCNR20
      LOGICAL DAYCNR,DRKCNR                                             CALCNR21
      COMMON/DEBUG/CHECK,REED,TABRD                                     CALCNR23
      COMMON/PARM/THRESH,PERITE,PPIP,IMP
      LOGICAL CHECK,REED,TABRD                                          CALCNR24
C     BLOCK TABL1 CONTAINS INFO FROM PROGRAM TABGEN
      COMMON /TABL1/ DBV(301,9,2),PERV(301,4,2),ENV(1501)    ,
     1 CSCF(601) ,FUN1(301,4,2),FUN2(301,4,2),FTAJ(151,2)
C                                                                       CALCNR27
C FUN1  FOR CASE ONE          DIST; F H C F; DAY  NIGHT
C FUN2 THE IS FOR CASE TWO        DIST;  F H C; DAY  NIGHT
C
C
C     X = X-AXIS VALUE IN THESE DISTANCES
C     Y = Y-AXIS VALUE IN THESE DISTANCES
      CNSDAY=CNSDAY=0.0                                                 CALCNR31
C                                                                       CALCNR32
```

```
C     CALCULATE THE ANGLE OF DIRECTION (PHI ANGLE) WITH RESPECT TO THE FIRICALCNR33
C     LINE TO X AND SHIFTED COORDINATE, THE TOTAL OVERPRESSURE DB FACTOR.  CALCNR34
C                                                                          CALCNR35
C CALCULATE DISTANCE AND ANGLE OF NOISE SOURCE
      DO 1 IS=1,NSHOS                                                      CALCNR36
C     CALCULATE X,Y DISTANCES
      DX=X-XLOC(IS)                                                        CALCNR42
      DY=Y-YLOC(IS)                                                        CALCNR43
C     DISTANCE FORMULA
      A=DX*DX+DY*DY                                                        CALCNR44
      D=SQRT(A)

C IF UNITS ARE IN FEET, CHANGE TO METERS
      IF(.NOT.METERS)D=D/3.28                                              CALCNR46
      IF(D.EQ.0..OR.ANGCOS(IS).EQ.99.)GO TO 34                             CALCNR47


C     ANGLE PHI IS THE ANGLE OF DIRECTIVITY SOUGHT.  ANGLE ALPHA IS THE
C     ANGLE FROM REFERENCE POINT TO X,Y COORDINATES.  ANGLE THETA IS THE
C     ANGLE FOR WHICH SINE AND COSINE CALCULATIONS HAVE ALREADY BEEN MADE.
C     THE FORMULAS USED IN CALCULATING ANGLE PHI WILL BE AS FOLLOWS:
C     SIN(PHI)=SIN(ALPHA-THETA) AND COS(PHI)=COS(ALPHA-THETA), WHICH BY
C     TRIGNOMETRIC SUBSTITUTION YIELDS:
C     SIN(PHI)=SIN(ALPHA)COS(THETA)-SIN(THETA)COS(ALPHA) AND
C     COS(PHI)=COS(ALPHA)COS(THETA)+SIN(ALPHA)SIN(THETA)
C     SIN(ALPHA)=DY/D AND COS(ALPHA)=DX/D
C     SIN(THETA)=ANGSIN(IS) AND COS(THETA)=ANGCOS(IS)
C     THEREFORE, THE FINAL FORMULAS WILL BE:
C     SIN(PHI)=(DY/D)*ANGCOS-(DX/D)*ANGSIN;
C     COS(PHI)=(DX/D)*ANGCOS+(DY/D)*ANGSIN.

      SINPHI=(DY/D)*ANGCOS(IS)-(DX/D)*ANGSIN(IS)
      COSPHI=(DX/D)*ANGCOS(IS)+(DY/D)*ANGSIN(IS)

C
C     TO COMPUTE THE ANGLE PHI, WE WILL USE THE ARCTANGENT FUNCTION
C     WITH DX(PHI)=COSPHI AND DY(PHI)=SINPHI.
C     CALCULATE PHI IN DEGREES
      PHI=ATAN2(SINPHI,COSPHI)*180.0/3.1415926


C CALCULATION OF DIRECTIVITY CORRECTION DBDEG.
C     FIND GUN TYPE IN GUN TYPE TABLE FROM FORMA(LIMIT=50)
C     USE INDEX FROM IDGUN TO FIND ROW IN CONTOR
      IROW=GUNT(IS)

C     FIND COLUMN IN CONTOR TO LOCATE CORRECTION FACTOR
      ITEMP=((PHI-180)/30)-0.5
      IF(PHI.LE.0.)I=(-ITEMP)-6
      IF(PHI.GT.0.)I=6-ITEMP
      IF(I.EQ.12)I=0
      ICOL=I+3

      DBDEG = CONTOR(IROW,ICOL)
      GO TO 32
   34 DBDEG=0.0                                                            CALCNR66
```

```
      C     COMBINE ANGLE CORR AND CHARGE CORR
       32   OVERPR=DHEFG+SOBAR(IS)                                       CALCNR67
      C
      C CALCULATE OVERPRESSURE DB FOR BASE,FOCUS,NEGATIVE GRADIENT CURVE.
      C                                                                  CALCNR68
      C     CALCULATE OVERPRESSURE DB FOR BASE CURVE, FOCUS, AND NEGATIVE CALCNR69
      C     GRADIENT CURVE.                                              CALCNR70
      C                                                                  CALCNR71
      C                                                                  CALCNR72
      C     DISTANCE IS KEPT BETWEEN 100 AND 100000.
            IF (D.LT.100.) D=100.
            IF(D.GT.100000.)D=100000.                                    CALCNR74
      C     FORMULA TO CALCULATE POSITION IN TANGEN TABLE
            L=ALOG10( [illegible] -193.) + 0.5
      C
            DO 1 DAY= 1,2
      C     TEST IF [illegible] ARE [illegible] JUST DAY OR JUST NIGHT
            CNT=1.
            IF (DAY.EQ.1.AND.DAYONR)  CNT=DAY.J(IS)
            IF (DAY.EQ.2.AND.[illegible]) CNT = DARKN(IS)
            IF (CNT.EQ. [illegible]) [illegible] 1
      C
      C CONVERT DB VALUES TO SOUND EXPOSURE
            DO 3 M= 1,4
      C     LOOP THRU [illegible] EXPES
      C
      C     GET DB AND ADD IN CHARGE CORR AND ANGLE CORRECTION
            DMAX=DHV( [illegible] , M*2-1 , DAY) + OVERPR
            DMEAN = DBV( [illegible] , M*2 , DAY) + OVERPR
            DMIN= DBV( [illegible] , M*2+1 , DAY) + OVERPR
      C
      C     ASSUME [illegible] CASE [illegible] ENERGY
            EN= [illegible]
      C     TEST FOR [illegible] CASES
      C     TAKE DB VALUE [illegible] TANGEN ANGLE CORRECTION
      C     FACTOR [illegible]
            IF (THRESH [illegible] DMAX) [illegible] CASE 4
            IF (THRESH [illegible] DMAX) [illegible]
            IF (THRESH [illegible] DMIN) [illegible] CASE 1
            IF (THRESH [illegible]) [illegible] 1
            IF (THRESH [illegible] DMEAN) [illegible] 3
            IF (THRESH [illegible] DMEAN) [illegible] 3
      C     OTHERWISE [illegible] CASE 2.
      C
      C     CASE 2
      C     [illegible]
            [illegible] (THRESH [illegible] M,DAY) *
           1 [illegible] THRESH [illegible]),1)
            IF [illegible] AY,DMAX,DMEAN,DMIN,EN
          * [illegible]
            [illegible]
      C
      C     CASE 3
        3   EN= [illegible] (DMAX-THRESH ) * 10.0) ,2)
            IF [illegible] AY, MAX,DMEAN,DMIN,EN
```

```
      GOTO 1
C     CASE 1
101   EN = EXP( IFIX( MEAN ) , -240. )
      IF(CHECK) WRITE(KPRINT,9 ) CUM,DAY,DMAX,DMEAN,DMIN,EN
C     CASE 1 ONLY WAS HERE DIRECTLY ...
C     USE IMPULSE CORRECTION FACTOR ONLY IF IMP WAS INPUT AS NONZERO IN
C     THE "IMPULSE" CARD.
1     IF(IMP.EQ.0) GOTO 12
C     IMPULSE FACTOR CORRECTION -- PAUL SCHOMER 10 JAN 80
C     10**9.25 = 10**9 * 10**(1/4) = 10**9 * SQRT(SQRT(10)) = 1778279410
C     D1=10**9.25  (=1778279410)
C     D10=10**10.25  (=17782794100  BY SIMILAR DERIVATION)
      D1=1778279410
      D10=17782794100
C     IF(EN.LE.10**9.25) EN=EN AND GOTO 120
      IF(EN.LE.D1) GOTO 120
C     IF(EN.GE.10**10.25) EN=EN*10 AND GOTO 120
      IF(EN.LT.D10) GOTO 110
      EN=EN*10
      GOTO 120
C     (10**9.25 .LT. EN .LT. 10**10.25)  IS TRUE AT THIS POINT
C     EN=EN * 10**(ALOG10(EN)-9.25)  BY THE FOLLOWING METHOD...
C     10**(ALOG10(EN)-9.25) = (10**ALOG10(EN)) / (10**9.25)
C                          = EN / 10**9.25 = EN / D1
110   EN=EN*EN/D1
120   IF (DAY.EQ.2) GOTO 105
C     ADD IN REPETITION COUNT
      CNSDAY= CNSDAY + PERV(LD,M,1) * EN *  CNT
      GOTO 3
105   CNSNI  = CNSNI + PERV(LD,M,2) * EN * CNT * PPIP
3     CONTINUE
1     CONTINUE


      TCNSDAY=0.0
      TCNSNI=0.0
      IF (CHECK.AND. CNSDAY.GT. 0.0) TCNSDAY=10.0* ALOG10(CNSDAY)
      IF (CHECK.AND.CNSNI.GT.0.0) TCNSNI= 10.0*ALOG10(CNSNI)
      IF (CHECK) WRITE(KPRINT,800) TCNSDAY,TCNSNI
C "POWER OF DAY SUM" TCNSDAY, "POWER OF NIGHT SUM", "TCNSNI"
      IF (CNSDAY .EQ. 0.0 .AND. CNSNI .EQ. 0.0) GOTO 20
C     CONVERT IT TO AN LDN
C     FINAL SOUND EXPOSURE CONVERTED BACK TO DECIBEL UNITS
      CNR=  10.0 *  ALOG10 (CNSDAY+ CNSNI) - 49.3651374
      RETURN                                                          CALCN167
   20 CONTINUE                                                        CALCN168
      CNR=-99.                                                        CALCN169
C WARNING...FOR X=_,Y=_, BOTH DAY +NIGHT CUMULATIVE NOISE SUMS=0; NEF SET
C TO -99.*
      WRITE(KPRINT,801)X,Y                                            CALCN170
```

```
                         RETURN                                                   CALCN171
    230             800  FORMAT(10X,*POWER OF DAY SUM *,F6.2,* POWER OF NIGHT SUM *,F6.2)  CALCN175
                     801 FORMAT(//10X,*..... WARNING .....F R X =*,F7.2,*, Y =*,F7.2,     CALCN176
                        1*, BOTH DAY + NIGHT CUMULATIVE NOISE SUMS = 0; NEF SET TO -99.*)  CALCN177
                         END                                                      CALCN178


          SYMBOLIC REFERENCE MAP (R=3)

ENTRY POINTS      DEF LINE      REFERENCES
   3  CALCNR          1          221         229

VARIABLES       SN  TYPE             RELOCATION
   451  A            REAL                              REFS        68    DEFINED       67
  3746  ANGCOS       REAL        ARRAY     GRID        REFS        25        30        72        89        90
  7666  ANGSIN       REAL        ARRAY     GRID        REFS        25        30        89        90
     0  CHECK        LOGICAL               DEBUG       REFS        40        42       165       171       176       213       214
                                                        215
     1  CNR          REAL                  FT          REFS        22    DEFINED      220       224
   444  CNSDAY       REAL                              REFS       202     2*213      217       220   DEFINED        53       202
   445  CNSNI        REAL                              REFS       204     2*214      217       220   DEFINED        53       204
   465  CNT          REAL                              REFS       136       202       204   DEFINED       133       134       135
    62  CONTOR       REAL        ARRAY     CONTR       REFS        28        29       111
   454  COSPHI       REAL                              REFS        96    DEFINED       90
 22157  CSCF         REAL        ARRAY     TABL1       REFS        44
   452  D            REAL                              REFS        71        72      2*89      2*90       126       127       129
                                                     DEFINED       68        71       126       127
 13562  DARKNO       REAL        ARRAY     FT          REFS        22        23       135
   443  DAY          INTEGER                           REFS        17       134       135       143       144       145       163
                                                        165       170       171       176       200   DEFINED       131
     0  DAYCNR       LOGICAL               CALC        REFS        38        39       134
  7642  DAYNO        REAL        ARRAY     FT          REFS        22        23       134
   462  DBDEG        REAL                              REFS       116    DEFINED      111       113
     0  DBV          REAL        ARRAY     TABL1       REFS        44       143       144       145
    14  DIST         REAL        ARRAY     GRID        REFS        25        30
   467  DMAX         REAL                              REFS       154       165       170       171       176
                                                     DEFINED      143
   470  DMEAN        REAL                              REFS       158       163       165       171       175       176
                                                     DEFINED      144
   471  DMIN         REAL                              REFS       156       163       165       171       176
                                                     DEFINED      145
     1  DRKCNR       LOGICAL               CALC        REFS        36        39       135
   447  DX           REAL                              REFS      2*67        89        90   DEFINED        64
   450  DY           REAL                              REFS      2*67        89        90   DEFINED        65
   473  D1           REAL                              REFS       188       198   DEFINED      185
   474  D10          REAL                              REFS       190   DEFINED      186
     1  EAS          REAL                  GRID        REFS        25
   472  EN           REAL                              REFS       165       171       176       188       190       191     2*198
                                                        202       204   DEFINED      149       163       170       175       191
                                                        196
 17222  ENV          REAL        ARRAY     TABL1       REFS        44       163       175
 23310  FUN1         REAL        ARRAY     TABL1       REFS        44       163
 30060  FUN2         REAL        ARRAY     TABL1       REFS        44       170
 34630  FTWO         REAL        ARRAY     TABL1       REFS        44       163       170
```

```
     SUBROUTINE CALCNR      74/175  OPT=2 ROUND=+-*/               FTN 4.8  508          80/10/30. 12.45.43        PAGE     6

VARIABLES       SN  TYPE              RELOCATION
     0  GROSZ       REAL                   FT          REFS         22
     0  GRNT        INTEGER     ARRAY      GRD         REFS         34        35        102
   461  I           INTEGER                            REFS        108       109     DEFINED        106        107        108
     2  IBOTH       INTEGER                METRIC      REFS         36
   461  ICLL        INTEGER                            REFS        111     DEFINED      109
     0  IDGON       INTEGER     ARRAY      CLNTR       REFS         28        29
     1  IMETER      INTEGER                METRIC      REFS         36
     3  IMP         INTEGER                PARM        REFS         41       180
   456  IROW        INTEGER                            REFS        111     DEFINED      102
   446  IS          INTEGER                            REFS         64        65         72       2*89       2*90        102        116
                                                        134        135     DEFINED       62
   457  ITEMP       INTEGER                            REFS        106       107     DEFINED        105
     0  KARD        INTEGER                IO          REFS         18
     1  KPRINT      INTEGER                IO          REFS         18     I/O REFS     165        171        176        215        228
   464  LD          INTEGER                            REFS        143       144        145        163        165        170        171
                                                        176        202       204     DEFINED        129
   466  M           INTEGER                            REFS        143       144        145        163        165        170        171
                                                        176        202       204     DEFINED        139
     0  METERS      LOGICAL                METRIC      REFS         36        37         71
     0  NRCNR       INTEGER                GRID        REFS         25
     0  NSRCS       INTEGER                SRCS        REFS         21        62
   463  OVERPR      REAL                               REFS        143       144        145     DEFINED        116
     1  PENITE      REAL                   PARM        REFS         41
 12452  PERV        REAL        ARRAY      TABL1       REFS         44       202        204
   455  PHI         REAL                               REFS        105       106        107     DEFINED         96
     2  PPIP        REAL                   PARM        REFS         41       204
     2  RCNR        REAL        ARRAY      GRID        REFS         25        30
     1  REED        LOGICAL                DEBUG       REFS         40        42
     0  RINV1       REAL                   FACTI       REFS         20
     1  RINV2       REAL                   FACTI       REFS         20
     2  RINV3       REAL                   FACTI       REFS         20
    26  SUMWH       REAL        ARRAY      GRID        REFS         25        30        116
   453  SINPHI      REAL                               REFS         96     DEFINED       89
     2  TABRD       LOGICAL                DEBUG       REFS         40        42
   475  TCNSDAY     REAL                               REFS        215     DEFINED      211        213
   476  TCNSNI      REAL                               REFS        215     DEFINED      212        214
     0  THRESH      REAL                   PARM        REFS         41       154        156        158        163        170
     0  X           REAL                   F.P.        REFS         64       228     DEFINED          1
     2  XLOC        REAL        ARRAY      FT          REFS         22        23         64
     0  Y           REAL                   F.P.        REFS         65       226     DEFINED          1
  3722  YLOC        REAL        ARRAY      FT          REFS         22        23         65
        VARIABLES USED AS FILE NAMES, SEE ABOVE

EXTERNALS             TYPE     ARGS          REFERENCES
        ALOG10        REAL      1 LIBRARY     129        213       214       220
        ATAN2         REAL      2 LIBRARY      96
        SQRT          REAL      1 LIBRARY      68

INLINE FUNCTIONS      TYPE     ARGS           DEF LINE  REFERENCES
        IFIX          INTEGER   1  INTRIN                 2*163      170       175

STATEMENT LABELS                DEF LINE    REFERENCES
   244  1                          207          62        131       136
   236  3                          206         139        203
   302  20                         223         217
    70  32                         116         112
    67  34                         113          72
```

STATEMENT LABELS

			DEF LINE	REFERENCES		
202	101		175	154		
170	103		170	157		
211	104		181	154	167	172
231	105		204	20		
221	110		194	190		
223	120		200	180	188	192
375	450	FMT	230	215		
402	501	FMT	231	228		
332	420	FMT	166	165	171	176

LOOPS

LABEL	INDEX	FROM-TO	LENGTH	PROPERTIES	
14	I	62 207	236H	EXT REFS	NOT INNER
116	I	131 207	132H	EXT REFS	NOT INNER
142	K	139 206	102H	EXT REFS	

COMMON BLOCKS

	LENGTH	MEMBERS + RIBS NAME(LENGTH)					
IO	2		KARC (1)	1 KPRINT (1)			
FACTI	3		MINV1 (1)	1 RINV2 (1)		2 RINV3 (1)	
SHOS	1		NSHOS (1)				
FT	4002		GRDSZ (1)	1 CNR (1)		2 XLOC (2000)	
		2002	YLOC (2000)	4002 DAYNO (2000)		6002 DARKNO (2000)	
GRID	6022		GRCNR (1)	1 EAS (1)		2 RCNR (10)	
		12	DIST (10)	22 SOBJM (2000)		2022 ANGCOS (2000)	
		4022	ANGSIN (2000)				
CONTR	800		IDGLM (50)	50 CONTOM (750)			
GUI	2007		GUNT (2000)				
METRIC	3		METERS (1)	1 IMETER (1)		2 IBOTH (1)	
CALC	2		DAYCNR (1)	1 DARKCNR (1)			
DEBUG	3		CHECK (1)	1 REED (1)		2 TABRC (1)	
PARM	4		THRESH (1)	1 PENITE (1)		2 PPIP (1)	
		3	IMP (1)				
TABLE	15346		PHI (5418)	5418 PERV (2408)		7826 ENV (1501)	
		9327	COEF (6 1)	9928 FUN1 (2408)		12336 FUN2 (2408)	
		14744	PIV (302)				

STATISTICS

PROGRAM LENGTH	532B	346
CM LABELED COMMON LENGTH	76227B	31888

60000B CM USED

```
      FUNCTION CFOUR       74/175   OPT=1   FTN 4.6+...

            FUNCTION CFOUR(WEIGHT)
      C     FUNCTION ... [illegible]
      C     THE WEIGHT ... [illegible]
      C     ... THE WEIGHT IS ... [illegible]

      C     IF LESS THAN ... [illegible]
            CFOUR = ...

            IF (WEIGHT.... [illegible]
            IF (WEIGHT.... [illegible]
            IF (WEIGHT.LT.... [illegible] 2
            IF (WEIGHT.LT.... [illegible] 3
      C     FOUR EQUATIONS ARE USED BECAUSE ... [illegible]

            CFOUR=98.3+11.5*ALOG10(WEIGHT)
            RETURN
        3   CFOUR=112.74+7.5*ALOG10(WEIGHT)
            RETURN
        2   CFOUR=112.67+3.33*ALOG10(WEIGHT)
            RETURN
        1   CFOUR=98.0+12.4*ALOG10(WEIGHT)
            RETURN
            END

    SYMBOLIC REFERENCE MAP (R=3)

 ENTRY POINTS    DEF LINE     REFERENCES
     4  CFOUR        1        1?       17       2?       23       26

 VARIABLES     SN  TYPE          RELOCATION
    53  CFOUR      REAL                        DEFINED     8     16     19    ..    25
     0  WEIGHT     REAL               F.P.     REFS       1?     11     ..    ..    ..    ..    22
                                               25      DEFINED    1

 EXTERNALS          TYPE    ARGS      REFERENCES
       ALOG10       REAL      1 LIBRARY    16       19       22       25

 STATEMENT LABELS              DEF LINE   REFERENCES
    33  1                         25         11
    26  2                         22         12
    21  3                         19         13

 STATISTICS
   PROGRAM LENGTH                 54B        44
          60000B CM USED
```

```
      SUBROUTINE PUTXR(XARRAY,MDIM,NDIM,N1,N2,AID1,AID2,DAYS,XREF,IJ)    PUTXR  2
      COMMON/KIO/KARD,KPRINT                                              PUTXR  3
      DIMENSION XARRAY(MDIM,NDIM),AID1(1),AID2(1),XREF(2)
      DATA STARS/3H***/                                                   PUTXR  5
      DO 20 I=1,N1                                                        PUTXR  6
      XREF(1)=AID1(I)                                                     PUTXR  7
      K=2                                                                 PUTXR  8
      DO 10 J=1,N2                                                        PUTXR  9
      IF(IJ.EQ.1)X=XARRAY(I,J)                                            PUTXR 10
      IF(IJ.EQ.2)X=XARRAY(J,I)                                            PUTXR 11
      IF(X.LE.0.)GO TO 10                                                 PUTXR 12
      XREF(K)=AID2(J)                                                     PUTXR 13
      XREF(K+1)=X/DAYS                                                    PUTXR 14
      K=K+2                                                               PUTXR 15
   10 CONTINUE                                                            PUTXR 16
      K=K-1                                                               PUTXR 17
      IF(K.GT.1)GO TO 15                                                  PUTXR 18
      XREF(2)=STARS                                                       PUTXR 19
      K=2                                                                 PUTXR 20
   15 WRITE(KPRINT,50)(XREF(N),N=1,K)                                     PUTXR 21
   20 CONTINUE                                                            PUTXR 22
      RETURN                                                              PUTXR 23
   50 FORMAT(/4X,A3,2X,7(3X,A3,F8.1,2X)/(9X,7(3X,A3,F8.1,2X)))            PUTXR 24
      END                                                                 PUTXR 25
```

```
SYMBOLIC REFERENCE MAP (R=1)

ENTRY POINTS     DEF LINE     REFERENCES
  3  PUTAR           1           22

VARIABLES    SN  TYPE             RELOCATION
     ALL 1        REAL     ARRAY
     AL 2         REAL     ARRAY
     DAYS         REAL
                  INTEGER
                  INTEGER
                  INTEGER
                  INTEGER

     KAR          INTEGER
     KPRINT       INTEGER
     MDIM         INTEGER
                  INTEGER
     NDIM         INTEGER
     N1           INTEGER
     N2           INTEGER
  65 STARS        REAL
     X            REAL
     XARRAY       REAL     ARRAY
     XREF         REAL     ARRAY                                                      13
     VARIABLES USED AS FILE NAMES, SEE ABOVE

     SUBROUTINE PUTAR      74/175  OPT=0 ROUND+-*/          FTN 4.8+508       80/10/30. 12.45.43       PAGE      2

STATEMENT LABELS                 DEF LINE    REFERENCES
   41  10                           15          4          11
   50  15                           20          17
       20                           21          5
   72  50         FMT               23          2

LOOPS  LABEL    INDEX      FROM-TO      LENGTH     PROPERTIES
   17  20       I          5 21         41B                    EXT REFS  NOT INNER
   33  1        J          8 15         1                INSTACK

COMMON BLOCKS   LENGTH      MEMBERS - BIAS NAME(LENGTH)
     IO            2                  KAR     (1)                  1 KPRINT (1)

STATISTICS
  PROGRAM LENGTH                 114B       76
  CM LABELED COMMON LENGTH         2B        2
         60000B CM USED
```

```
      SUBROUTINE FORMA                                                  FORMA  2
C     THIS SUBROUTINE TAKES INPUT FROM THE DATA BASE CONSISTING OF
C     AN ID NUMBER AND LOCATION FOR EACH FIRING POINT,
C     ALSO INCLUDED ARE DEFINITIONS FOR ALL THE NOISE SOURCES AT
C     THAT SITE, GIVING EACH TYPE, NUMBER OF DAY AND NIGHT FIRINGS,
C     MAXIMUM AND MINIMUM CHARGE ZONES, TARGET ID OR AN INDICATION
C     THAT THE SOURCE IS OMNIDIRECTIONAL, AND WEIGHT FOR EACH FIRING;
C     (2) CALCULATES THE SIZE AND LOCATION OF THE NOISE DIRECTION
C     (400.0 FOR OMNIDIRECTIONAL SOURCES, INCLUDING TARGETS), DB CHARGE
C     CORRECTION FACTOR FOR THAT WEIGHT CHARGE AND WEIGHT (DBWT)
C     (FOUND IN ARRAY SCHARG); (3) SORTS A TABLE WITH AN ENTRY FOR EACH
C     UNIQUE NOISE SOURCE, EACH UNIQUE FIRING POINT-TARGET/WEIGHT-CHARGE
C     OR TARGET=CHARGE COMBINATION, A TARGET ALSO BEING A SOURCE OF OMNIDIRECTIONAL
C     NOISE WHEN HIT FOR EACH ENTRY, FORMA ACCUMULATES THE NUMBER OF DAY
C     AND NIGHT FIRINGS WITH THAT CHARGE FOR A (FIRING PT,) OR TGMARG
C     (TARGET) THAT LOCATION.


      COMMON/IO/KARD,KPRINT                                             FORMA  3
      COMMON /SHOTS/ NSHOTS
      COMMON/PT/GRESZ,GMH,XLOC,YLOC,DAYNO,DARKNO                        FORMA  5
      DIMENSION XLOC(2000),YLOC(2000),DARKNO(2000),DAYNO(2000),NAME(100)
      COMMON/GRID/HCNR,EAS,HCNR,DIST,SCHARG,          ANGCOS,ANGSIN
      DIMENSION HCNR(10),DIST(10),SCHARG(2000),
     1ANGCOS(2000),ANGSIN(2000)                                         FORMA  9
      COMMON /CHECK/ CHECK,NEED,TARGD
      COMMON/CONTR/IDGUN,CONTOR
      DIMENSION IDGUN(50),CONTOR(50,15)
      LOGICAL CHECK,NEED,TARGD
      INTEGER GTYPE                                                     FORMA 12
      INTEGER GUNT(2000)                                                FORMA 13
      COMMON /GUN/GUNT                                                  FORMA 14
      LOGICAL FIRST                                                     FORMA 15
      COMMON /TABLE/ ENV(301,3,2) , [illegible] , ENV(1501) ,
     1 CSCF(401) , [illegible]
C     CHARGE ARRAY CONTAINS THE DB RELEASE CHARGES FOR TNT
C     FOR EACH WEAPON.
C     TCHARGE ARRAY CONTAINS THE DB RELEASE CHARGE FOR EACH WEAPON
      [illegible]
C     [illegible]
      [illegible]                                                       FORMA 17
      [illegible]                                                       FORMA 18
C     [illegible]
      [illegible]
C     [illegible]                                                       FORMA 19
C     [illegible]
C     [illegible]
      [illegible]                                                       FORMA 20
      [illegible]                                                       FORMA 21
      [illegible]                                                       FORMA 22
      [illegible]                                                       FORMA 23


      [illegible]
      [illegible]                                                       FORMA 24
```

```
C       TABLE LIMIT
        DATA ITABLIM/5
        DATA INLT/1H /                                                  FORMA 25


        CALL SEC OUT(1000)                                              FORMA 26
C IF TARGET HASN'T ALREADY BEEN READ, CALL REALTB.
        IF (.NOT.TARRD) CALL REALTB
C       INITIALIZE
        TARRD=.FALSE.
        XMAX=YMAX=-9999999.                                             FORMA 27
        XMIN=YMIN= 9999999.0                                            FORMA 28
C       HEADING FOR FORMA
        WRITE(KPRINT,701)                                               FORMA 29
        ASSIGN 211 TO JUMP                                              FORMA 31
        NDX=1                                                           FORMA 32

C READ FORM-A-2 CARD (AVERAGING TECHNIQUE, GROUND CORRECTION,
C AND REWIND INHIBIT FLAG).
        READ(KARD,603) IFUNC,X2,NOWIND                                  FORMA 33
        IF(X2.GT.0.)GHCLHR=X2                                           FORMA 34
        IF(NOWIND .EQ. 0) REWIND IN1

C ASSIGN THE GO-TO STATEMENT FOR EACH TECHNIQUE.
        IF (IFUNC .EQ. 10HMAX       ) GO TO 91                          FORMA 35
        IF (IFUNC .EQ. 10HIAVE      ) GO TO 92                          FORMA 36
        IF (IFUNC .EQ. 10HCAVE      ) GO TO 93                          FORMA 37
        GO TO 99                                                        FORMA 38
   91 ASSIGN 211 TO JUMP                                                FORMA 39
C       NDX IS THE INDEX USED WHEN WRITING THE HEADING FOR
C       THE AVERAGING TECHNIQUE.
        NDX=1                                                           FORMA 40
        GO TO 99                                                        FORMA 41
   92 ASSIGN 212 TO JUMP                                                FORMA 42
        NDX=2                                                           FORMA 43
        GO TO 99                                                        FORMA 44
   93 ASSIGN 213 TO JUMP                                                FORMA 45
        NDX=3                                                           FORMA 46
C WRITE HEADING ABOUT TECHNIQUE USING CHAT ARRAY.
   99  WRITE(KPRINT,609)(CHAT(NDX,I),I=1,3)                             FORMA 47


C       READ GUN TYPE TABLE                                             FORMA 48
        I=1                                                             FORMA 49
   50 READ(IN1,FMT)IGUN(I),ICHARG(I),(CHARGE(J,I),J=1,10),NAME(2*I-1),
      1IFLAG,(CONTUR(I,J),J=1,15)
C       IF END OF FILE STOP
        IF(EOF(IN1).NE.0.) GO TO 520                                    FORMA 51
C       IF *** READ TARGETS
        IF(IFLAG.EQ.1H*)GO TO 60                                        FORMA 52
C       COUNT GUN TYPES
        I=I+1                                                           FORMA 53
C       IF TABLE OVERFLOWS, THEN STOP.
        IF(I.GE.ITABLIM) GO TO 56
```

```
      WRITE(*PRINT,707) I
      STOP
C     READ TARGETS UPON FINDING * IN COL 1 PROCEED WITH SOURCES        FORMA 56
C     NTYPES=NUMBER OF GUN TYPES IN DATA BASE
   60 NTYPES=I                                                         FORMA 57


      I=1                                                              FORMA 58
      NTAR=0                                                           FORMA 59
C     READ TARGETS
  100 READ(IN1,600) IFLAG,J,XLOC(I),YLOC(I),PHCORR                     FORMA 61
C     IF END OF FILE, STOP
      IF(EOF(IN1).NE.0.) GO TO 500                                     FORMA 62
C     COUNT TARGET CARDS
      NTAR=NTAR+1                                                      FORMA 63
C     STORE TARGET ID IN IDTRG ARRAY
      IDTRG(I)=J                                                       FORMA 64
C     IF NO PHCORR GIVEN, IT TAKES A DEFAULT VALUE
      IF(PHCORR.EQ.0.)PHCORR=GHCORR                                    FORMA 65
C     CHECK TO SEE IF TARGET IS WITHIN SPECIFIED BOUNDS
      IF (XLOC(I) .GT. XMAX) XMAX=XLOC(I)                              FORMA 66
      IF (XLOC(I) .LT. XMIN) XMIN=XLOC(I)                              FORMA 67
      IF (YLOC(I) .GT. YMAX) YMAX=YLOC(I)                              FORMA 68
      IF (YLOC(I) .LT. YMIN) YMIN=YLOC(I)                              FORMA 69
C     INITIALIZE ARRAY IDTYPE WITH A NEGATIVE NUMBER TO DENOTE
C     A TARGET.
      IDTYPE(I)=-I                                                     FORMA 70
C     INITIALIZE GUNT TO 0.
      GUNT(I)=0.                                                       FORMA 71
C     INITIALIZE SDRAH TO DENOTE A TARGET NEVER ACCESSED
      SDRAH(I)=-100.                                                   FORMA 72
      HCORR(I)=PHCORR                                                  FORMA 73
C     IF *** READ FIRING POINT CARDS
      IF (IFLAG .EQ. 1H*) GO TO 200                                    FORMA 74
      I=I+1
C     IF OVERFLOW THEN STOP
      IF(I .LE. ITARLIM) GO TO 100
      WRITE(*PRINT,709) I
      STOP


C     COUNT FP CARDS
  200 I=I+1                                                            FORMA 76
C FIRING POINT CARDS.
C     READ FP=1 CARD
      READ(IN1,601) XLOC(I),YLOC(I),PHCORR                             FORMA 77
C     IF END OF FILE, THEN END OF DATA.
      IF(EOF(IN1).NE.0.) GO TO 300                                     FORMA 78
C     IF PHCORR DOESN'T HAVE A VALUE THEN GIVE IT DEFAULT
      IF(PHCORR.EQ.0.)PHCORR=GHCORR                                    FORMA 79
C     CHECK TO SEE IF FP IS WITHIN BOUNDARIES.
      IF (XLOC(I) .GT. XMAX) XMAX=XLOC(I)                              FORMA 80
      IF (XLOC(I) .LT. XMIN) XMIN=XLOC(I)                              FORMA 81
      IF (YLOC(I) .GT. YMAX) YMAX=YLOC(I)                              FORMA 82
      IF (YLOC(I) .LT. YMIN) YMIN=YLOC(I)                              FORMA 83
```

```
      C     IPTR POINTS TO THE BEGINNING OF DEF. CARDS FOR THIS FIRING PT.
            IPTR=1                                                          FORMA 84
            FIRST=.TRUE.                                                    FORMA 85
175
      C FP=2 CARDS
        210 READ(IN1,ERR=300,END=300)IFLAG,ITYPE,DAY,DARK,MIN,MAX,ID,XFLAG,MGT  FORMA 86
            IF(IFLAG(IN1).NE.1.) GO TO 300                                  FORMA 88
180   C     SEE IF GUN TYPE IS IN TABLE
            DO 201 IT=1,NTYPES                                              FORMA 89
            IF(ITYPE.EQ.IGUN(IT))GO TO 202                                  FORMA 90
          201 CONTINUE                                                      FORMA 91
      C     UNDEFINED GUN TYPE                                              FORMA 92
185         WRITE(KPRINT,705)ITYPE,XLOC(IPTR),YLOC(IPTR)                    FORMA 93
            STOP                                                            FORMA 94
      C     SET POINTER TO THE LOCATION IN GUN TYPE TABLE
      C     ITYPE IS NOW SUBSCRIPT INTO GUN TABLE

190         202 ITYPE=IT                                                    FORMA 95
            IF(ID.NE.IABS(MGT))GO TO 204                                    FORMA 96
            IF(XFLAG.EQ.1)GO TO 203                                         FORMA 97
            WRITE(KPRINT,708)XLOC(IPTR),YLOC(IPTR)                          FORMA 98
            STOP                                                            FORMA 99
195   C     HIT FLAG SET

            203 ID=0                                                        FORMA100
            GO TO 207                                                       FORMA101
          204 CONTINUE                                                      FORMA102
200   C     CHECK TO SEE IF THIS IS A LEGITIMATE TARGET
            DO 205 IT=1,NTAR                                                FORMA103
            IF(ID.EQ.IDT(IT)) GO TO 206                                     FORMA104
          205 CONTINUE                                                      FORMA105

205   C     UNDEFINED TARGET                                                FORMA106
            WRITE(KPRINT,709)ID,XLOC(IPTR),YLOC(IPTR)                       FORMA107
            STOP                                                            FORMA108
      C     UPON LOCATING THE TARGET ID, SET A POINTER TO THE LOCATION

210         206 ID=IT                                                       FORMA109
            207 CONTINUE                                                    FORMA110


      C IN THIS SECTION, THE SUBROUTINE JUMPS TO WHICHEVER METHOD OF AVERAGING IT
215   C IS GOING TO USE ACCORDING TO THE CAR DIRECTIVE.  IT CALCULATES THE EQUIVALENT
      C INT WEIGHTS, WITH THE INFORMATION STORED IN ARRAY CHARGE.
            GO TO JUMP,(211,212,213)                                        FORMA111
      C     AVE. TECHNIQUE MAX
          211 CONTINUE                                                      FORMA112
220         IAVE=MAX0(MIN,MAX)                                              FORMA113
            GO TO 214                                                       FORMA114
      C     AVE. TECHNIQUE IAVE
          212 CONTINUE                                                      FORMA115
            AVE=(MIN+MAX)*.5                                                FORMA116
225         IAVE=AVE                                                        FORMA117
            IF((AVE-IAVE).GT..01)IAVE=IAVE+1                                FORMA118
            GO TO 214                                                       FORMA119
      C     AVE. TECHNIQUE CAVE
```

```
                213  CONTINUE                                                  FORMA120
230                  CHAR=(CHARGE(MIN,ITYPE)+CHARGE(MAX,ITYPE))*.5             FORMA121
                     GO TO 216                                                 FORMA122
                214  CONTINUE                                                  FORMA123
                     CHAR=CHARGE(IAVE,ITYPE)                                   FORMA124
                 216 CONTINUE                                                  FORMA125
235                  TCHAR=TCHARG(ITYPE)                                       FORMA126

               C     IF FIRST DEFINITION CARD FOR (XLOC,YLOC) PROCESS          FORMA127
                     IF (FIRST) GO TO 250                                      FORMA128
240
               C     CHECK FOR OLD TARGET (ALSO OMNI)                          FORMA129
                     IF(ID.EQ.0)GO TO 217                                      FORMA130

               C     OLD TARGET SO CHECK TO SEE IF IT IS A DUPLICATE
245                  DO 215 IOLD=IPTR,I                                        FORMA131
               C     SAME TARGET, CHARGE, AND GUN; UPDATE
                     IF (IDTYPE(IOLD).EQ.ID.AND.SOBWM(IOLD).EQ.CHAR.AND.       FORMA132
                    1 GUNT(IOLD).EQ.ITYPE)  GOTO 220                           FORMA133
                 215 CONTINUE                                                  FORMA134
250
                     GO TO 219                                                 FORMA135

               C     SEE IF TARGET IS SAME
                   217 DO 218 IOLD=IPTR,I                                      FORMA136
255            C     OMNI AND GUN
                     IF(SOBWM(IOLD).EQ.CHAR.AND.HEIGHT(IOLD).EQ.HGT)GO TO 220  FORMA137
                   218 CONTINUE                                                FORMA138

260                219 CONTINUE                                                FORMA139
               C     NEW TARGET,NEW CHARGE,OR NEW HEIGHT FOUND FOR (X,Y)       FORMA140
                     I=I+1                                                     FORMA141
                     XLOC(I)=XLOC(IPTR)                                        FORMA142
                     YLOC(I)=YLOC(IPTR)                                        FORMA143
265                  SOBWM(I)=CHAR                                             FORMA144
                     GUNT(I)=ITYPE                                             FORMA145
                     IDTYPE(I)=ID                                              FORMA146
                     DAYNO(I)=DAY                                              FORMA147
                     DARKNO(I)=DARK                                            FORMA148
270                  HEIGHT(I)=HGT                                             FORMA149
                     HCORR(I)=PHCORR                                           FORMA150

               C     NO HIT AT TARGET; ALL DONE
               C     IF OMNI SOURCE DONE                                       FORMA151
275                221 IF (KFLAG .EQ. 1) GO TO 240                            FORMA152

               C     ON FIRST ACCESS TO A TARGET SET TCHARGE                   FORMA153
                     IF (SOBWM(IO) .LT. 0) GO TO 270                           FORMA154
               C     LINK TO TARGETS OF SAME TYPE BUT DIFFERENT CHARGE         FORMA155
280                  IOLD=IO                                                   FORMA156
                 222 IF (SOBWM(IOLD) .EQ. TCHAR.AND.HEIGHT(IOLD).EQ.HGT)GO TO 225  FORMA157
                     IF (IABS(ITYPE) .LT. IABS(GUNT(IOLD))) GO TO 227          FORMA158
                     IOLD=IABS(GUNT(IOLD))                                     FORMA159
                     GO TO 222                                                 FORMA160
285
```

```
                225  CONTINUE                                                  FORMA161
                C    UPDATE OLD CHARGE IF SOURCE                               FORMA162
                     DAYNO(IOLD)=DAYNO(IOLD)+DAY                               FORMA163
                     DARKNO(IOLD)=DARKNO(IOLD)+DARK                            FORMA164
      240            GO TO 240                                                 FORMA165

                C    SET TARGET CHARACTERISTICS FOR THIS FIRING                FORMA166
                C PROCESS FF=2 CARD
                 227  I=I+1                                                     FORMA167
      295            ITYPE(IOLD)=-1                                            FORMA168
                     ITYPE(I)=-1                                               FORMA169
                     SCHAR(I)=TCHAR                                            FORMA170
                     SGNT(I)=ITYPE                                             FORMA171
                     XLOC(I)=XLOC(IOLD)                                        FORMA172
      300            YLOC(I)=YLOC(IOLD)                                        FORMA173
                     HCHAR(I)=HCHAR(IOLD)                                      FORMA174
                     HEIGHT(I)=HGT                                             FORMA175
                     DAYNO(I)=DAY                                              FORMA176
                     DARKNO(I)=DARK                                            FORMA177
      305            GO TO 240                                                 FORMA178

                 220  CONTINUE                                                  FORMA179
                C    UPDATE OLD CHARGE IF SOURCE                               FORMA180
                     DAYNO(IOLD)=DAYNO(IOLD)+DAY                               FORMA181
      310            DARKNO(IOLD)=DARKNO(IOLD)+DARK                            FORMA182
                     GO TO 221                                                 FORMA183

                C    TOO MANY SOURCES
                 240  IF (I .GE. 2000) GO TO 510                               FORMA184
      315       C GO BACK AND READ NEW FF=1 CARD.
                C    STAR * INDICATES END OF FF=2 CARDS FOR A PARTICULAR FIRING PT.
                     IF (IFLAG .EQ. 1H*) GO TO 200                             FORMA185
                     GO TO 210                                                 FORMA186

      320
                C    SET FIRST SOURCE ACCESS                                   FORMA187
                C    STORE ALL INFORMATION OF FF=2 INTO APPROPRIATE ARRAYS
                 250  CONTINUE                                                  FORMA188
                     IDTYPE(I)=IT                                              FORMA189
      325            DAYNO(I)=DAY                                              FORMA190
                     DARKNO(I)=DARK                                            FORMA191
                     SGNT(I)=ITYPE                                             FORMA192
                     SCHAR(I)=CHAR                                             FORMA193
                     HEIGHT(I)=HGT                                             FORMA194
      330            HCHAR(I)=HCHAR                                            FORMA195
                     FIRST=.FALSE.                                             FORMA196
                     GO TO 221                                                 FORMA197

                C    SET FIRST SOURCE ACCESS                                   FORMA198
      335        270  CONTINUE                                                  FORMA199
                     SCHAR(IO)=ICHAR                                           FORMA200
                     HEIGHT(IO)=HGT                                            FORMA201
                     DAYNO(IO)=DAY                                             FORMA202
                     DARKNO(IO)=DARK                                           FORMA203
      340            GO TO 240                                                 FORMA204
```

```
  300  CONTINUE                                                          FORMA205
C      NUMBER OF SOURCES
       NSRCS=I-1                                                         FORMA206
C FORMA-2 DIRECTIVE CARD.
C      NUMBER OF DAYS CARD
       READ(KARD,607) DAYS                                               FORMA229
       IF(EOF(KARD).NE.0.) DAYS=0.0                                      FORMA230
       IF (DAYS .LE. 0.0) DAYS = 1.0                                     FORMA231
       DAYN=0.0                                                          FORMA232
       WRITE(KPRINT,610) DAYS                                            FORMA233
       DARKN=0.0                                                         FORMA234

C      BEGIN TO COMPRESS DATA
       DO 420 I=1,NSRCS                                                  FORMA235
C      IF GUN, CALCULATE ANGLE AT WHICH IT IS POINTING
       IF (IDTYPE(I).GT.0)  GOTO 410                                     FORMA236
C      FOR OMNIDIRECTIONAL FIRINGS, ANGSIN=ANGCOS=999.0
       ANGSIN(I)=999.0                                                   FORMA237
       ANGCOS(I)=999.0                                                   FORMA238
       GOTO 420                                                          FORMA239

  410  CONTINUE                                                          FORMA240
C      CALCULATE THE ANGLE OF FIRING TOWARDS A TARGET
C      CALCULATE DISTANCE FROM GUN TO TARGET
       A = XLOC(IDTYPE(I))-XLOC(I)                                       FORMA241
       B=  YLOC(IDTYPE(I))-YLOC(I)                                       FORMA242
       C = SQRT(A*A+B*B)                                                 FORMA243
       IF (C.NE.0)  GOTO 405                                             FORMA244
       WRITE(KPRINT,710) XLOC(I),YLOC(I)                                 FORMA245
  710   FORMAT(//T15,*.....  WARNING   GUN IS POINTING AT SELF* ,         FORMA246
     1   *...  LOCATION  (*,F8.0,*,*,F8.0,*)*)                           FORMA247
C      MAKE IT OMNIDIRECTIONAL
       ANGSIN(I)=999.0
       ANGCOS(I)=999.0
       GO TO 420
C      GET SINE AND COSINE OF THE ANGLE
  405  ANGSIN(I)=B/C                                                     FORMA248
       ANGCOS(I)=A/C                                                     FORMA249
  420  CONTINUE                                                          FORMA250


       I=0
  400  I=I+1
C      IF SDBAM<0 THEN ITEM IS A TARGET  NEVER ACCESSED
       IF(SDBAM(I).GT.0.)GO TO 401                                       FORMA253
  402  IF(SDBAM(NSRCS).GT.0.)GO TO 403                                   FORMA254
       NSRCS=NSRCS-1                                                     FORMA255
       IF(NSRCS.GT.I)GO TO 402                                           FORMA256
       NSRCS=I-1                                                         FORMA257
       GO TO 404

  403  CONTINUE                                                          FORMA259
C      SWITCH BOTTOM ELEMENTS WITH PRESENT ELEMENTS
       XLOC(I)=XLOC(NSRCS)                                               FORMA260
```

```
      [illegible]                                                   FORMA261
      [illegible]                                                   FORMA262
      [illegible]                                                   FORMA263
      [illegible]                                                   FORMA264
      [illegible]                                                   FORMA265
      [illegible]                                                   FORMA266
      [illegible]                                                   FORMA267
      [illegible]                                                   FORMA268
      HEIGHT(I)=[illegible]                                         FORMA269
      [illegible]                                                   FORMA270
      [illegible]                                                   FORMA271

 401  CONTINUE                                                      FORMA272
C     FOR GREATER THAN ONE DAY CALCULATION DAYNO MUST BE DIVIDED BY FORMA273
C        NUMBER OF DAYS                                             FORMA274
      DAYS [illegible](I)=DAYN[illegible](I)/DAYS                   FORMA275
      DARK [illegible](I)= ARK[illegible](I)/DAYS                   FORMA276
      DAYN=DAYN+[illegible](I)                                      FORMA277
      DARKN=DARK[illegible]+DARK[illegible](I)                      FORMA278
C                                                                   FORMA287

C     CALCULATE THE CORRECTION FACTOR DH OF WEIGHT                  FORMA288
C CHECK FOR DNI: IF SO, FIND DIFF FROM 5 LBS. OF C-4
      IF(ANGSIN(I).EQ.999.0) GO TO 4021

C     IF GUN USED, FIND IT IN THE TABLE
      K=GUNT(I)
      IF(K .GT. 0) GO TO 4020
C     GUN NOT FOUND
      WRITE(KPRINT,4011)
 4011 FORMAT(//T10,*.....ERROR.....GUN NOT FOUND IN GUN TYPE TABLE...JOB
     1 ABORTED.....*//)
      STOP

C     FIND EQUATION PARAMETERS FOR C-4 IN ARRAY CONTUR
 4020 PARAMA=CONTUR(K,1)
      PARAMB=CONTUR(K,2)
      AVG=CONTUR(K,15)
C     CALCULATE INNER RING ENERGY LEVEL
      A=PARAMA+(PARAMB*ALOG10(SDBWH(I)*16))
C     FIND DIFF FROM 5 LBS OF C-4
      DHWT=A-119.0-AVG
      GO TO 4022

C      FOR DNI
 4021 DHWT=CFOUR(SDBWH(I)*16)-119.0


 4022 DHHI=0.0
C CORRECTION FOR GROUND LEVEL OR HELO AND OMNIDIRECTIONAL SOURCES   FORMA292
      IF(ANGSIN(I).NE.999..OR.HEIGHT(I).GT.0.)GO TO 425             FORMA293
      H100=SDBWH(I)/100.                                            FORMA294
      DHHI=HH(LHH(I)+AMIN1(H100,HC HH(I)))                          FORMA295
      IF(HEIGHT(I).E[illegible])GO TO 425                           FORMA296

C     CALCULATE THE DH FOR HELO THE [illegible]                     FORMA297
```

```
      DEPTH = AMS(HEIGHT(I)/SDBWH(I)**.333333333)                 FORMA298
      DEP1 = DEPTH/(-1.362)                                       FORMA299
      DEP2 = (DEPTH+17.46)/(-11.933)                              FORMA300
      IF (DEPTH .GT. 0.0 .A. DEPTH .LE. 2.25) TD=10.0**DEP1       FORMA301
      IF (DEPTH .GT. 2.25) TD=10.0**DEP2                          FORMA302
      DHHI=20.0*ALOG10(TD)+DBHI                                   FORMA303

C     ADD HEIGHT AND HEIGHT/WEAPON CORRECTION FACTORS IN SDBWH
  425 SDBWH(I)=DWHT+DHHI                                          FORMA304

      IF(NSRCS.EQ.I)GO TO 4000
C     ANY MORE SOURCES?
      IF(I.LT.NSRCS) GO TO 400


C "NUMBER OF UNIQUE NOISE SOURCES COUNTED IS.."
 4000 WRITE(KPRINT,700) NSRCS
C "TOTAL PER DAY DAY FIRING IS..."
C "TOTAL PER DAY NIGHT FIRING IS..."
      WRITE(KPRINT,721) DAYN,DARKN                                FORMA307

      ENTRY FORMA2                                                FORMA308
C WRITE ALL OF THIS OUT TO TAPE8.
      WRITE(IOUT) NSRCS,NAYS
      WRITE(IOUT) XLOC                                            FORMA310
      WRITE(IOUT) YLOC                                            FORMA311
      WRITE(IOUT) SDBWH                                           FORMA312
      WRITE(IOUT) ANGSIN                                          FORMA313
      WRITE(IOUT) ANGCOS                                          FORMA314
      WRITE(IOUT) DAYNO                                           FORMA315
      WRITE(IOUT) DARKNO                                          FORMA317
      REWIND IOUT                                                 FORMA319
      REED=.TRUE.                                                 FORMA320

C COMPUTE TIME IN FORMA.
      CALL SECOND(T99)                                            FORMA321
      T99=T99-T100                                                FORMA322
      WRITE(KPRINT,900) T99                                       FORMA323
      RETURN                                                      FORMA324

C "ERROR.. ALL SOURCES ARE TARGETS...NUMBER OF SOURCES COUNTED ARE..."
  500 WRITE(KPRINT,702) I                                         FORMA325
      RETURN                                                      FORMA326
C "ERROR...NUMBER OF SOURCES WILL EXCEED SPACE.  NUMBER OF SOURCES
C NOW IS..."
  510 WRITE(KPRINT,703) I                                         FORMA327
      CALL EXIT                                                   FORMA328
C "ERROR...EOF ENCOUNTERED WHILE PROCESSING GUN TYPE DATA".
  520 WRITE(KPRINT,704)                                           FORMA329
      STOP                                                        FORMA330
```

```
S[illegible] [illegible] FORMA [illegible]        [illegible]        10-30, 12.45.43        PAGE   10

[illegible]                                                                         FORMA334
[illegible]                                                                         FORMA335
[illegible]                                                                         FORMA336
[illegible]
[illegible]                                                                         FORMA348
[illegible]                                                                         FORMA349
[illegible]                                                                         FORMA350
[illegible]                                                                         FORMA351
[illegible]                                                                         FORMA352
[illegible]                                                                         FORMA353
[illegible]                                                                         FORMA354
[illegible]                                                                         FORMA355
[illegible]                                                                         FORMA356
 [illegible] FORMAT(//,T15,*.....     ERROR     ......[illegible]                    FORMA357
    1[illegible] SPACE*,/,T25,*[illegible] IS *,I[illegible]                        FORMA358
 [illegible] FORMAT(//T15,*.....      ERROR     ..... UNDEFINED TARGET ID*,A5,      FORMA359
    1* FOR FIRING PT (*,F8.0,*,*,F8.0,*[illegible]                                  FORMA360
 [illegible]5 FORMAT(//T15,*.....     ERROR     ..... UNDEFINED GUN     ID*,3X,A2FORMA361
    1,*FOR FIRING PT [illegible],F8.0,*,*,F8.0,*)*[illegible]                       FORMA362
 [illegible]6 FORMAT(//T15,*..        ERROR     ......[illegible] ENCOUNTERED WHILE PROCESFORMA363
    1SING GUN TYPE TA[illegible]   DATA*//)                                          FORMA364
 707 FORMAT(//T15,*....       ERROR     .....*,I3,* GUN TYPES EXCEED TABFORMA365
    1LE LIMIT*//)                                                                    FORMA366
 708 FORMAT(//T15,*.....      ERROR     .....BLANK TARGET ID,OMNI/KILL FFORMA367
    1LAG NOT SET FOR SOURCE AT FIRING PT(*,F8.0,*,*,F8.0,*)*//)                     FORMA368
 709  FORMAT(//T15,* ...ERROR.  *,I3,* TARGET TYPES EXCEED TABLE*,
    1* LIMIT...*//)
 721  FORMAT(/,T15,*TOTAL PER DAY DAY EXPLOSIONS IS   *,F10.2,                       FORMA369
    1/,T15,*[illegible]TAL PER DAY NIGHT EXPLOSI[illegible]NS IS *,F10.2/           FORMA370
 9[illegible] FORMAT(//,T15,*.....     TIME FOR FORM-A SUBPROGRAM IS *,F8.3,         FORMA371
    1* SECONDS*)                                                                     FORMA372
    EN[illegible]                                                                    FORMA373


SYMBOLIC REFERENCE MAP (R=3)

ENTRY POINTS     DEF LINE     REFERENCES
    1  FORMA          1
  752  FORMA2       481        [illegible]      [illegible]

VARIABLES      SN  TYPE          RELOCATION
 1662  A            REAL                        REFS     2*371    382     442   DEFINED   369     440
 3746  ANGCOS       REAL     ARRAY     [illegible]  REFS   24      25      43     405      488
                                                DEFINED   363     378     382     405
 7666  ANGSIN       REAL     ARRAY     [illegible]  REFS   24      25      43     404      424     451     487
                                                DEFINED   362     377     381     404
 1652  AVE          REAL                        REFS      225     226   DEFINED   224
 1670  AVG          REAL                        REFS      442   DEFINED   438
 1663  B            REAL                        REFS     2*371    381   DEFINED   370
 1664  C            REAL                        REFS      372     381     382   DEFINED   371
 1653  CHAR         REAL                        REFS      247     256     265     328   DEFINED   230     233
 2045  CHARGE       REAL     ARRAY              REFS       40    2*230    233   DEFINED   105
 3175  CHAT         REAL     ARRAY              REFS       46      99   DEFINED    52
```

```
VARIABLES      SN  TYPE                 RELOCATION
    0  CHECK    LOGICAL             DEMON    REFS        27        30
    1  CWH      REAL                FT       REFS        22
   62  CONTOR   REAL       ARRAY    CONTH    REFS        28        29       436       437       438
                                             DEFINED    135
22157  CSCF     REAL       ARRAY    TABL1    REFS        35
 1641  DARK     REAL                         REFS       269       283       304       310       326       339
                                             DEFINED    178
 1661  DARKN    REAL                         REFS       414       478   DEFINED       354       418
13562  DARKNU   REAL       ARRAY    FT       REFS        22        23       289       310       407       416       418
                                                   47  DEFINED    269       289       304       310       326       339
                                                  417       416
 1640  DAY      REAL                         REFS       268       288       303       309       325       338
                                             DEFINED    176
 1660  DAYN     REAL                         REFS       417       478   DEFINED       352       417
 7642  DAYNU    REAL       ARRAY    FT       REFS        22        23       288       309       406       415       417
                                                  474  DEFINED    268       288       303       309       325       338
                                                  416       415
 1657  DAYS     REAL                         REFS       351       353       415       416       483
                                             DEFINED    349       350       351
 1672  DBHI     REAL                         REFS       462       466   DEFINED       449       453       462
    0  DBW      REAL       ARRAY    TABL1    REFS        35
 1671  DBHT     REAL                         REFS       466   DEFINED       442       446
 1674  DEPTH    REAL                         REFS       458       459     2*460       461   DEFINED       457
 1675  DEP1     REAL                         REFS       460   DEFINED       458
 1676  DEP2     REAL                         REFS       461   DEFINED       459
   14  DIST     REAL       ARRAY    GHI      REFS        24        25
    1  EAS      REAL                GHIL     REFS        24
17222  ENV      REAL       ARRAY    TABL1    REFS        35
 1616  FIRST    LOGICAL                      REFS        34       239   DEFINED       174       331
23310  FUN1     REAL       ARRAY    TABL1    REFS        35
30060  FUN2     REAL       ARRAY    TABL1    REFS        35
34630  FTWO     REAL       ARRAY    TABL1    REFS        35
 1042  GHCURR   REAL                         REFS       134       166   DEFINED        50        80
    0  GRLSZ    REAL                FT       REFS        22
 1615  GTYPE    INTEGER                      REFS        31       182   DEFINED       178
    0  GUNT     INTEGER    ARRAY    GUN      REFS        32        33       247       403       427
                                             DEFINED    144       266       298       327       403
 7126  HCURR    REAL       ARRAY             REFS        49       301       409     2*453   DEFINED        50       147
                                                  271       301       330       409
 3206  HEIGHT   REAL       ARRAY             REFS        49       256       281       408       451       454       457
                                             DEFINED     50       270       302       329       337       408
 1646  HGT      REAL                         REFS       256       270       281       302       329       337
                                             DEFINED    178
 1673  HIJO     REAL                         REFS       453   DEFINED       452
 1631  I        INTEGER                      REFS        99     5*105       112       114       115       119     2*126
                                                  132     2*136     2*137     2*138     2*139     2*142       144       146
                                                  147       150       152       153       159     2*162     2*168     2*169
                                                2*170     2*171       173       245       254       262       263       264
                                                  265       266       267       268       269       270       271       294
                                                  295     2*296       297       298       299       300       301       302
                                                  303       304       314       324       325       326       327       328
                                                  329       330       346       360       362       363     2*369     2*370
                                                2*372       377       378       381       382       388       390       393
                                                  394       399       400       401       402       403       404       405
                                                  406       407       408       409     2*415     2*416       417       418
                                                  424       427       440       446     2*451       452     2*453       454
                                                2*457       466       468       470       503       507
```

```
VARIABLES   SN  TYPE            RELOCATION
                                           DEFINED    94     [illegible] 112    [illegible] [illegible] 154    262
                                              294    354     387     388
 1651  IAVE      INTEGER                   REFS     2*226    233   DEFINED   [illegible] [illegible] 226
 1046  IBUNK     INTEGER                   REFS      191   DEFINED     60
 1044  IO        INTEGER                   REFS      191     212     216   [illegible]  247    267    278
                                              280    294     300     301   [illegible]  336    337    338
                                              334  DEFINED   178     191   [illegible]
    0  IDGUS     INTEGER   ARRAY  [illegible] REFS    28      29     162   DEFINED   [illegible]
 1656  IOPLO     INTEGER                   REFS     2*281   2*282    283   2*288   2*289   295
                                           DEFINED   280     283
 3113  IOTRK     INTEGER   ARRAY           REFS       46     272   DEFINED   [illegible]
 3746  IOTYPE    INTEGER   ARRAY  GRID     REFS       42      43     247   [illegible]  283    362    369
                                              377    402   DEFINED   142   [illegible]  295    296    324
                                              402
 1633  IFLAG     INTEGER                   REFS      110     149     317   DEFINED   125    126    178
 1626  IFUNC     INTEGER                   REFS       84      85      86   DEFINED    79
 1043  INI       INTEGER                   REFS      108     125     164     179   DEFINED   57
                                           I/O REFS   81     105     126     162     178
 1655  IOLD      INTEGER                   REFS     3*247   2*256   2*309   2*310  DEFINED  245    254
 1044  IOUT      INTEGER                   DEFINED    57   I/O REFS  483     484     485    486    487
                                              488    489     490     491
 1637  IPTR      INTEGER                   REFS     2*185   2*193   2*206    245     254    263    264
                                           DEFINED   173
 1647  IT        INTEGER                   REFS      182     193     202     210   DEFINED  181    201
 1045  ITABLIM   INTEGER                   REFS      114     152   DEFINED    54
 1650  ITYPE     INTEGER                   REFS      185   2*230     233     235     247    266    298
                                              327  DEFINED   190
 1632  J         INTEGER                   REFS     2*105    132   DEFINED   2*115    126
 1624  JUMP      INTEGER                   REFS      217   DEFINED    73      85      93     96
 1665  K         INTEGER                   REFS      426     436     437     438   DEFINED  427
    0  KARD      INTEGER          IO       REFS       20     355   I/O REFS   74     349
 1645  KFLAG     INTEGER                   REFS      192     275   DEFINED   178
    1  KPRINT    INTEGER          IO       REFS       20   I/O REFS   72      94     115    153    185
                                              193    206     353     373     430     475    478    497
                                              503    507     510
 1643  MAX       INTEGER                   REFS      220     224     230   DEFINED   178
 1642  MIN       INTEGER                   REFS      220     224     230   DEFINED   178
 1701  NAME      INTEGER   ARRAY           REFS       23   DEFINED   105
 1625  NDX       INTEGER                   REFS       99   DEFINED    74      91      94     97
 7666  NUPER     INTEGER   ARRAY  GRID     REFS       42      43
 1630  NUWIND    INTEGER                   REFS       81   DEFINED    79
    0  NRCVR     INTEGER          GRID     REFS       24
    0  NSRCS     INTEGER          SRCS     REFS       21     358     391     392     393    399    400
                                              401    402     403     404     405     406    407    408
                                              409    410     468     470     475     483
                                           DEFINED   346     392     394     410
 1635  NTAR      INTEGER                   REFS      130     201   DEFINED   124     130
 1634  NTYPES    INTEGER                   REFS      181   DEFINED   119
 1666  PARAMA    REAL                      REFS      440   DEFINED   436
 1667  PARAMB    REAL                      REFS      440   DEFINED   437
12452  PERV      REAL      ARRAY  TABLE1   REFS       35
 1636  PHCURR    REAL                      REFS      134     147     166     271     330
                                           DEFINED   126     134     162     166
    2  RCVR      REAL      ARRAY  GRID     REFS       24      25
    1  REFS      LOGICAL          DEBUG    REFS       27      30   DEFINED   492
   26  SCHWM     REAL      ARRAY  GRID     REFS       24      25     247     256     278    281    390
                                              391    401     440     446     452     457    486
```

VARIABLES	SN	TYPE	RELOCATION								
2	IANMG	LOGICAL		DEFINED	146	265	297	328	336	401	466
1654	ICMAG	REAL		REFS	27	30	68	DEFINED	68		
3031	ICMANG	REAL	ARRAY	REFS	281	297	336	DEFINED	235		
1677	IG	REAL		REFS	42	235	DEFINED	105			
1617	IIGG	REAL		REFS	462	DEFINED	461	461			
1700	IXX	REAL		REFS	64	496					
2	XLOC	REAL	ARRAY FT	REFS	493	496	497	DEFINED	496		
				REFS	22	23	2*136	2*137	2*168	2*169	185
					193	206	263	299	2*369	373	399	484
				DEFINED	126	162	263	299	399		
1620	XMAX	REAL		REFS	136	168	DEFINED	69	136	168	
1622	XMIN	REAL		REFS	137	169	DEFINED	70	137	169	
1627	X2	REAL		REFS	2*140	DEFINED	79				
3722	YLOC	REAL	ARRAY FT	REFS	22	23	2*138	2*139	2*170	2*171	185
					193	206	264	300	2*370	373	400	485
				DEFINED	126	162	264	300	400		
1621	YMAX	REAL		REFS	138	170	DEFINED	69	138	170	
1623	YMIN	REAL		REFS	139	171	DEFINED	70	139	171	

VARIABLES USED AS FILE NAMES, SEE ABOVE

EXTERNALS	TYPE	ARGS		REFERENCES				
ALOG10	REAL	1	LIBRARY	440	462			
CPUDUR	REAL	1		446				
EOF	REAL	1		10*	12M	164	174	35*
EXIT		0		5CM				
HEADTH		0		66				
SECOND		1		64	495			
SQRT	REAL	1	LIBRARY	371				

INLINE FUNCTIONS	TYPE	ARGS		DEF LINE	REFERENCES	
ABS	REAL	1	INTRIN		457	
AMIN1	REAL	0	INTRIN		453	
IABS	INTEGER	1	INTRIN		282	283
MAX0	INTEGER	0	INTRIN		220	

STATEMENT LABELS		DEF LINE	REFERENCES		
73	50	105	114		
135	60	119	117		
35	91	88	84		
40	92	93	85		
43	93	96	86		
45	94	99	87	92	95
142	1	124	152		
207	230	156	149	317	
0	231	183	181		
264	202	194	142		
300	203	197	148		
302	204	199	191		
0	205	202	201		
316	206	217	210		
317	207	211	194		
243	21	174	91		
321	211	219	73	88	217
325	212	223	98	217	
336	213	229	96	217	
345	214	236	221	227	
0	215	244	245		

```
STATEMENT LABELS                 DEF LINE   REFERENCES
  351  216                          234        231
  372  217                          254        242
  [illegible]  218                  257        254
  [illegible]  219                  [illegible]  251
  407  22[illegible]                3[illegible]  247      [illegible]
  423  221                          27[illegible]  311      35[illegible]
  432  222                          24[illegible]  [illegible]4
  4[illegible]2  225                26[illegible]  281
  4[illegible]7  227                294        282
  474  24[illegible]                314        [illegible]75     [illegible]   3[illegible]   [illegible]
  5[illegible]1  25[illegible]      323        239
  514  27[illegible]                33[illegible]  278
  523  3[illegible]                 3[illegible]  164      17[illegible]
  6[illegible]2  4[illegible]       388        47[illegible]
  637  4[illegible]1                412        39[illegible]
  61[illegible]  4[illegible]2      391        393
  615  4[illegible]3                347        391
  572  4[illegible]                 381        372
  552  41[illegible]                366        36[illegible]
  5[illegible]6  42[illegible]      383        358      36[illegible]   379
  737  425                          466        451      454
 1[illegible]1[illegible]  5[illegible]  5[illegible]3  128
 1013  510                          5[illegible]7  314
 1017  52[illegible]                51[illegible]  109
 1321  60[illegible]  FMT           51[illegible]  126
 1325  601  FMT                     515        162
 1330  602  FMT                     516        178
 1335  603  FMT                     517         79
 1340  604  FMT                     51[illegible]  105
 1345  6[illegible]7  FMT           51[illegible]  349
 1347  609  FMT                     52[illegible]   99
 1357  61[illegible]  FMT           523        353
 1366  70[illegible]  FMT           524        475
 1375  701  FMT                     52[illegible]   72
 1404  702  FMT                     526        503
 1420  703  FMT                     526        507
 1435  704  FMT                     53[illegible]  206
 1451  705  FMT                     532        185
 1465  706  FMT                     534        510
 1477  707  FMT                     536        115
 1510  708  FMT                     53[illegible]  193
 1526  709  FMT                     54[illegible]  153
 1210  710  FMT                     374        373
 1537  721  FMT                     542        478
 1552  90[illegible]  FMT           544        497
  745  4000                         475        395      468
 1226  4011  FMT                    431        43[illegible]
  655  4020                         436        428
  671  4021                         446        424
  677  4022                         44[illegible]  443

LOOPS  LABEL  INDEX  FROM-TO    LENGTH  PROPERTIES
   53         I       99  99     1[illegible]8            EXT REFS
  110         J      105 105     1[illegible]8            EXT REFS
  253  201    IT     181 183      28     INSTACK   EXITS
  305  205    IT     201 203      26     INSTACK   EXITS
  362  215    IOLO   245 249      78     INSTACK   EXITS
```

```
ERRORS  LABEL    INDEX     FROM-TO    LENGTH    PROPERTIES
  377   21H      I  CC     254 257      5       INSTACK     EXITS
  545   42       I         358 383      34                EXT REFS

COMMON BLOCKS   LENGTH     MEMBERS - BIAS NAME(LENGTH)
       I             2            0 KARD    (1)              1 KPRINT (1)
       NHCS          1            0 NSHCS   (1)
       FT         8002            0 GHTSZ   (1)              1 CNR     (1)               2 XLOC    (2000)
                               2002 YLOC    (2000)        4002 DATNO   (2000)         6002 DARKNO  (2000)
       SHIP       6022            0 NHCNR   (1)              1 EAS     (1)               2 RCNR    (10)
                                 12 LIST    (10)            22 SOHWH   (2000)         2022 ANGCOS  (2000)
                               4022 ANGSIN  (2000)
       LEMOS         3            0 CHECK   (1)              1 FEED    (1)               2 TABRD   (1)
       CONTR       800            0 IDGEN   (50)            50 CONTOR  (750)
       GUN        2000            0 GUNT    (2000)
       TABLE     15346            0 FAV     (5416)        5418 PERV    (2408)         7826 ENV     (1501)
                               9327 CSCF    (601)         9928 FUN1    (2408)        12336 FUN2    (2408)
                              14744 FTA     (392)

EQUIV CLASSES   LENGTH     MEMBERS - BIAS NAME(LENGTH)
NRCNR   ANGCOS    2000            0 IDTYPE  (2000)
NHCNR   ANGSIN    2000            0 NOPER   (2000)

STATISTICS
  PROGRAM LENGTH                 13046B     5670
  CM LABELED COMMON LENGTH       76204B    31876
       60000B CM USED
```

```
      [illegible]                                                       LOC   2
      [illegible]
      [illegible]
C * [illegible]  TARGET="A" FIRING [illegible]
      [illegible]                                                       LOC   3
      [illegible] FOR FIRING
      [illegible] ARGS.
      [illegible]                                                       LOC   4
C     FLAGS FOR [illegible] INPUT
      LOGICAL TARG,FPT,NAM,LOC                                          LOC   5
C     ARRAY [illegible] DEFAULT VALUES
      DIMENSION STARS(2)                                                LOC   6
C     INI=TAPE 7   IOUT=TAPE 3
      DATA INI/7/,IOUT/3/                                               LOC   7
      STARS(1)=2H      $ STARS(2)=2H                                    LOC   8
      CPTIME=SECOND(CP)                                                 LOC   9


C WRITE TITLE OF MODULE "LOCATOR"
      WRITE(KPRINT,25)                                                  LOC  10
 25   FORMAT(1H1,20X,*.....       LOCATOR       .....*)                 LOC  11

      REWIND IOUT                                                       LOC  12

C PHASE 4 PLOTS SPECIAL LINES AND TEXT ON THE MAP AND DRAWS A BORDER ON THE MAP.
      WRITE(IOUT,10)                                                    LOC  13
 10   FORMAT(4HPHS4)                                                    LOC  14
C                                                                       LOC  15
C          READ SPECIFICATION CARD                                      LOC  16
C                                                                       LOC  17
C THIS IS INFORMATION FOUND ONTHE LOC-2 CARD,
      READ(KARD,20)CHOICE,NAME,LOCATE   ,SIZE,ANGLE                     LOC  18
 20   FORMAT(3A10,2F10.0)                                               LOC  19
C                                                                       LOC  20
C          SPECIFICATION CARD CHECKS                                    LOC  21
C                                                                       LOC  22
C THIS IF-LOOP CHECKS TO SEE IF A DEFAULT VALUE IS NEEDED FOR THE VARIABLE
C SIZE OR IF THERE IS ONE GIVEN BY THE USER,
      IF(SIZE.NE.0) GO TO 27                                            LOC  23
      SIZE=.14                                                          LOC  24
      STARS(1)=2H**                                                     LOC  25


C THIS IF-LOOP CHECKS FOR THE DEFAULT FOR THE VARIABLE ANGLE
C IF A DEFAULT VALUE IS USED, IT IS DENOTED IN THE OUTPUT BY STARS.
 27   IF(ANGLE.NE.0.) GO TO 29                                          LOC  26
      ANGLE=0.0                                                         LOC  27
      STARS(2)=2H**                                                     LOC  28


C TARG,FPT,NAM, AND LOC ARE ALL LOGICAL VALUE TO SEE WHICH VALUES WERE
C SUBMITTED BY THE USER FOR THE TARGET, FIRING PT., NAME, AND LOCATE VARIABLES.
 29   TARG=((CHOICE.EQ.3HALL).OR.(CHOICE.EQ.6HTARGET))                  LOC  29
      FPT=((CHOICE.EQ.3HALL).OR.(CHOICE.EQ.6HFIRING))                   LOC  30
      NAM=(NAME.EQ.4HNAME)                                              LOC  31
      LOC=(LOCATE   .EQ.8HLOCATION)                                     LOC  32
```

```
      C                                                              LOC   33
      C HEADING," OPTIONS REQUESTED."
      C          PRINT REPORT HEADING AND OPTIONS REQUESTED          LOC   34
      C                                                              LOC   35
            WRITE(KPRINT,30)                                         LOC   36
       30   FORMAT(1H0,23X,*..OPTIONS REQUESTED..*/)                  LOC   37
            IF((TARG).OR.(FPT)) GO TO 34                             LOC   38
      C MESSAGE"NO OPTIONS REQUESTED-JOB ENDED"
            WRITE(KPRINT,32)                                         LOC   39
       32   FORMAT(1H0,23X,*NO OPTIONS REQUESTED - JOB ENDED*)       LOC   40
            REWIND IOUT                                              LOC   41
            GO TO 250                                                LOC   42

      C THIS IF-STATEMENTS WRITE THE OPTIONS
       34   IF(TARG)WRITE(KPRINT,35)10HTARGET                        LOC   43
            IF(FPT) WRITE(KPRINT,35) 20HFIRING POINT                 LOC   44
            IF(NAM)  WRITE(KPRINT,35)10HNAME                         LOC   45
            IF(LOC)WRITE(KPRINT,35) 10HLOCATION                      LOC   46
       35   FORMAT(25X,2A10)                                         LOC   47
            WRITE(KPRINT,40) SIZE,STARS(1),ANGLE,STARS(2)            LOC   48
       40   FORMAT(1H0,25X,5HSIZE=,F5.2,A2,5X,6HANGLE=,F6.2,A2)      LOC   49


      C MESSAGE "STARS INDICATE DEFAULT VALUES"
            WRITE(KPRINT,45)                                         LOC   50
       45   FORMAT(26X,*STARS INDICATE DEFAULT VALUES*)              LOC   51
            WRITE(KPRINT,36)                                         LOC   52
       36   FORMAT(1H0,24X,7HID CODE,8X,8HX COORD.,5X,8HY COORD.,5X,7HG COMP.,
           15X,6HBOUNDS)
      C     TEST IF BOUNDS SET?  NO THEN STOP
            IF(BDS) GO TO 90
            WRITE(KPRINT,85)
      85    FORMAT(*  ...NO BOUNDS CALL BEFORE LOCATOR...*)
            STOP
      C                                                              LOC   54
      C          READ GUN TYPES                                      LOC   55
      C                                                              LOC   56
      90    REWIND INI                                               LOC   57
       100  READ(INI,110)STAR                                        LOC   58
       110  FORMAT(A1)                                               LOC   59
            IF(EOF(INI).NE.0) GO TO 260                              LOC   60
            IF(STAR.NE.1H*) GO TO 100                                LOC   61
      C IF WE HAVE A TARGET,PLACE AN "X" ON THE POINT HEADING-"TARGET DATA"
      C                                                              LOC   62
      C          READ TARGET CARDS                                   LOC   63
            IF(TARG) GO TO 140                                       LOC   64
      C     FLUSH TARGET CARDS
       120  READ(INI,130) STAR                                       LOC   65
       130  FORMAT(A1)                                               LOC   66
            IF(EOF(INI).NE.0) GO TO 260
            IF(STAR.NE.1H*) GO TO 120                                LOC   67
            WRITE(KPRINT,135)                                        LOC   68
       135  FORMAT(1H0,23X,*TARGETS NOT REQUESTED*)                  LOC   69
            GO TO 205                                                LOC   70
      C                                                              LOC   71
      C          TARGET/FIRING POINT CARDS PRODUCED                  LOC   72
      C                                                              LOC   73
```

```
  140  POINT=1H*                                                       LOC   74
       WRITE(KPRINT,141)                                               LOC   75
  141  FORMAT(     11X,*..TARGET DATA..*/)                             LOC   76
  145  READ(IN1,150) EOI,IDCODE,XPOS,YPOS,SCORR                        LOC   77
  150  FORMAT(A1,1X,A3,1X,3F6.0)                                       LOC   78
       IF(EOF(IN1).NE.0) GO TO 250                                     LOC   79
C      TEST IF POINT OUT OF BOUNDS  IF S  FLAG
       IF (XPOS.LT.XMIN.OR.XPOS.GT.XMAX)  GOTO 164
       IF (YPOS.LT.YMIN.OR.YPOS.GT.YMAX)  GOTO  164
       WRITE(KPRINT,160)IDCODE,XPOS,YPOS,SCORR                         LOC   81
  160  FORMAT(26X,A3,10X,F8.0,5X,F8.0,5X,F8.0)                         LOC   82

       GOTO  163
C      OUT OF BOUNDS DATA,  TELL USER
  164  WRITE(KPRINT,161)  IDCODE,XPOS,YPOS,SCORR
  161  FORMAT(26X,A3,5X,3(5X,F8.0),7X,3HOUT)
       GOTO 200

C      L IS AN INPUT VARIABLE TO NASAPLOT
  163  L=1                                                             LOC   83
       IF(NAM) GO TO 165                                               LOC   84
       IDCODE=3H                                                       LOC   85
       GO TO 166                                                       LOC   86
  165  L=5                                                             LOC   87
  166  IF(LOC) GO TO 180                                               LOC   88
       WRITE(IOUT,170)XPOS,YPOS,SIZE,ANGLE,POINT,IDCODE,L              LOC   89
  170  FORMAT(4HTEXT,2X,2F8.0,8X,2F8.3,2H 1,A1,1X,A3,19X,I2)           LOC   90
       GO TO 200                                                       LOC   91
  180  L=23                                                            LOC   92
       WRITE(IOUT,190)XPOS,YPOS,SIZE,ANGLE,POINT,IDCODE,XPOS,YPOS,L    LOC   93
  190  FORMAT(4HTEXT,2X,2F8.0,8X,2F8.3,2H 1,A1,1X,A3,1X,F8.0,1X,F8.0,1X,I
      12)

C      ARE WE DOING FIRING PTS?
  200  IF(POINT.EQ.1H0) GO TO 206
C      DOING TARGETS--IS IS THE LAST ONE?
       IF(EOI.NE.1H*) GO TO 145
C                                                                      LOC   96
C          FIRING POINT CARDS                                          LOC   97
C                                                                      LOC   98
C IF WE HAVE A FIRING POINT PLACE AN "0" ON THE POINT HEADING-FIRING POINT
C DATA"
  205  IF(.NOT.FPT) GO TO 240                                          LOC   99
       WRITE(KPRINT,210)                                               LOC  101
  210  FORMAT(     11X,*..FIRING POINT DATA..*/)                       LOC  102
       POINT=1H0                                                       LOC  103
       GO TO 145                                                       LOC  104

C      FLUSH OUT FIRING POINT INFO FOR THIS FIRING PT.
  206  READ(IN1,130) STAR
       IF(EOF(IN1).NE.0) GO TO 260
       IF(STAR.EQ.1H*) GO TO 145
       GO TO 206
  240  WRITE(KPRINT,245)                                               LOC  105
  245  FORMAT(1H0,23X,*FIRING POINTS NOT REQUESTED*)                   LOC  106
```

```
      C THIS COMPUTES THE TIME IN LOCATR'S PROGRAM
       250  CPTIME1=SECOND(CP)                                               LOC  107
            CPTIME1=CPTIME1-CPTIME                                           LOC  108

            WRITE(KPRINT,251)CPTIME1                                         LOC  109
       251  FORMAT(1H0,20X,*......*,F8.3,* SECONDS IN LOCATOR  .....*)       LOC  110
            GO TO 300                                                        LOC  111
      C "ERROR--PREMATURE EOF ON DATA BASE FILE TAPE 3 NOT CREATED"
       260  WRITE(KPRINT,261)                                                LOC  112
       261  FORMAT(1H0,20X,35H** PREMATURE EOF ON DATA BASE FILE  ,          LOC  113
           1                                      22H**TAPE3 NOT CREATE      LOC  114
           1D **)                                                            LOC  115
            REWIND INPUT                                                     LOC  116
            STOP                                                             LOC  117
       300  RETURN                                                           LOC  118
            END                                                              LOC  119
```

SYMBOLIC REFERENCE MAP (R=3)

```
ENTRY POINTS      DEF LINE      REFERENCES
    1  LOCATR         1           186

VARIABLES     SN   TYPE            RELOCATION
  651  ANGLE       REAL                              REFS       47      77     140     144   DEFINED    33    48
    4  BDS         LOGICAL               BOUND       REFS        2       3      88
  645  CHOICE      REAL                              REFS     2*54    2*55  DEFINED     33
  644  CP          REAL                              REFS       16     173
  643  CPTIME      REAL                              REFS      174  DEFINED     16
  660  CPTIME1     REAL                              REFS      174     176  DEFINED    173     174
  652  EOI         REAL                              REFS      151  DEFINED    118
  640  FPT         LOGICAL                           REFS       10      64      73     157   DEFINED    55
  656  GCURR       REAL                              REFS      124     129  DEFINED    118
  653  ICOUNE      INTEGER                           REFS      124     129     140     144   DEFINED   118   136
  231  INI         INTEGER                           REFS       98     107     120     165   DEFINED    14
                                                   I/O REFS     95      96     105     118     164
  232  INPUT       INTEGER                         DEFINED      14  I/O REFS     24      27      68   140   144
                                                      184
    0  KARD        INTEGER               IO          REFS        5  I/O REFS     33
    1  KPRINT      INTEGER               IO          REFS        5  I/O REFS     21      62      66    72    73
                                                       74       75      77      82      84      89   109   116
                                                      124      129     158     165     176     180
  657  L           INTEGER                           REFS      140     144  DEFINED    134     138   143
  642  LOC         LOGICAL                           REFS       10      75     139  DEFINED     57
  647  LOCATE      INTEGER                           REFS       57  DEFINED     33
  641  NAM         LOGICAL                           REFS       10      74     135  DEFINED     56
  646  NAME        INTEGER                           REFS       56  DEFINED     33
  636  POINT       INTEGER                           REFS        8     140     144     149   DEFINED   115   160
  650  SIZE        REAL                              REFS       45      77     140     144   DEFINED    33    41
  635  STAR        INTEGER                           REFS        8      99     108     166   DEFINED    96   105
                                                      184
  661  STARS       REAL         ARRAY                REFS       12    2*77  DEFINED   2*15      42    49
  637  TARG        LOGICAL                           REFS       12      64      72     103   DEFINED    54
    2  XMAX        REAL                  BOUND       REFS        2     122
```

```
VARIABLES     SN  TYPE           RELOCATION
    0  XMIN       REAL            3 [illegible]     REFS        2       122
  654  XPOS       REAL                              REFS    2*122       124     129     14[illegible]   2*144
                                                 DEFINED      118
    3  YMAX       REAL            BOUND             REFS        2       123
    1  YMIN       REAL            BOUND             REFS        2       123
  655  YPOS       REAL                              REFS    2*123       124     129     14[illegible]   2*144
                                                 DEFINED      118
       VARIABLES USED AS FILE NAMES, SEE ABOVE

EXTERNALS          TYPE   ARGS      REFERENCES
       EOF         REAL    1            9[illegible]    107     120     165
       SECOND      REAL    1            15     173

STATEMENT LABELS               DEF LINE   REFERENCES
  247  10       FMT              28        27
  262  20       FMT              34        33
  236  25       FMT              22        21
   22  27                        47        40
   26  29                        54        47
  270  30       FMT              63        62
  300  32       FMT              67        66
   54  34                        72        64
  327  35       FMT              76        72       73      74      75
  360  36       FMT              85        84
  340  40       FMT              78        77
  350  45       FMT              83        82
  102  80                        95        86
  373  85       FMT              90        89
  104  100                       96        99
  407  110      FMT              97        96
  114  120                      105       108
  416  130      FMT             106       105      164
  423  135      FMT             110       109
  125  140                      115       103
  433  141      FMT             117       116
  130  145                      118       151      161     166
  450  150      FMT             119       118
  462  160      FMT             125       124
  475  161      FMT             130       129
  153  163                      134       127
  150  164                      129       122      123
  157  165                      138       135
  160  166                      139       137
  513  170      FMT             141       140
  164  180                      143       139
  535  190      FMT             145       144
  167  200                      149       131      142
  172  205                      157       111
  177  206                      164       149      167
  547  210      FMT             159       158
  206  240                      16[illegible]       157
  564  245      FMT             169       169
  210  250                      173        69      120
  575  251      FMT             177       176
  216  260                      18[illegible]        98      107     165
  607  261      FMT             181       180
  223  300                      186       178
```

COMMON BLOCKS LENGTH MEMBERS - BIAS NAME(LENGTH)

BLOCK	5	0	XMIN (1)	1	YMIN (1)	2	XMAX (1)
		3	YMAX (1)	4	BDS (1)		
IO	2	0	KARG (1)	1	KPRINT (1)		

STATISTICS

PROGRAM LENGTH	674B	444
CM LABELED COMMON LENGTH	76	7

60000B CM USED

```
      SUBROUTINE MAP                                                    MAP    2
C                                                                       MAP    3
C     THIS SUBROUTINE PRODUCES A MAP OF THE POINTS                      MAP    4
C                                                                       MAP    5
      COMMON/PT/IXPT
      COMMON/IO/KARD,KPRINT                                             MAP    7
      COMMON/GRID/GTXPT
      DIMENSION IDGUN(50),CHARGE(11,50),IDTRG(50),IDPT(300),NAME1(50),
     1 TCORDS(5,2),FCORDS(300,2),IXPT(50,300),TRGT(50,50),GTXPT(50,
     2  300)                                                           MAP   11
      DIMENSION ADGUN(50),ADTRG(50),ADPT(300),XREF(2500),NAME2(50)
      EQUIVALENCE (ADGUN(1),IDGUN(1)),(ADTRG(1),IDTRG(1)),(ADPT(1),    MAP   13
     1   IDPT(1))                                                       MAP   14
      DIMENSION NCOL(80),NTCOLS(50),NBCOLS(49)
      DIMENSION CUNTUR(50,15)
      DIMENSION IN(3)                                                   MAP   21
      COMMON/DEBUG/CHECK,REED                                           MAP   22
      LOGICAL CHECK,REED                                                MAP   23
      LOGICAL NEOF                                                      MAP   24
      LOGICAL PRNT                                                      MAP   25
      LOGICAL OVRFL,IMAGE,POS                                           MAP   26
      LOGICAL DUP                                                       MAP   27
      LOGICAL LOI                                                       MAP   28
C XREF FLAGS)0=NO TABLE;1= TABLE;2= INCOMPLETE TABLE DUE TO OVERFLOW    MAP   29
      INTEGER XTF,XTG,XGF                                               MAP   30
      INTEGER HOLLTA                                                    MAP   31
      DATA IBLNK/1H /                                                   MAP   32
      DATA IN/7,75,77/,MFILES/3/                                        MAP   33
C TABLE LIMITS,ETC.                                                     MAP   34
      DATA NUMTRG/50/,NUMPT/300/,NUMFNT/300/,NUMLIN/50/,NUMCHG/11/,     MAP   35
     1     BIGCHG/50./                                                  MAP   36
C BLANK CARD COLUMN MAPS                                                MAP   37
      DATA NTCOLS/2,6,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,  MAP   38
     1  41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,   56,57,58,59,60,  MAP   39
     2  61,62,63,64,65,66,67,68,69,70,71,72/,                          MAP   40
     3NBCOLS/2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,                MAP   41
     1  41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,   56,57,58,59,60,  MAP   42
     2  61,62,63,64,65,66,67,68,69,70,71,72/                           MAP   43
      CALL SECOND(TIC0)                                                 MAP   44
C SET TAPE9 READ FLAG--WILL HAVE TO BE CREATED OR REREAD FOR FURTHER     MAP   45
C   PROCESSING, SINCE MAP USES ITS COMMON STORAGE                       MAP   46
      REED=.FALSE.                                                      MAP   47
      NEOF=.FALSE.                                                      MAP   48
      WRITE(KPRINT,700)                                                 MAP   49
      NCARDS=0                                                          MAP   50
      NT=0                                                              MAP   51
      NS=0                                                              MAP   52
      NG=0                                                              MAP   53
      LINES=0                                                           MAP   54
      NERR=0                                                            MAP   55
      NLRG=-9999                                                        MAP   56
      NSML=9999                                                         MAP   57
      CHLRG=-999999999.                                                 MAP   58
      CHSML=999999999.                                                  MAP   59
      CHCUMD=0.                                                         MAP   60
      CHCUM=0.
      DAYNO=DAMAX=0.0                                                   MAP   61
```

```
      XLRG=YLRG=-99999999.0                                               MAP   62
      XSML=YSML= 99999999.0                                               MAP   63
      HARV=HREL=0.0                                                       MAP   64
      XTF=0                                                               MAP   65
      XTG=0                                                               MAP   66
      XGF=0                                                               MAP   67
C CHECK FOR DATA BASE FILES--WILL PROCESS BASIC(OR OUTPUT FROM MERGE)-   MAP   68
C        IN(1)- IF PRESENT;                                               MAP   69
C       IF NOT WILL CHECK FOR (1) IN(2)-OUTPUT FROM SORT,                 MAP   70
C        (2) IN(3)-OUTPUT FROM CMBINE                                     MAP   71
      DO 5 I=1,NFILES                                                     MAP   72
      IU=IN(I)                                                            MAP   73
      REWIND IU                                                           MAP   74
      READ(IU,609)                                                        MAP   75
      IF(EOF(IU).EQ.0.)GO TO 6                                            MAP   76
    5 CONTINUE                                                            MAP   77
C NO INPUT FILE                                                           MAP   78
      WRITE(KPRINT,899)                                                   MAP   79
      STOP                                                                MAP   80
    6 IN1=IU                                                              MAP   81
      REWIND IN1                                                          MAP   82
C     READ MAP-2 CARD WITH FLAGS FOR CROSS-REFERENCE TABLES
C     IFLAG IS A FLAG FOR PRINTING DATA BASE INFO
      READ(KARD,600) IFLAG,ITF,ITG,IGT,IGF,IOI                            MAP   83
C     LOI IS A FLAG FOR PRINTING ERROR MESSAGE FOR EXTRANEOUS DATA
      LOI=(IOI.EQ.0)                                                      MAP   84
C     PRINT FLAG PRNT FOR MAP MODULE OUTPUT
      PRNT=(IFLAG .GT. 0)                                                 MAP   85
C     CHECK FOR CROSS-REFERENCE TABLE PRINTING
      IF(ITF.GT.0)XTF=1                                                   MAP   86
      IF(ITG.GT.0.OR.IGT.GT.0)XTG=1                                       MAP   87
      IF(IGF.GT.0)XGF=1                                                   MAP   88
C CLEAR XREF TABLES                                                       MAP   89
      IF(XTF.LE.0.AND.XGF.LE.0)GO TO 15                                   MAP   90
      DO 14 J=1,NUMFPT                                                    MAP   91
      DO 10 I=1,NUMTRG                                                    MAP   92
   10 TXFPT(I,J)=0.0                                                      MAP   93
      DO 11 I=1,NUMGT                                                     MAP   94
   11 GTXFPT(I,J)=0.0                                                     MAP   95
   14 CONTINUE                                                            MAP   96
   15 IF(XTG.LE.0)GO TO 21                                                MAP   97
      DO 16 J=1,NUMGT                                                     MAP   98
      DO 16 I=1,NUMTRG                                                    MAP   99
   16 TXGT(I,J)=0.0                                                       MAP  100
   21 CONTINUE                                                            MAP  101
C                                                                         MAP  102
C READ [illegible] TYPE TABLE                                             MAP  103
C                                                                         MAP  104
C     WRITE HEADINGS FOR [illegible] TYPE TABLE
      WRITE(KPRINT,724)                                                   MAP  105
      [illegible]=.FALSE.                                                 MAP  106
      [illegible]                                                         MAP  107
      I=1                                                                 MAP  108
   50 READ(IN1,[illegible]) [illegible](I),(CHARGE(J,I),J=1,NUMCHG),NAME1(I),NAME2(I),
     1IFLAG,(C[illegible](I,J),J=1,15)
      IF(E[illegible]F(IN1).NE.0)GO TO 499                                MAP  110
C     C[illegible] DATA BASE CARDS
```

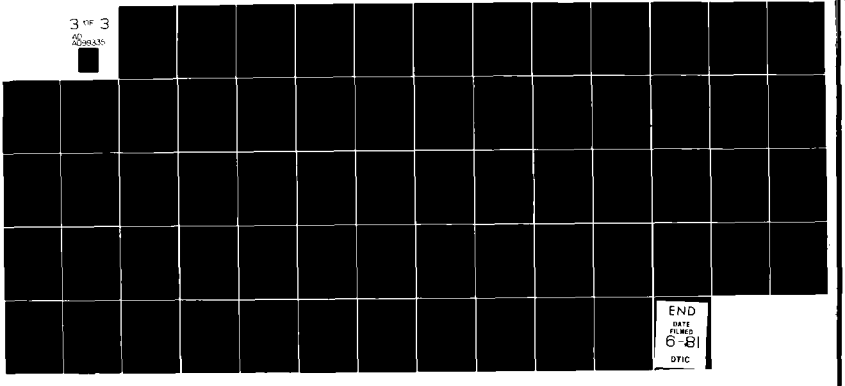
```
      NCAR [illegible]
C     COUNT [illegible]
      NG=NG+1                                                          MAP  112
C [illegible]                                                          MAP  113
      IF(I [illegible]                                                 MAP  114
     [illegible] J=1,NUMCH)                                            MAP  115
   55 IF(CHARGE(J,I) [illegible]                                       MAP  116
C     IF IFLAG [illegible] DATA [illegible]
      IF(KPRNT [illegible]),(CHARGE(J,I),J=1,NUMCH)                    MAP  117
      IMAGE=[illegible]                                                MAP  118
C CHECK CHARGE LIMITS                                                  MAP  119
      NEG=                                                             MAP  120
      LRG=                                                             MAP  121
      POS=.FALSE.                                                      MAP  122
      [illegible] J=1,NUMCH                                            MAP  123
      IF(CHARGE(J,I)) 58, [illegible]                                  MAP  124
C     CHARGE SIZE<
   58 NEG=NEG+1                                                        MAP  125
      KERR=KERR+1                                                      MAP  126
      GO TO 60                                                         MAP  127
C     CHARGE SIZE>
   59 POS=.TRUE.                                                       MAP  128
      IF(CHARGE(J,I).LE.HIGCHG) [illegible] TO 60                      MAP  129
      LRG=LRG+1                                                        MAP  130
      KERR=KERR+1                                                      MAP  131
   60 CONTINUE                                                         MAP  132
      IF(POS.AND.(NEG+LRG).E[illegible])GO TO 70                       MAP  133
      IF(IMAGE)GO TO 65                                                MAP  134
      WRITE(KPRINT,725)IFLAG,IGUN(I), (CHARGE(J,I),J=1,NUMCHG)         MAP  150
      IMAGE=.TRUE.                                                     MAP  136
C     NO POSITIVE CHARGES FOR THIS GUN TYPE
 65   IF(.NOT.POS)WRITE(KPRINT,731)                                    MAP  137
C     NEG CHARGES ENCOUNTERED
      IF(NEG.GT.0)WRITE(KPRINT,732)NEG                                 MAP  138
C     CHARGE SIZE TOO LARGE
      IF(LRG.GT.0)WRITE(KPRINT,733)LRG,HIGCHG                          MAP  139
   70 CONTINUE                                                         MAP  140
C CHECK FOR DUP. I FIRST OCCURRENCE USED FOR TABLE                     MAP  141
      IF(I.EQ.1)GO TO 80                                               MAP  142
      NTD=I-1                                                          MAP  143
      ID=I[illegible](I)                                               MAP  144
      DO 75 [illegible]=1,[illegible]                                  MAP  145
      IF(I[illegible].EQ.IGUN([illegible]))GO TO 76                    MAP  146
   75 CONTINUE                                                         MAP  147
      GO TO 80                                                         MAP  148
   76 IF(IMAGE)GO TO 77                                                MAP  149
      WRITE(KPRINT,725)IFLAG,I[illegible](I),(CHARGE(J,I),J=1,NUMCHG)  MAP  135
      IMAGE=.TRUE.                                                     MAP  151
C     DUPLICATE I[illegible]
   77 WRITE(KPRINT,734)                                                MAP  152
      KERR=KERR+1                                                      MAP  153
      I=NT[illegible]                                                  MAP  154
   80 CONTINUE                                                         MAP  155
C LAST [illegible] TABLE [illegible]                                   MAP  156
      IF(IFLAG.EQ.1[illegible])GO TO [illegible]                       MAP  157
      IF(I.GE.NU[illegible])GO TO 85                                   MAP  158
      I=I+1                                                            MAP  159
```

```
      GO TO 50                                                          MAP  160
C TABLE OVERFLOW                                                        MAP  161
   85 IF(OVRFLO) GO TO 5                                                MAP  162
      WRITE(KPRINT,7(H)?) ?GT                                           MAP  163
      I=N??GT+1                                                         MAP  164
      OVRFL=.TRUE.                                                      MAP  165
C FLAG OVERFLOW F?R ???T ??T ??EF TABLES                                MAP  166
      IF(XT?.E?.1)XT?=?                                                 MAP  167
      IF(XGF.E?.1)XGF=?                                                 MAP  168
      GO TO 50                                                          MAP  169
C END OF GUN TABLE INPUT                                                MAP  170
   90 NTYPES=MI??(1,???T)                                               MAP  171
      IF(.NOT. PRNT) GO TO 95
C     WRITE HEADING F?R SEC??? PART OF ??? TYPE TABLE FOR CONTOUR
C     INFORMATION.
      WRITE(KPRINT,7241)
 7241 FORMAT(////T2,*?TYPE*,?X,*?AME*,1?X,*EG PARA*,2X,*EG PARB*,29X,
     1*DIRECTIVITY VALUES*,3?X,*AVG*//)
C     WRITE OUT NAME AND CONTOUR INFORMATION
      DO 91 K=1,NTYPES
         WRITE(KPRINT,7251)IGGUN(K),NAME1(K),NAME2(K),
     1(CONTOR(K,J),J=1,15)
 7251 FORMAT(T4,A2,5X,2A10,2X,F?.2,?X,F5.2,3X,12F6.2,2X,F5.2)
   91 CONTINUE
C PRINT ERROR/WARNING COUNT                                             MAP  172
   95 WRITE(KPRINT,730)KERR                                             MAP  173
      NERR=NERR+KERR                                                    MAP  174
      KERR=0                                                            MAP  175
C                                                                       MAP  176
C READ TARGETS                                                          MAP  177
C                                                                       MAP  178
      ASSIGN 231 TO MAXMIN                                              MAP  179
C     WRITE HEADING FOR TARGET CARDS
      WRITE(KPRINT,6?7)                                                 MAP  180
      WRITE(KPRINT,714)                                                 MAP  181
      OVRFL=.FALSE.                                                     MAP  182
C     READ TARGET CARDS
      I=1                                                               MAP  183
  200 READ(I?1,?01)IFLAG,?GL(1),ID,?GL(2),XL?C,YLOC,PHCORR,             MAP  184
     1(?GL(J),J=3,50)                                                   MAP  185
      IF(EOF(IN1).NE.0.)GO TO 299                                       MAP  186
C     COUNT TARGET CARDS WITH DATA BASE CARDS
      NCARDS=NCARDS+1                                                   MAP  187
C     COUNT TARGET CARDS
      NT=NT+1                                                           MAP  188
C ?0(BLANK) TO ? FOR PRINTING                                           MAP  189
      IF(ID.?.?)ID=0                                                    MAP  190
      IF(XLOC.?.?.)XLOC=?.                                              MAP  191
      IF(YLOC.?.?.)YLOC=?.                                              MAP  192
      IF(PHCORR.?.?.)PHCORR=?.                                          MAP  193
C                                                                       MAP  194
      IF(PRNT)WRITE(KPRINT,71?)IFLAG,?GL(1),ID,?GL(2),XLOC,YLOC,PHCORR, MAP  195
     1         (?GL(J),J=3,50)                                          MAP  196
      IMAGE=PR?T                                                        MAP  197
C CHECK BLANK FIELDS                                                    MAP  198
      DO 210 J=1,50                                                     MAP  199
      IF(?GL(J).NE.IHL ?) GO TO 211                                     MAP  200
```

```
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]                                              MAP 204
      ASSIGN [illegible]                                       MAP 205
      GO TO 2999[illegible]                                    MAP 206
  [illegible] IF [illegible] WRITE(KPRINT,735 [illegible]      MAP 207
  [illegible] KERR=KERR+1                                      MAP 208
      [illegible]                                              MAP 209
  [illegible] IF [illegible] TO 233                            MAP 210
      IF(IMAGE) GO TO 222                                      MAP 211
      ASSIGN 222 TO KPRNTT                                     MAP 212
      GO TO 2999[illegible]                                    MAP 213
  [illegible] WRITE(KPRINT,736)                                MAP 214
      KERR=KERR+1                                              MAP 215
C [illegible] CHECK                                            MAP 216
  23[illegible] GO TO 2999[illegible]                          MAP 217
  231 CONTINUE                                                 MAP 218
C [illegible]                                                  MAP 219
      IF(I.GT.1)GO TO 234                                      MAP 220
      IT[illegible]S(1)=1                                      MAP 221
      TC[illegible]S(1,1)=XLOC                                 MAP 222
      TC[illegible]S(1,2)=YLOC                                 MAP 223
      GO TO 25[illegible]                                      MAP 224
C IF [illegible] FIRST OCCURRENCE FOR TABLE                    MAP 225
  234 NTG=I-1                                                  MAP 226
      DO 240 J=1,NTG                                           MAP 227
      IF(I[illegible] THG(J))GO TO 241                         MAP 228
  240 CONTINUE                                                 MAP 229
C [illegible]                                                  MAP 230
      IF([illegible]) GO TO 24[illegible]                      MAP 231
      I[illegible]S(I)=1                                       MAP 232
      TC[illegible]S(I,1)=XLOC                                 MAP 233
      TC[illegible]S(I,2)=YLOC                                 MAP 234
      GO TO 245                                                MAP 235
C DUP                                                          MAP 236
  241 IF(IMAGE)GO TO 242                                       MAP 237
      ASSIGN 242 TO KPRNTT                                     MAP 238
      GO TO 2999[illegible]                                    MAP 239
  242 WRITE(KPRINT,734)                                        MAP 240
      KERR=KERR+1                                              MAP 241
      IF(.NOT.[illegible])[illegible]                          MAP 242
C DUP, [illegible] CONSIDERED SEPARATE FOR TABLE               MAP 243
  245 CONTINUE                                                 MAP 244
      DO 246 J=1,NTG                                           MAP 245
      IF(XLOC.[illegible].TC[illegible](J,1))GO TO 246         MAP 246
      IF(YLOC.[illegible].TC[illegible](J,2))GO TO 247         MAP 247
  246 CONTINUE                                                 MAP 248
      GO TO 25                                                 MAP 249
  247 IF(IMAGE)GO TO 248                                       MAP 250
      ASSIGN 248 TO KPRNTT                                     MAP 251
      GO TO 2999[illegible]                                    MAP 252
C     WARNING [illegible] DUPLICATE [illegible]
  248 WRITE(KPRINT,737)                                        MAP 253
      KERR=KERR+1                                              MAP 254
C     LAST [illegible]                                         MAP 255
```

```
   25 IF(IFLAG.[illegible]) [illegible]                                  MAP 256
      IF([illegible]) [illegible]                                        MAP 257
      I=I+1                                                              MAP 258
      [illegible]                                                        MAP 259
C TABLE [illegible]                                                      MAP 260
  260 IF([illegible]) [illegible]                                        MAP 261
      WRITE([illegible],73A) [illegible]                                 MAP 262
      I=[illegible]+1                                                    MAP 263
      [illegible]                                                        MAP 264
C [illegible] TABLES                                                     MAP 265
      IF([illegible])                                                    MAP 266
      IF([illegible])                                                    MAP 267
      [illegible]                                                        MAP 268
C****************[illegible] TARGET AND FIRING [illegible] IMAGE         MAP 269
 2991 WRITE([illegible],IFLAG,[illegible](1),[illegible](2),XLOC,YLOC,PHCORR,  MAP 270
     1 [illegible](J),J=1,5)                                             MAP 271
      [illegible]                                                        MAP 272
      GO TO [illegible],(212,222,242,248,312,322,342,348)                MAP 273
C**************                                                          MAP 274
C [illegible] TARGET [illegible]                                         MAP 275
  241 [illegible]                                                        MAP 276
C PRINT [illegible]                                                      MAP 277
      WRITE([illegible],73[illegible])[illegible]                        MAP 278
      [illegible]                                                        MAP 279
      [illegible]                                                        MAP 280
C                                                                        MAP 281
C READ FIRING PT [illegible] CARDS                                       MAP 282
C                                                                        MAP 283
C READ FIRING PT. CARDS                                                  MAP 284
      ASSIGN 331 TO [illegible]                                          MAP 285
C     WRITE HEADING FOR FIRING POINT CARDS
      WRITE([illegible],716)                                             MAP 286
      WRITE([illegible],7161)                                            MAP 287
      [illegible]                                                        MAP 288
      I=1                                                                MAP 289
  300 ASSIGN 301 TO [illegible]                                          MAP 290
      NEG=1                                                              MAP 291
      READ(IN),[illegible]IFLAG,[illegible],XLOC,YLOC,PHCORR,            MAP 292
     1[illegible](J),J=1,5)                                              MAP 293
      IF([illegible]) [illegible]                                        MAP 294
C     [illegible] CARDS
      [illegible]                                                        MAP 295
C     [illegible] CARDS
      NS=NS+1                                                            MAP 296
C [illegible]                                                            MAP 297
      IF([illegible])                                                    MAP 298
      IF(XLOC.[illegible])                                               MAP 299
      IF(YLOC.[illegible])                                               MAP 300
      IF([illegible])                                                    MAP 301
C                                                                        MAP 302
      IF([illegible]) [illegible] 2991                                   MAP 303
  3[illegible] CONTINUE                                                  MAP 304
      IF([illegible],IFLAG,[illegible]1),IC,[illegible](2),XLOC,YLOC,[illegible],  MAP 305
     1 [illegible],J=[illegible])                                        MAP 306
      [illegible]                                                        MAP 307
C CHECK [illegible]                                                      MAP 308
      [illegible] 31[illegible] J=1,5                                    MAP 309
```

AD-A099 335 CONSTRUCTION ENGINEERING RESEARCH LAB (ARMY) CHAMPAIGN IL F/G 20/1
BLAST NOISE PREDICTION, VOLUME II. BNOISE 3.2 COMPUTER PROGRAM --ETC(
MAR 81 L L LITTLE, V I PAWLOWSKA, D L EFFLAND
UNCLASSIFIED CERL-TR-N-98-VOL-2 NL



```
          GO TO 345                                                          MAP  363
      C      DIFF COORDS                                                     MAP  364
      C   DUP ID, DIFF. COORDINATES
  344     WRITE(KPRINT,754)                                                  MAP  365
          KERR=KERR+1                                                        MAP  366
          LINES=LINES+1                                                      MAP  367
          IF(OVRFL)GO TO 345                                                 MAP  368
      C      GENERATED ID                                                    MAP  369
          ID=SHIFT(-1,42)                                                    MAP  370
          WRITE(KPRINT,755) ID                                               MAP  371
          LINES=LINES+1                                                      MAP  372
          GO TO 343                                                          MAP  373
      C DUP. COORDS;WARNING,BUT CONSIDERED SEPARATE FOR TABLE                MAP  374
  345 CONTINUE                                                               MAP  375
          DO 346 J=1,NTD                                                     MAP  376
          IF(XLOC.NE.FCORDS(J,1))GO TO 346                                   MAP  377
          IF(YLOC.EQ.FCORDS(J,2))GO TO 347                                   MAP  378
  346 CONTINUE                                                               MAP  379
          GO TO 350                                                          MAP  380
  347 IF(IMAGE)GO TO 348                                                     MAP  381
          ASSIGN 348 TO KPRNTT                                               MAP  382
          GO TO 29991                                                        MAP  383
  348     WRITE(KPRINT,737)                                                  MAP  384
          KERR=KERR+1                                                        MAP  385
          LINES=LINES+1                                                      MAP  386
  350 CONTINUE                                                               MAP  387
      C                                                                      MAP  388
      C READ FIRING PT. DEF. CARDS                                           MAP  389
      C                                                                      MAP  390
          ASSIGN 401 TO MORRTN                                               MAP  391
  400     READ(IN1,603)IFLAG,(NUL(J),J=1,17),ITYPE,DAY,DARK,MIN,MAX,IDT,     MAP  392
         1KFLAG,HGT,(NUL(J),J=18,56)                                         MAP  393
          IF(EOF(IN1).NE.0.)GO TO 399                                        MAP  394
      C      COUNT MORE DATA BASE CARDS
          NCARDS=NCARDS+1                                                    MAP  395
      C -0(BLANK) TO 0 FOR PRINTING                                          MAP  396
          IF(ITYPE.EQ.0)ITYPE=0                                              MAP  397
          IF(DAY.EQ.0.)DAY=0.                                                MAP  398
          IF(DARK.EQ.0.)DARK=0.                                              MAP  399
          IF(MIN.EQ.0)MIN=0                                                  MAP  400
          IF(MAX.EQ.0)MAX=0                                                  MAP  401
          IF(KFLAG.EQ.0)KFLAG=0                                              MAP  402
          IF(IDT.EQ.0)IDT=0                                                  MAP  403
          IF(HGT.EQ.0.)HGT=0.                                                MAP  404
      C                                                                      MAP  405
          IF(PRNT.AND.LINES.GE.NUMLIN)GO TO 99991                            MAP  406
  401 CONTINUE                                                               MAP  407
          IF(PRNT)WRITE(KPRINT,718)IFLAG,(NUL(J),J=1,17),ITYPE,DAY,DARK,MIN, MAP  408
         1MAX,IDT,KFLAG,HGT,(NUL(J),J=18,56)                                 MAP  409
          LINES=LINES+1                                                      MAP  410
          IMAGE=PRNT                                                         MAP  411
      C CHECK BLANK FIELDS                                                   MAP  412
          DO 405 J=1,49                                                      MAP  413
          IF(NUL(J).NE.IBLANK)GO TO 406                                      MAP  414
  405 CONTINUE                                                               MAP  415
          GO TO 409                                                          MAP  416
      C DATA IN BLANK FIELD                                                  MAP  417
```

```
      406   IF(.NOT. L(I)) GO TO 408
            IF(IMAGE)GO TO 407                                    MAP  418
            ASSIGN 407 TO KPRNTD                                  MAP  419
            GO TO 49991                                           MAP  420
      C     IF EXTRANEOUS DATA FLAG NOT ZERO,PRINT ERROR MESSAGE
       407  IF(L(J))WRITE(KPRINT,735) NOCOLS(J)                   MAP  421
            LINES=LINES+1
      408   KERR=KERR+1                                           MAP  422
      C GUN TYPE                                                  MAP  424
        409 DO 410 IG=1,NTYPES                                    MAP  425
            IF(ITYPE.EQ.IEGUN(IG))GO TO 415                       MAP  426
        410 CONTINUE                                              MAP  427
      C UNDEFINED GUN TYPE                                        MAP  428
            IF(IMAGE)GO TO 411                                    MAP  429
            ASSIGN 411 TO KPRNTD                                  MAP  430
            GO TO 49991                                           MAP  431
       411  WRITE(KPRINT,727)ITYPE,ID                             MAP  432
            KERR=KERR+1                                           MAP  433
            LINES=LINES+1                                         MAP  434
            IG=0                                                  MAP  435
        415 CONTINUE                                              MAP  436
      C FIRINGS                                                   MAP  437
            IF(DAY.LT.0..OR.DARK.LT.0.)GO TO 420                  MAP  438
            SHOTS=DAY+DARK                                        MAP  439
            IF(DUP)GO TO 419                                      MAP  440
            DAYN=DAYN+DAY                                         MAP  441
            DARKN=DARKN+DARK                                      MAP  442
        419 IF(SHOTS.GT.0.)GO TO 430                              MAP  443
        420 SHOTS=0.                                              MAP  444
            IF(IMAGE)GO TO 421                                    MAP  445
            ASSIGN 421 TO KPRNTD                                  MAP  446
            GO TO 49991                                           MAP  447
      C     NUMBER OF FIRINGS<=0
       421  WRITE(KPRINT,739)ID                                   MAP  448
            KERR=KERR+1                                           MAP  449
            LINES=LINES+1                                         MAP  450
      C CHARGE                                                    MAP  451
        430 IF(MIN.LT.NSML)NSML=MIN                               MAP  452
            IF(MIN.LE.0.OR.MIN.GE.NUMCHG)GO TO 434                MAP  453
            IF(IG.EQ.0)GO TO 440                                  MAP  454
      C FIRST ELEMENT IN CHARGE=TARGET CHARGE,SO OFFSET BY 1      MAP  455
            CHG=CHARGE(MIN+1,IG)                                  MAP  456
            IF(CHG.LE.0.)GO TO 434                                MAP  457
            IF(CHG.LT.CHSML)CHSML=CHG                             MAP  458
            GO TO 440                                             MAP  459
      C INVALID CHARGE                                            MAP  460
        434 IF(IMAGE)GO TO 435                                    MAP  461
            ASSIGN 435 TO KPRNTD                                  MAP  462
            GO TO 49991                                           MAP  463
        435 KERR=KERR+1                                           MAP  464
            IF(MIN.GT.0)GO TO 436                                 MAP  465
            NEG=NEG+1                                             MAP  466
            GO TO 440                                             MAP  467
      C     CHARGE SIZE NO GOOD
       436  WRITE(KPRINT,740)ID                                    MAP  468
            LINES=LINES+1                                         MAP  469
        440 CHG=0.                                                MAP  470
```

```
                IF(MAX.GT.MLRG)MLRG=MAX                                 MAP  471
515             IF(MAX.LE.0.OR.MAX.GE.NUMCHG)GO TO 444                  MAP  472
                IF(IG.EQ.0)GO TO 446                                    MAP  473
                CHG=CHANGE(MAX+1,IG)                                    MAP  474
                IF(CHG.LE.0.)GO TO 444                                  MAP  475
                IF(CHG.GT.CHLRG)CHLRG=CHG                               MAP  476
520             IF(.NOT.DUP) CHCUMD=CHCUMD+CHG*DAY                      MAP  477
                IF(.NOT. DUP) CHCUMN=CHCUMN+CHG*DARK
                GO TO 446                                               MAP  478
              C INVALID                                                 MAP  479
              444 IF(IMAGE)GO TO 445                                    MAP  480
525             ASSIGN 445 TO KPRNTD                                    MAP  481
                GO TO 9999                                              MAP  482
              445 KERR=KERR+1                                           MAP  483
                CHG=0.                                                  MAP  484
                IF(MAX.GT.0)GO TO 448                                   MAP  485
530             NEG=NEG+1                                               MAP  486
                GO TO 446                                               MAP  487
              448 WRITE(KPRINT,740)ID                                   MAP  488
                LINES=LINES+1                                           MAP  489
              446 IF(MIN.LE.MAX)GO TO 450                               MAP  490
535             IF(IMAGE)GO TO 447                                      MAP  491
                ASSIGN 447 TO KPRNTD                                    MAP  492
                GO TO 9999                                              MAP  493
              447 WRITE(KPRINT,741)ID                                   MAP  494
                KERR=KERR+1                                             MAP  495
540             LINES=LINES+1                                           MAP  496
              C HIT FLAG,TARGET ID                                      MAP  497
              C HIT FLAG MUST BE SET IF IDT BLANK                       MAP  498
                450  IF(IDT.NE.IBLNK)GO TO 460                         MAP  499
                IT=0                                                    MAP  500
545             IF(HFLAG.EQ.1)GO TO 470                                 MAP  501
                IF(IMAGE)GO TO 451                                      MAP  502
                ASSIGN 451 TO KPRNTD                                    MAP  503
                GO TO 9999                                              MAP  504
              C     BLANK TARGET ID
550           451 WRITE(KPRINT,728)ID                                   MAP  505
                KERR=KERR+1                                             MAP  506
                LINES=LINES+1                                           MAP  507
                GO TO 470                                               MAP  508
              C     FIND TARGET GIVEN IN TARGET ID TABLE
555           460 DO 465 IT=1,NTAR                                      MAP  509
                IF(IDT.EQ.IDTRG(IT))GO TO 468                           MAP  510
              465 CONTINUE                                              MAP  511
              C UNDEFINED TARGET ID                                     MAP  512
                IF (IMAGE)GO TO 467                                     MAP  513
560             ASSIGN 467 TO KPRNTD                                    MAP  514
                GO TO 9999                                              MAP  515
              467 WRITE(KPRINT,722)IDT,ID                               MAP  516
                KERR=KERR+1                                             MAP  517
                LINES=LINES+1                                           MAP  518
565             IT=0                                                    MAP  519
                GO TO 470                                               MAP  520
              C VALID TARGET CHANGE?                                    MAP  521
              468 IF(IG.LE.0)GO TO 470                                  MAP  522
                IF(CHG=0.
570             IF(HFLAG .EQ. 1) GO TO 470
```

```
                TCHG=CHARGE(1,IG)                                       MAP  523
                IF(TCHG.GT.0)GO TO 475                                  MAP  524
                TCHG=0.                                                 MAP  525
575
                IF(IMAGE)GO TO 469                                      MAP  526
                ASSIGN 469 TO KPRNTD                                    MAP  527
                GO TO 49991                                             MAP  528
580           C     NON POSITIVE TARGET CHARGE
              469 WRITE(KPRINT,757)ITYPE,ID                             MAP  529
                KERR=KERR+1                                             MAP  530
                LINES=LINES+1                                           MAP  531
              C HEIGHT                                                  MAP  532
585           470   IF(.NOT. DUP) CHCUMD=CHCUMD+TCHG*DAY
                    IF(.NOT. DUP) CHCUMN=CHCUMN+TCHG*DARK
                    IF(ABS(HGT).GT.999.)GO TO 475
                    IF(HGT.LT.HBEL)HBEL=HGT                             MAP  534
                    IF(HGT.GT.HABV)HABV=HGT                             MAP  535
590                 GO TO 480                                           MAP  536
              475 IF(IMAGE)GO TO 477                                    MAP  537
                    ASSIGN 477 TO KPRNTD                                MAP  538
                    GO TO 49991                                         MAP  539
              C     LARGE HEIGHT VALUE IN DEF. CARD FOR FIRING PT.
595           477 WRITE(KPRINT,742)ID                                   MAP  540
                KERR=KERR+1                                             MAP  541
                LINES=LINES+1                                           MAP  542
              C XREF TABLE ENTRIES                                      MAP  543
              480 IF(SHOTS.LE.0..OR.DUP)GO TO 490                       MAP  544
600             IF(IT.LE.0)GO TO 487                                    MAP  545
                IF(XTF.LE.0.OR.IFP.LE.0)GO TO 485                       MAP  546
              C TARGET X FIRING PT -- FIRINGS                           MAP  547
                TXFPT(IT,IFP)=TXFPT(IT,IFP)+SHOTS                       MAP  548
              C TARGETS X GUN TYPES -- FIRINGS*TARGET CHARGE            MAP  549
605           485 IF(IG.LE.0)GO TO 490                                  MAP  550
                IF(XTG.LE.0)GO TO 487                                   MAP  551
                TXGT(IT,IG)=TXGT(IT,IG)+ TCHG        *SHOTS             MAP  552
              C GUN TYPES X FIRING PT -- FIRINGS*CHARGE                 MAP  553
              487 IF(XGF.LE.0.OR.IFP.LE.0)GO TO 490                     MAP  554
610             GTXFPT(IG,IFP)=GTXFPT(IG,IFP)+CHG*SHOTS                 MAP  555
              C       LAST DEF CARD?                                    MAP  556
              490 IF(IFLAG.NE.1H*)GO TO 400                             MAP  557
              C     IF NEGATIVE CHARGES---WRITE ERROR MESSAGE
                IF(NEG.GT.0)WRITE(6,756)NEG,ID                          MAP  558
615             LINES=LINES+2                                           MAP  559
              C     CHECK FOR OVERFLOW
                IF(I.GE.NUMFPT)GO TO 491                                MAP  560
                I=I+1                                                   MAP  561
                GO TO 300                                               MAP  562
620           C TABLE OVERFLOW                                          MAP  563
              491 IF(OVRFL)GO TO 300                                    MAP  564
                WRITE(KPRINT,743)NUMFPT                                 MAP  565
                I=NUMFPT+1                                              MAP  566
                OVRFL=.TRUE.                                            MAP  567
625           C FLAG OVERFLOW FOR RELEVANT XREF TABLES                  MAP  568
                IF(XTF.EQ.1)XTF=2                                       MAP  569
                IF(XGF.EQ.1)XGF=2                                       MAP  570
```

```
      C NULL TABLE INDEX                                                    MAP  571
            IFP=0                                                           MAP  572
  630       GO TO 300                                                       MAP  573
      C**************PRINT DEF CARD IMAGE                                   MAP  574
      49991 WRITE(KPRINT,718)IFLAG,(NUL(J),J=1,17),ITYPE,DAY,DARK,MIN,MAX,  MAP  575
           1          INT,KFLAG,HGT,(NUL(J),J=18,56)                      MAP  576
            IMAGE=.TRUE.                                                    MAP  577
  635       GO TO KPRNTD,(407,411,421,435,445,447,451,469)                  MAP  578
      C*************                                                        MAP  579
      C****************** INTERNAL SUBROUTINES ********************        MAP  580
      C HEADER ROUTINE                                                      MAP  581
      99991 CONTINUE                                                        MAP  582
  640       WRITE(KPRINT,607)                                               MAP  583
            WRITE(KPPINT,7161)                                              MAP  584
            LINES=0                                                         MAP  585
            GO TO HDRTN,(301,401)                                           MAP  586
      C MAX,MIN COORDS CHECK                                                MAP  587
  645 99992 CONTINUE                                                        MAP  588
            IF(XLOC.LE.XLRG)GO TO 100                                       MAP  589
            XLRG=XLOC                                                       MAP  590
            YLRG1=YLOC                                                      MAP  591
        100 IF(YLOC.LE.YLRG)GO TO 105                                       MAP  592
  650       XLRG1=XLOC                                                      MAP  593
            YLRG=YLOC                                                       MAP  594
        105 IF(XLOC.GE.XSML)GO TO 115                                       MAP  595
            XSML=XLOC                                                       MAP  596
            YSML1=YLOC                                                      MAP  597
  655   115 IF(YLOC.GE.YSML)GO TO 120                                       MAP  598
            XSML1=XLOC                                                      MAP  599
            YSML=YLOC                                                       MAP  600
        120 GO TO MAXMIN,(231,331)                                          MAP  601
      C*************************************************************       MAP  602
  660 C END OF FIRING PT. CARDS                                             MAP  603
        500 IF(.NOT.OVRFL)I=I-1                                             MAP  604
            NSRCS=MIN0(I,NUMFPT)                                            MAP  605
      C PRINT ERROR/WARNING COUNT                                           MAP  606
            WRITE(KPRINT,730)KERR                                           MAP  607
  665       NERR=NERR+KERR                                                  MAP  608
      C                                                                     MAP  609
      C END OF INPUT PHASE                                                  MAP  610
      C                                                                     MAP  611
      C     TOTAL NUMBER OF DATA BASE ERRORS                                
  670       WRITE(KPRINT,744)NERR                                           MAP  612
            WRITE(KPPINT,607)                                               MAP  613
      C     TOTAL NUMBER OF DATA BASE CARDS                                 
            WRITE(KPRINT,701)NCARDS                                         MAP  614
      C     NUMBER OF GUN TYPES, TARGETS, AND FIRING POINTS                 
  675       WRITE(KPRINT,699)NG                                             MAP  615
            WRITE(KPRINT,702)NT                                             MAP  616
            WRITE(KPRINT,703)NS                                             MAP  617
            IF(NS.EQ.0)GO TO 299                                            MAP  618
            READ(KARD,505) DAYS                                             MAP  619
  680       IF(EOF(KARD).EQ.0.) GO TO 501                                   MAP  620
            DAYS=1.0                                                        MAP  621
            REOF=.TRUE.                                                     MAP  622
        501 CONTINUE                                                        MAP  623
      C     DATA BASE TIME PERIOD                                           
```

```
      WRITE(KPRINT,723)DAYS                                              MAP  624
C     TOTAL OF DAY FIRINGS AND TOTAL NIGHT FIRINGS
      WRITE(KPRINT,704)DAYN,DARKN                                        MAP  625
      DAYN=DAYN/DAYS                                                     MAP  626
      DARKN=DARKN/DAYS                                                   MAP  627
      WRITE(KPRINT,721)DAYN,DARKN                                        MAP  628
      CHCUMD=CHCUMD/DAYS                                                 MAP  629
      CHCUMN=CHCUMN/DAYS
C     CHARGE INFO
      WRITE(KPRINT,713)XSML,XLRG,CHSML,CHLRG,CHCUMD,CHCUMN               MAP  630
      IF(ABS(HABV).NE.0.0.OR.ABS(HBEL).NE.0.0)WRITE(KPRINT,705)HABV,HBEL MAP  631
      WRITE(KPRINT,708)XLRG,XLRG,YLRG1                                   MAP  632
      WRITE(KPRINT,709)YLRG,XLRG1,YLRG                                   MAP  633
      WRITE(KPRINT,710)XSML,XSML,YSML1                                   MAP  634
      WRITE(KPRINT,711)YSML,XSML1,YSML                                   MAP  635
      DX=ABS(XLRG-XSML)                                                  MAP  636
      DY=ABS(YLRG-YSML)                                                  MAP  637
C     READ NUMBER OF GRID SIZES TO BE TESTED
      READ(KARD,604) N                                                   MAP  638
      IF(EOF(KARD).NE.0.)GO TO 510                                       MAP  639
      DO 505 I=1,N                                                       MAP  640
C     READ GRID SIZES
      READ(KARD,605) GRDSZ                                               MAP  641
C     CALCULATE GRID DIMENSION
      DX1=DX/GRDSZ                                                       MAP  642
      DY1=DY/GRDSZ                                                       MAP  643
      WRITE(KPRINT,712)GRDSZ,DX1,DY1                                     MAP  644
  505 CONTINUE                                                           MAP  645
  510 CONTINUE                                                           MAP  646
C                                                                        MAP  647
C XREF TABLES                                                            MAP  648
C                                                                        MAP  649
C TARGETS X FIRING PTS                                                   MAP  650
C       ARRAY PROCESSING ORDER                                           MAP  651
C       BY COLUMN                                                        MAP  652
      IJ=1                                                               MAP  653
C       BY ROW                                                           MAP  654
      JI=2                                                               MAP  655
      IF(ITF.LE.0)GO TO 560                                              MAP  656
      WRITE(KPRINT,607)                                                  MAP  657
      IF(XTF.EQ.2)WRITE(KPRINT,746)                                      MAP  658
      WRITE(KPRINT,745)                                                  MAP  659
      CALL PUTXR(TXFPT,NUMTRG,NUMFPT,NTAR,NSHCS,ADTRG,ADFPT,DAYS,XREF,   MAP  660
     1           IJ)                                                     MAP  661
C TARGETS BY GUN TYPES AND VICE-VERSA                                    MAP  662
  560 IF(ITG.LE.0)GO TO 565                                              MAP  663
      WRITE(KPRINT,607)                                                  MAP  664
      IF(XTG.EQ.2)WRITE(KPRINT,746)                                      MAP  665
      WRITE(KPRINT,748)                                                  MAP  666
      CALL PUTXR(TXGT,NUMTRG,NUMGT,NTAR,NTYPES,ADTRG,ADGUN,DAYS,XREF,IJ)MAP  667
  565 IF(IGT.LE.0)GO TO 570                                              MAP  668
      WRITE(KPRINT,607)                                                  MAP  669
      IF(XTG.EQ.2)WRITE(KPRINT,746)                                      MAP  670
      WRITE(KPRINT,750)                                                  MAP  671
      CALL PUTXR(TXGT,NUMTRG,NUMGT,NTYPES,NTAR,ADGUN,ADTRG,DAYS,XREF,JI)MAP  672
C GUN TYPES X FIRING PT                                                  MAP  673
  570 IF(IGF.LE.0)GO TO 590                                              MAP  674
```

```
      WRITE(KPRINT,607)                                                 MAP  675
      IF(XGF.EQ.2)WRITE(KPRINT,746)                                     MAP  676
      WRITE(KPRINT,752)                                                 MAP  677
745   CALL PUTXR(GTXFPT,NUMGT,NUMFPT,NTYPES,NSHCS,ADGUN,ADFPT,DAYS,XREF,MAP  678
     1           IJ)                                                    MAP  679
  590 CONTINUE                                                          MAP  680
C     COMPUTE TIME IN MAP
      CALL SECOND(T99)                                                  MAP  681
750   T99=T99-T100                                                      MAP  682
      WRITE(KPRINT,900)T99                                              MAP  683
      RETURN                                                            MAP  684
C EOF EXITS                                                             MAP  685
  299 CONTINUE                                                          MAP  686
755   C TARGET CARDS                                                    MAP  687
      WRITE(KPRINT,719)                                                 MAP  688
      STOP                                                              MAP  689
  399 CONTINUE                                                          MAP  690
      C FIRING PT. CARDS                                                MAP  691
760   WRITE(KPRINT,720)                                                 MAP  692
      STOP                                                              MAP  693
C      GUN TABLE
  499 WRITE(KPRINT,729)
      STOP
765   600  FORMAT(6I1)                                                  MAP  697
  601 FORMAT(A1,A1,A3,A1,3F6.0,48A1)                                    MAP  698
  603 FORMAT(18A1,  A2,F4.0,F4.0,I2,I2,A3,I1,F5.0,39A1)                 MAP  699
  604 FORMAT (I2)                                                       MAP  700
  605 FORMAT(F10.0)                                                     MAP  701
770   607 FORMAT(1H1)                                                   MAP  702
  609 FORMAT(1X,A2,11F7.0/3X,2A10/A1,2X,F6.2,14F5.2)
  699 FORMAT(/T15,*NUMBER OF GUN TYPES READ IS *,I5)                    MAP  704
  700 FORMAT(1H1,///,T15,*.....     MAP OF SOURCE POINTS     .....*)    MAP  705
  701 FORMAT(//,T15,*NUMBER OF DATA BASE CARD IMAGES READ IS *,I5)      MAP  706
775   702 FORMAT(/,T15,*NUMBER OF TARGETS READ IS *,I5)                 MAP  707
  703 FORMAT(/,T15,*NUMBER OF SOURCES READ IS *,I5)                     MAP  708
  704 FORMAT(/,T15,*TOTAL DAY FIRINGS IS   *,F10.2,                     MAP  709
     1/,T15,*TOTAL NIGHT FIRINGS IS *,F10.2)                            MAP  710
  705 FORMAT(/,T15,*MAXIMUM HEIGHT IS *,F10.2,//,T15,*MAXIMUM DEPTH IS *,MAP  711
780  1F10.2,/)                                                          MAP  712
  708 FORMAT(/,T15,*MAXIMUM X IS *,F10.1,* IN PAIR  (*,F10.1,* , *,F10.1MAP  713
     1,* )*)                                                            MAP  714
  709 FORMAT(/,T15,*MAXIMUM Y IS *,F10.1,* IN PAIR  (*,F10.1,* , *,F10.1MAP  715
     1,* )*)                                                            MAP  716
785   710 FORMAT(/,T15,*MINIMUM X IS *,F10.1,* IN PAIR  (*,F10.1,* , *,F10.1MAP  717
     1,* )*)                                                            MAP  718
  711 FORMAT(/,T15,*MINIMUM Y IS *,F10.1,* IN PAIR  (*,F10.1,* , *,F10.1MAP  719
     1,* )*)                                                            MAP  720
  712 FORMAT(//T15,* FOR GRID SIZE*,F8.1,*, GRID DIMENSIONS =*,F7.1,    MAP  721
790  1* X *,F7.1)                                                       MAP  722
  713 FORMAT(/T15,*MINIMUM CHARGE NUMBER*,I5/T15,*MAXIMUM CHARGE NUMBER*MAP  723
     1,I5//                                                             MAP  724
     2T15,*MINIMUM CHARGE WEIGHT*,F9.1,* LBS*/T15,*MAXIMUM CHARGE WEIGHTMAP  725
     3*,F9.1,* LBS*//                                                   MAP  726
795  4 T15,*TOTAL DAY CHARGE WEIGHT PER DAY*,2X,F10.1,* LBS*/           MAP  727
     5 T15,*TOTAL NIGHT CHARGE WEIGHT PER DAY*,F10.1,* LBS*)
  714 FORMAT(//,T15,*.....     TARGET CARDS     .....*,//T10,*FLAG*,2X, MAP  728
     1*  IO *,2X,*   X   *,2X,*   Y   *,4X,*HT CORR*//)                 MAP  729
```

```
      715  FORMAT(T12,A1,2X,A1,A3,A1,1X,3(2X,F7.0),48A1)                    MAP  730
800   716  FORMAT(1H1/                                                      MAP  731
     1          T15,*.....     FIRING PT. SOURCE AND DEFINITION CARDS      MAP  732
     1  .....*//)                                                          MAP  733
     7161 FORMAT(                                                         MAP  734
     1   T3,*FLAG*,2X,*  ID *,2X,*   X    *,2X,*   Y    *,2X,*HT CORR*,  MAP  735
805  1 3X,*G TYPE*,                                                       MAP  736
     12X,*DAY NO*,2X,*NIGHT NO*,2X,*MIN*,2X,*MAX*,2X,*T ID*,2X,*FLAG*,    MAP  737
     12X,*HGT*,                                                          MAP  738
     2,//)                                                               MAP  739
      718 FORMAT                                                          MAP  740
810   1     (T5,A1,17A1,                                                  MAP  741
     1             22X,A2,4X,F5.0,3X,F5.0,4X,I2,3X,I2,3X,A3,3X,I2,F7.1,  MAP  742
     2              39A1)                                                MAP  743
      719  FORMAT(///,T3 ,*]]]]]      ERROR -- EOF ENCOUNTERED WHILE READING TMAP  744
     1ARGET CARDS*)                                                       MAP  745
815   720  FORMAT(///,T3 ,*]]]]]      ERROR -- SOURCE DEFINITION CARD ENDS IMPMAP  746
     1ROPERLY (WITH A EOF)     .....*,//)                                 MAP  747
      721  FORMAT(/,T15,*TOTAL PER DAY DAY FIRINGS IS   *,F10.2,          MAP  748
     1/,T15,*TOTAL PER DAY NIGHT FIRINGS IS *,F10.2)                      MAP  749
      722 FORMAT(T3,*.....      ERROR -- UNDEFINED TARGET ID*,A5,          MAP  750
820   1* FOR FIRING PT.*,A3)                                              MAP  751
      723 FORMAT(//T15,*.....      DATA BASE TIME PERIOD:*,F5.0,* DAY(S)*)  MAP  752
      724 FORMAT(//T15,*.....      GUN TYPE CARDS      .....*//T10,*FLAG*,2X, MAP  753
     1*G TYPE*,2X,*T CHARGE*,40X,*PROPELLANT WEIGHTS*//)                  MAP  754
      725 FORMAT(T12,A1,4X,A3,3X,F9.2,2X,10F9.2)                          MAP  755
825   726 FORMAT(  T3,*]]]]]      ERROR -- GUN TYPES EXCEED TABLE LIMIT IN MAMAP  756
     1P;ONLY FIRST*,I3,* TYPES USED FOR SUBSEQUENT CROSS-CHECKING*)       MAP  757
      727 FORMAT(  T3,*]]]]]      ERROR -- UNDEFINED GUN     ID*,A5,       MAP  758
     1* FOR FIRING PT.*,A3)                                              MAP  759
      728 FORMAT(  T3,*]]]]]      ERROR   -- BLANK TARGET ID, HIT FLAG NOT SEMAP  760
830   1T: FIRING PT.*,A3)                                                 MAP  761
      729 FORMAT(//T3,*]]]]]      ERROR -- EOF ENCOUNTERED WHILE READING GUN MAP  762
     1 TYPE DEFINITION CARDS*//)                                          MAP  763
      730 FORMAT(/T3,*.....*,I4,* ERROR/WARNING CONDITIONS DETECTED FOR THISMAP  764
     1 CARD TYPE*)                                                        MAP  765
835   731 FORMAT(T3,*.....      ERROR -- NO POSITIVE CHARGE FOR GUN TYPE*)  MAP  766
      732 FORMAT(T3,*.....      ERROR -- *,I3,* NEG. CHARGES ENCOUNTERED*)  MAP  767
      733 FORMAT(T3,*.....      WARNING -- *,I3,* CHARGES LARGER THAN*,F4.0, MAP  768
     1* LBS*)                                                             MAP  769
      734 FORMAT(T3,*.....      ERROR -- DUPLICATE ID;FIRST OCCURRENCE USED FMAP  770
840   1OR TABLE*)                                                         MAP  771
      735 FORMAT(T3,*.....      WARNING -- EXTRANEOUS DATA STARTING IN CARD CMAP  772
     1OL.*,                                                               MAP  773
     1I3,*) CHECK ALL FIELDS*)                                            MAP  774
      736 FORMAT(T3,*.....      ERROR -- HEIGHT CORRECTION DATA OUT OF RANGE*MAP  775
845   1)                                                                  MAP  776
      737 FORMAT(T3,*.....      WARNING -- DUP. POINT: IDENTICAL COORDINATES*MAP  777
     1)                                                                   MAP  778
      738 FORMAT(/T3,*.....      ERROR -- TARGETS EXCEED TABLE LIMIT IN MAP; MAP  779
     1ONLY FIRST*,I3,  USED FOR SUBSEQUENT CROSS-CHECKING*/)              MAP  780
850   739 FORMAT(T3,*.....      ERROR -- FIRINGS DATA NEGATIVE OR BOTH ZERO OMAP  781
     1N DEF. CARD FOR FIRING PT.*,A3)                                     MAP  782
      740 FORMAT(T3,*.....      ERROR -- INVALID CHARGE NO.:NONPOSITIVE OR NMAP  783
     1 GUN TABLE ENTRY; DEF CARD FOR FIRING PT. *,A3)                     MAP  784
      741 FORMAT(T3,*.....      ERROR -- HIT GREATER THAN MAX CHARGE NO.; DEFMAP  785
855   1 CARD FOR FIRING PT.*,A3)                                          MAP  786
```

```
                742 FORMAT(T5,*.....    WARNING -- LARGE WEIGHT VALUE IN DEF. CARD FORMAP  787
                   1FIRING PT. *,A3)                                                MAP  788
                743 FORMAT(/T3,*.....    ERROR -- FIRING PTS. EXCEED TABLE LIMIT IN MAMAP  789
                   1P; ONLY FIRST*,I3,* USED FOR SUBSEQUENT CROSS CHECKING*/)       MAP  790
      860       744 FORMAT(//T3,*..... END OF INPUT PHASE:*,I5,* ERROR/WARNING CONDITIMAP  791
                   1ONS DETECTED*)                                                  MAP  792
                745 FORMAT(/1X,*CROSS-REFERENCE: TARGETS BY FIRING POINTS ; DAILY FIRIMAP  793
                   1NGS*////                                                        MAP  794
                   1      1X,*TARGET ID*/10X,7(*FPT ID          *))                MAP  795
      865       746 FORMAT(/1X,*.....     WARNING -- DUE TO PREVIOUS TABLE OVERFLOW, TMAP  796
                   1HE FOLLOWING CROSS-REFERENCE TABLE IS INCOMPLETE*/)             MAP  797
                748  FORMAT(/1X,*CROSS-REFERENCE: TARGETS BY GUN TYPES;  DAILY *
                   1,*PROJECTILE CHARGE WEIGHT (LBS)*////
                   1              1X,*TARGET ID*/10X,7(*GUN ID          *))        MAP  800
      870       750   FORMAT(/1X,*CROSS-REFERENCE: GUN TYPES BY TARGETS;  DAILY *
                   1,*PROJECTILE CHARGE WEIGHT (LBS)*////
                   1              1X,*GUN ID*/10X,7(*TAR ID          * ))          MAP  803
                752 FORMAT                                                          MAP  804
                   1(/1X,*CROSS-REFERENCE: GUN TYPES BY FIRING POINTS; DAILY *
      875          2,*PROPELLENT CHARGE WEIGHT (LBS)*////
                   1                    1X,*GUN ID*/10X,7(*FPT ID          *))  MAP  807
                753 FORMAT(T3,*.....     ERROR -- DUPLICATE ID,COORDINATES; DEF CARDS MAP  808
                   1CHECKED FOR ERRORS,BUT OTHERWISE IGNORED*)                      MAP  809
                754 FORMAT(T3,*.....     ERROR -- DUPLICATE ID,DIFFERENT COORDINATES; MAP  810
      880          1TREATED AS SEPARATE ENTRY,*)                                    MAP  811
                755 FORMAT(T24,*GENERATED ID = *,A3)                                MAP  812
                756 FORMAT(/T3,*.....     ERROR --*,I3,* NONPOSITIVE CHARGE NOS. ENCOUMAP  813
                   1NTERED FOR FIRING PT. *,A3)                                     MAP  814
                757 FORMAT(T3,*.....     ERROR -- NONPOSITIVE TARGET CHARGE IN TABLE FMAP  815
      885          1OR GUN *,A3,* ;DEF CARD FOR FIRING PT.*,A3)                     MAP  816
                899 FORMAT(///T3,*)))))     ERROR -- MISSING DATA BASE FILE; EXECUTIONMAP  817
                   1 ABORTED*)                                                      MAP  818
                900  FORMAT(//,T15,*.....     TIME FOR MAPPING SUBPROGRAM IS *,F8.3,  MAP  819
                   1* SECONDS*)                                                     MAP  820
      890           END                                                           MAP  821
```

SYMBOLIC REFERENCE MAP (R=3)

ENTRY POINTS DEF LINE REFERENCES
1 MAP 1 752

VARIABLES	SN	TYPE	RELOCATION								
4335 ADEPT		REAL	ARRAY	REFS	5	12	727	745			
4171 ADGUN		REAL	ARRAY	REFS	8	12	734	739	745		
4253 ADTRG		REAL	ARRAY	REFS	F	12	727	734	739		
2011 BIGCHG		REAL		REFS	137	150	DEFINED	30			
5011 CHARGE		REAL	ARRAY	REFS	5	121	123	130	137	143	161
					498	517	571	DEFINED	111	121	
4072 CHGVAL		REAL		REFS	520	585	691	694	DEFINED	55	520
					585	691					
4073 CHGVAN		REAL		REFS	521	586	692	694	DEFINED	56	521
					586	692					
0 CHECK		LOGICAL		REFS	17	1F					

VARIABLES	SN	TYPE	RELOCATABLE							
4147	CMG	REAL		REFS	499	2*500	518	2*519	520	521	616
				DEFINED	498	513	517	528			
4070	CMLRG	REAL		REFS	519	694	DEFINED	53	519		
4071	CMSML	REAL		REFS	500	694	DEFINED	54	500		
21612	CONTUR	REAL	ARRAY	REFS	15	192	DEFINED	111			
4136	DARR	REAL		REFS	438	447	479	480	483	521	586
					632	DEFINED	430	436			
4075	DARRN	REAL		REFS	483	687	689	690	DEFINED	57	483
					689						
4135	DAY	REAL		REFS	437	447	479	480	482	520	585
					632	DEFINED	430	437			
4074	DAYN	REAL		REFS	482	687	688	690	DEFINED	57	482
					688						
4157	DAYS	REAL		REFS	685	688	689	691	692	727	734
					739	745	DEFINED	679	681		
4051	DUP	LOGICAL		REFS	22	481	520	521	585	586	599
				DEFINED	368	398					
4160	DX	REAL		REFS	709	DEFINED	700				
4164	DX1	REAL		REFS	711	DEFINED	709				
4161	DY	REAL		REFS	710	DEFINED	701				
4165	DY1	REAL		REFS	711	DEFINED	710				
6305	FCORDS	REAL	ARRAY	REFS	8	2*391	415	416	DEFINED	372	373
					384	385					
4163	GRDSZ	REAL		REFS	709	710	711	DEFINED	707		
0	GTxFPT	REAL	ARRAY GRID	REFS	7	8	610	745	DEFINED	96	618
4102	HABV	REAL		REFS	589	2*695	DEFINED	60	589		
4103	HBEL	REAL		REFS	588	2*695	DEFINED	60	588		
4056	HDRRTN	INTEGER		REFS	26	643	DEFINED	321	429		
4143	HGT	REAL		REFS	443	447	587	2*588	2*589	632	
				DEFINED	430	443					
4104	I	INTEGER		REFS	69	94	96	101	5*111	2*119	2*121
					2*123	130	137	2*143	153	154	155	2*161
					170	171	183	249	255	261	262	263
					287	288	306	369	376	382	383	384
					385	408	617	618	661	662		
				DEFINED	68	93	95	100	110	166	171	
					176	209	271	288	293	320	399	618
					623	661	705					
2002	IBLNK	INTEGER		REFS	228	343	453	543	DEFINED	27		
4122	ID	INTEGER		REFS	157	218	223	250	257	261	300	
					331	338	371	378	382	409	473	490
					511	532	538	550	562	581	595	618
				DEFINED	155	210	219	323	331	408		
4335	IDFPT	INTEGER	ARRAY	REFS	8	12	378	DEFINED	371	382		
4171	IDGUN	INTEGER	ARRAY	REFS	6	12	119	123	143	155	157	
					161	192	467	DEFINED	111	119		
4141	IDT	INTEGER		REFS	442	447	543	556	562	632		
				DEFINED	430	442						
4253	IDTRG	INTEGER	ARRAY	REFS	6	12	257	556	DEFINED	250	261	
4107	IFLAG	INTEGER		REFS	85	123	143	161	169	223	286	
					300	336	447	612	632	DEFINED	81	111
					210	323	430					
4133	IFP	INTEGER		REFS	601	2*603	609	2*610	DEFINED	370	383	
					597	629						
4145	IG	INTEGER		REFS	467	496	498	516	517	568	571	
					605	2*607	2*610	DEFINED	466	476		
4113	IGF	INTEGER		REFS	89	741	DEFINED	81				

VARIABLES	SN	TYPE	RELOCATION								
4112	IGT	INTEGER		REFS	88	735	DEFINED	81			
4166	IJ	INTEGER		REFS	727	734	745	DEFINED	720		
4047	IMAGE	LOGICAL		REFS	21	142	160	233	240	266	279
					348	356	388	419	458	470	486	503
					524	535	546	559	577	591		
				DEFINED	124	144	162	225	302	340	450
					634						
23170	IN	INTEGER	ARRAY	REFS	16	69	DEFINED	28			
4106	IN1	INTEGER		REFS	113	212	325	432	DEFINED	77	
				I/O REFS	75	111	210	323	430		
4114	IO1	INTEGER		REFS	83	DEFINED	81				
4150	IT	INTEGER		REFS	556	600	2*603	2*607	DEFINED	544	555
					565						
4110	ITF	INTEGER		REFS	87	723	DEFINED	81			
4111	ITG	INTEGER		REFS	88	730	DEFINED	81			
4134	ITYPE	INTEGER		REFS	436	447	467	473	581	632	
				DEFINED	430	436					
4105	IU	INTEGER		REFS	72	77	DEFINED	69	I/O REFS	70	71
4115	J	INTEGER		REFS	94	96	101	2*111	2*121	123	130
					137	143	161	192	210	223	228	236
					257	275	276	300	323	338	343	351
					378	2*391	415	416	2*430	2*447	453	462
					2*632	DEFINED	92	99	2*111	120	123	129
					143	161	192	210	223	227	256	274
					300	323	338	342	377	414	2*430	2*447
					452	2*632						
4167	JI	INTEGER		REFS	739	DEFINED	722				
4123	K	INTEGER		REFS	157	4*192	DEFINED	156	191		
0	KARD	INTEGER	IC	REFS	6	680	704	I/O REFS	81	679	703
				707							
4116	KERR	INTEGER		REFS	133	139	165	197	198	237	244
					270	284	308	309	353	361	394	404
					423	464	474	491	506	527	539	551
					563	582	596	664	665	DEFINED	109	133
					139	165	199	237	244	270	284	310
					353	361	394	404	423	464	474	491
					506	527	539	551	563	582	596	
4142	KFLAG	INTEGER		REFS	441	447	545	570	632			
				DEFINED	430	441						
1	KPRINT	INTEGER	IC	REFS	0	I/O REFS	44	75	107	123	143
					146	148	150	161	164	175	187	192
					197	205	206	223	236	243	269	283
					292	300	308	317	318	338	351	360
					393	403	409	422	447	462	473	490
					511	532	538	550	562	581	595	622
					632	640	641	664	670	671	673	675
					676	677	685	687	690	694	695	696
					697	698	699	711	724	725	726	731
					732	733	736	737	738	742	743	744
					751	756	760	763				
4144	KPRNTO	INTEGER		REFS	635	DEFINED	459	471	487	504	525
					536	547	560	578	592			
4131	KPRNTT	INTEGER		REFS	303	DEFINED	234	241	267	280	349
					357	389	420					
4064	LINES	INTEGER		REFS	336	352	362	395	405	410	424
					445	449	463	475	492	512	533	540
					552	564	583	597	615	DEFINED	49	352

VARIABLES	SN	TYPE	RELOCATION							
				382	395	405	419	424	449	363	475
				496	512	533	542	552	564	583	597
				615	642						
4052	LP1	LOGICAL		REFS	23	232	236	347	351	457	462
				DEFINED	83						
4120	LPS	INTEGER		REFS	138	141	2*150	DEFINED	127	138	
4140	MAX	INTEGER		REFS	440	447	2*514	2*515	517	529	534
				632	DEFINED	438	447				
4125	MAXPTS	INTEGER		REFS	658	DEFINED	203	315			
4137	MIN	INTEGER		REFS	439	447	2*494	2*495	498	507	534
				632	DEFINED	438	439				
4162	N	INTEGER		REFS	705	DEFINED	733				
6057	NAME1	INTEGER	ARRAY	REFS	8	192	DEFINED	111			
21245	NAME2	INTEGER	ARRAY	REFS	8	192	DEFINED	111			
4060	NCARDS	INTEGER		REFS	115	214	327	434	673		
				DEFINED	45	115	214	327	434		
21531	NDCOLS	INTEGER	ARRAY	REFS	14	462	DEFINED	33			
4117	NEG	INTEGER		REFS	132	141	2*148	508	530	2*614	
				DEFINED	126	132	322	508	530		
4065	NERR	INTEGER		REFS	198	309	665	670	DEFINED	50	198
				304	665						
2003	NFILES	INTEGER		REFS	68	DEFINED	28				
4063	NG	INTEGER		REFS	117	675	DEFINED	48	117		
4066	NLRG	INTEGER		REFS	514	694	DEFINED	51	514		
4062	NS	INTEGER		REFS	329	677	678	DEFINED	47	329	
4067	NSML	INTEGER		REFS	494	694	DEFINED	52	494		
4156	NS=CS	INTEGER		REFS	727	745	DEFINED	662			
4061	NT	INTEGER		REFS	216	676	DEFINED	96	216		
4132	NTAB	INTEGER		REFS	555	727	734	739	DEFINED	306	
21447	NTCOLS	INTEGER	ARRAY	REFS	14	236	351	DEFINED	33		
4121	NTD	INTEGER		REFS	156	166	256	271	274	377	399
				414	DEFINED	154	255	376			
4124	NTYPES	INTEGER		REFS	191	466	734	739	745		
				DEFINED	183						
21327	NUL	INTEGER	ARRAY	REFS	14	3*223	228	3*300	3*338	343	2*447
				453	2*632	DEFINED	3*210	3*323	2*430		
2010	NUMCHG	INTEGER		REFS	111	120	123	129	143	161	495
				515	DEFINED	30					
2006	NUMFPT	INTEGER		REFS	92	617	622	623	662	727	745
				DEFINED	30						
2005	NUMGT	INTEGER		REFS	95	99	173	175	176	183	734
				739	745	DEFINED	30				
2007	NUMLIN	INTEGER		REFS	336	345	DEFINED	30			
2004	NUMTRG	INTEGER		REFS	93	100	287	292	293	306	727
				734	739	DEFINED	30				
4046	OVNFL	LOGICAL		REFS	21	174	260	271	291	381	396
				406	621	661	DEFINED	108	177	207	298
				319	624						
4130	PMCORR	REAL		REFS	221	223	2*239	300	334	338	2*355
				DEFINED	210	221	323	334			
4050	POS	LOGICAL		REFS	21	141	146	DEFINED	128	136	
4045	PRNT	LOGICAL		REFS	20	123	124	184	223	225	336
				336	340	445	447	450	DEFINED	85	
1	REED	LOGICAL	DEMOD	REFS	17	18	DEFINED	42			
4044	REOF	LOGICAL		REFS	19	DEFINED	43	642			
4146	SMOIS	REAL		REFS	484	599	603	607	610		
				DEFINED	480	485					

VARIABLES	SN	TYPE	RELOCATION							
4151	TCHG	REAL		REFS	572	585	586	607	DEFINED	569	571
				573							
6141	TCOMDS	REAL	ARRAY	REFS	8	275	276	DEFINED	251	252	262
				263							
0	TXFPT	REAL	ARRAY FT	REFS	5	8	603	727	DEFINED	94	603
7435	TXGT	REAL	ARRAY	REFS	8	607	734	739	DEFINED	101	607
4057	T100	REAL		REFS	39	750					
4170	T99	REAL		REFS	749	750	751	DEFINED	750		
4055	XGF	INTEGER		REFS	25	91	180	609	627	743	
				DEFINED	63	89	180	627			
4126	XLOC	REAL		REFS	219	223	251	262	275	300	332
				336	372	384	391	415	646	647	650
				652	653	656	DEFINED	210	219	323	332
4076	XLRG	REAL		REFS	646	2*696	700	DEFINED	58	647	
4153	XLRG1	REAL		REFS	697	DEFINED	650				
14341	XREF	REAL	ARRAY	REFS	8	727	734	739	745		
4100	XSML	REAL		REFS	652	2*698	700	DEFINED	59	653	
4155	XSML1	REAL		REFS	699	DEFINED	656				
4053	XTF	INTEGER		REFS	25	91	297	601	626	725	
				DEFINED	61	87	297	626			
4054	XTG	INTEGER		REFS	25	98	179	296	606	732	737
				DEFINED	62	88	179	296			
4127	YLOC	REAL		REFS	220	223	252	263	276	300	333
				336	373	385	391	416	648	649	651
				654	655	657	DEFINED	210	220	323	333
4077	YLRG	REAL		REFS	649	2*697	701	DEFINED	58	651	
4152	YLRG1	REAL		REFS	696	DEFINED	648				
4101	YSML	REAL		REFS	655	2*699	701	DEFINED	59	657	
4150	YSML1	REAL		REFS	698	DEFINED	654				

FILE NAMES	MODE			
TAPE6	BUT	WRITES	614	
VARIABLES USED AS FILE NAMES, SEE ABOVE				

EXTERNALS	TYPE	ARGS	REFERENCES					
EOF	REAL	1	72	113	212	325	432	680	704
PUTBB		10	727	734	739	745			
SECOND		1	39	749					

INLINE FUNCTIONS	TYPE	ARGS	DEF LINE	REFERENCES			
ABS	REAL	1	INTRIN	587	2*695	700	701
MIN0	INTEGER	0	INTRIN	183	306	662	
SHIFT	NO TYPE	2	INTRIN	408			

STATEMENT LABELS			DEF LINE	REFERENCES		
0	5		73	68		
47	6		77	72		
0	10		94	93		
0	11		96	95		
0	14		97	92		
111	15		98	91		
0	16		101	99	100	
123	21		112	98		
137	50		111	172	174	191
0	55		121	120		
0	58	INACTIVE	132	130		
227	59		136	130		

STATEMENT LABELS		PER LINE REFERENCES				
232	60	128	129	130	134	137
251	65	140	142			
262	70	151	141			
0	75	154	156			
272	76	160	157			
302	77	164	160			
307	80	167	153	159		
315	85	174	176			
332	90	183	169			
0	91	195	191			
372	95	197	184			
1458	100	644	646			
1461	105	652	649			
1466	115	655	652			
1473	120	658	655			
405	200	210	289	291	298	
0	210	229	227			
436	211	232	228			
443	212	236	233	234	303	
451	213	237	232			
453	220	239	230			
461	222	243	240	241	303	
465	230	246	239			
466	231	247	203	658		
475	239	255	249			
0	240	258	256			
512	241	266	257			
516	242	269	266	267	303	
524	245	273	260	264		
533	246	277	274	275		
535	247	279	276			
541	248	283	279	280	303	
545	250	286	253	278		
553	260	291	287			
574	290	300	286			
1676	299	754	212	678		
612	300	321	619	621	630	
636	301	337	321	643		
0	310	344	342			
650	311	347	343			
655	312	351	303	348	349	
665	313	353	347			
667	320	355	345			
675	322	360	303	356	357	
702	330	365	355			
703	331	366	315	658		
713	339	376	369			
0	340	379	377			
731	341	388	378			
735	342	391	303	388	389	
723	343	382	411			
754	344	403	391			
767	345	413	381	386	400	406
776	346	417	414	415		
1000	347	419	416			
1004	348	422	303	419	420	
1011	350	425	374	418		

STATEMENT LABELS			OFF LINE	REFERENCES						
1701	394		758	432						
1912	400		430	612						
1043	401		446	429	643					
0	405		454	452						
1057	406		457	453						
1064	407		462	458	459	635				
1074	408		464	457						
1076	409		466	455						
0	410		468	466						
1107	411		473	470	471	635				
1115	415		477	467						
1125	419		484	481						
1127	420		485	479						
1133	421		490	486	487	635				
1140	430		494	484						
1156	434		503	495	499					
1161	435		506	503	504	635				
1166	436		511	507						
1172	440		513	496	501	509				
1221	444		524	515	518					
1224	445		527	524	525	635				
1235	446		534	516	522	531				
1242	447		538	535	536	635				
1231	448		532	529						
1247	450		543	534						
1256	451		550	546	547	635				
1263	460		555	543						
0	465		557	555						
1273	467		562	559	560					
1301	468		568	556						
1314	469		581	577	578	635				
1321	470		585	545	553	566	568	570	572	
1342	475		591	587						
1345	477		595	591	592					
1352	480		599	590						
1363	485		605	601						
1373	487		609	600	606					
1403	490		612	599	605	609				
1416	491		621	617						
1704	499		763	113						
1475	500		661	325						
1531	501		683	680						
0	505		712	705						
1611	510		713	704						
1626	560		730	723						
1641	565		735	730						
1654	570		741	735						
1667	590		747	741						
2735	600	FMT	765	81						
2737	601	FMT	766	210	323					
2743	603	FMT	767	430						
2751	604	FMT	768	703						
2753	605	FMT	769	679	707					
2755	607	FMT	770	695	640	671	724	741	736	742
2757	609	FMT	771	71	111					
2764	699	FMT	772	675						
2772	700	FMT	773	44						

STATEMENT	LABELS		DEF LINE	REFERENCES						
3001	701	FMT	774	673						
3010	702	FMT	775	676						
3015	703	FMT	776	677						
3022	704	FMT	777	687						
3033	705	FMT	779	695						
3043	708	FMT	781	696						
3053	709	FMT	783	697						
3063	710	FMT	785	698						
3073	711	FMT	787	699						
3103	712	FMT	789	711						
3113	713	FMT	791	694						
3147	714	FMT	797	206						
3163	715	FMT	799	223	300	338				
3170	716	FMT	800	317						
3221	718	FMT	809	447	632					
3231	719	FMT	813	756						
3242	720	FMT	815	760						
3255	721	FMT	817	690						
3270	722	FMT	819	562						
3301	723	FMT	821	685						
3310	724	FMT	822	107						
3325	725	FMT	824	123	143	161				
3331	726	FMT	825	175						
3347	727	FMT	827	473						
3360	728	FMT	829	550						
3371	729	FMT	831	763						
3403	730	FMT	833	197	308	664				
3414	731	FMT	835	146						
3423	732	FMT	836	148						
3432	733	FMT	837	150						
3442	734	FMT	839	164	269					
3453	735	FMT	841	236	351	462				
3466	736	FMT	844	243	360					
3476	737	FMT	846	283	422					
3506	738	FMT	848	292						
3523	739	FMT	850	490						
3536	740	FMT	852	511	532					
3552	741	FMT	854	538						
3564	742	FMT	856	595						
3575	743	FMT	858	622						
3612	744	FMT	860	670						
3623	745	FMT	862	726						
3640	746	FMT	865	725	732	737	743			
3655	748	FMT	867	733						
3674	750	FMT	870	738						
3713	752	FMT	873	744						
3732	753	FMT	877	393						
3746	754	FMT	879	403						
3760	755	FMT	881	409						
3764	756	FMT	882	614						
3776	757	FMT	884	581						
4012	899	FMT	886	75						
4023	900	FMT	888	751						
3201	7161	FMT	803	318	641					
2127	7241	FMT	188	187						
2153	7251	FMT	194	192						
567	29991		300	235	242	268	281	350	358	390	421

```
STATEMENT LABELS                DEF LINE   REFERENCES
 1433  49991                       632      460       472       488       505       526       537       548       561       579
                                              593
 1440  99991                       639      336       445
 1447  99992                       645      246       365

LOOPS  LABEL   INDEX    FROM-TO    LENGTH    PROPERTIES
   31  5       I          68 73       138              EXT REFS    EXITS
  101  14      J          92 97       108              NOT INNER
  103  10      I          93 94        28    INSTACK
  106  11      I.         95 96        28    INSTACK
  117  16      J          99 101       48              NOT INNER
  120  16      I         100 101       28    INSTACK
  154          J         111 111      108              EXT REFS
  201  55      J         120 121       38    INSTACK
  224  60      J         129 140       78    INSTACK
  267  75      K         156 158       28    INSTACK     EXITS
  344  91      K         191 195      268              EXT REFS  NOT INNER
  356          J         192 192      108              EXT REFS
  433  210     J         227 229       28    INSTACK     EXITS
  502  240     J         256 258       28    INSTACK     EXITS
  530  246     J         274 277       48    INSTACK     EXITS
  645  310     J         342 344       28    INSTACK     EXITS
  720  340     J         377 379       28    INSTACK     EXITS
  773  346     J         414 417       48    INSTACK     EXITS
 1054  405     J         452 454       28    INSTACK     EXITS
 1102  410     IG        466 468       38    INSTACK     EXITS
 1266  465     IT        555 557       28    INSTACK     EXITS
 1577  505     I         705 712      128              EXT REFS

COMMON BLOCKS   LENGTH     MEMBERS - BIAS NAME(LENGTH)
        FT       15000               0 TXFPT  (15000)
        IO           2               0 KARD   (1)                1 KPRINT (1)
        GPIO     15000               0 GTXFPT (15000)
        DEBUG        2               0 CHECK  (1)                1 PEED   (1)

EQUIV CLASSES   LENGTH     MEMBERS - BIAS NAME(LENGTH)
        ADGUN        50              0 IDGUN  (50)
        ADTRG        50              0 IDTRG  (50)
        ADFPT       300              0 IDFPT  (300)

STATISTICS
  PROGRAM LENGTH                 231738    9851
  CM LABELED COMMON LENGTH       724046   30004
          00000B CM USED
```

```
      SUBROUTINE READTB                                                 READTB 2
C READS INFORMATION FROM TABGEN (FOUND IN TAPE 20) AND MODIFIES IT ACCORDING TO
C THE INVERSION FACTORS GIVEN BY THE CALLING SUBROUTINE. THE TABLES ARE
C UNDER THE CONDITIONS OF STANDARD PERCENT TEMPERATURE INVERSION FACTORS.
      COMMON/I /FARG,PRINT                                              READTB 3
      COMMON/FACTI/ [illegible]
      COMMON/[illegible]/CHECK,REF,TAB                                  READTB 5
      COMMON/PARM/THRESH,RESULT
      LOGICAL [illegible],REF ,TAB                                      READTB 6
C     COMMON BLOCK TABLE CONTAINS THE TABLES OF PROGRAM TABGEN
C     FOUND IN TAPE 20.
C     OBV=TABLE OF [illegible] VALUES
C     PERV=TABLE OF PERCENTAGES
C     ENV=TABLE OF ENERGY VALUES
C     CSCF=TABLE OF CHARGE SIZE CORRECTION FACTORS
C     FON1 AND FON2 ARE USED IN F1 COMPUTATION
C     FTWO IS USED IN F2 COMPUTATION
      COMMON /TABL1/ OBV(301,9,2) , PERV(301,4,2) , ENV(1501)
     1 CSCF(601) ,FON1(301,4,2),FON2(301,4,2),FTWO(151,2)
      DATA IN1/20/                                                      READTB 8
      REWIND IN1                                                        READTB 9
C THIS READS THE STANDARD PERCENT TEMPERATURE INVERSION FACTORS FROM TAPE20.
      READ (IN1) PC1,PC2,PC3
C ERROR...OB,PERCENT CURVE TABLES MISSING--PROGRAM ABORTED*
      IF(EOF(IN1).GT.0.0) GO TO 999                                     READTB11
C     L=1 IS DAY
C     L=2 IS NIGHT
C     J=1 FOCUS MAX
C     J=2 FOCUS MEAN
C     J=3 BASE MAX
C     J=4 BASE MEAN
C     J=5 NEG MAX
C     J=6 NEG MEAN
C     J=7 EX NEG MAX
C     J=8 EX NEG MEAN
C     J=9 EX NEG MIN
      DO 20 L=1,2
      DO 20 J=1,9
C READ OB VALUES FORM TAPE 20 INTO ARRAY OBV.
      READ (IN1) (OBV(I,J,L),I=1,301)
20    CONTINUE
C     L=1 IS DAY
C     F=2 IS NIGHT
C     J=1 FOCUS
C     J=2 BASE
C     J=3 NEG
C     J=4 EX NEG
C
      DO 30 L=1,2
      DO 30 J=1,4
C READ PERCENT VALUES FROM TAPE 20 INTO ARRAY PERV.
      READ (IN1) (PERV(I,J,L),I=1,301)
30    CONTINUE
C CSCF=CHARGE SIZE CORRECTION FACTOR.
      READ (IN1) CSCF
C COMPUTE CORRECTION FACTORS FOR THE INVERSION FACTORS.  DISTANCE BETWEEN
C SOURCE AND POINT < 2 MILES.  DISTANCE> 10 MILES  2<DISTANCE<10
```

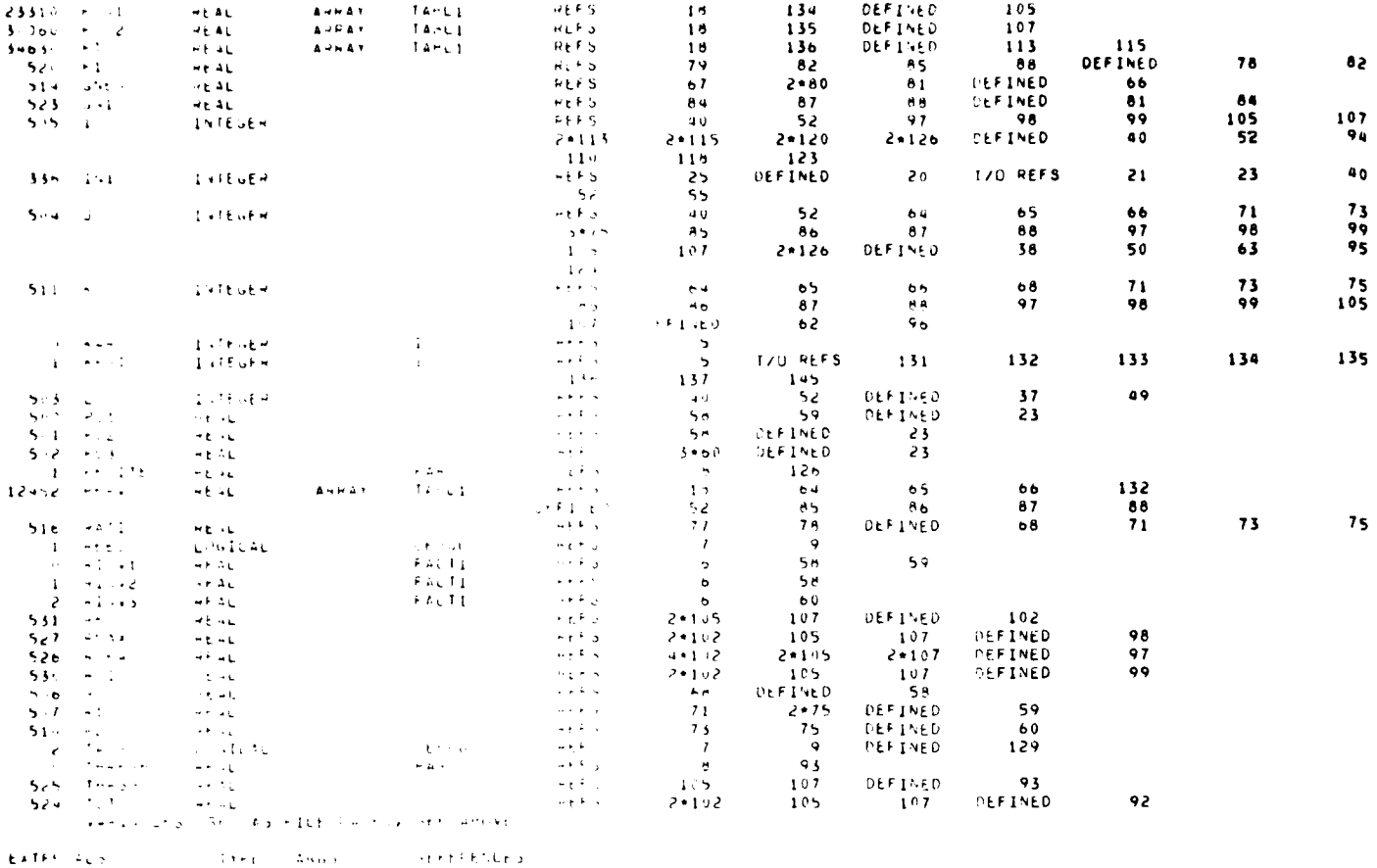
```
                R0=(RINV1+RINV2)/(PC1+PC2)                                      READTB14
                R1=RINV1/PC1                                                    READTB15
 60             R2= ((RINV3-PC3)/2. + PC3) /PC3
      C         CORRECT THE PERCENTAGE
                DO 100 K=1,2                                                    READTB17
                DO 100 J=1,301
                BASE=  PERV(J,2,K)
 65             FOCUS=  PERV(J,1,K)
                GNEG=  PERV(J,3,K)
                EXNEG=100.-(BASE+FOCUS+GNEG)                                    READTB22
                IF(K.EQ.1) RATIO=R0                                             READTB23
      C  2 MILE OR LESS         (152)                                           READTB24
 70   C         100*ALOG10((2 MILE)*(5280 FEET/MILE) *(.3048 METER/FEET)) =199
                IF(K.EQ.2.AND.J.LT.152)RATIO=R1
      C         10 OR GREATER        (22)
                IF (K.EQ.2.AND.J.GT.222) RATIO  =R2
      C BETWEEN 2 AND 10                                                        READTB28
 75             IF (K.EQ.2.AND. (J.LE.222.AND. J.GE.152))
               1  RATIO = (R2-R1) * (J-152) / 70.0  + R1
                B1=BASE*RATIO                                                   READTB30
                F1=FOCUS*RATIO                                                  READTB31
                DELB= BASE-B1 +FOCUS-F1                                         READTB32
 80             DELN=GNEG/(GNEG+EXNEG)*DELB                                     READTB33
                GN1=GNEG+DELN                                                   READTB34
                IF (F1.LT.0) F1=0.
                IF (B1.LT.0.) B1=0.
                IF (GN1.LT.0.) GN1=0.
 85             PERV(J,1,K)=  F1/100.
                PERV(J,2,K) = B1/100.
                PERV(J,3,K)= GN1/100.
                PERV(J,4,K)= (100.-F1-B1-GN1)/100.
            100 CONTINUE                                                        READTB47
 90   C         FUNE
      C F1 COMPUTATION
                TLT =10.0 / ALOG (10.0)
                THRSH=10.**(THRESH/10.)
                DO 50 I=1,301
 95             DO 50 J=1,4
                DO 50 K=1,2
                RMEAN  = DBV(I,J+2,K)
                RMAX=  DBV(I,J+2-1,K)
                RMIN=  DBV(I,J+2+1,K)
100   C         GET THE K FACTOR
      C GET THE K FACTOR
                RK= (TLT*( 10.**((RMAX-RMEAN)/10.0)-1.0) - (RMAX-RMEAN))  /
               1  ((RMEAN-RMIN) - (TLT * (1.0- 10.0**((RMIN-RMEAN)/10.0) )))
      C         CASE ONE
105             FUN1(I,J,K)=TLT*THRSH*RK/(RK*(RMEAN-RMIN)+(RMAX-RMEAN))
      C         CASE TWO
                FUN2(I,J,K)=TLT*THRSH/(RK*(RMEAN-RMIN)+(RMAX-RMEAN))
            50  CONTINUE
      C         FIN
110             DO 60 I= 1,151
      C         CASE 1
      C F2 COMPUTATION
                FUN2(I,1)= 1.0-10. **((1-I)/100.0)
      C         CASE 2
```

```
      FTAU(I,2) = 10.0**((I-1)/100.)-1.0
   60 CONTINUE
C     DB TO ENERGY
      DO 70 I=1,1501
C CHANGE DB TO ENERGY.  STORE IN ARRAY ENV.
      ENV(I) = 10.0 **((I+249) / 100.0)
   70 CONTINUE
C       CORRECT FOR NIGHT TIME VALUES
      DO 80 I=1,301
      DO 80 J=1,9
C NIGHTTIME CORRECTION FACTOR.
      DBV(I,J,2) = DBV(I,J,2) + PENITE
   80 CONTINUE
C LOGICAL FLAG IS TRUE AFTER THE EXECUTION OF THE SUBROUTINE
      TABHD=.TRUE.                                                    READTB48
      IF (.NOT.CHECK) RETURN
      WRITE(KPRINT,997)      DBV
      WRITE(KPRINT,996) PERV
      WRITE(KPRINT,995) CSCF
      WRITE(KPRINT,994) FON1
      WRITE(KPRINT,994) FON2
      WRITE(KPRINT,993) FTAU
      WRITE(KPRINT,992) ENV
  997 FORMAT(1H1,18(/,1H0,20(/,1H ,15F8.2),/,1H ,F8.2))
  996 FORMAT (1H1,8(/,1H0,20(/,1H ,15F8.4),/,1H ,F8.4))
  995 FORMAT(1H1,(15F8.2))
  994 FORMAT(1H1,8(/,1H0,30(/,1H ,10E12.5),/,1H ,E12.5))
  993 FORMAT(1H1,2(/,1H0,15(/,1H ,10E12.5),/,1H ,E12.5))
  992 FORMAT(1H1,151(/,1H ,10E12.5))
      RETURN                                                          READTB49
  999 WRITE(KPRINT,998)                                               READTB50
  998 FORMAT(//10X,*..... ERROR .....DB,PERCENT CURVE TABLES(TAPE20) MISREADTB51
     1SING -- PROGRAM ABORTED*)                                       READTB52
      STOP                                                            READTB53
      END                                                             READTB54
```

SYMBOLIC REFERENCE MAP (R=3)

ENTRY POINTS DEF LINE REFERENCES
1 READTB 1 130 144

VARIABLES	SN	TYPE		RELOCATION									
512	BASE	REAL			REFS	67	77	79	DEFINED	64			
517	B1	REAL			REFS	79	83	86	88	DEFINED	77	83	
0	CHECK	LOGICAL		DEBUG	REFS	7	9	130					
22157	CSCF	REAL	ARRAY	TABLI	REFS	18	133	DEFINED	55				
0	DBV	REAL	ARRAY	TABLI	REFS	18	97	98	99	126	131		
					DEFINED	40	126						
521	DELB	REAL			REFS	80	DEFINED	79					
522	DELN	REAL			REFS	81	DEFINED	80					
17222	ENV	REAL	ARRAY	TABLI	REFS	18	137	DEFINED	120				
515	ENEG	REAL			REFS	80	DEFINED	67					
513	FOCUS	REAL			REFS	67	78	79	DEFINED	65			



```
      SUBROUTINE READTB      74/175  OPT=2 ROUND=+-*/              FTN 4.8  508         80/10/30. 12.45.43       PAGE     5

STATEMENT LABELS                  DEF LINE   REFERENCES
    0  100                          89          62        63
  445  992      FMT                143         137
  440  993      FMT                142         136
  433  994      FMT                141         134       135
  430  995      FMT                140         133
  423  996      FMT                139         132
  416  997      FMT                138         131
  454  998      FMT                146         145
  323  999                         145          25

LOOPS  LABEL     INDEX       FROM-TO     LENGTH     PROPERTIES
   16  20        L            37  41       16B                EXT REFS  NOT INNER
   21  20        J            38  41       10B                EXT REFS
   41  30        L            49  53       16B                EXT REFS  NOT INNER
   44  30        J            50  53       10B                EXT REFS
   76  100       K            62  69       62B                NOT INNER
  103  100       J            63  69       51B        OPT
  170  50        I            94 108       55B                EXT REFS  NOT INNER
  174  50        J            95 108       46B                EXT REFS  NOT INNER
  200  50        K            96 108       35B                EXT REFS
  246  60        I           110 116       16B                EXT REFS
  265  70        I           118 121        7B                EXT REFS
  276  80        I           123 127        4B                NOT INNER
  277  80        J           124 127        2B     INSTACK

COMMON BLOCKS   LENGTH      MEMBERS - BIAS NAME(LENGTH)
       IO            2                 0 KARD    (1)              1 KPRINT (1)
       FACTI         3                 0 RINV1   (1)              1 RINV2  (1)              2 RINV3  (1)
       DEBUG         3                 0 CHECK   (1)              1 REED   (1)              2 TABRD  (1)
       PARM          2                 0 THRESH  (1)              1 PENITE (1)
       TABL1     15046                 0 DBV     (5418)        5418 PERV   (2408)        7826 ENV    (1501)
                                    9327 CSCF    (601)         9928 FON1   (2408)       12336 FON2   (2408)
                                   14744 FTWO    (302)

STATISTICS
  PROGRAM LENGTH                  532B     346
  CM LABELED COMMON LENGTH      35320B   15056
         60000B CM USED
```

```
      SUBROUTINE READIN                                                 READIN 2
C     THIS ROUTINE IS CALLED TO READ DATA FROM FORMA
      COMMON/IO/KARD,KPRINT                                             READIN 3
      COMMON /SRCS/ NSRCS
      COMMON/FT/GRDSZ,CNR,XLOC,YLOC,DAYNO,DARKNO                        READIN 5
      DIMENSION XLOC(2000),YLOC(2000),DARKNO(2000),DAYNO(2000)          READIN 6
      COMMON/GRID/NRCNR,EAS,RCNR,DIST,SDBRM,          ANGCOS,ANGSIN
      DIMENSION RCNR(10),DIST(10),SDBRM(2000),
     1ANGCOS(2000),ANGSIN(2000)                                         READIN 9
      COMMON/DEBUG/CHECK,REED                                           READIN10
      LOGICAL CHECK,REED                                                READIN11
      DATA IN1/8/                                                       READIN12
C     READ NUMBER OF NOISE SOURCES AND NUMBER OF DAYS
      READ(IN1) NSRCS,DAYS
C     READ XLOC AND YLOC FOR TARGETS AND FIRING POINTS
      READ(IN1) XLOC                                                    READIN16
      READ(IN1) YLOC                                                    READIN17
C     READ CHARGE SIZES
      READ(IN1) SDBRM                                                   READIN18
C     READ SINE AND COSINE FOR ANGLE BETWEEN GUN AND TARGET
      READ(IN1) ANGSIN                                                  READIN19
      READ(IN1) ANGCOS                                                  READIN20
C     NUMBER OF DAY FIRINGS AND NIGHT FIRINGS
      READ(IN1) DAYNO                                                   READIN21
      READ(IN1) DARKNO                                                  READIN23
C     REWIND FILE
      REWIND IN1                                                        READIN25
C     SET FLAG
      REED=.TRUE.                                                       READIN26
      RETURN                                                            READIN68
      END                                                               READIN69
```

```
      SYMBOLIC REFERENCE MAP (R=3)

ENTRY POINTS      DEF LINE      REFERENCES
    1  READIN           1        30

VARIABLES     SN  TYPE           RELOCATION
 3746  ANGCOS      REAL       ARRAY    GRID          REFS        7          8    DEFINED       22
 7666  ANGSIN      REAL       ARRAY    GRID          REFS        7          8    DEFINED       21
    0  CHECK       LOGICAL             DEBUG         REFS       10         11
    1  CNR         REAL                FT            REFS        5
13502  DARKNO      REAL       ARRAY    FT            REFS        5          6    DEFINED       25
 7642  DAYNO       REAL       ARRAY    FT            REFS        5          6    DEFINED       24
   71  DAYS      * REAL                              DEFINED     14
   14  DIST        REAL       ARRAY    GRID          REFS        7          8
    1  EAS         REAL                GRID          REFS        7
    0  GRDSZ       REAL                FT            REFS        5
   26  IN1         INTEGER                           DEFINED     12    I/O REFS       14         16         17         19         21
                                                                 22         25         27
    0  KARD        INTEGER             IO            REFS        3
    1  KPRINT      INTEGER             IO            REFS        3
    0  NRCNR       INTEGER             GRID          REFS        7

      SUBROUTINE READIN     74/175  OPT=2 ROUND=+-*/          FTN 4.8  508       80/10/30. 12.45.43       PAGE      2

VARIABLES     SN  TYPE           RELOCATION
    0  NSHCS       INTEGER             SHCS          REFS        4    DEFINED       14
    2  RCNR        REAL       ARRAY    GRID          REFS        7          8
    1  REED        LOGICAL             DEBUG         REFS       10         11    DEFINED       29
   26  SDBMM       REAL       ARRAY    GRID          REFS        7          8    DEFINED       19
    2  XLOC        REAL       ARRAY    FT            REFS        5          6    DEFINED       16
 3722  YLOC        REAL       ARRAY    FT            REFS        5          6    DEFINED       17
       VARIABLES USED AS FILE NAMES, SEE ABOVE

COMMON BLOCKS   LENGTH      MEMBERS - BIAS NAME(LENGTH)
       IO            2              0 KARD   (1)          1 KPRINT (1)
       SHCS          1              0 NSHCS  (1)
       FT         8002              0 GRDSZ  (1)          1 CNR    (1)          2 XLOC   (2000)
                                 2002 YLOC   (2000)     4002 DAYNO  (2000)     6002 DARKNO (2000)
       GRID       6022              0 NRCNR  (1)          1 EAS    (1)          2 RCNR   (10)
                                   12 DIST   (10)        22 SDBMM  (2000)     2022 ANGCOS (2000)
                                 4022 ANGSIN (2000)
       DEBUG         2              0 CHECK  (1)          1 REED   (1)

STATISTICS
  PROGRAM LENGTH                  728       58
  CM LABELED COMMON LENGTH     333158    14029
        600008 CM USED
```

```
      SUBROUTINE POINT
C     SEE ZDEF FOR THE DESCRIPTION OF THESE VARIABLES
      COMMON/IO/KARD,KPRINT                                          PGRID  3
      COMMON/BOUND/XMIN,YMIN,XMAX,YMAX,BDS                           PGRID  4
      LOGICAL BDS                                                    PGRID  5
      COMMON /SHLS/ NSHLS
      COMMON/FT/GRDSZ,CNR,XLOC,YLOC,DAYNO,DARKNO                     PGRID  7
      DIMENSION XLOC(2000),YLOC(2000),DARKNO(2000),DAYNO(2000)       PGRID  8
      COMMON/FACTI/ RINV1,RINV2,RINV3
      COMMON/DEBUG/CHECK,REED,TABRD                                  PGRID 10
      COMMON/CALC/DAYCNR,DRKCNR                                      PGRID 11
      COMMON/METRIC/METERS,IMETER,IBOTH                              PGRID 12
      LOGICAL METERS                                                 PGRID 13
      LOGICAL DAYCNR,DRKCNR                                          PGRID 14
      LOGICAL OUT
C     FLAGS FOR READING TAPE5 INFO(REED), TAPE20 INFO(TABRD)
      LOGICAL CHECK,REED,TABRD                                       PGRID 15
      CALL SECOND(T100)                                              PGRID 22
      WRITE(KPRINT,610)                                              PGRID 23
C IF THE DATA IS NOT ALREADY READ IN FOR USE IN POINT, THEN CALL SUBROUTINE
C READIN TO READ THE DATA FROM TAPE 5.
      IF (.NOT. REED) CALL READIN                                    PGRID 24
C IF GRID BOUNDARIES ARE NOT PROPERLY INITIALIZED, COMPLAIN.
C GRID BOUNDARIES NOT PROPERLY INITIALIZED -- ABORT                  PGRID 25
      IF(BDS) GO TO 10
      WRITE(KPRINT,5)
5     FORMAT(*  ... NO BOUNDS CALL BEFORE POINT CALL...*)
      STOP
C READ INPUT DIRECTIVE,SET DEFAULTS,IF NECESSARY                     PGRID 28
C THIS READS IN THE CARD CONTAINING THE INVERSION FACTORS, AND DAY/NIGHT/BOTH
10    READ(KARD,700) X1,X2,X4,X3
C ERROR MESSAGE:  ERROR--MISSING INPUT DIRECTIVE; JOB ABORTED
      IF(EOF(KARD).NE.0.)GO TO 300                                   PGRID 30
C INVERSION FACTORS                                                  PGRID 31
C THIS SECTION READS, VERIFIES OR FINDS OUT ABOUT THE INVERSION FACTORS USED
C FROM SUBROUTINE READIB OR THE DIRECTIVE CARD.
      IF(.NOT.TABRD)GO TO 50                                         PGRID 32
      IF(X1.EQ.RINV1.AND.X2.EQ.RINV2.AND.X4.EQ.RINV3)   GOTO 75
      TABRD=.FALSE.                                                  PGRID 34
   50 RINV1=X1                                                       PGRID 35
      RINV2=X2                                                       PGRID 36
      RINV3= X4
      CALL READTB                                                    PGRID 37
   75 CONTINUE                                                       PGRID 38
C DAY +/OR NIGHT CONSIDERED                                          PGRID 42
      DAYCNR=(X3.EQ.10HBOTH       .OR. X3.EQ.10HDAY          )     PGRID 43
      DRKCNR=(X3.EQ.10HBOTH       .OR. X3.EQ.10HNIGHT        )     PGRID 44
      IF(DAYCNR.OR.DRKCNR)GO TO 90                                   PGRID 45
      DAYCNR=.TRUE.                                                  PGRID 46
      DRKCNR=.TRUE.                                                  PGRID 47
   90 CONTINUE                                                       PGRID 48
      IF (DRKCNR) IBOTH=10HNIGHT ONLY                                PGRID 49
      IF (DAYCNR) IBOTH=10H  DAY ONLY                                PGRID 50
      IF (DAYCNR .A. DRKCNR) IBOTH=10HBOTH D + N                     PGRID 51
C PRINT PARAMETERS                                                   PGRID 60
      WRITE(KPRINT,620) RINV1,RINV2,RINV3
      WRITE(KPRINT,622) IBOTH                                        PGRID 64
```

```
      C     LOOP THRU THE DATA CARDS UNTIL A STAR IS READ
   60       WRITE(NPRINT,599)
            J=0
  150   CONTINUE
            WRITE(NPRINT,600)
            DO 200 I=1,5
   65       J=J+1
            READ(NARE,601) IFLAG,NAME,X,Y
      C     TEST IF THERE WAS A CARD
            IF (EOF(NARE).NE.0.) GO TO 3000
      C TEST IF POINT IS IN BOUNDS
   70       OUT=.FALSE.
            IF (X.LT.XMIN.OR.X.GT.XMAX) OUT=.TRUE.
            IF (Y.LT.YMIN.OR.Y.GT.YMAX) OUT=.TRUE.
      C THIS CALLS CALCNR TO COMPUTE THE LOSS FOR GRID COORDINATES X AND Y AND STORES
      C THE VALUES AS NREF, CNR CNR.
   75       CALL CALCNR(X,Y)
            IF (CNR.GT.99.) CNR=99.
            IF (OUT) WRITE(NPRINT,602) J,NAME,X,Y,CNR
            IF (.NOT.OUT) WRITE(NPRINT,603) J,NAME,X,Y,CNR
            WRITE(10,800) CNR
   80       IF (IFLAG.EQ.1H*)  GOTO 400
  200   CONTINUE
            WRITE(NPRINT,640)
            GOTO 150
   85   400   CONTINUE
      C THIS CALCULATES THE TIME SPENT IN POINT
            CALL SECOND(T99)                                          PGRID132
            T99=T99-T100                                              PGRID133
            WRITE(NPRINT,900) T99                                     PGRID134
   90       RETURN                                                    PGRID135
      C     EARLY END OF FILE
      3000  WRITE(NPRINT,699)
            STOP
      C EOF EXIT                                                      PGRID136
        300 WRITE(NPRINT,698)                                         PGRID137
            STOP                                                      PGRID138
      599   FORMAT(///3H    )
      600   FORMAT(11X,4HNAME,9X,1HX,9X,1HY,8X,5HLEVEL  )
      601   FORMAT(1A1,1A9,2F10.0)
  100 602   FORMAT(I5,4X,1A9,2F10.0,F10.2,5X,13HOUT OF BOUNDS  )
      603   FORMAT(I5,4X,1A9,2F10.0,F10.2)
       610  FORMAT(1H1,///,T15,*.....      POINT      .....*)
      620   FORMAT(//T16,*  INVERSION =*, 3F8.2 //)
       622  FORMAT(T12,*CALCULATIONS FOR NRF WILL USE *,A10,* OF DAY AND NIGHPGRID156
  105     1 CALCULATIONS*,///)                                        PGRID157
      640   FORMAT(1H1)
        698 FORMAT(///   50H *****     ERROR -- MISSING INPUT DIRECTIVE; JOB APGRID160
           1ORTED  )                                                  PGRID161
      699   FORMAT(///   50H *****     ERROR -- EARLY END OF FILE ; JOB AB
  110     1RTED  )
      700   FORMAT(6F10. ,A1 )
        800 FORMAT(F1 .2)
      900   FORMAT(///,T15,*.....   TIME FOR THE POINT SUBPROGRAM IS *,F8.5
           1* SECONDS*)                                               PGRID165
```

```
SUBROUTINE POINT       74/175  OPT=2 ROUND=+-*/             FTN 4.8  508        83/10/30. 12.45.43       PAGE     3

  115                END                                                          PGRID166


      SYMBOLIC REFERENCE MAP (R=3)

ENTRY POINTS       DEF LINE       REFERENCES
    1  POINT           1            90

VARIABLES       SN  TYPE              RELOCATION
    4  BDS          LOGICAL                 ROUND      REFS           4        5      25
    0  CHECK        LOGICAL                 DEBUG      REFS          10       17
    1  CNR          REAL                    FT         REFS           7       76      77       78       79
                                                    DEFINED          76
13562  DARKNC       REAL          ARRAY     FT         REFS           7        8
    0  DAYCNR       LOGICAL                 CALC       REFS          11       14      48       53       54
                                                    DEFINED          46       49
 7642  DAYND        REAL          ARRAY     FT         REFS           7        8
    1  DRKCNR       LOGICAL                 CALC       REFS          11       14      48       52       54
                                                    DEFINED          47       50
    0  GRDSZ        REAL                    FT         REFS           7
  433  I          * INTEGER                         DEFINED          64
    2  IBOTH        INTEGER                 METRIC     REFS          12       57  DEFINED      52       53       54
  434  IFLAG        INTEGER                            REFS          80  DEFINED      66
    1  IMETER       INTEGER                 METRIC     REFS          12
  432  J            INTEGER                            REFS          65       77      78  DEFINED       61       65
    0  KARD         INTEGER                 IO         REFS           3       33      68  I/O REFS      31       66
    1  KPRINT       INTEGER                 IO         REFS           3  I/O REFS     19       26       56       57       60
                                                        63           77       78      82       89       92       95
    0  METERS       LOGICAL                 METRIC     REFS          12       13
  435  NAME         INTEGER                            REFS          77       78  DEFINED      66
    0  NSRCS        INTEGER                 SRCS       REFS           6
  424  OUT          LOGICAL                            REFS          15       77      78  DEFINED       70       71       72
    1  REED         LOGICAL                 DEBUG      REFS          10       17      22
    0  RINV1        REAL                    FACTI      REFS           9       38      56  DEFINED       40
    1  RINV2        REAL                    FACTI      REFS           9       38      56  DEFINED       41
    2  RINV3        REAL                    FACTI      REFS           9       38      56  DEFINED       42
    2  TAHRD        LOGICAL                 DEBUG      REFS          10       17      37  DEFINED       39
  425  T100         REAL                               REFS          18       88
  440  T99          REAL                               REFS          87       88      89  DEFINED       88
  436  X            REAL                               REFS        2*71       75      77       78  DEFINED       66
    2  XLOC         REAL          ARRAY     FT         REFS           7        8
    2  XMAX         REAL                    ROUND      REFS           4       71
    0  XMIN         REAL                    ROUND      REFS           4       71
  426  X1           REAL                               REFS          38       40  DEFINED      31
  427  X2           REAL                               REFS          38       41  DEFINED      31
  431  X3           REAL                               REFS        2*46     2*47  DEFINED      31
  430  X4           REAL                               REFS          38       42  DEFINED      31
  437  Y            REAL                               REFS        2*72       75      77       78  DEFINED       66
 3722  YLOC         REAL          ARRAY     FT         REFS           7        8
    3  YMAX         REAL                    ROUND      REFS           4       72
    1  YMIN         REAL                    ROUND      REFS           4       72
```

```
FILE NAMES          MODE
       TAPE10     FMT                    WRITES        79
       VARIABLES USED AS FILE NAMES, SEE ABOVE

EXTERNALS          TYPE    ARGS        REFERENCES
       CALCNR                2              75
       EOF        REAL       1              33        68
       READIN                0              22
       READTB                0              43
       SECOND                1              18        87

STATEMENT LABELS                   DEF LINE   REFERENCES
  175  5        FMT                   27          26
   15  10                             31          25
   33  50                             40          37
   41  75                             44          38
   52  90                             51          48
   73  150                            62          83
    0  200                            81          64
  153  300                            95          33
  141  400                            85          80
  304  599      FMT                   97          60
  306  600      FMT                   98          63
  312  601      FMT                   99          66
  315  602      FMT                  100          77
  323  603      FMT                  101          78
  327  610      FMT                  102          19
  335  620      FMT                  103          56
  342  622      FMT                  104          57
  353  640      FMT                  106          82
  355  698      FMT                  107          95
  365  699      FMT                  109          92
  375  700      FMT                  111          31
  400  800      FMT                  112          79
  402  900      FMT                  113          89
  150  3000                           92          68

LOOPS  LABEL    INDEX      FROM-TO     LENGTH     PROPERTIES
  76   200      I           64 81       41B                   EXT REFS    EXITS

COMMON BLOCKS   LENGTH    MEMBERS - BIAS NAME(LENGTH)
       IO          2                0 KARD    (1)           1 KPRINT (1)
       BOUND       5                0 XMIN    (1)           1 YMIN   (1)           2 XMAX    (1)
                                    3 YMAX    (1)           4 BUS    (1)
       SRCS        1                0 NSRCS   (1)
       FT       8002                0 GROSZ   (1)           1 CNR    (1)           2 XLOC    (2000)
                                 2002 YLOC    (2000)     4002 DAYNO  (2000)     6002 DARKNO  (2000)
       FACTI       3                0 RINV1   (1)           1 RINV2  (1)           2 RINV3   (1)
       DEBUG       3                0 CHECK   (1)           1 REED   (1)           2 TABRD   (1)
       CALC        2                0 DAYCNR  (1)           1 DRKCNR (1)
       METRIC      3                0 METERS  (1)           1 IMETER (1)           2 IBOTH   (1)

STATISTICS
  PROGRAM LENGTH                 441B      289
  CM LABELED COMMON LENGTH     17525B     8021
       60000B CM USED
```

```
      SUBROUTINE PGRID                                                  PGRID  2
C SETS UP A GRID OF LCON VALUES FOR A SPECIFIED AREA.
C     SEE ZNEF FOR THE DESCRIPTION OF THESE VARIABLES
      COMMON/IO/KARD,KPRINT                                             PGRID  3
      COMMON/BOUND/XMIN,YMIN,XMAX,YMAX,BDS                              PGRID  4
      LOGICAL BDS                                                       PGRID  5
      COMMON /SRCS/ NSRCS
      COMMON/FT/GRDSZ,CNR,XLOC,YLOC,DAYNO,DARKNO                        PGRID  7
      DIMENSION XLOC(2000),YLOC(2000),DARKNO(2000),DAYNO(2000)          PGRID  8
      COMMON/FACTI/ RINV1,RINV2,RINV3                                   PGRID  9
      COMMON/DEBUG/CHECK,REED,TABRD                                     PGRID 10
      COMMON/CALC/DAYCNR,DRKCNR                                         PGRID 11
      COMMON/METRIC/METERS,IMETER,IBOTH                                 PGRID 12
      LOGICAL METERS                                                    PGRID 13
      LOGICAL DAYCNR,DRKCNR                                             PGRID 14
C     FLAGS FOR READING TAPE8 INFO(REED), TAPE20 INFO(TABRD)
      LOGICAL CHECK,REED,TABRD                                          PGRID 15
C     FLAG TO DENOTE WHETHER OR NOT THE ORIGINAL BOUNDS WERE CHANGED
      LOGICAL DELBDS                                                    PGRID 16
C     ARRAYS FOR LABELS(IL,JL) XARRY CONTAINS LCON VALUES
      DIMENSION IL(100),JL(50),XARRY(50)                                PGRID 17
      EQUIVALENCE (JL(1),XARRY(1))                                      PGRID 18
C     DEFGSZ=DEFAULT GRID SIZE
C     KOLS=MAX. NUMBER OF COLUMNS
C     GMULT=MULTIPLE VALUE
C     TOL = TOLERENCE
      DATA DEFGSZ/2000./,KOLS/15/,GMULT/250./,TOL/.01/                  PGRID 19
C     NSKIP AIDS IN SETTING UP TABLE FOR OUTPUT
      DATA BLANK/1H /,ZERO/1H0/,NSKIP/5/                                PGRID 20
C     IOUT=TAPE1
      DATA IOUT/1/                                                      PGRID 21
      CALL SECOND(T100)                                                 PGRID 22
      WRITE(KPRINT,610)                                                 PGRID 23
C IF THE DATA IS NOT ALREADY READ IN FOR USE IN PGRID, THEN CALL SUBROUTINE
C READIN TO READ THE DATA FROM TAPE 8.
      IF (.NOT. REED) CALL READIN                                       PGRID 24
C IF GRID BOUNDARIES ARE NOT PROPERLY INITIALIZED, COMPLAIN.
C GRID BOUNDARIES NOT PROPERLY INITIALIZED -- ABORT                     PGRID 25
      IF(BDS) GO TO 10
      WRITE(KPRINT,5)
5     FORMAT(*  ...NO BOUNDS CALL BEFORE PUDDLE GRID CALL...*)
      STOP
C READ INPUT DIRECTIVE,SET DEFAULTS,IF NECESSARY                        PGRID 28
C THIS READS PGRID=2 CARD CONTAINING THE INVERSION FACTORS, GRID SIZE, AND DAY
C AND/OR NIGHT.
10    READ(KARD,700) X1,X2,X4,GRDSZ,X3,GRDNAME
C ERROR MESSAGE:  ERROR**MISSING INPUT DIRECTIVE; JOB ABORTED
      IF(EOF(KARD).NE.0)GO TO 300                                       PGRID 30
C INVERSION FACTORS                                                     PGRID 31
C THIS SECTION READS, VERIFIES OR FINDS OUT ABOUT THE INVERSION FACTORS USED
C FROM SUBROUTINE READIN OR THE DIRECTIVE CARD.
      IF(.NOT.TABRD)GO TO 50                                            PGRID 32
      IF(X1.EQ.RINV1.AND.X2.EQ.RINV2.AND.X4.EQ.RINV3)   GOTO 75
      TABRD=.FALSE.                                                     PGRID 34
50    RINV1=X1                                                          PGRID 35
      RINV2=X2                                                          PGRID 36
      RINV3= X4
```

```
                CALL READTB                                                     PGRID 37
              75 CONTINUE                                                       PGRID 38
60          C GRID SIZE                                                       PGRID 39
            C IF THERE IS NO GRID SIZE ON THE DIRECTIVE CARD, THEN IT ASSUMES A DEFAULT
            C VALUE (HERE DEFGSZ=2500)
                IF(GRDSZ.LE.0.) GRDSZ=DEFGSZ                                PGRID 40
            C THIS IS A WARNING STATING THAT THE INPUT GRID SIZE IS NOT A MULTIPLE OF SOME
65          C NUMBER,GMULT.  (HERE, GMULT=2500)
                IF(AMOD(GRDSZ,GMULT).GT.TOL)WRITE(KPRINT,697)GRDSZ,GMULT    PGRID 41
            C DAY +/OR NIGHT CONSIDERED                                     PGRID 42
                DAYCNR=(X3.EQ.10HBOTH        .OR. X3.EQ.10HDAY          )   PGRID 43
                DRKCNR=(X3.EQ.10HBOTH        .OR. X3.EQ.10HNIGHT        )   PGRID 44
70              IF(DAYCNR.OR.DRKCNR)GO TO 90                                PGRID 45
                DAYCNR=.TRUE.                                               PGRID 46
                DRKCNR=.TRUE.                                               PGRID 47
              90 CONTINUE                                                   PGRID 48
                IF (DRKCNR) IBOTH=10HNIGHT ONLY                             PGRID 49
75              IF (DAYCNR) IBOTH=10H  DAY ONLY                             PGRID 50
                IF (DAYCNR .A. DRKCNR) IBOTH=10HBOTH D + N                  PGRID 51
            C THE NEXT FEW LINES CHECK TO MAKE SURE EACH BOUND SUBMITTED BY THE USER IS
            C CORRECT BY CALLING SUBROUTINE BDSET.
                PGXMIN=XMIN
80              PGYMIN=YMIN
                PGXMAX=XMAX
                 PGYMAX=YMAX
                DELBDS=.FALSE.                                              PGRID 55
            C THESE CALLS SEND BOTH THE MINIMUM AND MAXIMUM BOUNDARY VALUES TO
85          C BDSET SO AS TO FIND THE AREA THEY ENCOMPASS.
                CALL BDSET(PGXMIN,PGXMAX,GRDSZ,NJ,DELBDS)
                CALL BDSET(PGYMIN,PGYMAX,GRDSZ,NI,DELBDS)
            C PRINT PARAMETERS                                              PGRID 60
            C IF THE FLAG IS TRUE THEN THE BOUNDS ARE MODIFIED,  IF NOT, PGRID WILL USE
90          C THE PRESENT BOUNDS.
                IF(DELBDS) WRITE(KPRINT,612) XMIN,YMIN,XMAX,YMAX            PGRID 61
                WRITE(KPRINT,611)PGXMIN,PGYMIN,PGXMAX,PGYMAX                PGRID 62
                WRITE(KPRINT,620) MINV1,MINV2,MINV3
                WRITE(KPRINT,622) IBOTH                                     PGRID 64
95              WRITE(KPRINT,621) GRDSZ,IMETER                              PGRID 65
            C NASAPLOT PROGRAM INPUT -- WRITE RUN INFO                      PGRID 80
                REWIND IOUT                                                 PGRID 81
                WRITE(IOUT,701)GRNAME,GRDSZ,PGXMIN,PGYMIN,PGXMAX,PGYMAX,IMETER  PGRID 82
                WRITE(IOUT,605)                                             PGRID 83
100             WRITE(IOUT,603)NI,NJ                                        PGRID 84
            C ADD ONE TO DIFFERENCES TO GET ACTUAL ROW,COL. COUNTS FOR LOOPING;  PGRID 85
            C   NASAPLOT ROWS,COLS = ACTUAL +1                              PGRID 86
                NI=NI+1                                                     PGRID 87
                NJ=NJ+1                                                     PGRID 88
105         C STORE ROW LABELS                                              PGRID 89
                IGRDSZ=GRDSZ                                                PGRID 90
            C THIS LOOP INITIALIZES THE IL ARRAY TO THE ROW LABLES USED IN SETTING UP THE
            C GRID.
                IL(1)=PGYMAX                                                PGRID 91
110             DO 130 I=2,NI                                               PGRID 92
            C     CHECK FOR OVERFLOW
                  IF(I .GT. 100) GO TO 300
130         130 IL(I)= IL(I-1) - IGRDSZ
            C IAP = STARTING X VALUE FOR CURRENT PAGE                       PGRID 98
```

```
      IXP=0
C     LOOP HERE FOR NEXT PAGE
  140 IX=IXP                                                          PGRID 96
      IY=NI-1
      LINECT=0                                                        PGRID 98
      CC=ZERO                                                         PGRID 99
C SET UP PAGING                                                       PGRID100
      NJP=NJ                                                          PGRID101
      IF(NJ.LE.NOLS)GO TO 150                                         PGRID102
      NJP=NOLS                                                        PGRID103
  150 CONTINUE                                                        PGRID104
C PRINT COL. LABELS FOR THIS PAGE                                     PGRID105
C THIS LOOP INITIALIZES THE JL ARRAY TO THE COLUMN LABLES USED IN SETTING UP
C THE GRID.  NEXT, IT PRINTS ALL OF THE COLUMN LABELS.
      JL(1)=PGXMIN+ IX*GRDSZ
      DO 160 J=2,NJP                                                  PGRID107
      IF(J .EQ. 50) GO TO 3000
  160 JL(J)=JL(J-1) + IGRDSZ
      WRITE(KPRINT,600) (JL(J),J=1,NJP)                               PGRID109
C THIS DO-LOOP CALCULATES THE VALUES ON THE GRID FOR X AND Y TO BE SENT TO
C SUBROUTINE CALCNR TO COMPUTE THE LCOR VALUES.
      DO 180 I=1,NI                                                   PGRID110
C     THIS SETS UP THE ROW LABEL (Y COORDINATE) USED BY NASAPLOT
      II=NI-I+1                                                       PGRID111
      DO 170 J=1,NJP                                                  PGRID112
C THIS SETS UP THE COLUMN LABEL (OR X COORDINATE USED).
      X=IX*GRDSZ+PGXMIN
      Y=IY*GRDSZ+PGYMIN
C THIS CALLS CALCNR TO COMPUTE THE LCOR FOR GRID COORDINATES X AND Y AND STORES
C THE VALUES AS VNEF, OR CNR.
      CALL CALCNR(X,Y)
C     VNEF,II,JJ--INPUT TO NASAPLOT
      VNEF=XARRY(J)=CNR
      IF(VNEF.GT.80.)VNEF=80.                                         PGRID116
      JJ = IX+1
C     OUTPUT TO NASSAPLOT   ROW,COL,VALUE
C     WRITE PGRID INFO TO  TAPE1
      WRITE(IOUT,604) II,JJ,VNEF
  170 IX=IX+1
      IF(MOD(LINECT,4*IP).EQ.0)CC=ZERO                                PGRID119
C THIS PRINTS OUT THE LCOR VALUES FOR THE POODLE GRID OUTPUT.
      WRITE(KPRINT,601)CC,IL(I),(XARRY(J),J=1,NJP)                    PGRID120
      LINECT=LINECT+1                                                 PGRID121
      CC=BLANK                                                        PGRID122
      IY=IY-1
  180 IX=IXP                                                          PGRID124
  190 CONTINUE                                                        PGRID125
      NJ=NJ-NJP                                                       PGRID126
C ANOTHER PAGE?
C THIS CHECKS TO SEE IF THERE IS A NEED FOR ANOTHER PAGE OF OUTPUT SINCE THERE
C IS A LIMIT PER PAGE OF OUTPUT.
      IF(NJ.LE.0)GO TO 200                                            PGRID128
      IXP=IXP+NJP                                                     PGRID129
C     START A NEW PAGE
      GO TO 140                                                       PGRID130
  200 CONTINUE                                                        PGRID131
C THIS CALCULATES THE TIME SPENT IN PGRID.
```

```
                      CALL SECOND(T99)                                                  PGRID132
                      T99=T99-T100                                                      PGRID133
                      WRITE(KPRINT,900) T99                                             PGRID134
  175                 RETURN                                                            PGRID135
                 3000 WRITE (KPRINT,702)
                      STOP
                    C EOF EXIT
                      300 WRITE(KPRINT,698)                                             PGRID137
  180                 STOP                                                              PGRID138
                      600 FORMAT(1H1///8X,15(I7,1X) /                                   PGRID139
                      601 FORMAT(A1,I7,15(F8.1,2X))                                     PGRID140
                      603 FORMAT(4HGRID,2X,2I8)                                         PGRID141
                      604 FORMAT(4HGRDI,2X,2I8,F8.3)                                    PGRID142
  185                 605 FORMAT(4HPHS2)                                                PGRID143
                      610  FORMAT(1H1,///,T15,*.....     POODLE GRID      .....*)      PGRID144
                      611 FORMAT(                                  ///T12,*START        PGRID145
                         1AT DATA BASE COORDINATES (*,F9.1,*,*,F9.1,* )*                PGRID146
                         2                                        //T12,*STOP          PGRID147
  190                    1AT DATA BASE COORDINATES (*,F9.1,*,*,F9.1,* )* )               PGRID148
                      612 FORMAT(///   40H *****     WARNING -- SPECIFIED BOUNDS ( ,F9.1,*,*PGRID149
                         1,F9.1,* ):(*,F9.1,*,*,F9.1,* ) DO NOT CORRESPOND TO INTEGRAL GRID PGRID150
                         2BOUNDS.*/21X,**MODIFIED BOUNDS WILL BE USED TO PRODUCE THE GRID ANDPGRID151
                         3 TO DEFINE ANY PLOT UTILIZING THIS GRID*)                     PGRID152
  195                 620   FORMAT(//T16,*  INVERSION =*, 3F8.2 //)
                        621 FORMAT(/12X,*GRID SIZE =*,F8.1,                 * ) DISTANCES IN PGRID154
                         1*,A10//)                                                      PGRID155
                      622  FORMAT(T12,*CALCULATIONS FOR REF WILL USE *,A10,* OF DAY AND NIGHTPGRID156
                         1 CALCULATIONS*,//)                                            PGRID157
  200                   697 FORMAT(///   31H *****     WARNING -- GRID SIZE ,F8.1,* NOT MULTIPPGRID158
                         1LE OF*,F8.1)                                                  PGRID159
                        698 FORMAT(///   56H *****     ERROR -- MISSING INPUT DIRECTIVE; JOB APGRID160
                         1BORTED  )                                                     PGRID161
                      700   FORMAT(4F10.0,2A10)
  205                   701 FORMAT(A10,5F10.0,A10)                                      PGRID163
                        702 FORMAT(/// 48H *ARRAY INDEX LARGER THAN THE DIMENSIONS; ABORT*)
                      900  FORMAT(//,T15,*.....     TIME FOR POODLING SUBPROGRAM IS *,F8.3,  PGRID164
                         1* SECONDS*)                                                   PGRID165
                          END                                                           PGRID166
```

SYMBOLIC REFERENCE MAP (R=3)

ENTRY POINTS DEF LINE REFERENCES
1 PGRID 1 175

VARIABLES	SN	TYPE									
4	BDS	LOGICAL		ROUND	REF	5	6	39			
347	BLANK	REAL			REF	156	DEFINED	29			
743	CC	REAL			REF	156	DEFINED	120	154	158	
0	CHECK	LOGICAL		DEBUG	REF	11	17				
1	CNR	REAL		FT	REF	8	147				
13562	DAMAG	REAL	ARRAY	FT	REF	6	9				
0	DAYCHR	LOGICAL		CALL	REF	12	15	70	75	76	
					DEFINED	68	71				

VARIABLES	SN	TYPE		RELOCATION								
7642	DAYNO	REAL	ARRAY	FT	REFS	8	9					
343	DEFGSZ	REAL			REFS	63	DEFINED	27				
720	DELBUS	LOGICAL			REFS	14	86	87	91	DEFINED	83	
1	DMRCNR	LOGICAL		CALC	REFS	12	15	70	74	76		
					DEFINED	69	72					
345	GMULT	REAL			REFS	2*66	DEFINED	27				
726	GNUNAME	REAL			REFS	98	DEFINED	46				
0	GRDSZ	REAL		FT	REFS	8	63	2*66	86	87	95	98
						106	129	141	142	DEFINED	46	63
736	I	INTEGER			REFS	112	2*113	138	156	DEFINED	110	136
2	IBUTM	INTEGER		METRIC	REFS	13	94	DEFINED	74	75	76	
735	IGRDSZ	INTEGER			REFS	113	132	DEFINED	106			
746	II	INTEGER			REFS	152	DEFINED	138				
1036	IL	INTEGER	ARRAY		REFS	21	113	156	DEFINED	109	113	
1	IMETER	INTEGER		METRIC	REFS	13	95	98				
352	IOUT	INTEGER			DEFINED	31	I/O REFS	97	98	99	100	152
740	IX	INTEGER			REFS	129	141	149	153	DEFINED	117	153
						160						
737	IXP	INTEGER			REFS	117	160	167	DEFINED	115	167	
741	IY	INTEGER			REFS	142	159	DEFINED	118	159		
745	J	INTEGER			REFS	131	2*132	133	147	156		
					DEFINED	130	133	139	156			
752	JJ	INTEGER			REFS	152	DEFINED	149				
754	JL	INTEGER	ARRAY		REFS	21	22	132	133	DEFINED	129	132
0	KARD	INTEGER		IO	REFS	4	48	I/O REFS	46			
344	KULS	INTEGER			REFS	123	124	DEFINED	27			
1	KPRINT	INTEGER		IO	REFS	4	I/O REFS	33	40	66	91	92
						93	94	95	133	156	174	176	179
742	LINECT	INTEGER			REFS	154	157	DEFINED	119	157		
0	METERS	LOGICAL		METRIC	REFS	13	14					
734	NI	INTEGER			REFS	87	100	103	110	118	136	138
					DEFINED	105						
733	NJ	INTEGER			REFS	86	100	104	122	123	162	166
					DEFINED	104	162					
744	NJP	INTEGER			REFS	130	133	139	156	162	167	
					DEFINED	122	124					
351	NSKIP	INTEGER			REFS	154	DEFINED	29				
0	NSRCS	INTEGER		SRCS	REFS	7						
731	PGXMAX	REAL			REFS	86	92	98	DEFINED	81		
727	PGXMIN	REAL			REFS	80	92	98	129	141		
					DEFINED	79						
732	PGYMAX	REAL			REFS	87	92	98	109	DEFINED	82	
730	PGYMIN	REAL			REFS	87	92	98	142	DEFINED	80	
1	REED	LOGICAL		DEBUG	REFS	11	17	36				
0	RINV1	REAL		FACTI	REFS	10	53	93	DEFINED	55		
1	RINV2	REAL		FACTI	REFS	10	53	93	DEFINED	56		
2	RINV3	REAL		FACTI	REFS	10	53	93	DEFINED	57		
2	TABRD	LOGICAL		DEBUG	REFS	11	17	52	DEFINED	54		
346	TOL	REAL			REFS	66	DEFINED	27				
721	T100	REAL			REFS	32	173					
753	T99	REAL			REFS	172	173	174	DEFINED	173		
751	VREF	REAL			REFS	148	152	DEFINED	147	148		
747	X	REAL			REFS	145	DEFINED	141				
754	XARRY	REAL	ARRAY		REFS	21	22	156	DEFINED	147		
2	XLOC	REAL	ARRAY	FT	REFS	8	9					
2	XMAX	REAL		ROUND	REFS	5	81	91				
0	XMIN	REAL		ROUND	REFS	5	79	91				

VARIABLES	SN	TYPE	RELOCATION						
722	X1	REAL			REFS	55	55	DEFINED	46
723	X2	REAL			REFS	55	56	DEFINED	46
725	X3	REAL			REFS	2*08	2*09	DEFINED	46
724	X4	REAL			REFS	55	57	DEFINED	46
750	Y	REAL			REFS	145	DEFINED	142	
3722	YLOC	REAL	ARRAY	X1	REFS	8	9		
3	YMAX	REAL		BOUND	REFS	5	82	91	
1	YMIN	REAL		BOUND	REFS	5	80	91	
350	ZERO	REAL			REFS	120	154	DEFINED	29

VARIABLES USED AS FILE NAMES, SEE ABOVE

EXTERNALS	TYPE	ARGS	REFERENCES	
BUSET		5	80	87
CALCNR		2	145	
EOF	REAL	1	48	
READIN		0	36	
READTB		0	58	
SECOND		1	32	172

INLINE FUNCTIONS	TYPE	ARGS		DEF LINE	REFERENCES
AMOD	REAL	2	INTRIN		66
MOD	INTEGER	2	INTRIN		154

STATEMENT LABELS			DEF LINE	REFERENCES	
361	5	FMT	41	40	
15	10		46	39	
33	50		55	52	
41	75		59	53	
65	90		73	70	
0	130		113	110	
157	140		117	109	
170	150		125	123	
0	160		132	130	
0	170		135	139	
0	180		161	136	
273	200		170	166	
305	300		179	48	
521	600	FMT	181	133	
524	601	FMT	182	156	
527	603	FMT	183	100	
532	604	FMT	184	152	
535	605	FMT	185	94	
537	610	FMT	188	35	
545	611	FMT	187	92	
563	612	FMT	191	91	
614	620	FMT	195	93	
621	621	FMT	196	95	
630	622	FMT	198	94	
641	697	FMT	200	66	
651	698	FMT	202	179	
661	700	FMT	204	46	
664	701	FMT	205	98	
667	702	FMT	206	176	
676	900	FMT	207	174	
302	3000		176	112	131

LOOPS	LABEL	INDEX	FR	TO	LENGTH	BRANCH TO	
145	130	I	110	113	5B	INSIDE	EXITS
176	160	J	130	132	4B	INSIDE	EXITS
215	190	I	136	161	52B		EXT REFS
222	170	J	139	153	2AB		EXT REFS

COMMON BLOCKS	LENGTH	MEMBERS					
IO	2	0 XMIN	(1)	1 XMAXT	(1)		
BOUND	5	0 XMIN	(1)	1 YMIN	(1)	2 XMAX	(1)
		3 YMAX	(1)	4 NLS	(1)		
SMLS	1	0 NSMLS	(1)				
PT	8002	0 GMDSZ	(1)	1 ENR	(1)	2 XLOC	(2000)
		2002 YLOC	(2000)	4002 DATNO	(2000)	6002 CARRNO	(2000)
FACTI	3	0 MINV1	(1)	1 MINV2	(1)	2 MINV3	(1)
DEBUG	3	0 CHECK	(1)	1 NEED	(1)	2 TABRD	(1)
CALC	2	0 DATCHK	(1)	1 CARCHK	(1)		
METRIC	3	0 METERS	(1)	1 IMETER	(1)	2 IBOTH	(1)

EQUIV CLASSES	LENGTH	MEMBERS	
JL	50	0 XARAY	(50)

STATISTICS

PROGRAM LENGTH 12428 842

CM LABELED COMMON LENGTH 175250 8021

000008 CM USED

```
      SUBROUTINE PLOT                                                   PLOT   2
C     FILE  SAGE  AR[illegible]                                         PLOT   3
C        PLOG=P[illegible]                                              PLOT   4
C        [illegible]                                                    PLOT   5
C        SCAT=SCATTER                                                   PLOT   6
C        RAS=RAS[illegible]                                             PLOT   7
C     TAPE NAMES                                                        PLOT   8
C        IN1 =P[illegible] TAPE                                         PLOT   9
C        IN2 =PHASE TAPE                                                PLOT  10
C        IN3 =L[illegible] TAPE                                         PLOT  11
C        IN4 =SCATTER TAPE                                              PLOT  12
      COMMON/[illegible]PRINT                                           PLOT  13
      COMMON/[illegible],SATABLE,STTABLE,LARGE                          PLOT  14
      COMMON/[illegible],YMIN1,XMAX1,YMAX1,NOS                          PLOT  15
      LOGICAL [illegible],LARGE                                         PLOT  16
      COMMON /METRIC/ METERS                                            PLOT  17
      LOGICAL METERS                                                    PLOT  18
      REAL MA[illegible]                                                PLOT  19
      DIMENSION STARS(1[illegible])                                     PLOT  20
      DIMENSION [illegible]                                             PLOT  21
      DIMENSION [illegible]                                             PLOT  22
      DIMENSION [illegible](5)                                          PLOT  23
      INTEGER PL[illegible]                                             PLOT  24
      INTEGER TEXT(2[illegible]STAR                                     PLOT  25
      DIMENSION [illegible](20)                                         PLOT  26
      INTEGER START,STOP,[illegible],LABEL,LINC                         PLOT  27
      INTEGER PLOT,PLOG,SCAT,LOC,RAS                                    PLOT  28
      DIMENSION MEASURE(2)                                              PLOT  29
      DIMENSION IUNIT(4)                                                PLOT  30
      EQUIVALENCE (IUNIT(1),IN1),(IUNIT(2),IN2),(IUNIT(3),IN3),(IUNIT(4)PLOT  31
     1,IN4)                                                             PLOT  32
      EQUIVALENCE (IFILE(1),[illegible]),(IFILE(2),RAS),(IFILE(3),LOC),(IFILE(4PLOT  33
     1),SCAT)                                                           PLOT  34
      DATA MEASURE/10HMETERS    ,10H[illegible]    /                    PLOT  35
      DATA HEA[illegible]/10HP[illegible]   ,10HRAS[illegible]   ,10HLOCATOR   ,9HSCATTER  / PLOT  36
C                                                                       PLOT  37
C     VARIABLE I[illegible] MUST BE INITIALIZE[illegible] INPUT UNIT NUMBER PLOT  38
C                                                                       PLOT  39
      DATA [illegible]/[illegible],IN3/3/,IN4/4/,IC1/[illegible]/       PLOT  40
C     TAPE55 [illegible]                                                
      [illegible]                                                       PLOT  41
      WRITE([illegible])                                                PLOT  42
 150  FORMAT([illegible]........   [illegible]   ........)              PLOT  43
C     [illegible] NUMBER [illegible]                                    
      [illegible]                                                       PLOT  43
      [illegible]                                                       PLOT  44
C     [illegible]                                                       
      CALL [illegible]                                                  PLOT  45
      CALL TIME[illegible]                                              PLOT  46
      CALL [illegible]                                                  PLOT  47
      WRITE([illegible])                                                PLOT  48
 [illegible] FORMAT [illegible],I3)                                     PLOT  49
      [illegible]                                                       PLOT  50
      [illegible]                                                       PLOT  51
 [illegible] START [illegible]                                          PLOT  52
      REA[illegible] STATE                                              
      READ [illegible]                                                  PLOT  53
```

```
                                                                      PLOT  56
                                                                      PLOT  57
                                                                      PLOT  58
                                                                      PLOT  59
                                                                      PLOT  60
                                                                      PLOT  61
                                                                      PLOT  62
                                                                      PLOT  63
                                                                      PLOT  64
                                                                      PLOT  65
                                                                      PLOT  66
                                                                      PLOT  67
                                                                      PLOT  68
                                                                      PLOT  69
                                                                      PLOT  70
                                                                      PLOT  71
                                                                      PLOT  72
                                                                      PLOT  73
                                                                      PLOT  74
                                                                      PLOT  75
                                                                      PLOT  76
                                                                      PLOT  77
                                                                      PLOT  78
                                                                      PLOT  79
                                                                      PLOT  80
                                                                      PLOT  81
                                                                      PLOT  82
                                                                      PLOT  84
                                                                      PLOT  87
                                                                      PLOT  88
                                                                      PLOT  89
                                                                      PLOT  90
                                                                      PLOT  91
                                                                      PLOT  92
                                                                      PLOT  93
                                                                      PLOT  94
                                                                      PLOT  95
                                                                      PLOT  96
                                                                      PLOT  98
                                                                      PLOT  99
                                                                      PLOT 100
                                                                      PLOT 101
                                                                      PLOT 102
                                                                      PLOT 103
                                                                      PLOT 104
```

```
          MAG = DEFAULT 1.                                                PLOT 105
           IF((MAG.GT.9.).OR.(MAG.LT. .1)) MAG=1.0                         PLOT 106
           IF(MAG.EQ.1.0) STARS(4)=2H**                                    PLOT 107
C         PERCSM = DEFAULT .333                                           PLOT 108
           IF((PERCSM.GT.4.).OR.(PERCSM.LT. 0.01)) PERCSM=.333             PLOT 109
           IF(PERCSM.EQ..333) STARS(5)=2H**                                PLOT 110
C         START = DEFAULT 55
           IF((START.GT.999).OR.(START.LT.1)) START=55
           IF(START.EQ.55) STARS(6)=2H**
C         STOP = DEFAULT 75
           IF((STOP.GT.999).OR.(STOP.LT.1)) STOP=75
           IF(STOP.EQ.75) STARS(7)=2H**
C         LSTART = DEFAULT 55
           IF((LSTART.GT.999).OR.(LSTART.LT.1)) LSTART=55
           IF(LSTART.EQ.55) STARS(8)=2H**
C         LSTOP = DEFAULT 75
           IF((LSTOP.GT.999).OR.(LSTOP.LT.1)) LSTOP=75
           IF(LSTOP.EQ.75) STARS(9)=2H**
C         LABEL = DEFAULT 1                                               PLOT 123
           IF((LABEL.NE.-1).AND.(LABEL.NE.0).AND.(LABEL.NE.1)) LABEL=1     PLOT 124
           IF(LABEL.EQ.1)      STARS(10)=2H**                              PLOT 125
C         INC = DEFAULT 5                                                 PLOT 126
           IF((INC.GT.99).OR.(INC.LE.0)) INC=5                             PLOT 127
           IF(INC.EQ.5) STARS(11)=2H**                                     PLOT 128
C         LINC= DEFAULT5                                                  PLOT 129
           IF((LINC.GT.99).OR.(LINC.LE.0)) LINC=5                          PLOT 130
           IF(LINC.EQ.5) STARS(12)=2H**                                    PLOT 131
C         INPUT VALIDITY CHECK                                            PLOT 132
           WRITE(KPRINT,151)                                               PLOT 133
      151  FORMAT(1H0,20X,30HFOLLOWING FILES WERE REQUESTED/)              PLOT 134
           DO 153 I=1,4                                                    PLOT 135
           IF(IFILE(I).EQ.1) WRITE(KPRINT,152) HEAD(I)                     PLOT 136
      152  FORMAT(25X,A10)                                                 PLOT 137
      153  CONTINUE                                                        PLOT 138
C          CHECK PLOT BOUNDS AND LABEL BOUNDS
           IF(START.GE.STOP) GO TO 300                                     PLOT 139
           IF(LSTART.GE.LSTOP) GO TO 320                                   PLOT 140
           IS=0                                                            PLOT 141
      200  J1=MOD(START,INC)                                               PLOT 142
           J2=MOD(STOP,INC)                                                PLOT 143
           IF(J1.EQ.J2) GO TO 210                                          PLOT 144
           IS=1                                                            PLOT 145
           STOP=STOP+1                                                     PLOT 146
           GO TO 200                                                       PLOT 147
      210  IF(IS.EQ.1) WRITE(KPRINT,220) STOP                              PLOT 148
      220  FORMAT(15X,"WARNING..START + (INC * X ) .NE. STOP",/15X,"STOP INCRPLOT 149
          1EASED TO ",I3)                                                  PLOT 150
      230  J1=MOD(LSTART,INC)                                              PLOT 151
           J2=MOD(LSTOP,INC)                                               PLOT 152
           IF(J1.EQ.J2) GO TO 240                                          PLOT 153
           IS=2                                                            PLOT 154
           LSTOP=LSTOP+1                                                   PLOT 155
           GO TO 230                                                       PLOT 156
      240  IF(IS.EQ.2) WRITE(KPRINT,250) LSTOP                             PLOT 157
      250  FORMAT(15X,"WARNING..LSTART+(INC*X) .NE. LSTOP",/15X,"LSTOP INCREAPLOT 158
          1SED TO ",I3)                                                    PLOT 159
           GO TO 105                                                       PLOT 160
```

```
      300  WRITE(KPRINT,310)                                                 PLOT 161
      310  FORMAT("START ,GE,STOP...JOB ENDED")                             PLOT 162
           READ*D IOUT                                                       PLOT 163
175        GO TO 900                                                         PLOT 164
      320  WRITE(KPRINT,330)                                                 PLOT 165
      330  FORMAT("LSTART ,GE, LSTOP...JOB ENDED")                          PLOT 166
           REWIND IOUT                                                       PLOT 167
           GO TO 900                                                         PLOT 168
180   C                                                                      PLOT 169
      C                                                                      PLOT 170
      C    DEFAULT AND ERRORS CHECKED - CREATE CARDS                         PLOT 171
      C                                                                      PLOT 172
      C                                                                      PLOT 173
185   C                                                                      PLOT 174
      C       THE FOLLOWING COMPARES BOUNDARY VALUES FROM COMMON             PLOT 175
      C       WITH THOSE READ FROM PGRID TAPE. IF DIFFERENT PGRID            PLOT 176
      C       VALUES USED.                                                   PLOT 177
      C                                                                      PLOT 178
190    105  IF(PUDG.NE.1) GO TO 108                                          PLOT 179
      C     PGRID INFORMATION IS FOUND ON TAPE 1(IN1)
           REWIND IN1                                                        PLOT 180
           READ(IN1,166)GRDNAME,GRDSZ,XMIN1,YMIN1,XMAX1,YMAX1                PLOT 181
      166  FORMAT(A10,5E10.0)                                                PLOT 182
195        IF((XMIN.EQ.XMIN1).AND.(YMIN.EQ.YMIN1).AND.(XMAX.EQ.XMAX1)        PLOT 183
          1.AND.(YMAX.EQ.YMAX1)) GO TO 109                                   PLOT 184
           WRITE(KPRINT,107) XMIN,XMIN1,YMIN,YMIN1,XMAX,XMAX1,YMAX,YMAX1     PLOT 185
      107  FORMAT(1H0, 7X,*WARNING...PUDGLE GRID BOUNDS DO NOT MATCH SPECIFIEPLOT 186
          1D BOUNDS...*//25X,*PGRID VALUES USED*,10X,*SPECIFIED BOUNDS*/     PLOT 187
200        115X,*XMIN=*,6X,F10.2,17X,F10.2/                                  PLOT 188
           115X,*YMIN=*,6X,F10.2,17X,F10.2/                                  PLOT 189
           115X,*XMAX=*,6X,F10.2,17X,F10.2/                                  PLOT 190
           115X,*YMAX=*,6X,F10.2,17X,F10.2)                                  PLOT 191
           GO TO 109                                                         PLOT 192
205    108  XMIN=XMIN1                                                       PLOT 193
           YMIN=YMIN1                                                        PLOT 194
           XMAX=XMAX1                                                        PLOT 195
           YMAX=YMAX1                                                        PLOT 196
      109  XSCALE=SCALE/PERCX/12.0
210        YSCALE=SCALE/PERCY/12.0
           IF (METERS) XSCALE=XSCALE*0.02540 *12.0
           IF (METERS) YSCALE=YSCALE*0.02540 *12.0
           XOR=5.0*MAG                                                       PLOT 199
           YOR=5.5                                                           PLOT 200
215   C    WRITE OUT TAPE 55 INFORMATION ABOUT BOUNDS, MAGNIFICATION,
      C    AND SCALING.
           WRITE(IOUT ,110)NMAPS,XSCALE,YSCALE,XMIN,XMAX,YMIN,YMAX,XOR,YOR   PLOT 201
      110  FORMAT(A4,2X,2E8.3,4E8.3,2E8.3)                                   PLOT 202
           WRITE(IOUT ,115) NMSCLE,MAG                                       PLOT 203
220   115  FORMAT(A4,2X,F4.3)                                                PLOT 204
           IF(SYTABLE.GE. 0.) GO TO 116                                      PLOT 205
           SYTABLE=((YMAX-YMIN)/XSCALE*PERCX*MAG*XOR*5.                      PLOT 206
           SYTABLE=11.
           IF(.NOT. LANDS) GO TO 117
225        IF((YMAX-YMIN)/XSCALE*PERCY*MAG*XOR*5) .GT. 11.) SYTABLE=14.5
      C
      117  WRITE(IOUT ,120) NMDIMT,SXTABLE,SYTABLE                          PLOT 211
      120  FORMAT(A4,2X,2E8.3)                                               PLOT 212
```

```
      C                                                               PLOT 215
      C                                                               PLOT 216
      C     IF 2ND PLOT                                               PLOT 217
      C                                                               PLOT 218
      C                                                               PLOT 219
      116   IF(PLTCT.EQ.1) GO TO 395                                  PLOT 220
            WRITE(IOUT ,301) 4HPAGE                                   PLOT 221
       301  FORMAT(A4)                                                PLOT 222
      C                                                               PLOT 223
      C                                                               PLOT 224
      C     IF PGDDLEGRID                                             PLOT 225
      C                                                               PLOT 226
      C                                                               PLOT 227
       395  IF(PGDG.NE.1) GO TO 495                                   PLOT 228
       400  READ(IN1,99)                                              PLOT 229
            IF(EOF(IN1))430,410                                       PLOT 230
      410   WRITE(IOUT,99)
            IF (IDUMP.NE.0)  WRITE(KPRINT,99)
            GO TO 400                                                 PLOT 232
       430  WRITE(IOUT ,440) 4HPHS3                                   PLOT 233
       440  FORMAT(A4)                                                PLOT 234
            ZMIN=START                                                PLOT 235
            ZLMIN=LSTART                                              PLOT 236
            D2=INC                                                    PLOT 237
            D2L=LINC                                                  PLOT 238
            NLEVS=(STOP-START)/D2+1.0001                              PLOT 239
            KLEV=0                                                    PLOT 240
            L1=LABEL                                                  PLOT 241
            L2=1                                                      PLOT 242
            ID=-1                                                     PLOT 243
            DISL=2.0                                                  PLOT 244
            HGTL=.167                                                 PLOT 245
            DIST=0                                                    PLOT 246
            TLNG=0                                                    PLOT 247
            TLER=(PERCSM*2)/(XSCALE+YSCALE)*GROSZ                     PLOT 248
            I=2                                                       PLOT 249
            SKIP=0                                                    PLOT 250
      C     WRITE OUT PGRID INFO INTO PHS3 OF TAPE 55
            WRITE(IOUT ,441)4HPLOT,ZMIN,ZLMIN,D2,D2L,NLEVS,KLEV,L1,L2,ID,DISL,PLOT 251
           1HGTL,DIST,TLNG,TLER,I,SKIP                                PLOT 252
       441  FORMAT(A4,2X,2F7.0,2F6.0,I5,I4,3I2,5F4.2,I1,F4.2)         PLOT 253
      C                                                               PLOT 254
      C                                                               PLOT 255
      C     IF LOCATION                                               PLOT 256
      C                                                               PLOT 257
      C                                                               PLOT 258
       495  IF(LOC.NE.1) GO TO 530                                    PLOT 259
      C     TAPE 3 CONTAINS INFORMATION FROM LOCATOR
            REWIND IN3                                                PLOT 260
            READ(IN3,99)                                              PLOT 261
            WRITE(IOUT,99)                                            PLOT 262
       530  READ(IN3,531) XPOS,YPOS,HITE,ANG,(ITEXT(I),I=1,3),L       PLOT 263
       531  FORMAT(6X,2F8.3,2X,2F8.3,2X,3A10,A4,I2)                   PLOT 264
            IF(EOF(IN3).NE.0) GO TO 530                               PLOT 265
      C                                                               PLOT 266
      C          THE FOLLOWING CENTERS THE X OR O ON THE POINT, OTHERWISE PLOT 267
      C          THE POINT WOULD BE AT THE LOWER LEFT CORNER OF THE MARK. PLOT 268
```

```
      C                                                                 PLOT 269
            ANG1=ANG*3.1416/180.0                                       PLOT 270
            XOFF=.5*(6./7.*WHITE)*XSCALE                                PLOT 271
            YOFF=.5*(WHITE)*YSCALE                                      PLOT 272
290         YPRIME=(SQRT(XOFF**2+YOFF**2))*                             PLOT 273
           1       (SIN(ANG1+(ATAN(YOFF/XOFF))))                        PLOT 274
            XPRIME=(SQRT(XOFF**2+YOFF**2))*                             PLOT 275
           1       (COS(ANG1+(ATAN(YOFF/XOFF))))                        PLOT 276
            XPOS=XPOS-XPRIME                                            PLOT 277
295         YPOS=YPOS-YPRIME                                            PLOT 278
      C                                                                 PLOT 279
      C        THE FOLLOWING CHECKS FOR ENOUGH ROOM TO PRINT THE        PLOT 280
      C        LABELING INFORMATION WITHIN THE BOUNDARY.                PLOT 281
      C                                                                 PLOT 282
300         LENTH=LEN*6./7.*WHITE*XSCALE                                PLOT 283
      C        IF ENOUGH ROOM GIVEN ANGLE                               PLOT 284
            IF(((XPOS+LENTH*COS(ANG1)).GT.XMAX).OR.                     PLOT 285
           1((YPOS+LENTH*SIN(ANG1)).GT.YMAX).OR.                        PLOT 286
           1((YPOS+LENTH*SIN(ANG1)).LT.YMIN).OR.                        PLOT 287
305        1((XPOS+LENTH*COS(ANG1)).LT.XMIN)) GO TO 21                  PLOT 288
            GO TO 51                                                    PLOT 289
      C        IF ENOUGH ROOM 90 DEGREES                                PLOT 290
       21   IF((YPOS+LENTH).GT.YMAX) GO TO 22                           PLOT 291
            ANG=90.0                                                    PLOT 292
310         GO TO 51                                                    PLOT 293
      C        IF ENOUGH ROOM 270 DEGREES                               PLOT 294
       22   IF((YPOS-LENTH).LT.YMIN) GO TO 23                           PLOT 295
            ANG=270.0                                                   PLOT 296
            GO TO 51                                                    PLOT 297
315   C        IF ENOUGH ROOM 180 DEGREES                               PLOT 298
       23   IF((XPOS-LENTH).LT.XMIN) GO TO 99                           PLOT 299
            ANG=180.0                                                   PLOT 300
            GO TO 51                                                    PLOT 301
       99   GO TO 51                                                    PLOT 302
320   C     WRITE OUT LOC INFO TO TAPE 55                               PLOT 303
         51 WRITE(IOUT,502)XPOS,YPOS,WHITE,ANG,,(ITEXT(I),I=1,3)        PLOT 304
        502 FORMAT(4HTEXT,2X,2F8.0,F8.0,2F8.3,2H 1,2A10,A4)             PLOT 305
            GO TO 500                                                   PLOT 306
      C                                                                 PLOT 307
325   C                                                                 PLOT 308
      C     IF SCATTER                                                  PLOT 309
      C                                                                 PLOT 310
      C                                                                 PLOT 311
       530  IF(SCAT.NE.1) GO TO 560                                     PLOT 312
330   C     TAPE 4 CONTAINS INFORMATION FROM SCATTER                    PLOT 313
            REWIND I44                                                  PLOT 314
       535  READ(I44,99)                                               PLOT 315
            IF(EOF(I44)) 560,545                                        PLOT 316
      C     WRITE OUT SCATTER INFO TO TAPE 55                           PLOT 317
335    545  WRITE(IOUT,99)                                             PLOT 318
            GO TO 535                                                   PLOT 319
      C                                                                 PLOT 320
      C                                                                 PLOT 321
      C     IF HASF                                                     PLOT 322
340   C                                                                 PLOT 323
      C                                                                 PLOT 324
       560  IF(HAS.NE.1) GO TO 580                                      PLOT 325
```

```
      C     TAPE 2 CONTAINS INFORMATION FROM BASE                                  PLOT 323
            REWIND IN2                                                             PLOT 324
345    565  READ(IN2,99)                                                          PLOT 325
            IF(EOF(IN2))580,572                                                    PLOT 326
       572  WRITE(IOUT,99)                                                        PLOT 327
       99   FORMAT(A0H                                                            PLOT 328
           (                    )                                                 PLOT 329
350         GO TO 565                                                             PLOT 330
       580  WRITE(KPRINT,575)                                                     PLOT 331
       575  FORMAT(1H0,25X,20HVALUES USED BY PLOT /)                              PLOT 332
            WRITE(KPRINT,590) STARS(1),SCALE,STARS(2),PERCX,STARS(3),PERCY        PLOT 333
           1,STARS(4),MAG,STARS(5),PERCSM,STARS(6),START,STARS(7),STOP,           PLOT 334
355        1STARS(8),LSTART,STARS(9),LSTOP,STARS(10),LABEL,STARS(11),INC,         PLOT 335
           1STARS(12),LINC,GRDSZ                                                  PLOT 336
       590  FORMAT(18X,A2,10HSCALE     =,F7.1//                                  PLOT 337
           118X,A2,10HPERCENT X=,F4.2,10X,A2,10HPERCENT Y=,F4.2//                 PLOT 338
           118X,A2,10HMAG       =,F4.2,10X,A2,10HPERC SMTH=,F4.2//                PLOT 339
360        118X,A2,10HSTART     =,I3,11X,A2,10HSTOP      =,I3//                   PLOT 340
           118X,A2,10HL START   =,I3,11X,A2,10HL STOP    =,I3//                   PLOT 341
           118X,A2,10HLABEL     =,I2,12X,A2,10HINCREMENT=,I2//                    PLOT 342
           118X,A2,10HL INCREMT=,I2,14X,10HGRID SIZE=,F6.0)                       PLOT 343
            WRITE(KPRINT,13)                                                       PLOT 344
365    13   FORMAT(1H0,18X,29HSTARS INDICATE DEFAULT VALUES)                      PLOT 345
      C                                                                            PLOT 346
      C                                                                            PLOT 347
      C     PRINT ANY USER TEXT                                                    PLOT 348
      C                                                                            PLOT 349
370   C                                                                            PLOT 350
            WRITE(IOUT  ,690) 4HPHS4                                               PLOT 351
            WRITE(IOUT,690)4HBRDR                                                  PLOT 352
       690  FORMAT(A4)                                                            PLOT 353
            IF(IFLAG.EQ.1) GO TO 800                                               PLOT 354
375         WRITE(KPRINT,156)                                                      PLOT 355
       156  FORMAT(1H0,18X,36HFOLLOWING CARDS WERE USER TEXT INPUT)                PLOT 356
      C     READ PLT-4 CARD WITH USER TEXT INPUT                                   PLOT 357
       700  READ(KARD,705)X,Y,HT,ANGLE,IC,(TEXT(I),I=1,38),STAR                   PLOT 358
            ANGLE1=ANGLE*3.1416/180.0                                              PLOT 359
380         DO 701 I=1,200                                                         PLOT 360
       701  HGT(I)=0.0                                                            PLOT 361
       705  FORMAT(4F10.0,I1,39A1)                                                PLOT 362
            IF(EOF(KARD).NE.0.)GO TO 800                                           PLOT 363
            IF((STAR.EQ.1H*).AND.(X.EQ.0)) GO TO 800                               PLOT 364
385         IFLAG=2                                                                PLOT 365
            WRITE(KPRINT,157)X,Y,HT,ANGLE,IC,(TEXT(I),I=1,38),STAR                 PLOT 366
       157  FORMAT(1H0,18X,1H",4F10.3,I1,39A1,1H")                                PLOT 367
            DO 720 J=1,38                                                          PLOT 368
      C     IF $ CONTINUE TEXT ON NEXT CARD(PLT-5)                                 PLOT 369
390         IF(TEXT(J).EQ.1H$) GO TO 730                                           PLOT 370
       720  CONTINUE                                                              PLOT 371
            J=38                                                                   PLOT 372
       730  HGT(1)=HT                                                             PLOT 373
            IF (STAR.EQ.1H*) GO TO 801                                             PLOT 374
395         K=J+68                                                                 PLOT 375
      C     READ PLT-5 CARD                                                        PLOT 376
            READ(KARD,740) HT1,(TEXT(I),I=J,K),STAR                                PLOT 377
            WRITE(KPRINT,158)HT1,(TEXT(I),I=J,K),STAR                              PLOT 378
       158  FORMAT(1H ,18X,F10.3,70A1)                                            PLOT 379
```

```
                740  FORMAT(F10.0,72A1)                                              PLOT 377
                     IF((HT1.EQ.HT).OR.(HT1.EQ.0)) GO TO 745                         PLOT 378
                     HGT(J)=HT1                                                      PLOT 379
                745  DO 750 J2=J,K                                                   PLOT 380
                     HT=HT1                                                          PLOT 381
                     IF(TEXT(J2).EQ.1H$) GO TO 760                                   PLOT 382
                750  CONTINUE                                                        PLOT 383
                     J2=K                                                            PLOT 384
                760  J=J2                                                            PLOT 385
               C     MAYBE MORE TEXT INPUT
                     IF(STAR.NE.1H*) GO TO 735                                       PLOT 386
                801  K1=1                                                            PLOT 387
                     DO 850 K=2,J                                                    PLOT 388
                     I1=0 $ I2=0 $ I3=0                                              PLOT 389
                     IF(K.EQ.J) GO TO 815                                            PLOT 390
                     IF(HGT(K).EQ.0) GO TO 850                                       PLOT 391
                815  K2=K-K1                                                         PLOT 392
               C    K2=NUMBER OF CHARACTERS - THIS HEIGHT                           PLOT 393
               C     K1= POSITION OF HEIGHT AND STARTING CHARACTER                  PLOT 394
                     IF(K2.GT.24) GO TO 820                                          PLOT 395
                     I1=K1+K2-1                                                      PLOT 396
                     GO TO 845                                                       PLOT 397
                820  IF(K2.GT.90) GO TO 830                                          PLOT 398
                     I1=K1+23                                                        PLOT 399
                     I2=K2+K1-1                                                      PLOT 400
                     GO TO 845                                                       PLOT 401
                830  I1=23+K1                                                        PLOT 402
                     I2=K1+89                                                        PLOT 403
                     I3=K2                                                           PLOT 404
               C CHOOSE FORMAT ACCORDING TO WHETHER OR NOT IC=0 OR IC=1
                     IF(IC.NE.0) GO TO 779                                           PLOT 405
                845  WRITE(IOUT  ,780)4HTEXT,X,Y,Z,HGT(K1),ANGLE,IC,(TEXT(I),I=K1,I1)  PLOT 406
                780  FORMAT(A4,2X,5F8.3,I2,24A1)
                     GO TO 782
                 779 WRITE(IOUT,781)4HTEXT,X,Y,Z,HGT(K1),ANGLE,IC,(TEXT(I),I=K1,I1)
                 781 FORMAT(A4,2X,2F8.0,3F8.3,I2,24A1)
                 782 IF(I2.EQ.0) GO TO 840
                     I1=I1+1                                                         PLOT 408
               C     WRITE USER TEXT TO TAPE 55
                     WRITE(IOUT,790) 4HCTEX,(TEXT(I),I=I1,I2)                        PLOT 409
                     IF(I3.EQ.0) GO TO 840                                           PLOT 410
                     I2=I2+1                                                         PLOT 411
                     WRITE(IOUT  ,790)4HCTEX,(TEXT(I),I=I2,I3)                       PLOT 412
                790  FORMAT(A4,2X,66A1)                                             PLOT 413
                840  X=X+K2*(6./7.) *HGT(K1)*COS(ANGLE1)                            PLOT 414
                     Y=Y+K2*(6./7.)*HGT(K1)*SIN(ANGLE1)                              PLOT 415
                     K1=K                                                            PLOT 416
                850  CONTINUE                                                        PLOT 417
                     GO TO 700                                                       PLOT 418
               C                                                                     PLOT 419
               C                                                                     PLOT 420
               C     PRINT IDENTIFYING TEXT (TIME OF DAY)                            PLOT 421
               C                                                                     PLOT 422
               C                                                                     PLOT 423
                900  PLOTY=((YMAX-YMIN)*GROSZ)/YSCALE*PERCY*MAG+YOR                  PLOT 424
                     PLOTX=(XMAX-XMIN)/XSCALE*PERCX*MAG+XOR                          PLOT 425
                     IF(IFLAG.EQ.1) WRITE(XPRINT,902)                                PLOT 426
```

```
                      IF((IFLAG.NE.2).AND.(IFLAG.NE.1)) WRITE(KPRINT,802)                PLOT 427
                  802 FORMAT(1H0,18X,24HNO USER TEXT CARDS INPUT)                        PLOT 428
                      WRITE(KPRINT,159)PLOTX,PLOTY                                       PLOT 429
      460             FORMAT(1H0,18X,12HTHIS PLOT IS,I4,10H INCHES BY,I4,7H INCHES)      PLOT 430
                      L10=2                                                              PLOT 431
                C     SCALE IF MEASUREMENTS ARE IN METERS
                      IF(METERS) L10=1                                                   PLOT 432
                      FAKTOR=(XSCALE+YSCALE)/2/MAG                                       PLOT 433
      465             WRITE(KPRINT,149)FAKTOR,MEASURE(L10)                               PLOT 434
                  149 FORMAT(1H0,18X,20HONE INCH IS EQUAL TO,F7.0,1X,A10)                PLOT 435
                      DO 630 J1=1,50                                                     PLOT 436
                      IF(PLOTX.LE.J1*8.5) GO TO 633                                      PLOT 437
                  630 CONTINUE                                                           PLOT 438
      470         633 DO 634 J=1,10                                                      PLOT 439
                      IYSZ=SYTABLE+0.4
                      IF(PLOTY.LE.J*IYSZ) GO TO 635                                      PLOT 440
                  634 CONTINUE                                                           PLOT 441
                  635 Z1=0 $ X1=1.0*MAG $ ANGLE=90.0                $ HT=.3*MAG          PLOT 442
      475             X2=2.0*MAG                                                         PLOT 443
                      CALL DATE(B)                                                       PLOT 444
                      CALL GETJN(K)                                                      PLOT 445
                      DO 645  I=1,J                                                      PLOT 446
                      Y1=((I-1)*IYSZ+1)                                                  PLOT 447
      480       C     WRITE OUT HEIGHT, ANGLE MEASUREMENTS TO TAPE 55
                      WRITE(IOUT,640)X1,Y1,Z1,HT,ANGLE,IC1,K,B                           PLOT 448
                  640 FORMAT(4HTEXT,2X,5F8.3,I2,2A10,2X,2HPL)                            PLOT 449
                C     WRITE OUT NUMBER OF PLOTS REQUESTED TO TAPE 55
                      WRITE(IOUT,641)PLTCT,I,J                                           PLOT 450
      485         641 FORMAT(4HCTEX,2X,2HOT,I2,3X,I1,1X,2HOF,1X,I1)                      PLOT 451
                      WRITE(IOUT,642)X2,Y1,Z1,HT,ANGLE,IC1,SCALE                         PLOT 452
                  642 FORMAT(4HTEXT,2X,5F8.3,I2,8HSCALE 1:,F7.0,2X,7H1 INCH=)            PLOT 453
                      WRITE(IOUT,643)FAKTOR,MEASURE(L10)                                 PLOT 454
                  643 FORMAT(4HCTEX,2X,F6.0,1X,A10)                                      PLOT 455
      490         645 CONTINUE                                                           PLOT 456
                      WRITE(IOUT,301)4HEND                                               PLOT 457
                      WRITE(KPRINT,160) J1,J                                             PLOT 458
                  160 FORMAT(1H0,18X,14HIT CONSISTS OF,I3,* PAGES IN THE X DIRECTIO*,    PLOT 459
                     1*N AND*,I3,* SECTIONS IN THE Y DIRECTION*)
      495       C     COMPUTE TIME IN PLOT
                      CPTIME1=SECOND(CP)                                                 PLOT 461
                      CPTIME=CPTIME1-CPTIME                                              PLOT 462
                      WRITE(KPRINT,101)CPTIME                                            PLOT 463
                  101 FORMAT(1H0,21X,21H..... TIME IN PLOT IS,F8.3,6H .....)             PLOT 464
      500         900 RETURN                                                             PLOT 465
                      END                                                                PLOT 466
```

SYMBOLIC REFERENCE MAP (R=3)

ENTRY POINTS	DEF LINE	REFERENCES
1 PLOT	1	500

VARIABLES	SN	TYPE	RELOCATION								
2644	ANG	REAL		REFS	287	321	DEFINED	280	309	313	317
2677	ANGLE	REAL		REFS	379	386	431	434	481	486	
				DEFINED	378	474					
2701	ANGLE1	REAL		REFS	444	445	DEFINED	379			
2666	ANG1	REAL		REFS	290	292	4*302	DEFINED	287		
2611	B	REAL		REFS	48	51	476	481			
2726	BAS	INTEGER		REFS	27	32	342	DEFINED	57		
4	BDS	LOGICAL	BOUND	REFS	14	15	59				
2612	C	REAL		REFS	49	51					
2607	CP	REAL		REFS	41	496					
2606	CPTIME	REAL		REFS	497	498	DEFINED	41	497		
2720	CPTIME1	REAL		REFS	497	DEFINED	496				
2653	DISL	REAL		REFS	267	DEFINED	259				
2655	DIST	REAL		REFS	267	DEFINED	261				
2644	D2	REAL		REFS	254	267	DEFINED	252			
2645	D2L	REAL		REFS	267	DEFINED	253				
2712	FAKTOR	REAL		REFS	465	488	DEFINED	464			
2631	GRDNAME *	REAL		DEFINED	193						
2614	GRDSZ	REAL		REFS	263	353	454	DEFINED	53	193	
2745	HEAD	REAL	ARRAY	REFS	20	84	146	DEFINED	35		
3264	HGT	REAL	ARRAY	REFS	25	415	431	434	444	445	
				DEFINED	381	393	402				
2654	HGTL	REAL		REFS	267	DEFINED	260				
2663	HITE	REAL		REFS	288	289	300	321	DEFINED	280	
2676	HT	REAL		REFS	386	393	401	481	486		
				DEFINED	378	404	474				
2703	HT1	REAL		REFS	398	2*401	402	404	DEFINED	397	
2615	I	INTEGER		REFS	55	70	71	84	2*146	267	280
				321	378	381	386	397	398	431	434
				439	442	479	484	DEFINED	54	68	145
				264	280	321	378	380	386	397	398
				431	434	439	442	478			
2700	IC	INTEGER		REFS	386	430	431	434	DEFINED	378	
1323	IC1	INTEGER		REFS	481	486	DEFINED	39			
2652	ID	INTEGER		REFS	267	DEFINED	258				
2625	IDUMP	INTEGER		REFS	246	DEFINED	93				
2725	IFILE	INTEGER	ARRAY	REFS	21	4*32	70	146			
2610	IFLAG	INTEGER		REFS	374	456	2*457	DEFINED	46	100	385
2624	INC	INTEGER		REFS	2*137	138	153	154	162	163	252
				353	DEFINED	93	99	137			
2721	IN1	INTEGER		REFS	30	244	DEFINED	39	I/O REFS	192	193
				243							
2722	IN2	INTEGER		REFS	30	346	DEFINED	39	I/O REFS	344	345
2723	IN3	INTEGER		REFS	30	282	DEFINED	39	I/O REFS	277	278
				280							
2724	IN4	INTEGER		REFS	30	333	DEFINED	39	I/O REFS	331	332
1322	IOUT	INTEGER		DEFINED	39	I/O REFS	89	90	174	178	217
				219	227	235	245	248	267	279	321
				335	347	371	372	431	434	439	442
				481	484	486	488	491			
2617	ITEST	INTEGER		REFS	75	DEFINED	71	I/O REFS	72	73	76
2751	ITEXT	INTEGER	ARRAY	REFS	22	321	DEFINED	280			
2721	IUNIT	INTEGER	ARRAY	REFS	29	4*30	71				
2713	IYSZ	INTEGER		REFS	472	479	DEFINED	471			
2705	I1	INTEGER		REFS	431	434	437	439	DEFINED	413	420
				423	426	437					

VARIABLES	SN	TYPE	RELOCATION							
2706	I2	INTEGER		REFS	436	439	441	442	DEFINED	413	424
				427	441						
2707	I3	INTEGER		REFS	440	442	DEFINED	413	428		
2626	IS	INTEGER		REFS	159	168	DEFINED	152	156	165	
2702	J	INTEGER		REFS	390	395	397	398	402	403	412
				414	472	478	484	492	DEFINED	388	392
				408	470						
2616	JFLAG	INTEGER		REFS	78	80	DEFINED	67	86		
2627	J1	INTEGER		REFS	155	164	468	492	DEFINED	153	162
				467							
2630	J2	INTEGER		REFS	155	164	405	408	DEFINED	154	163
				403	407						
2613	K	INTEGER		REFS	50	51	397	398	403	407	414
				415	416	446	477	481	DEFINED	395	412
0	KARD	INTEGER	IO	REFS	12	96	383	I/O REFS	57	93	378
				397							
2647	KLEV	INTEGER		REFS	267	DEFINED	255				
1	KPRINT	INTEGER	IO	REFS	12	I/O REFS	42	51	60	81	84
				143	146	159	168	172	176	197	246
				351	353	364	375	386	398	456	457
				459	465	492	498				
2704	K1	INTEGER		REFS	416	420	423	424	426	427	2*431
				2*434	444	445	DEFINED	411	446		
2710	K2	INTEGER		REFS	419	420	422	424	428	444	445
				DEFINED	416						
2665	L	INTEGER		REFS	300	DEFINED	280				
2604	LABEL	INTEGER		REFS	26	3*134	135	256	353		
				DEFINED	93	99	134				
3	LARGE	LOGICAL	PLOTCM	REFS	13	15	224				
2673	LENTH	INTEGER		REFS	4*302	308	312	316	DEFINED	300	
2605	LINC	INTEGER		REFS	26	2*140	141	253	353		
				DEFINED	93	99	140				
2727	LOC	INTEGER		REFS	27	32	275	DEFINED	57		
2602	LSTART	INTEGER		REFS	26	2*128	129	151	162	251	353
				DEFINED	93	99	128				
2603	LSTOP	INTEGER		REFS	26	2*131	132	151	163	166	168
				353	DEFINED	93	99	131	166		
2650	1	INTEGER		REFS	267	DEFINED	256				
2711	L10	INTEGER		REFS	465	488	DEFINED	461	463		
2651	L2	INTEGER		REFS	267	DEFINED	257				
2574	MAG	REAL		REFS	18	2*116	117	213	219	222	225
				353	454	455	464	2*474	475		
				DEFINED	93	98	116				
3574	MEASURE	INTEGER	ARRAY	REFS	28	465	488	DEFINED	34		
0	METERS	LOGICAL	METRIC	REFS	16	17	211	212	463		
2644	NLEVS	INTEGER		REFS	267	DEFINED	254				
2623	PERCSM	REAL		REFS	2*119	120	263	353	DEFINED	93	98
				114							
2621	PENCX	REAL		REFS	2*110	111	209	222	353	455	
				DEFINED	93	98	110				
2622	PENCY	REAL		REFS	2*113	114	210	225	353	454	
				DEFINED	93	98	113				
2575	PLOTX	INTEGER		REFS	23	459	468	DEFINED	455		
2576	PLOTY	INTEGER		REFS	23	459	472	DEFINED	454		
0	PLTCT	INTEGER	PLOTCM	REFS	13	27	45	51	234	484	
				DEFINED	45						
2725	PUDG	INTEGER		REFS	27	32	59	190	242		

VARIABLES	SN	TYPE		RELOCATION							
					DEFINED	57					
2620	SCALE	REAL			REFS	107	108	209	210	353	486	
					DEFINED	93	98	107				
2730	SCAT	INTEGER			REFS	27	32	329	DEFINED	57		
2660	SKIP	REAL			REFS	267	DEFINED	265				
2577	STAR	INTEGER			REFS	24	384	386	394	398	410	
					DEFINED	378	397					
2731	STARS	REAL	ARRAY		REFS	19	12*353	DEFINED	55	108	111	114
					117	120	123	126	129	132	135	138
					141							
2600	START	INTEGER			REFS	26	2*122	123	150	153	250	254
					353	DEFINED	93	98	122			
2601	STOP	INTEGER			REFS	26	2*125	126	150	154	157	159
					254	353	DEFINED	93	99	125	157	
1	SXTABLE	REAL		PLOTCM	REFS	13	221	227	DEFINED	222		
2	SYTABLE	REAL		PLOTCM	REFS	13	227	471	DEFINED	223	225	
2754	TEXT	INTEGER	ARRAY		REFS	24	386	390	398	405	431	434
					439	442	DEFINED	378	397			
2657	TLER	REAL			REFS	267	DEFINED	263				
2656	TLNG	REAL			REFS	267	DEFINED	262				
2674	X	REAL			REFS	384	386	431	434	444		
					DEFINED	378	444					
2634	XMAX	REAL			REFS	195	197	217	222	302	455	
					DEFINED	193	207					
2	XMAX1	REAL		BOUND	REFS	14	195	197	207			
2632	XMIN	REAL			REFS	195	197	217	222	302	316	455
					DEFINED	193	205					
0	XMIN1	REAL		BOUND	REFS	14	195	197	205			
2667	XOFF	REAL			REFS	2*290	2*292	DEFINED	288			
2640	XOR	REAL			REFS	217	222	455	DEFINED	213		
2661	XPOS	REAL			REFS	294	2*302	316	321	DEFINED	280	294
2672	XPRIME	REAL			REFS	294	DEFINED	292				
2636	XSCALE	REAL			REFS	211	217	222	263	288	300	455
					464	DEFINED	209	211				
2715	X1	REAL			REFS	481	DEFINED	474				
2716	X2	REAL			REFS	486	DEFINED	475				
2675	Y	REAL			REFS	386	431	434	445	DEFINED	378	445
2635	YMAX	REAL			REFS	195	197	217	225	302	308	454
					DEFINED	193	208					
3	YMAX1	REAL		BOUND	REFS	14	195	197	208			
2633	YMIN	REAL			REFS	195	197	217	225	302	312	454
					DEFINED	193	206					
1	YMIN1	REAL		BOUND	REFS	14	195	197	206			
2670	YOFF	REAL			REFS	2*290	2*292	DEFINED	289			
2641	YOR	REAL			REFS	217	225	454	DEFINED	214		
2662	YPOS	REAL			REFS	295	2*302	308	312	321		
					DEFINED	290	295					
2671	YPRIME	REAL			REFS	295	DEFINED	290				
2637	YSCALE	REAL			REFS	212	217	225	263	289	454	464
					DEFINED	210	212					
2717	Y1	REAL			REFS	481	486	DEFINED	479			
1321	Z	REAL			REFS	431	434	DEFINED	39			
2643	ZLMIN	REAL			REFS	267	DEFINED	251				
2642	ZMIN	REAL			REFS	267	DEFINED	250				
2714	Z1	REAL			REFS	481	486	DEFINED	474			

VARIABLES USED AS FILE NAMES, SEE ABOVE

EXTERNALS	TYPE	ARGS	REFERENCES							
ATAN	REAL	1	LIBRARY	290	292					
COS	REAL	1	LIBRARY	292	2*302	444				
DATE		1		48	476					
EOF	REAL	1		75	96	244	282	333	346	383
GETJN		1		50	477					
SECOND	REAL	1		41	496					
SIN	REAL	1	LIBRARY	290	2*302	445				
SQRT	REAL	1	LIBRARY	290	292					
TIME		1		49						

INLINE FUNCTIONS	TYPE	ARGS	DEF LINE	REFERENCES			
MOD	INTEGER	2	INTRIN	153	154	162	163

STATEMENT LABELS			DEF LINE	REFERENCES							
1402	2	FMT	74	73							
53	3		77	68	70	87					
62	6		80	75							
1410	7	FMT	82	81							
66	8		84	80							
1426	9	FMT	85	84							
1440	10	FMT	91	89	90						
0	11		55	54							
2170	13	FMT	365	364							
1366	15	FMT	61	60							
34	17		67	59							
1361	20	FMT	58	57							
627	21		308	302							
635	22		312	308							
643	23		316	312							
651	49		319	316							
651	51		321	306	310	314	318	319			
2047	99	FMT	348	243	245	246	278	279	332	335	345	347
1463	100	FMT	95	93							
350	105		190	171							
0	106	INACTIVE	98	96							
1612	107	FMT	198	197							
377	108		205	190							
404	109		209	195	204						
1657	110	FMT	218	217							
115	111		107	96							
1670	115	FMT	220	219							
452	116		234	221							
450	117		227	224							
1701	120	FMT	228	227							
2005	149	FMT	466	465							
1327	150	FMT	43	42							
1471	151	FMT	144	143							
1503	152	FMT	147	146							
0	153		148	145							
2212	156	FMT	376	375							
2250	157	FMT	387	386							
2271	158	FMT	399	398							
2371	159	FMT	460	459							
2506	160	FMT	493	492							
2525	161	FMT	499	498							
1574	166	FMT	194	193							

```
STATEMENT LABELS                    DEF LINE   REFERENCES
 1344  181    FMT                       52        51
  313  200                             153       158
  320  210                             159       155
 1511  220    FMT                      160       159
  330  230                             162       167
  335  240                             168       164
 1526  250    FMT                      169       168
  342  300                             172       150
 1710  301    FMT                      236       235       491
 1542  310    FMT                      173       172
  347  320                             176       151
 1553  330    FMT                      177       176
  456  395                             242       234
  460  400                             243       247
    0  410          INACTIVE           245       244
  472  430                             248       244
 1730  440    FMT                      249       248
 1756  441    FMT                      269       267
  525  495                             275       242
  535  500                             280       323
 2006  501    FMT                      281       280
 2022  502    FMT                      322       321
  654  530                             329       275       282
  660  535                             332       336
    0  545          INACTIVE           335       333
  667  560                             342       329       333
  673  565                             345       350
    0  572          INACTIVE           347       346
 2064  575    FMT                      352       351
  702  580                             351       342       346
 2125  590    FMT                      357       353
    0  630                             469       467
 1207  633                             470       468
    0  634                             473       470
 1220  635                             474       472
 2426  640    FMT                      482       481
 2440  641    FMT                      485       484
 2457  642    FMT                      487       486
 2472  643    FMT                      489       488
    0  645                             490       478
 2265  690    FMT                      373       371       372
  726  700                             378       448
    0  701                             381       380
 2233  705    FMT                      382       378
    0  720                             391       388
  755  730                             393       390
  762  735                             395       410
 2274  740    FMT                      400       397
 1002  745                             403       401
    0  750                             406       403
 1013  760                             418       405
 1063  779                             434       430
 2312  780    FMT                      432       431
 2330  781    FMT                      435       434
 1074  782                             436       433
 2346  790    FMT                      443       439       442
 1135  800                             454       374       383       384
```

STATEMENT LABELS			DEF LINE	REFERENCES		
1017	801		411	390		
2357	802	FMT	458	456	457	
1026	815		416	414		
1034	820		422	419		
1043	830		426	422		
1114	840		435	436	440	
1051	845		431	421	425	
1132	850		447	412	415	
1266	900		500	175	174	
74	1111		89	79		

LOOPS	LABEL	INDEX	FROM-TO	LENGTH	PROPERTIES			
23	11	I	54 55	2B	INSTACK			
40	3	I	6B 77	16B		EXT REFS	ENTRIES	EXITS
272	153	I	145 148	11B		EXT REFS		
733	701	I	380 385	2B	INSTACK			
752	720	J	386 391	3B	INSTACK	EXITS		
1007	750	J2	403 406	3B	INSTACK	EXITS		
1021	850	K	412 447	114B		EXT REFS		
1204	630	J1	467 469	3B	INSTACK	EXITS		
1215	634	J	470 473	3B	INSTACK	EXITS		
1237	645	I	478 490	16B		EXT REFS		

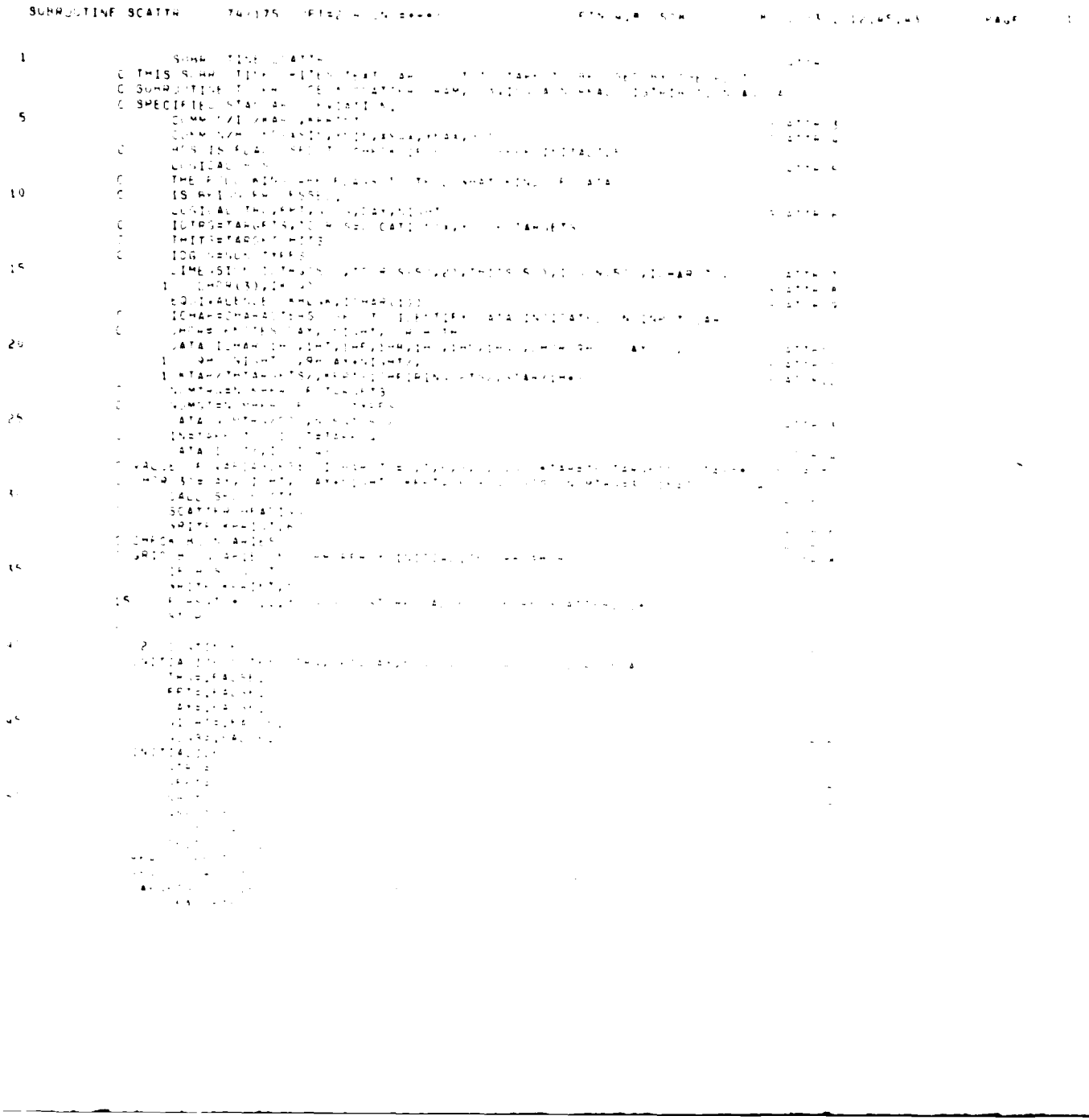
COMMON BLOCKS	LENGTH	MEMBERS - BIAS NAME(LENGTH)					
IO	2	0 NAPO (1)		1 KPRINT (1)			
PLOTCM	4	0 PLTCT (1)		1 SXTABLE(1)		2 SYTABLE(1)	
		3 LARGE (1)					
BOUND	5	0 XMIN1 (1)		1 YMIN1 (1)		2 XMAX1 (1)	
		3 YMAX1 (1)		4 BDS (1)			
METRIC	1	0 METERS (1)					

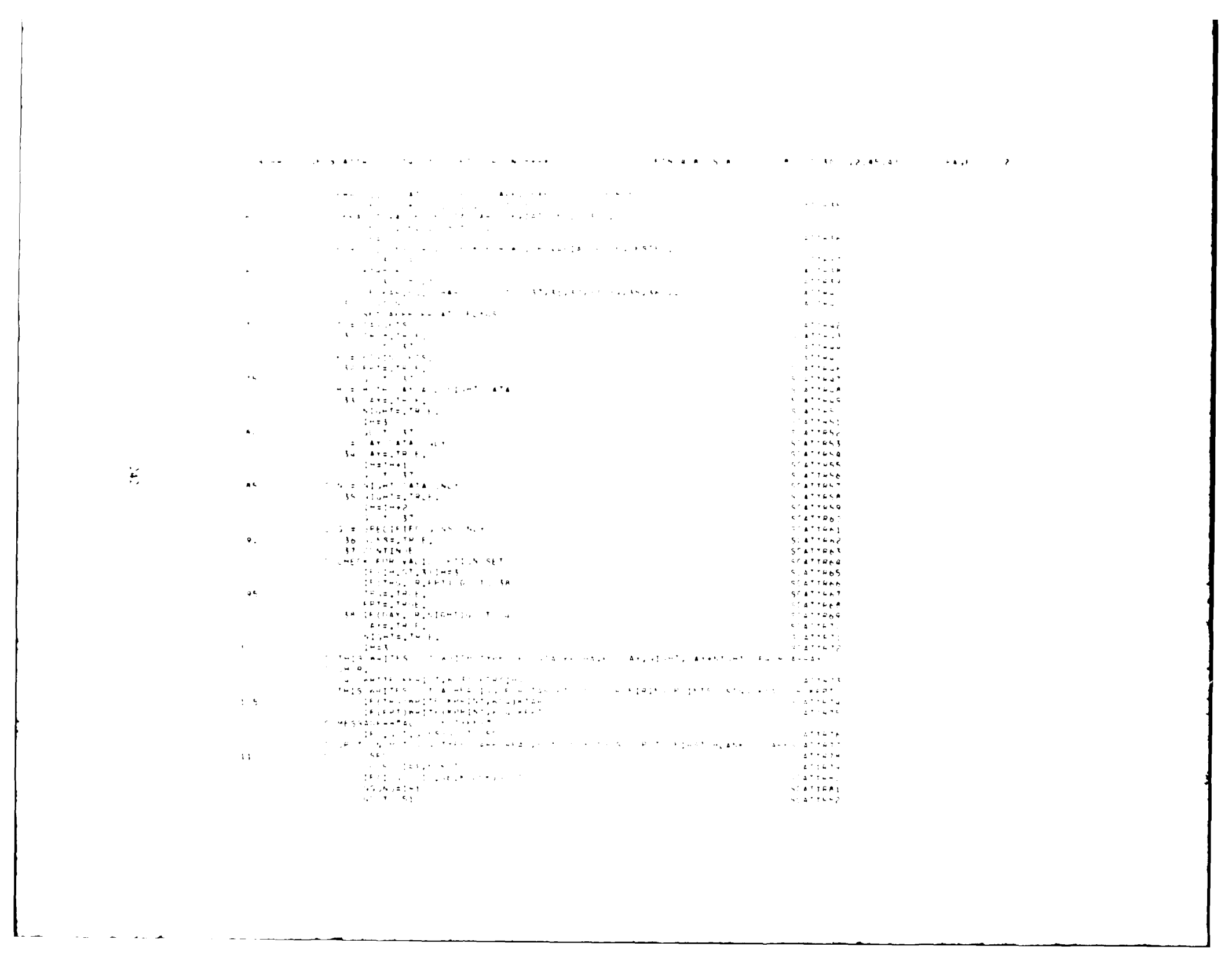
EQUIV CLASSES	LENGTH	MEMBERS - BIAS NAME(LENGTH)					
IUNIT	4	0 IN1 (1)		1 IN2 (1)		2 IN3 (1)	
		3 IN4 (1)					
IFILE	4	0 PUDG (1)		1 BAS (1)		2 LOC (1)	
		3 SCAT (1)					

STATISTICS

PROGRAM LENGTH	36348	1948
CM LABELED COMMON LENGTH	14B	12

600008 CM USED





```
```

```
C READ DEF CARD                                                   SCATT131
1     READ(IN,901)COL1,ITG,DAYNO,DARKNO,IDT,KFLAG                 SCATT132
      IF(.NOT.NGUNS)GO TO 320                                     SCATT133
C CHECK IF GUN IS REQUESTED GUN TYPE;IF NOT,READ NEXT DEF CARD    SCATT134
      DO 310 I=1,NGUNS                                            SCATT135
      IF(ITG.EQ.ITGUN(I))GO TO 320                                SCATT136
  310 CONTINUE                                                    SCATT137
      GO TO 390                                                   SCATT138
C IF TARGETS REQUESTED ,CHECK HIT FLAG;IF.NE.0,NO HIT             SCATT139
  320 IF(.NOT.TRG.OR.KFLAG.NE.0)GO TO 350                         SCATT140
      DO 325 I=1,NTRG                                             SCATT141
      IF(IDT.EQ.IETRG(I))GO TO 330                                SCATT142
  325 CONTINUE                                                    SCATT143
C ERROR -- UNDEFINED TARGET                                       SCATT144
      WRITE(KPRINT,902)IDT                                        SCATT145
      STOP                                                        SCATT146
C CUMULATE TARGET HITS                                            SCATT147
  330 IF(DAY) THITS(I)=THITS(I)+DAYNO                             SCATT148
      IF(NIGHT) THITS(I)=THITS(I)+DARKNO                          SCATT149
  350 IF(.NOT.FPT)GO TO 390                                       SCATT150
C CUMULATE FIRING PT SHOTS                                        SCATT151
      IF(DAY)FSHOTS=FSHOTS+DAYNO                                  SCATT152
      IF(NIGHT)FSHOTS=FSHOTS+DARKNO                               SCATT153
C LAST DEF CARDN                                                  SCATT154
  390 IF(COL1.NE.STAR)GO TO 305                                   SCATT155
      IF(.NOT.FPT)GO TO 300                                       SCATT156
C     TEST IF DATA POINT IS IN BOUNDS  IF NOT DON'T DO
      IF (X.LT.XMIN.OR.X.GT.XMAX)  GOTO 300
      IF (Y.LT.YMIN.OR.Y.GT.YMAX)  GOTO 300
C PROCESS ALL POINTS FOR 1 FIRING PT                              SCATT157
      N=FACT*FSHOTS/DAYS+.5                                       SCATT158
C     TOTAL NUMBER OF FIRINGS FOR 1 FIRING POINT
      NSHOTS=NSHOTS+N                                             SCATT159
C     CALCULATE STANDARD DEVIATION
      SS=(SD/10.)*(SQRT(FLOAT(N)))
      IF(N.GT.0)CALL SCATPL(N,X,Y,SS)                             SCATT160
      GO TO 300                                                   SCATT161
C PROCESS TARGET HITS,IF REQUESTED                                SCATT162
  400 IF(.NOT.TRG)GO TO 500                                       SCATT163
      DO 450 I=1,NTRG                                             SCATT164
      IF(THITS(I).EQ.0.)GO TO 450                                 SCATT165
C     TEST IF DATA POINT IS IN BOUNDS
      IF [illegible],1).LT.XMIN.OR.[illegible](I,1).GT.XMAX  GOTO 450
      IF [illegible],2).LT.YMIN.OR.[illegible],2).GT.YMAX) GOTO 450
C     TOTAL [illegible] TARGET HITS [illegible]
      N=FACT*[illegible]                                          SCATT166
      NHIT=NHIT+N                                                 SCATT167
C COMPUTE THE STANDARD DEVIATION [illegible] TO THE
C NUMBER OF [illegible]
      SS=[illegible]
      [illegible]                                                 SCATT168
  450 CONTINUE                                                    SCATT169
C [illegible]                                                     SCATT170
C WRITE THAT [illegible]                                          SCATT171
5     WRITE([illegible]                                           SCATT172
      [illegible]                                                 SCATT173
C STATI[illegible]                                                SCATT174
```

```
      WRITE(KPRINT,611)IOUT,NTRG,NHITS,NFPT,NSHOTS,FACT             SCATT175
C COMPUTE TIME SPENT IN SCATTER,
      CALL SECOND(T99)                                               SCATT176
      T99=T99-T100                                                   SCATT177
      WRITE(KPRINT,904)T99                                           SCATT178
      RETURN                                                         SCATT179
C EOF -- INPUT DIRECTIVES                                            SCATT180
  550 WRITE(KPRINT,905) KARD                                         SCATT181
      STOP                                                           SCATT182
C EOF - DATA BASE                                                    SCATT183
  590 WRITE(KPRINT,905)IN                                            SCATT184
      STOP                                                           SCATT185
  600 FORMAT(1H1///T15,*.....      SCATTER      .....*)              SCATT186
  601 FORMAT(4A1,26A2,2F10.0)                                        SCATT187
  602 FORMAT(F10.0)                                                  SCATT188
  603 FORMAT(////* THE SCATTER DIAGRAM WILL REPRESENT ..*,A9,*.. DATA FOSCATT189
     1R:*//)                                                         SCATT190
  604 FORMAT(/10X,*...*,A10)                                         SCATT191
  605 FORMAT(/10X,*...GUN TYPES:*,33(1X,A2))                         SCATT192
  606 FORMAT(/10X,*...ALL GUN TYPES*)                                SCATT193
  607 FORMAT(//5X,*...STANDARD DEVIATION =*,F10.3)                   SCATT194
  608 FORMAT(//5X,*...*,F4.0,* DAYS DATA IN DATA BASE*)             SCATT195
  609 FORMAT(A1,1X,A3,1X,2F6.0)                                      SCATT196
  610 FORMAT(A1,17X,A2,2F4.0,4X,A3,I1)                               SCATT197
  611 FORMAT(////* OUTPUT FILE TAPE*,I1,* CONTAINS 'ASAPLOT PHS4 INPUT DSCATT198
     1ATA REPRESENTING:*                                             SCATT199
     1                 ///5X,I5,* TARGETS    :*,I6,* SCATTER PTS.*   SCATT200
     1                  //5X,I5,* FIRING PTS:*,I6,* SCATTER PTS.*    SCATT201
     1                 ///5X,*MULTIPLIER USED =*,F10.1)              SCATT202
  612 FORMAT(////5X,*...MULTIPLIER =*,F10.1)                         SCATT203
  613 FORMAT(//* BOUNDED BY (*,F9.1,*,*,F9.1,*) - (*,F9.1,*,*,F9.1,*)*) SCATT204
  700 FORMAT(4HFHS4)                                                 SCATT205
  701 FORMAT(4HECHO)                                                 SCATT206
  901 FORMAT(///6OH *****      ERROR -- TARGET TABLE OVERFLOW: EXECUTION SCATT207
     1ABORTED        )                                               SCATT208
  902 FORMAT(///H H *****      ERROR -- ERROR IN DATA BASE: UNDEFINED TARSCATT209
     1GET ID   ,A3,*; EXECUTION ABORTED *)                           SCATT210
  904 FORMAT(///T15,*.....      TIME IN SCATTER IS*,F8.3,* SECONDS*)   SCATT213
  905 FORMAT(///41H *****      ERROR -- FORMAT OF EOF ON TAPE ,I1,*; EXECSCATT214
     1UTION ABORTED*)                                                SCATT215
  906 FORMAT(///HHH *****      ERROR -- ... SELECTED BUT NO GUN TYSCATT216
     1PES SPECIFIED: EXECUTION ABORTED                               SCATT217
      END                                                            SCATT218
```

```
SYMBOLIC REFERENCE MAP (R=3)

ENTRY POINTS      DEF LINE      REFERENCES
    1  SCATT2          1          23.

VARIABLES       SN   TYPE              RELOCATION
     4  ACS          LOGICAL
  1136  CELL         REAL                          ...     ...     15.     1..     196    DEFINED    146     15.
                                                   16.     1..     17.
```

VARIABLES SN TYPE RELOCATION

1145	DARKNS	REAL		REFS	190	194	DEFINED	173				
1117	DAY	LOGICAL		REFS	11	97	189	193	DEFINED	44	77	
					82	98						
1144	DAYNS	REAL		REFS	189	193	DEFINED	173				
1135	DAYS	REAL		REFS	136	137	202	217	DEFINED	132	136	
1554	DHOR	REAL	ARRAY	REFS	15	103	DEFINED	20				
1127	FACT	REAL		REFS	128	130	202	217	229			
				DEFINED	57	128						
1115	FPT	LOGICAL		REFS	11	94	106	191	197			
				DEFINED	43	74	96					
1142	FSHOTS	REAL		REFS	193	194	202	DEFINED	171	193	194	
1116	GUNS	LOGICAL		REFS	11	108	174	DEFINED	46	90		
1126	I	INTEGER		REFS	53	57	65	112	113	122	3*155	
					177	183	2*189	2*190	212	2*214	2*215	217
					2*222	DEFINED	52	57	64	111	122	154
					176	182	211					
1153	ICHAR	INTEGER	ARRAY	REFS	15	17	67	DEFINED	20			
1137	ID	* INTEGER		DEFINED	168							
1143	IDG	INTEGER		REFS	177	DEFINED	173					
1472	IDGUN	INTEGER	ARRAY	REFS	15	112	122	177	DEFINED	57		
1146	IDT	INTEGER		REFS	183	186	DEFINED	173				
1162	IDTRG	INTEGER	ARRAY	REFS	15	183	DEFINED	155				
1131	IH	INTEGER		REFS	83	87	93	103	DEFINED	62	79	
					83	87	93	100				
1557	IK	INTEGER	ARRAY	REFS	15	65	DEFINED	57				
461	IN	INTEGER		REFS	148	156	165	169	239			
				DEFINED	27	I/O REFS	144	146	155	164	168	
					173							
462	IOUT	INTEGER		REFS	229	DEFINED	27	I/O REFS	139	141	142	
					226	227						
1133	J	INTEGER		REFS	2*67	DEFINED	66					
1132	KAR	INTEGER		REFS	67	DEFINED	65					
0	KARD	INTEGER	IO	REFS	5	59	134	236	I/O REFS	57	132	
1153	KBLNK	INTEGER		REFS	17	112						
1147	KFLAG	INTEGER		REFS	181	DEFINED	173					
455	KFPT	INTEGER		REFS	106	DEFINED	20					
1	KPRINT	INTEGER	IO	REFS	5	I/O REFS	32	36	103	105	106	
					119	122	124	126	130	131	137	161
					196	229	233	236	239			
454	KTAR	INTEGER		REFS	105	DEFINED	20					
1150	L	INTEGER		REFS	204	206	2*207	218	221	2*222		
				DEFINED	202	217						
1123	NFPT	INTEGER		REFS	170	229	DEFINED	49	170			
1134	NGUNS	INTEGER		REFS	117	122	176	DEFINED	113	116		
1124	NHITS	INTEGER		REFS	218	229	DEFINED	50	218			
1120	NIGHT	LOGICAL		REFS	11	97	190	194	DEFINED	45	78	
					84	99						
1125	NSHOTS	INTEGER		REFS	204	229	DEFINED	51	204			
1122	NTRG	INTEGER		REFS	157	182	211	229	DEFINED	48	153	
					157							
460	NUMGT	INTEGER		REFS	111	116	DEFINED	25				
457	NUMTRG	INTEGER		REFS	52	154	DEFINED	25				
1130	SD	REAL		REFS	61	131	206	221	DEFINED	57	61	
1151	SS	REAL		REFS	207	222	DEFINED	206	221			
456	STAR	REAL		REFS	149	158	186	196	DEFINED	20		
1244	TCOORDS	REAL	ARRAY	REFS	15	2*214	2*215	2*222	DEFINED	2*155		
1410	THITS	REAL	ARRAY	REFS	15	189	190	212	217			

VARIABLES	SN	TYPE	RELOCATION						
				DEFINED	53	189	190			
1114	TRG	LOGICAL		REFS	11	94	105	151	181	210
				DEFINED	42	71	95			
1121	T100	REAL		REFS	30	232				
1152	T99	REAL		REFS	231	232	233	DEFINED	232	
1140	X	REAL		REFS	2*199	207	DEFINED	168		
2	XMAX	REAL	ARGUMENT	REFS	6	126	199	214		
0	XMIN	REAL	ARGUMENT	REFS	6	126	199	214		
1141	Y	REAL		REFS	2*200	207	DEFINED	168		
3	YMAX	REAL	ARGUMENT	REFS	6	126	200	215		
1	YMIN	REAL	ARGUMENT	REFS	6	126	200	215		

VARIABLES USED AS FILE NAMES, SEE ABOVE

EXTERNALS	TYPE	ARGS		REFERENCES					
EOF	REAL	1		59	134	148	156	165	169
SCATPL		4		207	222				
SECOND		1		30	231				
SQRT	REAL	1	LIBRARY	206	221				

INLINE FUNCTIONS	TYPE	ARGS		DEF LINE	REFERENCES					
FLOAT	REAL	1	INTRIN		206	221				

STATEMENT LABELS			DEF LINE	REFERENCES					
0	10		53	52					
471	15	FMT	37	36					
12	20		40	35					
0	30		68	66					
54	31		71	67					
56	32		74	67					
60	33		77	67					
63	34		82	67					
66	35		86	67					
71	36		90	67					
72	37		91	67	72	75	80	84	88	
101	38		97	94						
105	40		103	97						
127	50		115	111	112					
132	51		117	114						
136	52		122	117						
145	55		124	124						
147	60		126	126						
212	100		148	149						
0	210		159	154						
241	290		164	151	166					
250	300		168	158	197	199	200	208		
257	305		173	196						
0	310		178	176						
270	320		181	174	177					
0	325		184	182						
308	330		189	183						
311	350		191	181						
322	390		196	179	191					
331	400		210	159						
342	450		223	211	212	214	215			
435	510		228	210						
422	550		238	59	134					
425	590		239	148	156	165				

```
STATEMENT LABELS                DEF LINE    REFERENCES
   701  600      FMT               241         32
   707  601      FMT               242         57
   712  602      FMT               243        132
   714  603      FMT               244        103
   724  604      FMT               246        105        106
   727  605      FMT               247        122
   734  606      FMT               248        124
   740  607      FMT               249        131
   745  608      FMT               250        137
   753  609      FMT               251        146        155        164        168
   756  610      FMT               252        173
   762  611      FMT               253        229
  1007  612      FMT               258        130
  1014  613      FMT               259        126
  1024  700      FMT               260        141
  1026  701      FMT               261        142        226
  1030  901      FMT               262        161
  1040  902      FMT               264        186
  1053  904      FMT               266        233
  1062  905      FMT               267        236        239
  1073  906      FMT               269        119

LOOPS  LABEL    INDEX     FROM-TO    LENGTH    PROPERTIES
   21  10       I           52 53        28     INSTACK
   35  37       I           64 91       368                NOT INNER
   37  30       J           66 68       158     INSTACK      EXITS
  124  50       I          111 115       48     INSTACK      EXITS
  217  210      I          154 159      208                EXT REFS     EXITS
  265  310      I          176 178       28     INSTACK      EXITS
  276  325      I          182 184       28     INSTACK      EXITS
  353  450      I          211 223      328                EXT REFS

COMMON BLOCKS   LENGTH     MEMBERS - BIAS NAME(LENGTH)
       IO            2              0 KARD   (1)                  1 KPRINT (1)
       BOUND         5              0 XMIN   (1)                  1 YMIN   (1)                  2 XMAX   (1)
                                    3 YMAX   (1)                  4 BDS    (1)

EQUIV CLASSES   LENGTH     MEMBERS - BIAS NAME(LENGTH)
       KBLNK         7              0 ICHAR  (7)

STATISTICS
  PROGRAM LENGTH              15638      883
  CM LABELED COMMON LENGTH       78        7
        600008 CM USED
```

```
      SUBROUTINE SCATPL(N,X,Y,SD)                                          SCATPL 2
C THIS ROUTINE PLOTS POINTS ACCORDING TO A NORMAL DISTRIBUTION AROUND THE
C TARGET AREA.  (CALLED FROM SUBROUTINE SCATTER)
      COMMON/IO/KARD,KPRINT                                                SCATPL 3
      COMMON/BOUND/XMIN,YMIN,XMAX,YMAX                                     SCATPL 4
C     XD AND YD STORE THE COORDINATES OF THE DOTS
C     NCALLS= NUMBER OF TIMES THE RANDOM NO. GENERATOR IS CALLED
C     NLEFT=NUMBER OF CALLS LEFT
      DIMENSION XD(1000),YD(1000),REP(1)                                   SCATPL 5
      DATA IOUT/4/                                                         SCATPL 6
      DATA MAXRN/1000/,LIMOUT/5/,NREP/1/                                   SCATPL 7
      DATA HGT/.1/,NVAR/1/,IVAR/1/,IPRNT/0/,IX/123456789/,IY/987654321/    SCATPL 8
C** ADDITIONAL DATA TO CONVERT TO NCS "RAND" (FROM NSROC "NRAND")           DE100379
      DATA RNOPT/1/,FMX/16777213/,FMY/16777219/                            DE100379
C                                                                          DE100379
C VALUE OF VARIABLES:  IOUT=4  LIMOUT=5  HGT=.1  IVAR=1  IX=123456789
C MAXRN=1000  NREP=1  NVAR=1  IPRNT=0  IY=987654321
      NCALLS=(N-1)/MAXRN                                                   SCATPL 9
C N IS THE VALUE SENT FROM SCATTER, WHILE PROCESSING ALL NOISE POINTS
C FOR ONE NOISE SOURCE.  NCALLS IS THE INTEGER VALUE OF THIS DIVISION STATEMENT.
      NLEFT=(N-1)-NCALLS*MAXRN                                             SCATPL10
      NRN=MAXRN                                                            SCATPL11
C GENERATE A DOT AT THE NOISE SOURCE.
      XDOT=X
      YDOT=Y
      WRITE(IOUT,1)XDOT,YDOT,HGT
      IF(N.LE.1)RETURN
   50 IF(NCALLS.GT.0)GO TO 100                                             SCATPL12
C     NO SOURCES LEFT-RETURN
      IF(NLEFT .LE. 0) RETURN
      NRN=NLEFT                                                            SCATPL14
      NLEFT=0                                                              SCATPL15
C**************************************************                        DE100379
C LIBRARY ROUTINE NRAND                                                    SCATPL16
C GENERATES PSEUDO-RANDOM NOS. - NORMALLY DISTRIBUTED                      SCATPL17
C ..... SOURCE                                                             SCATPL18
C MATH SCIENCE LIBRARY -- CDC PROPRIETARY PRODUCT                          SCATPL19
C VOL. 7, P. 7-151                                                         SCATPL20
C .....PARAMETERS                                                          SCATPL21
C NRAND(N,M,I,XM,SIG,IU,X,IP)                                              SCATPL22
C  N   = TOTAL NO. RANDOM NOS. TO BE GENERATED                             SCATPL23
C  M   = TOTAL NO. VARIABLES IN DATA ARRAY X                               SCATPL24
C  I   = RANDOM NOS. WILL BE STORED AS VARIABLE I IN THE MULTIPLEXED       SCATPL25
C        ARRAY X                                                           SCATPL26
C  XM  = MEAN VALUE                                                        SCATPL27
C  SIG = STANDARD DEVIATION                                                SCATPL28
C  IU = START MULTIPLIER -- MUST BE ODD                                    SCATPL29
C  IP  = PRINT INDICATOR -- .GT.0 = NOS. WILL BE PRINTED                   SCATPL30
C  X   = MULTIPLEXED DATA ARRAY                                            SCATPL31
C**************************************************                        DE100379
C** ABOVE ROUTINE "NRAND" IS NO LONGER IN USE                              DE100379
C** HAS BEEN REPLACED WITH NCS ROUTINE "RANDU"                             DE100379
C** WILL DOCUMENT IN FULL AT LATER DATE                                    DE100379
C                                                                          DE100379
C** OLD SUBROUTINE CALLS                                                  DE100379
C*100 CALL NRAND(NRN,NVAR,IVAR,X,SD,IX,XD,IPRNT)                           SCATPL32
C*    CALL NRAND(NRN,NVAR,IVAR,Y,SD,IY,YD,IPRNT)                           SCATPL33
```

```
      C                                                                 DE100379
      C** NEW SUBROUTINE CALLS                                          DE100379
   60   100 CALL RAND(IX,FMX,RNOPT,NRN,XD)                             DE100379
            CALL RAND(IY,FMY,RNOPT,NRN,YD)                             DE100379

      C** CONVERT NORMALLY DISTRIBUTED RANDOM NUMBERS W/ MEAN=0 AND
      C** SD=1, TO NORMALLY DISTRIBUTED RANDOM NUMBERS W/ MEAN=X(OR Y)
   65 C** AND SD="SD" (INPUT TO SUBROUTINE THROUGH PARAM LIST).
            DO 105 I=1,NRN
            XD(I)=SD*XD(I)+X
      105   YD(I)=SD*YD(I)+Y
      C                                                                 DE100379
   70 C KEEPS TRACK OF HOW MANY POINTS WE MUST COMPUTE.
            NCALLS=NCALLS-1                                            SCATPL34
            DO 200 I=1,NRN                                             SCATPL35
            NOUT=0                                                     SCATPL36
            XDOT=XD(I)                                                 SCATPL37
   75 C CHECK IF PT. IS WITHIN PLOT BOUNDARY                           SCATPL38
      110   IF((XDOT.GE.XMIN).AND.(XDOT.LE.XMAX))GO TO 150             SCATPL39
      C XDOT OUT OF BOUNDS -- TRY AGAIN 2                              SCATPL40
            NOUT=NOUT+1                                                SCATPL41
            IF(NOUT.GT.LIMOUT)GO TO 190                                SCATPL42
   80 C                                                                DE100379
      C** OLD DOCUMENTATION                                            DE100379
      C************************************************               DE100379
      C THE PARAMETERS SENT TO NRAND HAVE THE FOLLOWING USE IN THAT LIBRARY ROUTINE:
      C     NRN=N=TOTAL NO. OF RANDOM NUMBERS TO BE GENERATED
   85 C     NVAR=M=TOTAL NO. OF VARIABLES IN DATA ARRAY X
      C     IVAR=I=RANDOM NOS. WILL BE STORED AS I IN ARRAY X.
      C     X(Y)=XM=MEAN VALUE
      C SD=SIG=STANDARD DEVIATION
      C XD(YD)=PRINT INDICATOR--GT. 0--NOS. WILL BE PRINTED.
   90 C IX(IY)=START MULTIPLIER--MUST BE ODD.
      C IPRNT=X=MULTIPLEXED ARRAY
      C************************************************               DE100379
      C                                                                 DE100379
      C** OLD SUBROUTINE CALL                                          DE100379
   95 C*    CALL NRAND(NREP,NVAR,IVAR,X,SD,IX,REP,IPRNT)               SCATPL43
      C                                                                 DE100379
      C** NEW SUBROUTINE CALL                                          DE100379
            CALL RAND(IX,FMX,RNOPT,NREP,REP)                           DE100379
      C** CONVERT MEAN AND SD
  100       REP=SD*REP+X
      C                                                                 DE100379
            XDOT=REP(1)                                                SCATPL44
            GO TO 110                                                  SCATPL45
      150   NOUT=0                                                     SCATPL46
  105       YDOT=YD(I)                                                 SCATPL47
      160   IF((YDOT.GE.YMIN).AND.(YDOT.LE.YMAX))GO TO 180             SCATPL48
      C YDOT OUT OF BOUNDS -- TRY AGAIN                                SCATPL49
            NOUT=NOUT+1                                                SCATPL50
      C "WARNING--GENERATED SCATTER POINT FOR LOCATION (X,Y) OUT OF BOUNDS
  110 C AFTER (LIMOUT) TRIES; POINT IGNORED.
            IF(NOUT.GT.LIMOUT)GO TO 190                                SCATPL51
      C                                                                 DE100379
      C** OLD SUBROUTINE CALL                                          DE100379
      C*    CALL NRAND(NREP,NVAR,IVAR,Y,SD,IY,REP,IPRNT)               SCATPL52
```

```
115          C                                                                  DE100379
             C** NEW SUBROUTINE CALL                                            DE100379
                   CALL RANC(IY,FMY,RNOPT,NREP,REP)                             DE100379
             C** CONVERT MEAN AND SD
                   REP=SD*REP+Y
120          C                                                                  DE100379
                   YDOT=REP(1)                                                  SCATPL53
                   GO TO 180                                                    SCATPL54
             C CREATE NASAPLOT TEXT CARD TO PUT OUT AT (XDOT,YDOT)              SCATPL55
               180 WRITE(IOUT,1)XDOT,YDOT,HGT                                   SCATPL56
125              1 FORMAT(*TEXT*,2X,2F8.0,6X,F8.3,T48,*1,*)                     SCATPL57
                   GO TO 200                                                    SCATPL58
             C UNRECOVERABLE OUT OF BOUNDS CONDITION                            SCATPL59
               190 WRITE(KPRINT,191)X,Y,LIMOUT                                  SCATPL60
               191 FORMAT(1H0***** ,* WARNING -- GENERATED SCATTER POINT FOR LOCATIONSCATPL61
130               * (*,F8.0,*,*,F8.0,*) OUT OF BOUNDS AFTER*,I3,* TRIES: PT. IGNORED*SCATPL62
                  2)                                                            SCATPL63
               200 CONTINUE                                                     SCATPL64
                   GO TO 50                                                     SCATPL65
                   END                                                          SCATPL67
```

```
CARD NR. SEVERITY  DETAILS    DIAGNOSIS OF PROBLEM

   100      I        REP       ARRAY NAME OPERAND NOT SUBSCRIPTED, FIRST ELEMENT WILL BE USED.
   100      I        REP       ARRAY NAME OPERAND NOT SUBSCRIPTED, FIRST ELEMENT WILL BE USED.
   119      I        REP       ARRAY NAME OPERAND NOT SUBSCRIPTED, FIRST ELEMENT WILL BE USED.
   119      I        REP       ARRAY NAME OPERAND NOT SUBSCRIPTED, FIRST ELEMENT WILL BE USED.
```

```
       SYMBOLIC REFERENCE MAP (R=3)

ENTRY POINTS      DEF LINE      REFERENCES
   3  SCATPL          1            27        30

VARIABLES      SN  TYPE              RELOCATION
  157  FMX         REAL                           REFS       60       98    DEFINED      14
  160  FMY         REAL                           REFS       61      117    DEFINED      14
  150  HGT         REAL                           REFS       26      124    DEFINED      12
  233  I           INTEGER                        REFS     2*67     2*69       74       105   DEFINED     66     72
  144  IOUT        INTEGER                     DEFINED       10   I/O REFS      26       124
  153  IPRNT    *  INTEGER                     DEFINED       12
  152  IVAR     *  INTEGER                     DEFINED       12
  154  IX          INTEGER                        REFS       60       98    DEFINED      12
  155  IY          INTEGER                        REFS       61      117    DEFINED      12
    0  KARD        INTEGER           IO           REFS        4
    1  KPRINT      INTEGER           IO           REFS        4   I/O REFS     128
  146  LIMOUT      INTEGER                        REFS       79      111       128   DEFINED      11
  145  MAXNG       INTEGER                        REFS       18       21        22   DEFINED      11
    0  [illegible] INTEGER           F.P.         REFS       13       21        27   DEFINED       1
  226  NCALLS      INTEGER                        REFS       21       24        71   DEFINED      18     71
  227  NLEFT       INTEGER                        REFS       30       31    DEFINED      21      32
```

```
      SUBROUTINE SCATPL     74/175  OPT=2 ROUND=+-*/             FTN 4.8  508          80/10/30. 12.45.43       PAGE     4

VARIABLES       SN  TYPE              RELOCATION
  234  NOUT         INTEGER                          REFS      78       79      108       111    DEFINED      73        78
                                                      104     108
  147  NEP          INTEGER                          REFS      98      117    DEFINED      11
  23   NN           INTEGER                          REFS      60       61       66        72    DEFINED      22        31
  151  NRAN       * INTEGER                        DEFINED     12
 4155  NFP          REAL         ARRAY               REFS       9       98      100       102       117      119       121
                                                   DEFINED    100      119
  156  NRUNT        REAL                             REFS      60       61       98       117    DEFINED      14
    0  SD           REAL                F.P.         REFS      67       68      100       119    DEFINED       1
    0  X            REAL                F.P.         REFS      24       67      100       128    DEFINED       1
  235  XD           REAL         ARRAY               REFS       9       60       67        74    DEFINED      67
  231  XOUT         REAL                             REFS      26     2*76      124   DEFINED        24       74       102
    2  XMAX         REAL                BOUND        REFS       5       76
    0  XMIN         REAL                BOUND        REFS       5       76
    0  Y            REAL                F.P.         REFS      25       68      119       128    DEFINED       1
 2205  YD           REAL         ARRAY               REFS       9       61       68       105    DEFINED      68
  232  YOUT         REAL                             REFS      26    2*106      124   DEFINED        25      105       121
    3  YMAX         REAL                BOUND        REFS       5      106
    1  YMIN         REAL                BOUND        REFS       5      106
       VARIABLES USED AS FILE NAMES, SEE ABOVE

EXTERNALS          TYPE   ARGS        REFERENCES
       RAND                5               60       61       98       117

STATEMENT LABELS                 DEF LINE   REFERENCES
  175  1          FMT              125         26        124
   24  50                           28        133
   30  100                          60         28
    0  105                          68         66
   51  110                          76        103
   65  150                         108         76
   70  160                         106        122
  104  180                         124        106
  107  190                         128         79        111
  210  191        FMT              129        128
  111  200                         132         72        126

LOOPS  LABEL     INDEX      FROM-TO     LENGTH     PROPERTIES
   40  105       I           66 68        4B       INSTACK
   46  200       I           72 132     464B                     EXT REFS

COMMON BLOCKS    LENGTH      MEMBERS - BIAS NAME(LENGTH)
       IO           2                0 NAPI    (1)                1 KPRINT  (1)
       ROUND        4                0 XMIN    (1)                1 YMIN    (1)                  2 XMAX     (1)
                                     3 YMAX    (1)

STATISTICS
  PROGRAM LENGTH                  4166B     2166
  CM LABELED COMMON LENGTH          6B         6
        60000B CM USED
```

```
SUBROUTINE STOPP       74/175  OPT=2 ROUND=+-*/               FTN 4.8  508        80/10/30. 12.45.43       PAGE      1

 1                 SUBROUTINE STOPP                                                          STOPP  2
                   COMMON /IO/KARD,KPRINT                                                    STOPP  3
                   DATA IOUT/55/                                                             STOPP  4
                   WRITE(KPRINT,10)                                                          STOPP  5
 5              10 FORMAT(1H1///T15,*.....      STOP      .....*)                            STOPP  6
                   WRITE(IOUT,100)                                                           STOPP  7
               100 FORMAT(4HSTOP)                                                            STOPP  8
                   WRITE(KPRINT,200)                                                         STOPP  9
               200 FORMAT(////T13,*NASAPLOT INPUT FILE COMPLETE.*)                           STOPP 10
10                 RETURN                                                                    STOPP 11
                   END                                                                       STOPP 12


       SYMBOLIC REFERENCE MAP (R=3)

ENTRY POINTS       DEF LINE      REFERENCES
    1  STOPP           1             10

VARIABLES       SN  TYPE             RELOCATION
   11  IOUT         INTEGER                          DEFINED        3    I/O REFS       6
    0  KARD         INTEGER               IO          REFS          2
    1  KPRINT       INTEGER.              IO          REFS          2    I/O REFS       4         8
       VARIABLES USED AS FILE NAMES, SEE ABOVE

STATEMENT LABELS              DEF LINE   REFERENCES
   15  10        FMT              5          4
   25  100       FMT              7          6
   32  200       FMT              9          8

COMMON BLOCKS    LENGTH      MEMBERS - BIAS NAME(LENGTH)
       IO            2                 0 KARD    (1)                 1 KPRINT (1)

STATISTICS
  PROGRAM LENGTH                 40B        32
  CM LABELED COMMON LENGTH        2B         2
         60000B CM USED
```

CERL DISTRIBUTION

Chief of Engineers
ATTN: DAEN-ASI-L (2)
ATTN: DAEN-CCP
ATTN: DAEN-MPE-I
ATTN: DAEN-ZCE

US Government Printing Office
Receiving Section/Depository Copies (2)

Defense Technical Information Center
ATTN: DDA (12)

Army Environmental Hygiene Agency (5)

FAA
ATTN: AEE-120

USAFHQ
ATTN: AF-Lee

NAFAC
ATTN: Code 09PB (5)

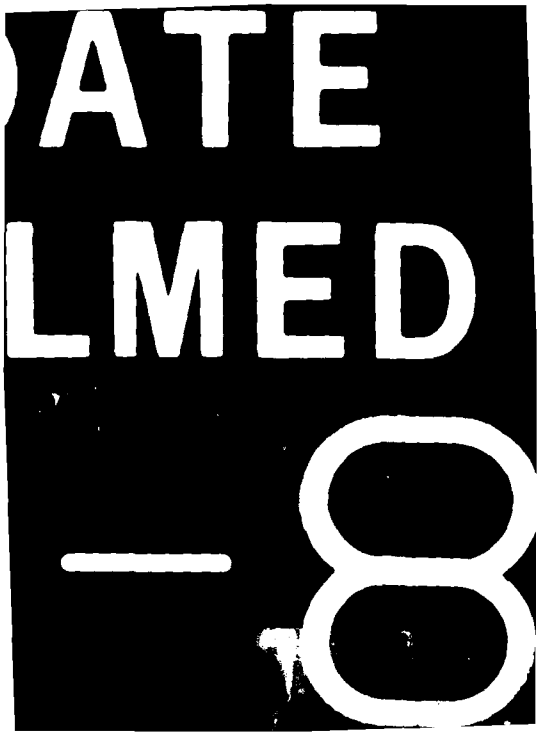
AFESC
Tyndall AFB, FL

Wright-Patterson AFB, OH 45433 (3)

Blast Noise Prediction/by Paul D. Schomer...[et al.] -- Champaign, Ill. : Construction Engineering Research Laboratory ; available from NTIS, 1981.
iv, [2] leaves ; 28 cm. -- (Technical report ; N-98)

Contents: v.1. Data bases and computational procedures. -- v.2. BNoise 2.2 computer program description and program listing.

1. Noise pollution. 2. Ordnance - noise. 3. BNOISE (computer program) I. Schomer, Paul D. II. Little, Lincoln L. III. Efland, David L. IV. Pawlowska, Violet I. V. Reobik, Steven J. VI. Series. VII. Series: Army Construction Engineering Research Laboratory. Technical report ; N-98.


ATE
LMED
8