**PONDER: A FORTRAN CODE THAT GRAPHICALLY
REPRESENTS NUCLEAR BLACKOUT EFFECTS**

Michael D. Maples

August 1, 1973

Prepared for U.S. Atomic Energy Commission under contract No. W-7405-Eng-48

**LAWRENCE
LIVERMORE
LABORATORY**

University of California/Livermore

MASTER

NOTICE

This document was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability for the accuracy, completeness, or usefulness of any information, suggestion, product or process disclosed, or represents that its use would not infringe privately owned rights.

Available in the United States of America
Available from

National Technical Information Service
U. S. Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22151

Hard Copy \$ _____; Microfiche \$0.95

NTIS Selling Price

\$4.00
\$5.45
\$7.60
\$10.60
\$13.60

TID-4500, UC-32
Mathematics and Computers

LAWRENCE LIVERMORE LABORATORY
University of California, Livermore, California, 94550

UCRL-51391

**PONDER: A FORTRAN CODE THAT GRAPHICALLY
REPRESENTS NUCLEAR BLACKOUT EFFECTS**

Michael D. Maples

MS. date: August 1, 1973

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Contents

Introduction	1
PONDER Code S/N Option	2
Spherical Mode	7
Constant Altitude Mode	10
Planar Mode	13
PONDER Code Refractive Option	15
PONDER Code CLUTTER Option	15
Acknowledgments	18
References	19
Appendices	
1. PONDER Code S/N Option (Listing)	20
2. Sample PMIV Output	50
3. PMIV Program (Listing)	56
4. DDTV Writeup	60
5. PONDER Code Refractive Option (Listing)	73
6. PONDER Code CLUTTER Option (Listing)	105

PONDER: A FORTRAN CODE THAT GRAPHICALLY REPRESENTS NUCLEAR BLACKOUT EFFECTS

Introduction

The purpose of this report is to document the PONDER Code (Pictures of Nuclear Device Effects from RANC),^{1,2} which has been used at Lawrence Livermore Laboratory for performing radar blackout analyses.³ PONDER is an executive routine in which the well-known radar blackout code RNAC (Radar Absorption Noise and Clutter)^{4,5} is a major subroutine.

"Radar blackout" refers to ionization levels that are produced by a nuclear burst and that cause sufficient absorption or refraction of a signal to make the radar system under study ineffective.

In general, PONDER outputs two types of data: computer-generated pictures of blackout, and calculations of blackout area in square kilometers. These computer-generated pictures consist of either a "projection" of blackout onto a hemispherical shell or a "projection" onto a constant-altitude surface or a "projection" onto a plane.

The hemispherical-shell concept is based on a spherical-coordinate system generally located with the radar at the origin. At detonation time, each action of interest is assumed to occur on a particular hemispherical shell. This shell system was chosen because it automatically fixes the slant range from the viewing point (which is generally the radar) to the targets. By the systematic selection of a number of target sample points, the spatial extent and degree of blackout is determined and is displayed as a two-dimensional picture. The combination of high-speed computing capabilities and efficient programming techniques makes this approach feasible. At present, pictures can be obtained for 1) signal-to-noise (S/N) or clutter-to-noise (C/N) ratios at the receiver, 2) radar distance errors caused by refraction, and 3) radar angular errors caused by refraction. These are cumulative or integrated effects. In addition, cause of the blackout (e.g., fireball ionization, x-ray ionization, γ -ray ionization, etc.) can be obtained as part of the picture. All output data available for the shell pictures are available for the constant-altitude and plane pictures.

The second section of this report discusses the three techniques for displaying S/N ratios at the receiver. The third and fourth sections, respectively, present means for displaying refractive effects and C/N calculations. The appendices to the report contain program listings of PONDER and of certain modified RANC routines.

PONDER Code S/N Option

Fig. 1. PONDER block diagram, emphasizing the relationship between PONDER and RANC IV.

Fig. 2. PONDER flow listing. PONDER subroutines are indicated by asterisks.

The PONDER package consists of an executive routine, PONDER, and eight subroutines (see Appendix 1 for a listing). The rest of the operating package is made up of the standard RANC IV routines.

Briefly, PONDER queries RANC for the S/N calculation of a certain coordinate, at a certain time, given certain conditions such as radar type, fission fraction (F. F.), and location of burst. PONDER takes RANC's responses and stores them in an n by m matrix, one row at a time. This matrix is coded and used to produce a PONDER picture, the plotting function being performed by subroutine PLOTSN. See Fig. 1 for a simplistic block diagram of PONDER and Fig. 2 for the flow of subroutine calls in PONDER.

The input subroutine of PONDER is based on the input subroutine of RANC IV (RANCIN). Figure 3 shows the unmodified RANC input-deck format, and Fig. 4 shows the modified input format for PONDER. The two formats are identical, with the exceptions of the 7th, 8th, and 9th cards.

The variable CONA in Column 72, Card 8, of RANCIN is a switch that sets PONDER into one of its three modes (spherical, constant altitude, or planar) of operation. Table 1 gives a brief description of the three modes, and Figs. 5, 6, and 7 give pictorial presentations of each mode. The input decks for the three modes differ slightly. The spherical mode will be discussed first, then the constant altitude mode, and finally the planar mode. Unless otherwise specified, the cards referred to in the following descriptions are contained in the modified RANC IV subroutine (Fig. 4).

CARD COLUMNS								
1	10	20	30	40	50	60	70 72 75-80	
1	RADAR FREQUENCY (MHz)	PEAK POWER (Mw)	RANGE RESOLUTION (m)	VELOCITY RESOLUTION (m/sec)	TRANSMITTER ANTENNA GAIN(db)	3 db BEAMWIDTH (deg)	SIDELobe LEVEL (db down)	OPTIONAL ALPHA LABEL
2	MINIMUM S/N (b)	SYS. NOISE TEMPERATURE (K)	RECEIVER NOISE BWTH (KHz)	PULSE COMPRESSION RATIO	RECEIVER ANTENNA GAIN(db)	3 db BEAMWIDTH (deg)	SIDELobe LEVEL (db down)	OPTIONAL ALPHA LABEL
3	ANTENNA SIDELOBE SPEC.	ARRAY SCAN LIMIT (deg)	ARRAY FACE ELEVATION (deg)	FIRST FACE AZIMUTH (deg)	SECOND FACE AZIMUTH (deg)	THIRD FACE AZIMUTH (deg)	FOURTH FACE AZIMUTH (deg)	
4	N LATITUDE OF ORIGIN (deg)	E LONGITUDE OF ORIGIN (deg)	DAY OR NIGHT (1 or 0)	CALCULATION OPTION	NORMAL OUTPUT OPTION	BEAM POINTING OPTION	DETAILED OUTPUT OPTION	
RADAR LOCATION								
5	X(N)	Y(N)	Z(N)	MAGNETIC DIP AT RADAR (deg)	MAGNETIC VARIATION AT RADAR (deg)			OPTIONAL ALPHA LABEL
BULB LOCATION (UP TO 50 BURSTS)								
6	X(N)	Y(N)	Z(N)	DETONATION TIME (sec)	TOTAL YIELD (MT)	FISSION FRACTION	DEVICE TYPE	OPTIONAL ALPHA LABEL
TRAJECTORY PARAMETERS				IMPACT LOCATION				
7	NOMINAL SURFACE RANGE(km)	NOMINAL LAUNCH, IMPACT ELEV. (deg)	NOMINAL LAUNCH, IMPACT VEL. (km/sec)	FINAL APPROACH AZ (deg)	SURFACE RANGE	TRUE AZIMUTH	EARTH ROTATION (0 = yes, 1 = no)	OPTIONAL ALPHA LABEL
					N LATITUDE	E LONGITUDE		
RV PARAMETERS								
8	RADAR CROSS SEC. (m ²)	BALLISTIC COEFFICIENT (lb/ft ²)	REFERENCE ALTITUDE (km)	TIME AT REFERENCE (sec)	REFERENCE ON DESCENT OR ASCENT (1 or 0)			OPTIONAL ALPHA LABEL
CALCULATION TIMES (SEC) (UP TO 100 TIMES, 7 PER CARD)								
9	t ₁	t ₂	t ₃	ETC.				

BLANK CARD AT END OF EACH CASE

Fig. 3. RANC IVA input format.

CARD COLUMNS									
1	10	20	30	40	50	60	70	72	75-80
1	RADAR FREQUENCY (MHz)	PEAK POWER (Mw)	RANGE RESOLUTION (m)	VELOCITY RESOLUTION (m/sec)	TRANSMITTER ANTENNA GAIN(db)	2.6 BEAMWIDTH (deg)	SIDELobe LEVEL (db down)	4	OPTIONAL ALPHA LABEL
2	MINIMUM S/N (b)	SYS. NOISE TEMPERATURE (K)	RECEIVER NOISE BNWTH (KHz)	PULSE COMPRESSION RATIO	RECEIVER ANTENNA GAIN(db)	3.6 BEAMWIDTH (deg)	SIDELobe LEVEL (db down)		OPTIONAL ALPHA LABEL
3	ANTENNA SIDELobe SPEC.	ARRAY SCAN LIMIT (deg)	ARRAY FACE ELEVATION (deg)	FIRST FACE AZIMUTH (deg)	SECOND FACE AZIMUTH (deg)	THIRD FACE AZIMUTH (deg)	FOURTH FACE AZIMUTH (deg)		
4	N LATITUDE OF ORIGIN (deg)	E LONGITUDE OF ORIGIN (deg)	DAY OR NIGHT (1 or 0)	CALCULATION OPTION	NORMAL OUTPUT OPTION	BEAM POINTING OPTION	DETAILED OUTPUT OPTION		
RADAR LOCATION									
5	X(N)	Y(N)	Z(N)	MAGNETIC DIP AT RADAR (deg)	MAGNETIC VARIATION AT RADAR (ecc)			N	OPTIONAL ALPHA LABEL
BURST LOCATION (UP TO 50 BURSTS)									
6	X(N)	Y(N)	Z(N)	DETONATION TIME (sec)	TOTAL YIELD (MT)	FISSION FRACTION	DEVICE TYPE	N	OPTIONAL ALPHA LABEL
7	TSEC (NT)	MAXNMX	NCRCT						
8	AZRF	RTG	RCS	SBOCRIT	DSNBOCR			CON A	OPTIONAL ALPHA LABEL
CALCULATION TIMES (SEC) (UP TO 100 TIMES, 7 PER CARD)									
9	PLDEN	RDEN	UNITHB	UNITVB	NMXB	NPXB	NHRT	NHR	

BLANK CARD AT END OF EACH CASE

Fig. 4. RANC IVA input format for PONDER code S/N option.

Fig. 5. Spherical mode.

Table 1. CONA PONDER mode switch.

CONA	Mode	Description
0	Spherical	Target shell at a constant radius from radar.
1	Constant altitude	Target shell at a constant altitude from the Earth's surface.
2	Planar	Target plane perpendicular to the Earth's surface with the radar observing the plane from any position on the Earth's surface.

Spherical target shell
(center coincident
with the Earth's)

Fig. 6. Constant altitude mode.

Fig. 7. Planar mode. Target heading $AZREF0$.

SPHERICAL MODE

In the spherical mode only the first three fields of Card 7, the first four fields plus Column 72 of the eighth card, and all seven fields of the ninth card are used. The definitions of Cards 7, 8, and 9 follow.

Card Seven

FORMAT (I1, E9.0, 6E10.0, 1X11)
NCARD card identifier (equals seven).
TSEC(NT) time that the RANC calculation is to be made, relative to the time of burst (NT).
MAXNX number of columns to the left of the center column on the printout (horizontal translation of the PONDER picture).
NCRCT print grid correction (zero disables) for blackout projections on the PONDER picture, i.e., used to try to encompass complete blackout area by extending the print area at the appropriate portions of the PONDER picture.

Card Eight

FORMAT (I1, E9.0, 6E10.0, 1X11)
NCARD card identifier (equals eight).
AZRF true heading from radar to center column of PONDER picture (in degrees).
RTG target shell radius (in kilometers): distance from the radar to the center of the spherical target shell being observed.
RCS radar cross section (meters square).
SBOCRIT radar threshold for S/N ratio (in decibels).
DSNBOCR (in dB) separation between the final PONDER picture S/N contours. This variable is used in the plotting subroutine PLOTSN.
CONA Switch that selects the mode in which PONDER will run [equals zero (0) for the spherical mode].

Card Nine

FORMAT (11, E9.0, 6E10.0, 1X11)
NCARD card identifier (equals nine).
PLDEN ratio: print grid increment/area grid increment.
RDEN ratio: search grid increment/area grid increment.
UNITHØ print grid horizontal increment (in degrees).
UNITVØ print grid vertical increment (in degrees).
NMXØ print grid: number of columns to the left of the center column that will comprise the PONDER picture.
NPXØ print grid: number of columns to the right of the center column that will comprise the PONDER picture.
NHR variable located in Columns 66 through 70. NHR is the total number of degrees of elevation from the Earth's surface to the topmost point to be observed, i.e., to be depicted in the final PONDER picture. NHR is specified in units of UNITVØ (i.e., NHR=20 means that the topmost point is 20 times the UNITVØ degrees to the topmost point of the PONDER picture).
NHR1 variable located in Columns 61 through 65. NHR1 is the total number of degrees to be covered in the PONDER picture from the topmost point (NHR) toward the bottom of the picture. NHR1 is also given in terms of UNITVØ and therefore designates the number of print rows that will compose the PONDER picture. If NHR1 is set to zero or blank, then the program automatically sets NHR1 equal to NHR, causing the NHR number (or print rows) to be depicted in the final PONDER picture.
Note: UNITHØ = 3/5 UNITVØ, to correct for printer distortion caused by using printers for the PONDER picture output.
UNITHØ = 1/2 UNITVØ, to correct for DD80 and/or FR80 (display devices used at Lawrence Livermore Laboratory) distortion when the DD80 or FR80 is used for PONDER picture output.

To execute PONDER, the first set of nine cards must be followed by a blank card or a card that has only a zero in its Column 1.

Table 2. Input variable settings for PONDER's automatic mode.

UNITHØ	= zero for automatic run with vertical sweep down to zero degrees elevation. = number less than zero (negative number) for automatic run with vertical sweep stopped just below the fireball.
UNITVØ	= zero for automatic run, using debris in RANC calculations. = negative number for automatic run, not using debris in RANC calculations.
NPXØ	= columns 51 through 60. This variable is replaced by three variables: NM, NP, and NH.
NM	= columns 51-54. This variable represents the number of print columns added to the left of the selected search grid.
NP	= columns 55-57. This variable represents the number of print grid columns added to the right of the selected search grid.
NH	= columns 58-60. This variable represents the number of print grid rows added to the selected search grid. NH/2 are added to the top of the PONDER picture, and the other half are added to the bottom of the PONDER picture.

PONDER ceases execution only when it reads the end of a file mark on its input file; therefore, multiple runs can be made by placing the appropriately modified cards after the zero card, followed by another zero card for as many runs as desired (see Fig. 8 for sample input deck).

The number nine card can also be used to set PONDER to the automatic search mode. In this mode the program automatically searches out the area of blackout along the target shell, sets UNITHØ and UNITVØ to appropriate values, and produces a PONDER picture that encompasses the complete blackout area (see Table 2 for the input variable settings needed to operate in the automatic mode).

Overlay

Of the three modes of PONDER, only the spherical mode has an overlay written for it. The overlay routine PMIV simply takes the information from the PONDER run (files PLOTSN, DOUT, and INPUT) and produces a grid for its plots (see Appendices 2 and 3). The DD80 is used to plot the PMIV overlay.

Two numbers are placed at the bottom of the grids produced, one number at the bottom left (x) and one at the bottom right (y). The x refers to the elevation line at the lowest grid line, and y refers to the increment in degrees between the grid lines.

1	10	20	30	40	50	60	70	72
1								
2								
3								
4								
5								
6								
7								
8								
9								
0								
6								
0								
7								
8								
9								
0								
1								
2								
3								
0								
7								
8								
9								
0								

END OF FILE

Fig. 8. Sample of input deck.

Note: for the PMIV routine to work correctly,

1. the ratio $UNITH\theta/UNITV\theta$ must equal $1/2$,
2. both $UNITH\theta$ and $UNITV\theta$ must be in powers of 2, and
3. the dimensions of the PONDER picture to be produced must not exceed 40 lines (rows) in the vertical direction and 85 lines (columns) in the horizontal direction.

Adherence to these three rules will cause PMIV to produce a DD80 file with the designators DX as its first two letters. To display this file and/or to produce a hard copy, the utility routine DDTV (unique to the LLL computer system) may be used. See Appendix 2 for an example of a PMIV output file as would be displayed on the TMDS. Refer to Table 3 for a short set of rules on the use of the routine DDTV, and to Appendix 4 for a writeup on DDTV.

CONSTANT ALTITUDE MODE

When $CONA = 1$, PONDER executes in its constant altitude mode. This mode uses a spherical shell that encircles the Earth at the desired altitude. On this shell, PONDER will map the desired blackout pictures. To allow this mode to interface with the present RANC IV, a few changes are made to the coordinate referencing system. First, the coordinates of the origin and the radar are made

Table 3. Instructions to DDTV.

Instruction	Action
Linefeed	Display the next frame. (The first frame is either the ID frame or blank; DDTV identifies a blank frame by displaying the word BLANK.)
END	End DDTV and release the monitor(s).
S <u>n</u>	Skip <u>n</u> frames and display the next frame.
S	Skip. DDTV then asks for <u>n</u> by typing TYPE NO. OF FRAMES TO BE SKIPPED.
- <u>n</u>	Back up <u>n</u> frames. <u>n</u> must be space-delimited from the minus sign.
GIVE	Give the DD80 file to the system to be printed. The <u>filename</u> must be in the proper format. (See Restrictions.)
. (period)	Send the same frame again (useful when a TMDS error results on the first try).
ERASE or E	Erase the frame. However, the frame still remains in the buffer and may be sent again by typing a period.
ADD	Superimpose the next frame to be displayed on the frame currently displayed. You may superimpose as many frames as you like, but you must type ADD each time.
P	Change screen polarity (reverses the screen background color of this frame and the following frames).
STATUS	Display the TMDS status frame. To return to the DD80 file frame, type STATUS again.
LINK <u>n</u>	Simultaneously displays your frames on another TMDS monitor: monitor <u>n</u> .
UNLINK <u>n</u>	Erase and release monitor <u>n</u> .
D <u>name</u>	The name of a DD80 file beginning with D. This instr allows you to begin scanning a second DD80 file without having scanned the entire first DD80 file.
SRD	Display an SRD reminder on this frame and all following frames. This does not flag the file or cause any label to appear on the hardcopy. The SRD labels for the hardcopy are generated by the sub-routine call CRTID in the DD80 program.
CONFIRM <u>nnnn</u>	<u>nnnn</u> is the four-digit confirm number displayed on the monitor you reserved. The confirm feature is intended for those users operating at an access level greater than P. If you do not confirm within one min, your monitor will be released.
HELP	Display a "help" message.

to coincide, and they should be located at 0 degrees latitude and 90 degrees longitude. Second, the heading of the center column of the PONDER picture must be a longitudinal line (this will be explained in more detail later); i.e., AZREF = 0 degrees.

All input data are given relative to the new set of coordinates. The magnetic dip in the input deck is that which occurred at the original (actual) unshifted radar position. The magnetic variation is derived by taking the angle between the center-column azimuth and the magnetic variation at the actual radar position. With the exceptions of the above stipulations, the input deck for this mode is the same as that for the spherical mode.

Fig. 9. Ponder surface for the constant altitude mode.

central column passes from the radar to the burst point (BP). This means that true north (at the radar) will be in the direction of the line running from the radar to the burst point. To preserve the character of the effects of the Earth's magnetic field on the RANC calculations, a new magnetic variation must be calculated. In this case it will be $VARR + \alpha$ (see Fig. 10). All that remains is to locate the radar. To avoid confusion, the radar can be arbitrarily placed at latitude zero, and the RANC origin placed at the radar location. To summarize, the following would be the Ponder/RANC input:

RANC origin: latitude ϕ , longitude 90 degrees (for instance)

Radar location: on top of RANC origin

Dip at radar: same as in physical problem

Magnetic variation: $VARR - \alpha$

Heading to burst point: ϕ degrees

Center column location/longitude: $AZRF = 90$ (because of RANC origin and radar location).

Fig. 10. Sample problem in the Ponder constant altitude mode.

Implementation of the constant altitude mode can best be explained by an example. The center column of the surface (i.e., the concentric sphere, see Fig. 9) must coincide with one of the Earth's longitude lines (this requirement is due to the use of unaltered RANC IV geometry routines). The particular longitude line that the center column is aligned with is specified by the variable $AZRF$ (Card 8, Field 1).

Consider the physical problem presented in Fig. 10. Assume that the grid

Implicit in this shift of the entire physical problem from its actual location to the Earth's equator and the magnetic variation is the assumption that all RANC calculations are independent of the Earth's rotation.

Another example might be useful. Instead of running the center column from the radar to the burst point, run it from the radar to the magnetic north pole. Then the Ponder/RANC inputs would be

RANC origin: latitude 0 degrees, longitude 90 degrees
 Radar location: on top of origin
 Dip at radar: same as in physical problem
 Magnetic variation: 0 degrees (true north and magnetic north will be the same)
 Heading to burst point: $-VARR + \alpha$
 Center column longitude: $AZRF - 90$.

The actual construction of the grid is now possible. The "vertical" grid increments (UNITV0) are increments in latitude. The "horizontal" grid increments (UNITH0) are increments in longitude. This is not strictly true and will be in error for a grid that covers a large portion of the Earth's surface. (These increments are actually the same as for the regular PONDER spherical surface having the center of the sphere as the vantage point.) NHR is counted from the Earth's equator northward. NMX0 and NPX0 are counted from the grid center column, left and right respectively, as seen from inside the sphere.

PLANAR MODE

When CONA = 2, PONDER goes into the PLANAR mode (see Fig. 7). The input package is the same as for the spherical mode. A brief description of the seventh, eighth, and ninth cards used in the planar mode follows.

Card Seven

FORMAT	(I1, E9.0, 6E10.0, 1X1)
NCARD	card identifier (equals seven).
TSEC(NT)	time that the RANC calculation is to be made, relative to the time of burst (NT).
MAXNMX	number of columns left of the center column on the printout (horizontal translation of the PONDER picture).
NCRCT	print grid correction (zero disables) for blackout projections.

Card Eight

FORMAT	(I1, E9.0, 6E10.0, 1X1)
NCARD	card identifier (equals eight).
AZREF	true heading of target plane (in degrees).
RTG	displaces grid away from center column.
RCS	radar cross section (meters square).
SBOCRIT	radar threshold for detection of targets (in decibels).
DSNBOCR	separation between contour lines in the PONDER pictures (in decibels).
CONA	switch that selects the mode in which PONDER will run (equals 2).

Card Nine

FORMAT (I1, E9.0, 6E10.0, 1X11)
NCARD card identifier (equals seven).
PLDEN ratio: print grid increments/area grid increments.
RDEN ratio: search grid increments/area grid increments.
UNITHØ print grid increment in horizontal direction (in kilometers).
UNITVØ print grid increment in vertical direction (in kilometers).
NMXØ number of columns to the left of the center column, where the center column of the plane is located by the normal line at the point of tangency.
NPXØ number of columns to the right of the center column.
NHR variable located in Columns 66 through 70. NHR is the total number of kilometers from the Earth's surface to the topmost point to be observed. NHR is specified in units of UNITVØ (i.e., NHR = 20 means that the topmost point is 20 times the UNITVØ kilometers to the topmost point of the PONDER picture).
NHR1 variable located in Columns 61-65. NHR1 is the total number of kilometers to be covered in the PONDER picture, from the topmost point (NHR) to the bottom of the picture. NHR1 is also given in terms of UNITVØ and therefore designates the number of print rows that will compose the PONDER picture. If NHR1 is set to zero or blank, the program automatically sets NHR1 equal to NHR, causing NHR number of print rows to be depicted in the final PONDER picture.
Note: UNITHØ = 3/5 UNITVØ, to correct for the picture distortion caused by using printers for the PONDER picture output.

Note that the origin, in Fig. 7, is the point of tangency to the Earth for the planar mode. The base line in the planar mode is defined as that line tangent to the origin. Thus, for large target distances, the position of the target altitude must be calculated by taking the difference between the base-line altitude at the said range and the target altitude at said range.

In the planar mode of operation, some difficulty arises as to which direction NPXØ pertains. The following is a summary of procedures that specify NPXØ's direction.

1. When the radar and origin (tangency point) are the same, or when the heading of the plane (relative to the origin) is 180 degrees different from the heading of the radar from the origin, then NPXØ is in the direction of AZREF (plane heading) away from the radar (AROSP).
2. When the heading of the radar and the plane (both relative to the origin) are the same, then NPXØ is in the direction opposite to AZREF (away from the radar).
3. When the condition in neither 1 or 2 (above) is satisfied, i.e., the headings of the radar and the plane are neither the same nor 180 degrees out of phase and the radar's location and that of the origin do not coincide, then the following conditions hold.

- a. With the radar located in a southern direction from the origin and
 1. the angle of the plane heading larger than the angle of the heading of the radar, then $NPX\theta$ is in the direction opposite to that of the plane heading;
 2. the angle of the plane heading less than that of the angle of the heading of the radar, then $NPX\theta$ is in the direction of the plane heading.
- b. With the radar located in a northern direction from the origin and
 1. the angle of the plane heading larger than the angle of the heading of the radar, then $NPX\theta$ is in the direction of the heading of the plane;
 2. the angle of the heading of the plane less than the angle of the heading of the radar, then $NPX\theta$ is in the direction opposite to the heading of the plane.

PONDER Code Refractive Option

This option operates in the same manner as the S/N option, except that 1) it has only a spherical mode, and 2) it provides for refractive calculations and plots along with S/N calculations and plots. Figure 11 gives the input variables of this option, and the variations from the S/N option are present here. All other information on the use of this option can be obtained from the description of the S/N option. Appendix 5 gives a listing of the refractive option.

PONDER Code CLUTTER Option

The plots performed in the Clutter option are

PLOTSN - signal/noise
 PLOTCL - clutter/noise
 PLOTDC - signal/(noise + clutter)
 PLOTDF - signal/clutter.

All of the above are plotted by using the same grid matrix and overlay setup. The Clutter option runs in the same three modes as the S/N model and uses the same overlay techniques.

The Clutter input format (Fig. 12) is essentially the same as that in the S/N option (Fig. 4). The main differences on the input formats occur on the seventh and eighth cards. That is,

1	10	20	30	40	50	60	70	72	75-80
---	----	----	----	----	----	----	----	----	-------

1	RADAR FREQUENCY (MHz)	PEAK POWER (Mw)	RANGE RESOLUTION (m)	VELOCITY RESOLUTION (m/sec)	TRANSMITTER ANTENNA GAIN(db)	3 db BEAMWIDTH (deg)	SIDELobe LEVEL (db down)	OPTIONAL ALPHA LABEL
---	-----------------------------	-----------------------	----------------------------	-----------------------------------	------------------------------------	----------------------------	--------------------------------	----------------------------

2	MINIMUM S/N (b)	SYS. NOISE TEMPERATURE (K)	RECEIVER NOISE BNWTH (KHz)	PULSE COMPRESSION RATIO	RECEIVER ANTENNA GAIN(db)	3 db BEAMWIDTH (deg)	SIDELobe LEVEL (db down)	OPTIONAL ALPHA LABEL
---	-----------------------	----------------------------------	----------------------------------	-------------------------------	---------------------------------	----------------------------	--------------------------------	----------------------------

3	ANTENNA SIDELOBE SPEC.	ARRAY SCAN LIMIT (deg)	ARRAY FACE ELEVATION (deg)	FIRST FACE AZIMUTH (deg)	SECOND FACE AZIMUTH (deg)	THIRD FACE AZIMUTH (deg)	FOURTH FACE AZIMUTH (deg)	
---	------------------------------	------------------------------	----------------------------------	--------------------------------	---------------------------------	--------------------------------	---------------------------------	--

4	N LATITUDE OF ORIGIN (deg)	E LONGITUDE OF ORIGIN (deg)	DAY OR NIGHT (1 or 0)	CALCULATION OPTION	NORMAL OUTPUT OPTION	BEAM POINTING OPTION	DETAILED OUTPUT OPTION	
---	----------------------------------	-----------------------------------	--------------------------	-----------------------	----------------------------	----------------------------	------------------------------	--

RADAR LOCATION

5	X(N)	Y(N)	Z(N)	MAGNETIC DIP AT RADAR (deg)	MAGNETIC VARIATION AT RADAR (deg)			N OPTIONAL ALPHA LABEL
---	------	------	------	-----------------------------------	---	--	--	---------------------------------

BURST LOCATION (UP TO 50 BURSTS)

6	X(N)	Y(N)	Z(N)	DETONATION TIME (sec)	TOTAL YIELD (MT)	FISSION FRACTION	DEVICE TYPE	N OPTIONAL ALPHA LABEL
---	------	------	------	-----------------------------	------------------------	---------------------	----------------	---------------------------------

7	TSEC (NT)	MAXNMX	NCRCT					1 0 OPTIONAL ALPHA LABEL
---	-----------	--------	-------	--	--	--	--	--------------------------------------

8	AZRF	RTG	RCS	SBOCRIT	DSNBOCR			0 OPTIONAL ALPHA LABEL
---	------	-----	-----	---------	---------	--	--	---------------------------------

CALCULATION TIMES (SEC) (UP TO 100 TIMES, 7 PER CARD)

9	PLDEN	RDEN	UNITH	UNITV	NMX	NPX	NHRT	NHR	
---	-------	------	-------	-------	-----	-----	------	-----	--

BLANK CARD AT END OF EACH CASE

Fig. 11. RANC IVA input format for PONDER code refractive option.

CARD COLUMNS								
1	10	20	30	40	50	60	70	75-80
1	RADAR FREQUENCY (MHz)	PEAK POWER (Mw)	RANGE RESOLUTION (m)	VELOCITY RESOLUTION (m/sec)	TRANSMITTER ANTENNA GAIN(db)	3 db BEAMWIDTH (deg)	SIDELobe LEVEL (db down)	OPTIONAL ALPHA LABEL
2	MINIMUM S/N (b)	SYS. NOISE TEMPERATURE (K)	RECEIVER NOISE BWIDTH (KHz)	PULSE COMPRESSION RATIO	RECEIVER ANTENNA GAIN(db)	3 db BEAMWIDTH (deg)	SIDELobe LEVEL (db down)	OPTIONAL ALPHA LABEL
3	ANTENNA SIDELOBE SPEC.	ARRAY SCAN LIMIT (deg)	ARRAY FACE ELEVATION (deg)	FIRST FACE AZIMUTH (deg)	SECOND FACE AZIMUTH (deg)	THIRD FACE AZIMUTH (deg)	FOURTH FACE AZIMUTH (deg)	
4	N LATITUDE OF ORIGIN (deg)	E LONGITUDE OF ORIGIN (deg)	DAY OR NIGHT (1 or 0)	CALCULATION OPTION	NORMAL OUTPUT OPTION	BEAM POINTING OPTION	DETAILED OUTPUT OPTION	
RADAR LOCATION								
5	X(N)	Y(N)	Z(N)	MAGNETIC DIP AT RADAR (deg)	MAGNETIC VARIATION AT RADAR (deg)			N OPTIONAL ALPHA LABEL
BURST LOCATION (UP TO 50 BURSTS)								
6	X(N)	Y(N)	Z(N)	DETONATION TIME (sec)	TOTAL YIELD (MT)	FISSION FRACTION	DEVICE TYPE	N OPTIONAL ALPHA LABEL
7	TSEC (NT)	MAXNMX	NCRCT	AZRF	RTG	RCS	VTR	OPTIONAL ALPHA LABEL
8	SBOCRIT	DSNBOCR	THCLTR	DTHCLTR	THDIFF	DTHDIFF		C O N A OPTIONAL ALPHA LABEL
9	PLDEN	RDEN	UNITHØ	UNITVØ	NMXØ	NPXØ	NHRI	NHR

BLANK CARD AT END OF EACH CASE

Fig. 12. RANC IVA input format for PONDER code clutter option.

Card Seven

FORMAT (I1, E9.0, 6E10.0, 1X11)
NCARD card identifier (equals seven).
TSEC(NT) time that the RANC calculation is to be made, relative to the time of burst (NT).
MAXMX number of columns to the left of the center column on the printout (horizontal translation of the PONDER picture).
NCRCT print grid correction (zero disables) for blackout projections on the PONDER picture, i.e., used to try to encompass complete blackout area by extending the print area at the appropriate portions of the PONDER picture.
AZRF true heading from radar to center column of PONDER picture (in degrees).
RTG target shell radius (in kilometers). Distance from the radar to the center of the spherical target shell being observed.
RCS radar cross section (meters square).
UTR velocity of the reentry vehicles toward their targets; needed to distinguish debris from reentry vehicles.

Card Eight

FORMAT (I1, E9.0, 6E10.0, 1X11)
NCARD card identifier (equals eight).
SBOCRIT maximum threshold of signal/noise and for signal/(noise plus clutter) (in decibels) for radar recognition
DSNBOCR (in decibels) separation between the final PONDER picture S/N and S/(N+C) contours.
THCLTR minimum threshold for clutter (in decibels) for radar recognition.
DTHCLTR separation between the final PONDER picture clutter contours.
THDIFF maximum threshold for signal/noise minus clutter/noise for radar recognition.

Acknowledgments

The author wishes to acknowledge the invaluable assistance of R. N. Castleton, who programmed the PONDER code, in the collection and composition of the information that exists within this document. The author also greatly appreciated the constant reviews, critiques, and suggestions of D. R. Dunn.

References

1. L. L. Cleland and R. N. Castleton, Spatial Pictures of Radar and Nuclear Weapon Interactions, Lawrence Livermore Laboratory, Rept. UCRL-50913 (1970) (title U, report SRD).
2. L. L. Cleland and R. N. Castleton, PONDER: Pictures Of Nuclear Device Effects from RANC, Lawrence Livermore Laboratory, Rept. UCRL-73043 (1971) (title U, report SRD).
3. L. L. Cleland, R. N. Castleton, and D. R. Dunn, PONDER Data: Fission Fraction Effect and RANC III $\frac{1}{2}$ -IV Comparisons, Lawrence Livermore Laboratory, Rept. UCRL-51384 (1973) (title U, report SRD).
4. R. L. Bogusch, S. R. Boyle, and R. A. Reynolds, Radar Absorption, Noise, and Clutter in a Nuclear Environment: A FORTRAN Code, General Electric Co., TEMPO, Santa Barbara, California, Rept. RM64 TMP-46 (1964) (title U, report SRD).
5. R. L. Bogusch, R. H. Christian, P. G. Fischer, R. J. Jordano, W. S. Knapp, W. McNamara, et al. (sic), RANC IV: Computer Simulation of Radar Propagation in a Nuclear Environment, General Electric Co., TEMPO, Santa Barbara, California, Rept. (July 1970)(title U; Vol. I, C-FRD; Vol. II, U; Vol. III, SRD).

Appendix 1

PONDER Code Signal/Noise Option (Listing)

Note that the subroutines in Appendices 1, 5, and 7 have the same names. Subtle differences may exist in some of the routines; therefore, these routines should not be interchanged but should be used only with the PONDER options indicated.

PONDER

CONA = mode switch
GLATØ = latitude of origin
GLONØ = longitude of origin
HHR = altitude of origin
XRØ = (east) x coordinate of radar with respect to origin
YRØ = (north) y coordinate of radar with respect to origin
ZRØ = (vert) z coordinate of radar with respect to origin
ARØ = azimuth coordinate of radar with respect to origin
AROSP = radar heading
ITOT = total number of targets used to produce PONDER picture

Listing

```
PROGRAM PONDER(INPUT,TAPE5=INPUT,OUTPUT,TAPE6=OUTPUT,DDUT,  
1TAPE7=DDUT,PLOTSN,TAPE8=PLOTSN)  
CALL CHANGE(7H+PONDER)  
ILEN=10000  
CALL CREATE(6HOUTPUT,ILEN,M1WD)  
ILEN=10000  
CALL CREATE(4HDDUT,ILEN,M1WD)  
ILEN=40000  
CALL CREATE(6HPLOTSN,ILEN,M1WD)  
COMMON /EXEC/CLTR(50),SRCL(50),DSRCL(50),VC(50),DVC(50),  
1 SIGMAC(50),ETA(50),FBNE(50),FBDB(50),FBRHQ(50),  
2 KEYC(50),TANT(50),AZFB(50),ELFB(50),DBTOFB(50),  
3 CBS(50),OMEGAB(50),KEYN(50),ACLTR,SRAC,DSRAC,VAC,  
4 CVAC,ASIGMA,ETAA,ENEA,ADBPXM,KEYA,AZAC,ELAC,DBAC,  
5 TASUM  
COMMON /BURST/NB,W(50),TSX(50),FF(50),HB(50),XBR(50),YBR(50),  
1 ZBR(50),TB(50),CUSCLB(50),RZERO(50),RHZERO(50),  
2 RUZERO(50),LHVB(50),RHOB(50),HSB(50),PKTERM(50),  
3 RLK(50),RTK(50),DISTK(50),XKR(50),YKR(50),ZKR(50),  
4 TILTK(50),WUV(3,50),HR,COSCLR,SINCLR,KALCH,KALCN,  
5 KALLR,DAY,CIRAX,TEM150,NSW1,NSW2,NSW3,NSW4,NSW5,NSW6  
COMMON/REGION/NF,INOXF(50),FLABL(50),KINDF(50),RFB(50),AFB(50),  
1 HF(50),XFR(50),YFR(50),ZFR(50),RLF(50),RTF(50),  
2 TILTF(50),HMINF(50),HMAXF(50),TF0(50),TF(50),  
3 RHOF(50),HSNF(50),HUGT(50),RDOT(50)  
COMMON /RADAR/FREQ,POWER,DELTAR,DELTAV,TAG,TBW,TSL,SNMDB,TSYS,BWN,  
1 PCOMP,RAG,RBW,RSL,KSPEC,SCAN,EFACE,AFACEM(4),NFACE  
COMMON/TARGET/RCS,STR,ATR,ETR,TSEC(1),SIGNAL,SNCTRC,OBTA,AER,EER,  
1RER,DISP,KEYT,KOUT,LFOUT  
COMMON/RNCIN/PLDEN,RDEN,NMXR,NPXR,NHR1,NHR,UNITH,UNITV,UNITHO,  
1UNITVQ,UNITHR,UNITVR,AZREF,AZRF,RTG,MAXNMX,NCRCT,VAKR,SBOCRIT,  
2DSNBDCR,AREAO  
COMMON/CONST/RE,PI,RADIAN  
COMMON/RNC/ICONF  
COMMON/CONAL/CONA,GLATO,GLONO,HHR  
COMMON/CMSPRAY/COSCL0,SINCL0,XRO,YRO,ZRO,ARO,AROSP,ISPF  
COMMON/TARCNT/ITOT  
DIMENSION SNPRES(500),SNPREV(500),KYT(500),S(150)  
DDOUT=7
```

```

NPLTSN=8
AREAO=1.
ICONF=1
GO TO 3
1 ICONF=2
3 CALL RANC
ICONF=3
ITOT=0
IAREA=0
AREA1=0.
AREAO=0.
ICONA=CONA+1.
MXTANT=0.
MXDBTF=0.
MXDBS=0.
MXTF=0.
MXFBR=0.
MXFBE=0.
MXFBOB=0.
SBOCR=SBOCRIT+DSNBOCR
IF(CONA.EQ.1.) GO TO 20
AZREF=AZRF-VARR
GO TO 21
20 CONTINUE
AZREF=AZRF
21 CONTINUE
CALL GRID(AXNHR)
WRITE(INDOUT,3003)UNITHR,UNITVR,UNITHO,UNITVO,UNITH,UNITV,NMXR,NPXR
1,NHRI,NHR,MAXNMX,CONA,AZREF,SBOCRIT,DSNBOCR
IPLDEN=PLDEN
IRDEN=RDEN
IRP=IRDEN*IPLDEN
NMXIRD=NMXR*IRDEN
NPXIRD=NPXR*IRDEN
MAXNRV=(NMXR+NPXR)*IRP+1
AXNHR=INTF(85./(2.*RDEN))-1.
NHRMO=(AXNHR-NHRI)*RDEN
IF(NHRMO.LE.0) GO TO 8401
S(1)=1H
DO 8400 I=1,NHRMO
WRITE(NPLTSN,3001) S(1)
8400 CONTINUE
8401 CONTINUE
DO 8100 I=1,MAXNRV
SNPRES(I)=0.
8100 SNPREV(I)=0.
IF(CONA.NE.2.) GO TO 8101
IF((SQRT(XRO**2+YRO**2+ZRO**2).LT..1).OR.(ABSF(ARD+180.-AZRF).LT.
1.1).OR.(ABSF(ARO-180.-AZRF).LT..1)) GO TO 8102
IF(ABSF(ARD-AZRF).LT..1) GO TO 8103
IF(YRO)8104,8105,8106
8104 CONTINUE
AROSP=ATANF(XRO/YRO)/RADIAN
ISPF=3
GO TO 8101
8106 CONTINUE
AROSP=ATANF(XRO/YRO)/RADIAN
8107 CONTINUE
ISPF=4
GO TO 8101
8105 CONTINUE
IF(XRO)8108,8101,8110
8108 CONTINUE

```

```

      AROSP=-90.
      GO TO 8107
8110  CONTINUE
      AROSP=90.
      GO TO 8107
8102  CONTINUE
      ISPF=1
      GO TO 8101
8103  CONTINUE
      ISPF=2
8101  CONTINUE
      IPLOT=2
      IRUN=2
      NPRES=0
      NLPRES=0
      NRPRES=0
      ELREF=NHR*UNITVR
      DO 6000 I1=1,NHR1
      GO TO(7001,7002),IRUN
7002  GO TO(7701,7702),IPLOT
7702  CONTINUE
      IPLOT=1
      IRUN=1
      NMX=NMXR
      NPX=NPXR
      UNH=UNITHR
7001  CONTINUE
      ELREF=ELREF-UNITVR
      IO=1
      GO TO 7003
7701  CONTINUE
      ELR=ELREF
      UNH=UNITH
      IO=IRP
7003  DO 6003 I2=1,IO
      GO TO(7006,7005),IRUN
7005  ELREF=ELR+(IRP-I2)*UNITV
      ILNCT=0
      IRNCT=0
7006  CONTINUE
      IL=0
      IR=0
      ILEFT=1
      NR=NMX+1
      NRV=NMX+NPX+1
      NRVPREV=0
7703  CONTINUE
      DO 309 ITRGT=1,NRV
      GO TO(5001,5002,5003),ILEFT
5001  CONTINUE
      NR=NR-1
      GO TO 5004
5002  CONTINUE
      NR=-NPX
      ILEFT=3
      GO TO 5004
5003  CONTINUE
      NR=NR+1
5004  CONTINUE
      CALL SPRAY(RTG,UNH,NR,ELREF,AZREF,VARR,ATR,STR,ETR)

```

```

CALL RANC
DO 8800 INB=1,NF
MXTANT=MAX1F(MXTANT,TANT(INB))
MXDBTF=MAX1F(MXDBTF,DBTOFB(INB))
MXDBS=MAX1F(MXCBS,DBS(INB))
MXTF=MAX1F(MXTF,TF(INB))
MXFBR=MAX1F(MXFBR,FBRHQ(INB))
MXFBE=MAX1F(MXFBE,FONE(INB))
MXFBDH=MAX1F(MXFBCB,FBDH(INB))
8800 CONTINUE
GO TO(5005,308),IRUN
5005 CONTINUE
IF((SIGNAL-SNBDCR).GE.0.) GO TO 309
5006 GO TO(5007,5008,5008),ILEFT
5007 ILEFT=2
IL=ITRGT
GO TO 309
5008 ITRGT=NRV
IR=NPX+NR+1
GO TO 309
308 CONTINUE
SNPRES(ITRGT+NRVPREV)=SIGNAL
KYT(ITRGT+NRVPREV)=KEYT
309 CONTINUE
GO TO(7007,7008),IRUN
7007 CONTINUE
IF((IL.GT.0).AND.(IR.EQ.0)) IR=NRV+1-IL
CALL FIT(NPRES,NLPRES,NRPRES,IRUN,NMXR,NPXR,NMXO,NPXO,NMX,NPX,
IPLDEN,IRDEN,MAXNMX,IL,IR)
GO TO 6003
7008 CONTINUE
IF(NCRCT.EQ.0) GO TO 9003
IF(IRNCT.EQ.1) GO TO 7801
IF(ILNCT.EQ.1) GO TO 7802
ILCT=0
IF((SNPRES(1)-SNBDCR).LT.0.) ILCT=1
IRCT=0
IF((SNPRES(NRV)-SNBDCR).LT.0.) IRCT=1
IF((ILCT.EQ.0).AND.(IRCT.EQ.0)) GO TO 9003
CALL CRCT(NMXIRC,NPXIRD,NMXO,NPXO,NMX,NPX,NPXRI,IPLDEN,NCRCT,ILCT,
IRCT,NRVL,NRVR,NRL,NRR)
ILNCT=1
IF(NRVL.GT.0) GO TO 7803
7804 CONTINUE
IRNCT=1
IF(NRVR.GT.0) GO TO 7805
7806 CONTINUE
ILCT=0
IF((SNPRES(1)-SNBDCR).LT.0.) ILCT=1
IRCT=0
IF((SNPRES(NRV)-SNBDCR).LT.0.) IRCT=1
CALL ANALYZE(NRVL,NRVR,ILCT,IRCT,ELREF)
GO TO 9003
7801 CONTINUE
NRV=NMX+NPX+1
GO TO 7806
7802 CONTINUE
NRV=NMX+NPX+1
GO TO 7804
7803 CONTINUE
NRVPREV=NRV
NRV=NRVL
NR=NRL

```

```

DO 7900 I=1,NRVPREV
SNPRES(NRVPREV+NRV+1-I)=SNPRES(NRVPREV+1-I)
KYT(NRVPREV+NRV+1-I)=KYT(NRVPREV+1-I)
7900 CONTINUE
NRVPREV=0
GO TO 7703
7805 CONTINUE
NRVPREV=NRV
NRV=NRVR
NR=NRN
NPX=NPXRI
GO TO 7703
9003 I3=MOD(I2,IPLDEN)
IF(I3)8001,8000,8001
8000 CALL PLOTSN(SNPRES,KYT,SNBOCR,DSNBOCR,NPXO,NMXO,0,0,MAXNMX,IPLDEN)
IPLDT=2
8001 CALL AREA(SNPRES,SNPREV,SBOCRIT,UNITH,UNITV,NMX,NPX,NRV,MAXNRV,
11AREA,AREAI,AREAD,ICONA)
6003 CONTINUE
IF((IRUN.EQ.2).AND.(IPLDT.EQ.1)) GO TO 7701
6000 CONTINUE
JO=MAXNMX+1
DO 9100 J=1,JO
9100 S(J)=1H
J1=JO+1
S(J1)=1H+
WRITE(NPLTSN,3001) (S(J),J=1,J1)
AREAAV=(AREAI+AREAD)/2.
AA=AREAAV*(STR**2)
IF(AREAI)3501,3502,3501
3502 APPERR=100.
GO TO 3507
3501 APPERR=(AREAD/AREAAV-1.)*100.
3507 AI=AREAI*(STR**2)
AD=AREAD*(STR**2)
GCDH=STR*UNITH0*RADIAN
WRITE(NDOUT,3005) TSEC(1)
DO 3503 I=1,NB
IF(TB(I)-TSEC(1))3504,3504,3503
3504 CONTINUE
CALL COORDX(0.,2,XBR(I),YBR(I),ZBR(I),X1,ABR,X2,X3,X4,X5,
1SBR,EBR)
WRITE(NDOUT,3006)TB(1),W(1),FF(1),TSX(1),HB(1),SBR,EBR,ABR
CALL COORDX(0.,2,XFR(I),YFR(I),ZFR(I),X1,AFR,X2,X3,X4,X5,SFR,EFR)
WRITE(NDOUT,3106)SFR,EFR,AFR
3503 CONTINUE
WRITE(NDOUT,3007) STR,VTR,SBOCRIT,RCS,FREQ
WRITE(NDOUT,3008) AA,AREAAV,APPERR
TANT(1)=MXTANT
DBTOFB(1)=MXDBTF
DBS(1)=MXDBS
TF(1)=MXTF
FBRHO(1)=MXFBR
FBNE(1)=MXFBE
FBDB(1)=MXFBDB
CALL RNCOT
WRITE(NDOUT,3009)
WRITE(NPLTSN,3009)
GO TO 1
3001 FORMAT(120A1)
3003 FORMAT(3(/2F20.10),/5I10,F2.0,/F20.3,/2F20.10)
3005 FORMAT(1H1,F51.10)
3006 FORMAT(F52.3,/F57.5,/F60.3,/F56.3,/F57.3,/F57.3,/F52.3,/F55.3)

```

```
3007 FORMAT(F57.3,/F53.3,/F57.3,/F46.3,/F56.3)
3008 FORMAT(F54.3,/F54.6,/F41.3)
3009 FORMAT(3HEND)
3106 FORMAT(3(/F15.7))
      END
```

RANC

This is an excerpt from the RANC routine, showing the modifications needed to run PONDER.

Listing

```
SUBROUTINE RANC
COMMON /EXEC/CLTR(50),SRCL(50),DSRCL(50),VC(50),DVC(50),
1 SIGMAC(50),ETA(50),FBNE(50),F8DB(50),F8RHD(50),
2 KEYC(50),TANT(50),AZFB(50),ELFB(50),DBTOFB(50),
3 DBS(50),OMEGAB(50),KEYN(50),ACLTR,SRAC,DSRAC,VAC,
4 DVAC,ASIGMA,ETAA,ENEA,ADBPKM,KEYA,AZAC,ELAC,DBAC,
5 TASUM
COMMON /RADAR/FREQ,POWER,DELTAR,DELTAV,TAG,TBW,TSL,SNMOB,TSYS,BWN,
1 PCOMP,RAG,RBW,RSI,KSPEC,SCAN,EFACE,AFACEM(4),NFACE
COMMON/TARGET/RCS,STR(1),ATR(1),ETR(1),TSEC(1),SIGNAL(1),SNCTRC(1)
1,DBTAR(1),AER(1),EER(1),RER(1),DISP(1),KEYT(1),KOUT,LFDUT
COMMON /BURST/NB,W(50),TSX(50),FF(50),HB(50),XBR(50),YBR(50),
1 ZBR(50),TB(50),COSCLB(50),RDZERO(50),RHZERO(50),
2 RUZERO(50),LHVB(50),RHOB(50),HSB(50),PKTERM(50),
3 RLK(50),RTK(50),DISTK(50),XKR(50),YKR(50),ZKR(50),
4 TILTK(50),WUV(3,50),HR,COSCLR,SINCLR,KALCH,KALCN,
5 KALCR,DAY,CIRAX,TEM150,NSW1,NSW2,NSW3,NSW4,NSW5,NSW6
COMMON/REGION/NF,INDXF(50),FLABL(50),KINDF(50),RFB(50),AFB(50),
1 HF(50),XFR(50),YFR(50),ZFR(50),RLF(50),RTF(50),
2 TILTF(50),HMINF(50),HMAXF(50),TFO(50),TF(50),
3 RHOF(50),HSNF(50),HDOT(50),RDOT(50)
COMMON/DEBRIS/ND,INDXD(50),DLABL(50),RDB(50),ADB(50),HD(50),
1 XDR(50),YDR(50),ZDR(50),RLD(50),RTD(50),COSCLD(50),
2 DISTD(50),WDI(50),PDI(50),RHDD(50),HSD(50),HBETA(50)
COMMON /PARAM/ALPHAR,GAMMA,UKOEA,UN,UBETA,UGAMMA,N1,N2,DMAX,HBMAX,
1 DHSTEP,RSTOP,OBSTOP,DEL,EDMINO,ENMINF,KINT,DENRAT,
2 FTRAIN,HTRAIN
COMMON/CNEXEC/GXMTR,BXMTR,XXMTR,KXMTR,GRCVR,BRCVR,SRGVR,KRCVR,
1 GTM,GRM,GM,COSCAN,PHIB,TAMB,NFB,BEAM,SNTERM
COMMON /CONST/RE,PI,RADIAN
COMMON/RNC/ICONF
DIMENSION INDEXT(50)
J=1
GO TO (81,1,80),ICONF
81 CONTINUE
RE = 6370.
PI = 3.14159265
RADIAN = 1.74532925E-2
ALPHAR = 1.E-12
GAMMA = 1.5
UKOEA = 1.E4
UN = 0.015
UBETA = 2.0
UGAMMA = 0.025
N1 = 3
N2 = 3
DMAX = 100.
HBMAX = 200.
DHSTEP = 2.5
RSTOP=0.01
OBSTOP=0.1
```

```

DEL = 0.001
EDMIND = 5.4E-9
ENMINF = 5.E4
KINT=1
DENRAT=2.
FTRAIN=1.
HTRAIN=1.
KASE=0
KSPEC=0
NFACE=0
NB=0
BOLTZ=1.3805E-20
C=2.998E+5
DBCON=4.34294482
FOURPI=4.*PI
TAMB=250.
HACMIN=90.
HACMAX=130.
RHOAC=1.E-10
1 CALL RANCIN(KASE)
  IF (POWER) 2, 730, 2
2  SNMIN=10.** (SNMDB/10.)
  GXMTR=10.** (TAG/10.)
  BXMTR=TBW*RADIAN
  SXMTR=10.** (-TSL/10.)
  KXMTR=MCD(KSPEC,100)/10
  GRCVR=10.** (RAC/10.)
  BRCVR=RBW*RADIAN
  SRCVR=10.** (-RSL/10.)
  KRCVR=MCD(KSPEC,10)
  DELRC=1.E-3*DELTAR
  DELRU=DELRC*PCOMP
  SNTERM=POWER*(C/FREQ)**2*PCOMP/(FOURPI)**3
  IF (NB) 11, 20, 11
11 DO 12 I=1,NB
12 COSCLB(I)=(COSCLR*(RE+HR+ZBR(I))+SINCLR*YBR(I))/(RE+HB(I))
20 CONTINUE
  JTIME=1
  TIME=TSEC(JTIME)
  CALL PHENOM(TSEC,JTIME)
  RETURN
80 CONTINUE
  TASUM=TSYS
  AZBEAM=ATR(J)
  ELBEAM=ETR(J)

```


GRID

This subroutine is used as an automatic searching routine. When PONDER is run in its auto mode (UNITH ≤ 0), Grid will compute the limits of the search grid based on fireball and debris dimension, which is used by PONDER to produce a picture enclosing this total region. To perform this function Grid normally need only be presented PLDEN and RDEN plus the auto run control values from the number 9 card. All other variables related to PONDER picture production are internally generated.

Listing

```
SUBROUTINE GRID(MAXNHR)
COMMON/RNCIN/PLDEN,RDEN,NMXR,NPXR,NHR1,NHR,UNITH,UNITV,UNITHO,
1 UNITVO,UNITHR,UNITVR,AZREF,AZRF,RTG,MAXNMX,NCRCT,VARR,SBOCRIT,
2 ZDSNBOCR,AREAD
COMMON/REGION/NF,INDXF(50),FLABL(50),KINDF(50),RF8(50),AF8(50),
1 HF(50),XFR(50),YFR(50),ZFR(50),RLF(50),RTF(50),
2 TILTF(50),HMINF(50),HMAXF(50),TF0(50),TF(50),
3 RHOF(50),HSNF(50),HDOOT(50),RDOT(50)
COMMON/DEBRIS/ND,INDXD(50),DLABL(50),ROB(50),AOB(50),HO(50),
1 XDR(50),YDR(50),ZDR(50),RLD(50),RTD(50),COSCLD(50),
2 DISTD(50),WDI(50),PD(50),RHOD(50),HSD(50),HBETA(50)
REAL MAXNHR,MAXNMXR,MAXNPXR,MINRBR,MXHOR,MXTOP,MNBOT
DIMENSION X(12),Y(3),Z(3)
DIMENSION RBR(50),ABR(50)
DATA X /-.00390625, .0078125, .015625, .03125, .0625, .125, .25, .5, 1., 2.,
14., 8./
DATA Y /-.6, .590909, .5/
DATA Z /120., 120., 85./
IF(UNITH.GT.0.) GO TO 50
PI=3.1415926536
RACIAN = PI/180.
RO=6364.
MXHOR=MAXNMX
MAXNHR=INTF(Z(NMXR)/(2.*RDEN))-1.
MAXNMXR=MAXIF(INTF(MXHOR+1.)/RDEN)-1.,0.)
MAXNPXR=INTF(Z(NMXR)/RDEN)-2.-MAXNMXR
RATIO=Y(NMXR)
NM=NPXR/1000000
NPXR=NPXR-NM*1000000
NP=NPXR/1000
NH=NPXR-NP*1000
IF(MOD(NH,2).NE.0) NH=NH+1
MINRBR=10000.
DO 2 I=1,NF
CALL COORDX(0.,2,XFR(I),YFR(I),ZFR(I),RBR(I),ABR(I),A,B,MXTCP,
1 MNBOT,MXHOR,THETA0)
MINRBR=MINIF(MINRBR,RBR(I))
2 CONTINUE
THETA0=MINRBR/RO
MXHOR=0.
MXTOP=0.
MNBOT=10000.
DO 1 I=1,NF
MXTOP=MAXIF(MXTOP,HF(I)+RLF(I))
MNBOT=MINIF(MNBOT,HF(I)-RLF(I))
```

```

IF(KIND(I).GT.2) GO TO 32
MXHOR=MAXIF(MXHOR,RTF(I))
GO TO 1
32 CONTINUE
MXHOR=MAXIF(MXHOR,RTF(I)+RLF(I))
1 CONTINUE
IF(UNITV.LT.0.) GO TO 4
DO 3 I=1,ND
MXTOP=MAXIF(MXTOP,HD(I)+RLD(I))
MNBOT=MINIF(MNBOT,HD(I)-RLD(I))
MXHOR=MAXIF(MXHOR,RTD(I))
3 CONTINUE
4 CONTINUE
A=.5*(MXTOP+MNBOT)
B=MXHOR
AO=RO*(1.-COSF(THETA0))/COSF(THETA0)
IF(A.GT.AO) GO TO 5
WRITE(6,20)
A=1.25*A0
5 CONTINUE
SO=SQRTF((RO+A)**2+RO**2-2.*RO*(RO+A)*COSF(THETA0))
PHI=B/(SO*RADIAN)
PHIM=-90.
PHIP=-90.
DO 6 I=1,NF
PHIP=MAXIF(PHIP,PHI+ABR(I)-AZREF)
PHIM=MAXIF(PHIM,PHI-ABR(I)+AZREF)
6 CONTINUE
A2PRO=SQRTF(MXHOR**2+(MXTOP+RO)**2)
THETA2=THETA0-ATANF(MXHOR/(MXTOP+RO))
S2=SQRTF(RO**2+A2PRO**2-2.*RO*A2PRO*COSF(THETA2))
ALPHA2=ACOSF(A2PRO*SINF(THETA2)/S2)/RADIAN
IF(UNITV.EQ.0.) GO TO 30
A1PRO=SQRTF(MXHOR**2+(MNBOT+RO)**2)
THETA1=THETA0+ATANF(MXHOR/(MNBOT+RO))
S1=SQRTF(RO**2+A1PRO**2-2.*RO*A1PRO*COSF(THETA1))
ALPHA1=ACOSF(A1PRO*SINF(THETA1)/S1)/RADIAN
GO TO 31
30 CONTINUE
ALPHA1=0.
31 CONTINUE
ALPHA=ALPHA2-ALPHA1
UNVR=ALPHA/MAXNHR
UNHRM=PHIM/MAXNMXR
UNHRP=PHIP/MAXNPXR
DO 7 I=1,12
IV=I
IF(UNVR.GE.X(I)) GO TO 7
I=12
7 CONTINUE
DO 9 I=1,12
IH=I
IF((UNHRP.GE.X(I)*RATIO).OR.(UNHRM.GE.X(I)*RATIO)) GO TO 9
I=12
9 CONTINUE
8 CONTINUE
UNITVR=MAXIF(X(IV),X(IH))
UNITHR=UNITVR*RATIO
IF(((ALPHA/UNITVR)+NH).GT.MAXNHR) GO TO 10
IF(((PHIM/UNITHR)+NM).GT.MAXNMXR).OR.(((PHIP/UNITHR)+NP).GT.
1MAXNPXR)) GO TO 11
71 CONTINUE
NHR=(ALPHA2/UNITVR)+NH/2

```

```

NHRI=(ALPHA/UNITVR)+NH
NHRI=MINOF(NHRI,NHR)
NMXR=(PHIM/UNITHR)+NM
NPXR=(PHIP/UNITHR)+NP
GO TO 12
10 CONTINUE
IF (IV.EQ.12) GO TO 71
IV=IV+1
GO TO 8
11 CONTINUE
IF (IH.EQ.12) GO TO 71
IH=IH+1
GO TO 8
12 CONTINUE
UNITVO = UNITVR/RDEN
UNITHO = UNITHR/RDEN
UNITV = UNITVO/PLDEN
UNITH = UNITHO/PLDEN
GO TO 51
50 UNITVO = PLDEN*UNITV
UNITHO = PLDEN*UNITH
UNITVR = RDEN*UNITVO
UNITHR = RDEN*UNITHO
51 CONTINUE
RETURN
20 FORMAT(16HFB BELOW HORIZON)
END

```

SPRAY

This subroutine produces the target matrix point by point, left to right, row by row. Each target point produced by SPRAY is then checked by RANC to find its S/N.

FTG = target shell radius
UNH = grid horizontal increment either
NR = position along horizontal line at which target is now located
ELREF = elevation angle (degrees cona = 0, km cona = 2) set to topmost row of PONDER picture and decremented by the vertical print grid increment until desired region of sky plotted
AZREF = azimuth from radar to grid center column (for case cona = 0)
VARR = magnetic variation
ATR = azimuth radar to target
STR = slant range radar to target
ETR = elevation angle to target relative to radar

Listing

```
SUBROUTINE SPRAY(RTG,UNH,NR,ELREF,AZREF,VARR,ATR,STR,ETR)
COMMON/CONAL/CONA,GLATO,GLONO,HHR
COMMON/CMSPRAY/COSCLO,SINCLO,XRD,YRD,ZRD,ARD,AROSP,ISPF
COMMON/TARCNT/ITOT
ITOT=ITOT+1
PI=3.14159265359
RADIAN=PI/180.
RN=NR
IF(CONA.EQ.2.) GO TO 6
PHIO=AZREF
THETA0=ELREF
B=SINF(THETA0*RADIAN)
IF(THETA0-90.)1,2,3
1 THEBAR=RN*UNH
5 A=CGSF(THETA0*RADIAN)
C=SINF(THETA0*RADIAN)
THETA=ASINF(A*B)/RADIAN
ALPHA=ASINF(C/SQRTF(1.-(A**2)*(B**2)))/RADIAN
PHI=PHIO-ALPHA
GO TO 4
2 THEBAR=RN*UNH
THETA=90.+SIGNF(-1.,RN)*THEBAR
PHI=PHIO+SIGNF(1.,RN)*90.
GO TO 4
3 PHIO=PHIO+180.
THEBAR=-RN*UNH
GO TO 5
4 ETR=THETA
ATR=PHI
STR=RTG
IF(CONA.EQ.1.) GO TO 17
16 CONTINUE
RETURN
```

```

17 GLAT2=ETR
   H2= STR
   GLON2 = ATR
   GLON1 = GLON0
   GLAT1 = GLAT0
   H1=HHR
   CALL XYZ(GLAT1,GLON1,H1,GLAT2,GLON2,H2,X21,Y21,Z21)
   MODE = 2
   HO= H1
   CALL COORDX(HO,MODE,X21,Y21,Z21,R,A,H,X,Y,Z,S,E)
   ETR = E
   ATR=A-VARR
   STR = S
   GO TO 16
6  CONTINUE
   A1=RN*UNH
   A2=AROSP-AZREF-VARR
   YTP=A1+RTG
   GO TO(7,11,9,10),ISPF
7  CONTINUE
   YTP=-YTP+2.*RTG
   GO TO 11
9  CONTINUE
   IF(A2.LT.0.) YTP=-YTP+2.*RTG
   GO TO 11
10 CONTINUE
   IF(A2.GT.0.) YTP=-YTP+2.*RTG
11 CONTINUE
   ZTO=ELREF
   YTO=YTP*COSEF((AZREF+VARR)*RADIAN)
   XTO=YTP*SINF((AZREF+VARR)*RADIAN)
   CALL ASPECT(COSCLO,S INCLO,0.,XRD,YRD,ZRD,1,XTO,YTO,ZTO,XTR,YTR,
1ZTR)
   CALL COORDX(HHR,2,XTR,YTR,ZTR,A,ATR,B,C,THETA,PHI,STR,ETR)
   ATR=ATR-VARR
   GO TO 16
END

```

RANCIN

This subroutine reads the input for PONDER calculations.

Listing

```
SUBROUTINE RANCIN(KASE)
COMMON /RADAR/FREQ,POWER,DELTAR,DELTAV,TAG,TBW,TSL,SNMDB,TSYS,BWN,
1      PCOMP,RAG,RBW,RSL,KSPEC,SCAN,EFACE,AFACEM(4),NFACE
COMMON/TARGET/RCS,STR(1),ATR(1),ETR(1),TSEC(1),SIGNAL(1),SNCTRC(1)
1,DBTAR(1),AER(1),EER(1),RER(1),QISP(1),KEYT(1),KOUT,LFOUT
COMMON /BURST/NB,W(50),TSX(50),FF(50),HB(50),XBR(50),YBR(50),
1      ZBR(50),TB(50),COSCLB(50),RDZERO(50),RHZERO(50),
2      RUZERO(50),LHVB(50),RHOB(50),HSB(50),PKTERM(50),
3      KLB(50),RTK(50),DISTK(50),XKR(50),YKR(50),ZKR(50),
4      TILTK(50),WUV(3,50),HR,CQSCLR,SINCLR,KALCH,KALCN,
5      KALCR,DAY,CIRAX,TEM150,NSW1,NSW2,NSW3,NSW4,NSW5,NSW6
COMMON /CONST/RE,PI,RADIAN
COMMON/RNCIN/PLDEN,KDEN,NMXX,NPXX,NHR1,NHR,UNITH,UNITV,UNITHO,
1UNITVO,UNITHR,UNITVR,AZREF,AZRF,RTG,MAXNMX,NCRCCT,VARR,SBOCRIT,
2DSNBOCR,AREAD
COMMON/CONAL/CONA,GLATO,GLOND,HHR
COMMON/CMSPRAY/CQSCLQ,SINCLQ,XRQ,YRQ,ZRQ,ARQ,AROSP,ISPF
DIMENSION X(7),AFACE(4)
DIMENSION RBO(50),ABO(50),XBO(50),YBO(50),ZBO(50),SBO(50),EBO(50),
1GLATB(50),GLONB(50),KLB(50),ALPHA6(50)
DIMENSION RTO(1),ATO(1),HT(1),XTO(1),YTO(1),ZTO(1),STO(1),ETO(1),
1GLATT(1),GLONT(1),KLT(1),ALPHA7(1)
GO TO 80
8700 AREAD=1.
GO TO 1
80 CONTINUE
LOGICAL ENFIL
LFIN=5
LFOUT=6
GO=980.0
RGAS=8.3143E+7
NHELP=0
IN6=0
NRV=0
NT=1
1 READ(LFIN,1001) NCARD,{X(I),I=1,7},KODE,ALPHA
IF(EOF,5) 4,5
4 CALL EXIT
5 IF((NCARD.EQ.0.).AND.(KODE.EQ.9)) NCARD=10
IF(AREAD)8701,8702,8701
8702 IF(NCARD.EQ.10)8700,1
8701 IF(NCARD)2,101,2
2 GO TO (11,21,31,41,51,61,71,81,91,1),NCARD
11 FREQ=X(1)
POWER=X(2)
DELTAR=X(3)
DELTAV=X(4)
TAG=X(5)
TBW=X(6)
TSL=X(7)
ALPHA1=ALPHA
IF(KODE)12,1,12
```

```

12 CALL DATAIN(KODE,AFACE)
   ALPHA2=ALPHA
   GO TO 1
21 SNMDB=X(1)
   TSYS=AMAX1(X(2),1.)
   BWN=AMAX1(X(3),1.E-3)
   PCOMP=X(4)
   RAG=X(5)
   RBW=X(6)
   RSL=X(7)
   ALPHA2=ALPHA
   GO TO 1
31 KSPEC=X(1)
   SCAN=X(2)
   EFACE=X(3)
   AFACE(1)=X(4)
   NFACE=1
   DO 33 I=5,7
     IF(X(I))32,33,32
32 NFACE=NFACE+1
   AFACE(NFACE)=X(I)
33 CONTINUE
   IF(SCAN)34,34,1
34 NFACE=0
   GO TO 1
41 GLATQ=X(1)
   GLONO=X(2)
   DAY=X(3)
   KALC=X(4)
   KALCH=MAX0(1-MOD(KALC,1000)/100,0)
   KALCN=MAX0(1-MOD(KALC,100)/10,0)
   KALCR=MAX0(1-MOD(KALC,10),0)
   KOUT=X(5)
   KTRACK=MIN1(ABS(X(6)),10.)
   KODE=X(7)
   NSW1=MOD(KODE,1000000)/100000
   NSW2=MOD(KODE,100000)/10000
   NSW3=MOD(KODE,10000)/1000
   NSW4=MOD(KODE,1000)/100
   NSW5=MOD(KODE,100)/10
   NSW6=MOD(KODE,10)
   IF(NSW6.EQ.1) NSW5=1
   IF(NSW5.EQ.1) NSW4=1
   IF(NSW4.EQ.1) NSW3=1
   GO TO 1
51 IF(KODE)52,53,52
52 CALL COORDX(0.,KODE,X(1),X(2),X(3),RRQ,ARQ,HR,XRQ,YRQ,ZRQ,SRQ,ERO)
   GO TO 54
53 GLATR=X(1)
   GLONR=X(2)
   HR=X(3)
54 KLR=KODE
   DIPR=X(4)
   VARR=X(5)
   ALPHA5=ALPHA
   HHR=HR
   GO TO 1
61 IF(IN6)62,65,62
62 IF(IN6-50)63,69,63
63 IF(X(4)-TB(NB))64,65,65
64 NHELP=63
65 IN6=IN6+1
   NB=IN6

```

```

IF(KODE)66,67,66
66 CALL COORDX(0.,KODE,X(1),X(2),X(3),RBO(NB),ABO(NB),HB(NB),XBO(NB),
  1YBO(NB),ZBO(NB),SBO(NB),EBO(NB))
GO TO 68
67 GLATB(NB)=X(1)
GLONB(NB)=X(2)
HB(NB)=X(3)
68 KLB(NB)=KODE
TB(NB)=X(4)
W(NB)=X(5)
FF(NB)=X(6)
TSX(NB)=X(7)
ALPHA6(NB)=ALPHA
GO TO 1
69 NHELP=62
GO TO 1
71 TSEC(NT)=X(1)
MAXNMX=X(2)
NCRCT=X(3)
GO TO 1
81 AZRF=X(1)
RTG=X(2)
RCS=X(3)
SBOCRIT=X(4)
DSNBDCR=X(5)
CONA=KODE
GO TO 1
91 PLDEN = X(1)
RDEN =X(2)
UNITH =X(3)
UNITHO=X(3)
UNITVO=X(4)
NMXO=X(5)
NPXO=X(6)
NHR=X(7)
UNITH=UNITHO*PLDEN
UNITV=UNITVO*PLDEN
NMXR=NMXO/RDEN
NPXR=NPXO/RDEN
NHR=NHR/RDEN
PLDEN=1./PLDEN
NHR1=NHR
IF(NHR.LT.100000) GO TO 1
NHR1=NHR/100000
NHR=NHR-NHR1*100000
GO TO 1
101 CONTINUE
WRITE OUTPUT TAPE 6,4009
IF(NCRCT.EQ.0) WRITE OUTPUT TAPE 6,4015
IF(KLR)102,103,102
102 CALL LATLON(GLATO,GLONO,0.,XRO,YRO,ZRO,GLATR,GLONR,X(1))
GO TO 104
103 CALL XYZ(GLATO,GLONO,0.,GLATR,GLONR,HR,XRO,YRO,ZRO)
CALL COORDX(0.,2,XRO,YRO,ZRO,RRO,ARO,X(1),XRO,YRO,ZRO,SRO,ERO)
104 IF(NB)105,110,105
105 DO 106 1=1,NB
  IF(KLB(1))106,107,106
106 CALL LATLON(GLATO,GLONO,0.,XBO(1),YBO(1),ZBO(1),GLATB(1),GLONB(1),
  1X(1))
GO TO 108
107 CALL XYZ(GLATO,GLONO,0.,GLATB(1),GLONB(1),HB(1),XBO(1),YBO(1),
  1ZBO(1))
CALL COORDX(0.,2,XBO(1),YBO(1),ZBO(1),RBO(1),ABO(1),X(1),XBO(1),

```



```

      LYBO(I),ZBO(I),SBO(I),EBO(I))
108 CONTINUE
110 IF(NRV)111,115,111
111 DO 114 I=1,NRV
      IF(KLT(I))112,113,112
112 CALL LATLON(GLATO,GLONO,0.,XTO(I),YTO(I),ZTO(I),GLATT(I),GLONT(I),
      IX(I))
      GO TO 114
113 CALL XYZ(GLATO,GLONO,0.,GLATT(I),GLONT(I),HT(I),XTO(I),YTO(I),
      IZTO(I))
      CALL COORDX(0.,2,XTO(I),YTO(I),ZTO(I),RTO(I),ATO(I),X(I),XTO(I),
      YTO(I),ZTO(I),STO(I),ETD(I))
114 CONTINUE
115 CONTINUE
      CLO=(90.-GLATO)*RADIAN
      COSCLO=COS(CLO)
      SINCLC=SIN(CLO)
      IF(NB)303,311,303
303 CALL ASPECT(COSCLO,SINCLC,0.,XRO,YRO,ZRO,NB,XBO,YBO,ZBO,XBR,YBR,
      IZBR)
      DO 308 I=1,NB
      CALL COORDX(HR,2,XBR(I),YBR(I),ZBR(I),RBR,ABR,X(1),X(2),X(3),X(4),
      ISBR,EBR)
308 CONTINUE
311 IF(NRV)312,400,312
312 DO 317 I=1,NRV
      CALL ASPECT(COSCLO,SINCLC,0.,XRO,YRO,ZRO,1,XTO(I),YTO(I),ZTO(I),
      IXTR,YTR,ZTR)
      CALL COORDX(HR,2,XTR,YTR,ZTR,RTR,ATR(I),X(1),X(2),X(3),X(4),STR(I),
      I,ETR(I))
317 CONTINUE
400 CONTINUE
402 CALL ATMOS0(-LFOUT,0.,X(1),X(2),X(3),X(4),X(5),X(6),X(7))
      CALL CIRA65(-LFOUT,150.,X(1),X(2),TEM150,X(3),X(4),X(5),EM150)
      G150=GO*(RE/(RE+150.))**2
      CIRAX=1.+150.E+5*G150*EM150/RGAS/TEM150
      X(1)=SIN(DIPR*RADIAN)
      COSCLR=X(1)/SQRT(4.-3.*X(1)**2)
      SINCLR=SQRT(1.-COSCLR**2)
      VMR=VARR*RADIAN
      COSVMR=COS(VMR)
      SINVMR=SIN(VMR)
      IF(NB)702,711,702
702 DO 703 I=1,NB
      X(1)=XBR(I)*COSVMR-YBR(I)*SINVMR
      YBR(I)=XBR(I)*SINVMR+YBR(I)*COSVMR
703 XBR(I)=X(1)
711 IF(NRV)712,721,712
712 DO 713 I=1,NRV
713 ATR(I)=ATR(I)-VARR
721 IF(NFACE)722,724,722
722 DO 723 I=1,NFACE
723 AFACEM(I)=AFACE(I)-VARR
724 CONTINUE
      IF(NHELP.NE.0) CALL HELPI6HRANCIN,NHELP,I)
      RETURN
1001 FORMAT(11,E9.0,6E10.0,1X11,2XA6)
4009 FORMAT(13H1RUN COMMENTS)
4015 FORMAT(41HNO FINE CORRECTIONS MADE TO TAILORED GRID)
      END

```

FIT

This subroutine plots blank horizontal lines for the PONDER picture, for cases in which no blackout was found at that elevation. This subroutine defines the boundaries of the print and area grids using the search grid calculations. The flag IRUN that switches the executive routine from search to area grid calculations and vice versa is set in this routine.

IRUN = Grid Selection Flag (IRUN = 1:search grid, IRUN = 2:area grid)
NMXP = number of targets on print grid row to the left of center column
NPXP = number of targets on print grid row to the right of center column
NMX = number of targets on area grid row to the left of center column
NPX = number of targets on area grid to the right of center column
NMXR = number of targets on search grid row to the left of center column
NPXR = number of targets on search grid row to the right of center column
UNITHP = print grid horizontal increment
UNITVP = print grid vertical increment
UNITH = area grid horizontal increment
UNITV = area grid vertical increment
UNITHR = search grid horizontal increment
UNITVR = search grid vertical increment

Listing

```
SUBROUTINE FIT(NPRES,NLPRES,NRPRES,IRUN,NMXP,NPXP,NMX,NPX,NMXR,NPXR,NMPO,NPPO,NMX,
  INPX,IPLDEN,IRDEN,MAXNMX,IL,IR)
  DIMENSION S(150)
  NPREV=NPRES
  NLPREV=NLPRES
  NRPREV=NRPRES
  IF(IL)3,4,3
4  NPRES=1
  IF(NPREV-2)5,6,5
6  NMPO=NLPREV*IRDEN
  NPPO=NRPREV*IRDEN
  NMX=NMPO*IPLDEN
  NPX=NPPO*IPLDEN
  IRUN=2
  RETURN
5  DO 13 I=1,IRDEN
  IO=MOD(I,IRDEN)
  IF(IO)14,15,14
14  S(I)=IH
  JI=1
  GO TO 13
15  JO=MAXNMX+1
  DO 16 J=1,JO
16  S(J)=IH
  JI=JO+1
  S(JI)=IH-
13  WRITE OUTPUT TAPE 8,3001,(S(I),I=1,JI)
```

```

3001 FORMAT(120A1)
      NMX=NMXR
      NPX=NPXR
      IRUN=1
      RETURN
  3 IF(NPRES111,12,11
12 WRITE OUTPUT TAPE 6,3009
3009 FORMAT(28H VERTICAL ESTIMATE TOO SMALL)
11 NPRES=2
   NL=NMXR+2-IL
   NR=NPXR+2-IR
   NLPRES=MINOF(NL,NM XR)
   NRPRES=MINOF(NR,NP XR)
   IF(NPREV-219,10,9
  9 NMXO=NLPRES*IRDEN
    NPXO=NRPRES*IRDEN
    NMX=NMXO*IPLDEN
    NPX=NPXO*IPLDEN
    IRUN=2
    RETURN
10 NMXO=MAXOF(NLPREV,NLPRES)*IRDEN
    NPXO=MAXOF(NRPREV,NRPRES)*IRDEN
    NMX=NMXO*IPLDEN
    NPX=NPXO*IPLDEN
    IRUN=2
    RETURN
END

```

CRCT

This subroutine performs horizontal grid corrections, as specified by NCRCT. When the blackout region extends past the PONDER picture limits, the subroutine CRCT sets in and extends the bounds by "NCRCT" number of grids to see if it can reach the outer limit of the blackout region.

NMX = (NMXR) number of targets per row to the left of center column
NPX = (NPXR) number of targets per row to the right of center column
NCRCT = grid correction factor (in units of print grid increments)

Listing

```
SUBROUTINE CRCT(NMXIRD,NPXIRD,NMXO,NPXO,NMX,NPX,NPXRI,IPLDEN,NCRCT
1, ILCT,IRCT,NTL,NTR,NRL,NRR)
  NTL=0
  NTR=0
  IF(ILCT.EQ.0) GO TO 1
  NMXOPRV=NMXO
  NMXO=MINO(NMXIRD,NMXO+NCRCT)
  IF(NMXOPRV.EQ.NMXO) GO TO 1
  NMX=NMXO*IPLDEN
  NTL=(NMXO-NMXOPRV)*IPLDEN
  NRL=NMX+1
1 CONTINUE
  IF(IRCT.EQ.0) GO TO 2
  NPXOPRV=NPXO
  NPXO=MINO(NPXIRD,NPXO+NCRCT)
  IF(NPXOPRV.EQ.NPXO) GO TO 2
  NRR=-NPX
  NPXRI=NPXO*IPLDEN
  NTR=(NPXO-NPXOPRV)*IPLDEN
2 CONTINUE
  RETURN
END
```

ANALYZE

This subroutine checks for the horizontal corrections made by subroutine CRC'T and prints out the appropriate information concerning each correction.

ELREF = increment in the vertical direction. ELREF set to topmost point to be observed and decremented to lowest point to be observed.

Listing

```
SUBROUTINE ANALYZE(NTL,NTR,ILCT,IRCT,ELREF)
  IF(NTL.EQ.0) GO TO 1
  IF(ILCT.EQ.0) GO TO 3
  WRITE OUTPUT TAPE 6,102,ELREF
  GO TO 4
3 CONTINUE
  WRITE OUTPUT TAPE 6,101,ELREF
4 CONTINUE
  IF(NTR.EQ.0) GO TO 2
  IF(IRCT.EQ.0) GO TO 5
  WRITE OUTPUT TAPE 6,104,ELREF
  GO TO 6
5 CONTINUE
  WRITE OUTPUT TAPE 6,103,ELREF
6 CONTINUE
  GO TO 99
1 CONTINUE
  IF(ILCT.EQ.1) WRITE OUTPUT TAPE 6,105,ELREF
  GO TO 4
2 CONTINUE
  IF(IRCT.EQ.1) WRITE OUTPUT TAPE 6,106,ELREF
99 CONTINUE
  RETURN
101 FORMAT(47HSUCCESSFUL CORRECTION MADE ON LEFT AT ELEVATION,F10.5)
102 FORMAT(49HUNSUCCESSFUL CORRECTION MADE ON LEFT AT ELEVATION,F10.5)
103 FORMAT(48HSUCCESSFUL CORRECTION MADE ON RIGHT AT ELEVATION,F10.5)
104 FORMAT(50HUNSUCCESSFUL CORRECTION MADE ON RIGHT AT ELEVATION,
1F10.5)
105 FORMAT(73HCORRECTION ATTEMPTED ON LEFT BUT STOPPED BY S. G. BOUNDARY.
ELEVATION IS,F10.5)
106 FORMAT(74HCORRECTION ATTEMPTED ON RIGHT BUT STOPPED BY S. G. BOUNDARY.
ELEVATION IS,F10.5)
  END
```

PLOTSN

This is the plotting subroutine. It takes the S/N calculations produced elsewhere, codes them alpha numerically point by point, and stores them in a row matrix left to right for one row. The coded line is next written into the output file called PLOTSN.

SIGNAL = number generated by RANC for target presently being observed (this is the completed signal/noise ratio)
KEYT = number generated by RANC for target presently being observed (target calculation code)
DSNBOCR = signal/noise increment between output contours
NPX = number of target per row in unrotated system to the right of center column
NMX = number of targets per row in unrotated system to the left of center column
NPY = number of rows in unrotated system above the row containing reference point
NMY = number of rows in unrotated system below the row containing reference point

Listing

```
SUBROUTINE PLOTSN(SIGNAL,KEYT,BOCR,DBOCR,NPX,NMX,NPY,NMY,MAXNMX,IP-  
ILDEN)  
  DIMENSION SIGNAL(2),KEYT(2),R(150)  
  NO=MAXNMX-NMX+1  
  N1=MAXOF(1,NO)  
  J1=NPY+NMY+1  
  K1=NPX+NMX+1  
  K2=N1+K1  
  I=1  
  DO 1016 J=1,J1  
  DO 1017 N=1,N1  
1017 R(N)=IH  
  DO 1015 K=1,K1  
  N=N1+K  
  SIG=SIGNAL(I)-BOCR  
  IF(SIG)1001,1002,1002  
1002 R(N)=IH  
  GO TO 1015  
1001 IF(SIG+DBOCR)1003,1004,1004  
1004 R(N)=IH.  
  GO TO 1015  
1003 IF(SIG+2.*DBOCR)1005,1006,1006  
1006 R(N)=IHC  
  GO TO 1015  
1005 IF(SIG+3.*DBOCR)1007,1008,1008  
1008 R(N)=IHE  
  GO TO 1015  
1007 IF(SIG+4.*DBOCR)1009,1010,1010  
1010 R(N)=IHA  
  GO TO 1015
```

```

1009 CONTINUE
      KEY=KEYT(I)
      CALL SLCT(KEY)
      KEY=KEY+3
      GO TO(2001,2002,2003,2004,2005,2006,2007,2008,2009),KEY
2001 R(N)=1H8
      GO TO 1015
2002 R(N)=1H7
      GO TO 1015
2003 R(N)=1H0
      GO TO 1015
2004 R(N)=1H1
      GO TO 1015
2005 R(N)=1H2
      GO TO 1015
2006 R(N)=1H3
      GO TO 1015
2007 R(N)=1H4
      GO TO 1015
2008 R(N)=1H5
      GO TO 1015
2009 R(N)=1H6
1015 I=I+1PLDEN
1016 WRITE OUTPUT TAPE 8,3001,(R(K),K=1,K2)
      RETURN
3001 FORMAT(120A1)
      END

```

AREA

This subroutine calculates the area of radar blackout on the 1KM target shell for the spherical mode; otherwise, it calculates the blackout area on the target surface.

Listing

```
SUBROUTINE AREA(SIGNAL,SSIGNAL,BOCRIT,UNITH,UNITV,NMX,NPX,NT,MAXNT
1,IAREA,AREAI,AREAU,ICONA)
DIMENSION SIGNAL(2),SSIGNAL(2)
PI=3.14159265359
RADIAN=PI/180.
NTP1=NT+1
MXNTP1=MAXNT+1
DO 100 I=1,NT
  IF(SIGNAL(1)-BOCRIT)101,102,102
101 SIGNAL(I)=1.
  GO TO 100
102 SIGNAL(I)=0.
100 CONTINUE
  DO 103 I=NTP1,MXNTP1
103 SIGNAL(I)=0.
  IF(IAREA)1,2,1
    2 NMXPREV=NMX
    NPXPREV=NPX
    1 IGAP=NMX-NMXPREV
    IF(IGAP)10,13,12
    12 DO 20 I=1,NTPREV
    20 SSIGNAL(NTPREV+IGAP+1-I)=SSIGNAL(NTPREV+1-I)
    DO 30 I=1,IGAP
    30 SSIGNAL(I)=0.
    13 ISHIFT=0
    GO TO 11
    10 IGAP=-IGAP
    DO 40 I=1,NT
    40 SIGNAL(NT+IGAP+1-I)=SIGNAL(NT+1-I)
    DO 50 I=1,IGAP
    50 SIGNAL(I)=0.
    ISHIFT=IGAP
    11 NM=MAXOF(NMX,NMXPREV)+MAXOF(NPX,NPXPREV)
    NMP2=NM+2
    DO 120 I=1,NM
    NO=SIGNAL(I)+SIGNAL(I+1)+SSIGNAL(I)+SSIGNAL(I+1)+1.
    GO TO(120,121,121,121,122),NO
    121 IFLAG=1
    GO TO 123
    122 IFLAG=2
    123 CONTINUE
    GO TO(301,302,303),ICONA
    301 CONTINUE
    MXNMX=MAXOF(NMX,NMXPREV)
    IF(I-(MXNMX+1))124,125,125
    124 THETA=(MXNMX-I)*UNITH
    GO TO 126
    125 THETA=(I-(MXNMX+1))*UNITH
    126 A=UNITV*RADIAN*(SINF((THETA+UNITH)*RADIAN)-SINF(THETA*RADIAN))
    202 CONTINUE
    GO TO(127,128),IFLAG
    128 AREAI=AREAI+A
```



```

127 AREAD=AREAD+A
GO TO 120
302 CONTINUE
A=0.
GO TO 202
303 CONTINUE
A=UNITV*UNITH
GO TO 202
120 CONTINUE
NTPREV=NT
NMXPREV=NMX
NPXPREV=NPX
IAREA=1
DO 60 I=1,NT
60 SSIGNAL(I)=SIGNAL(ISHIFT+I)
DO 70 I=NTPI,NMP2
70 SSIGNAL(I)=0.
RETURN
END
FUNCTION ASIN(X)
PI=3.14159265359
IF(X)1,2,3
1 Z=X+1.0
IF(Z)4,5,6
4 WRITE(6,1001)
5 ASIN=-PI/2.
GO TO 7
3 Z=1.0-X
IF(Z)8,9,6
8 WRITE(6,1002) Z
9 ASIN=PI/2.
GO TO 7
2 ASIN=0.
GO TO 7
6 Y=SQRT(1.0-X*X)
ASIN=ATANF(X/Y)
7 RETURN
1001 FORMAT(28HARGUMENT OF ARCSINE PLUS ONE,1PE20.12)
1002 FORMAT(29HCONE MINUS ARGUMENT OF ARCSINE,1PE20.12)
END
FUNCTION ACOS(X)
PI=3.14159265359
ACCS=PI/2.-ASINF(X)
RETURN
END

```

COORDX

This subroutine accepts coordinates of a point in any of four systems and transforms them to all of the other systems.

Inputs

HO = altitude of origin of coordinate systems (km)

MODE = 1 2 3 4 5
RXSE = GND RANGE(km) X(EAST, km) SLNT RANGE(km) ELEV.(deg) SLNT RANGE(km)
AYAA = AZIMUTH(deg) Y(NORTH, km) AZIMUTH(deg) AZIMUTH(deg) AZIMUTH(deg)
HZEH = ALTITUDE(km) Z(UP, km) ELEV.(deg) ALTITUDE(km) ALTITUDE(km)

Outputs

R, A, H = surface range, azimuth, altitude coordinates of point

X, Y, Z = rectangular tangent-plane coordinates of point

S, E = slant range, elevation coordinates of point

Listing

```
SUBROUTINE COORDX(HO,MODE,RXSE,AYAA,HZEH,R,A,H,X,Y,Z,S,E)
COMMON /CONST/RE,PI,RADIAN
REH=RE+HO
GO TO(11,21,31,41,51),MODE
51 U1=(HZEH+RE)**2+RE**2-RXSE**2
   U2=2.*RE*(HZEH+RE)
   RXSE=ACOS(U1/U2)*RE
11 R=RXSE
   A=AYAA
   H=HZEH
   U0=R/RE
   U2=A*RADIAN
   U3=RE+H
   U1=U3*SIN(U0)
   X=U1*SIN(U2)
   Y=U1*COS(U2)
   Z=U3*COS(U0)-REH
   IF(U0.EQ.0.) Z=H-HO
   IF(U1)23,25,23
21 X=RXSE
   Y=AYAA
   Z=HZEH
   U1=X**2+Y**2
   IF(U1)22,24,22
22 U2=REH+Z
   U3=SQRT(U1+U2**2)
   U1=SQRT(U1)
   R=ASIN(U1/U3)*RE
   IF(U2.LT.0.0) R=PI*RE-R
   A=ASIN(X/U1)/RADIAN
   IF(Y.LT.0.0) A=180.-A
```

```

      H=U3-RE
23  S=SQR(T(U1**2+Z**2)
      E=ASIN(Z/S)/RADIAN
      GO TO 99
24  R=0.
      A=0.
      H=HQ+Z
25  S=ABS(Z)
      E=90.
      IF(Z.LT.0.0) E=-90.
      GO TO 99
31  S=RXSE
      A=AYAA
      E=HZEH
      U2=A*RACIAN
      U3=E*RACIAN
      U1=S*COS(U3)
      X=U1*SIN(U2)
      Y=U1*COS(U2)
      Z=S*SIN(U3)
      U2=REH+Z
      U3=SQR(T(U1**2+U2**2)
      R=ASIN(U1/U3)*RE
      IF(U2.LT.0.0) R=PI*RE-R
      H=U3-RE
      GO TO 99
41  F=RXSE
      A=AYAA
      H=HZEH
      U2=A*RACIAN
      U3=E*RACIAN
      COSE=COS(U3)
      SINE=SIN(U3)
      S=HTOS(HQ,H,COSE,SINE)
      U1=S*COSE
      X=U1*SIN(U2)
      Y=U1*COS(U2)
      Z=S*SINE
      U3=RE+H
      R=ASIN(U1/U3)*RE
      IF(REH+Z.LT.0.0) R=PI*RE-R
99  RETURN
      END

```

RNCOT

This subroutine is listed and described in Refs. 4 and 5.

Listing

```
SUBROUTINE RNCOT  
END
```

SLCT

This subroutine reduces the number KEYT to a 1 digit representing its most significant digit. It then returns this digit to the plotting subroutine for appropriate action.

KEYT = number generated by RANC for target presently being observed (target calculation code)

Listing

```
SUBROUTINE SLCT(KEY)
  IF(KEY.LT.100) GO TO 10
  DO 1 I=1,9
  KEY=KEY-100
  IF(KEY.LT.100) I=9
1 CONTINUE
10 CONTINUE
  IF(KEY.LT.10) GO TO 99
  DO 2 I=1,9
  KEY=KEY-10
  IF(KEY.LT.10) I=9
2 CONTINUE
99 CONTINUE
  RETURN
  END
```

Appendix 2
Sample PMIV Output

1

10.0000000000

.0

.50000

.328

1.000

400.000

574.449

42.306

18.000

715.8468543

52.4561296

18.0000000

999.000

.0

20.000

20.500

527.000

1052253.058

1.054361

4.708

1 1 1

>>>FAACD

FFC	TCN (DB)	R (KM)	EX (KM)	V (KS)	SPRD (KS)
1 CE	.0	404.824	489.684	2.024	2.439

EPT (M2)	SC	SOLID ANGLE (STERAD)	RR (KS)
.0E+00	1.045E-02	6.499E-01	-2.439E+00

>>>>FNTO

FFC	N-T (K)	MAG AZ(DEG)	FB 1-W ABS EL(DEG)	DB	BS-ABS DB
1 CE	.0	18.000	52.456	.0	.0

FB-T (K)	D (CGS)	E-D (CGS)	ABS (DB/KM)
1.045E+04	.0E+00	2.358E+08	.0E+00

>>>>>FP

FFC	LOCATION RANGE(KM)	MAG AZ(DEG)	A	RADIUS T	RADIUS L
1 CE	.0	.0	581.286	350.611	358.140

AXIS TILT (RAD)	MIN A	MAX A
2.436E-01	2.780E+02	9.394E+02

SK. SPHEROID

>>>>>DP

DDC	LOCATION RANGE(KM)	MAG AZ(DEG)	A	RADIUS T	RADIUS L
1 IZ	.0	.0	581.286	281.381	289.943
1 ES	.0	.0	1023.060	365.817	365.817

FY	DISTRIBUTION PARAMETER	B-A
1.230E-01	8.000E+00	4.001E+02
4.100E-02	6.116E+00	1.023E+03

1	527	20	16	1.65	22	2	25
2	-15	350	7000	20	50	2	25
3	1	50	35	30	90	180	270
4	11	51	1	0	0	0	0
5	0	0	0	74	12		1
6	400	30	400	0	.5	.328	1 1
7	10	40	4				
8	30	999	20.5	20	5		0
9	1	1	1	2	38	38	40
0							

Appendix 3

PMIV Program (Listing)

PMIV

This program queries the user for a two-character ID to use as an identifier for the DD80 file produced. The user provides a box number to be used on all DD80 files produced. This box number is assigned when the program is compiled under the CHAT compiler.

Listing

```
PROGRAM PMIV(INPUT,TAPE5=INPUT,DOUT,TAPE7=DOUT,PLC1=PLCTSN
1,PLOTCL,TAPE9=PLOTCL,PLDTDF,TAPE1G=PLOTDF,PLOTPC,TAPE11=PLOTPC,
2TAPE59,TAPE100)
CALL CHANGE(5H+PMIV)
DIMENSION C1(2)
WRITE(59,401)
401 FORMAT(14HBOX NO. AND ID)
READ(59,402) C1
402 FORMAT(2A10)
CALL DD80ID(D1,1)
CALL KEEP80(1)
1 CONTINUE
READ(7,101) UNITHR,UNITVR
101 FORMAT(/ZF20.10)
IF(EOF,7) 4,5
5 CONTINUE
READ(7,102) UNITHO,UNITVO,UNITH,UNITV,NMXR,NPXR,NHR1,NHR,MAXNMX,
1CNA,AZREF,SROCRIT,DSNBOCR,THCLTR,DTHCLTR,THDIFF,DTHDIFF
102 FORMAT(2F20.10,/ZF20.10,/5I10,2X,F2.0,/F20.3,3(/ZF20.10))
CALL FRAME(1)
CALL SETCH(1.,128.,1,0,1,0,0)
WRITE(100,1C3) UNITHR,UNITVR,UNITHO,UNITVO,UNITH,UNITV,NMXR,NPXR,
1NHR1,NHR,MAXNMX,CNA,AZREF,SROCRIT,DSNBOCR,THCLTR,DTHCLTR,THDIFF,
2DTHDIFF
103 FORMAT(3(/ZF20.10),/5I10,2X,F2.0,/F20.3,3(/ZF20.10))
RDEN=UNITVR/UNITVO
AXNHR=INTF(85./(2.*RDEN))-1.*NHR-NHR1
UPR=.04
F1=(3.*(AXNHR*RDEN-1.-UPR))/128.
F2=F1-1.
CALL MAP(0.,1.,F2,F1,0.,1.,0.,1.)
DO 3 K=8,11
CALL FRAME(1)
IF(CNA.EQ.0.) CALL OVRLY(UNITHR,UNITVR,UNITVO,NMXR,NPXR,NHR1,NHR,
1MAXNMX)
CALL CCPIC1(K)
3 CONTINUE
K=7
CALL CCPIC1(K)
GO TO 1
4 CONTINUE
CALL FRAME(1)
K=5
CALL CCPIC1(K)
CALL PLCTE
CALL EXIT
END
```

```

SUBROUTINE DDPICT(NTAPE)
DIMENSION AA(85)
CALL SETCH(1.,128.,1,0,1,0,0)
6 CONTINUE
READ(NTAPE,101)AA
101 FORMAT(85A1)
IF(EOF,NTAPE) 4,5
5 CONTINUE
IF(AA(12).EQ.1HN) GO TO 4
IF(AA(1).EQ.1H1) CALL SETCH(1.,128.,1,0,1,0,0)
WRITE OUTPUT TAPE 100,101,AA
GO TO 6
4 RETURN
END
SUBROUTINE OVRLY(UNITHR,UNITVR,UNITVO,NMXR,NPXR,NHR1,NHR,MAXNMX)
DIMENSION X(52),Y(52)
CPF=.5
DYPROW=.0234375
CENTER=(MAXNMX+2.-CPF)/85.333
DEGPDIV=5.
IF(UNITVO.LT..9) DEGPDIV=8.
DEGPDIV=UNITVO*DEGPDIV
DEGV=NHR1*UNITVR
DEGH=MAXOF(NMXR,NPXR)*UNITHR
DEGMX=MIN1F(90.,MAX1F(DEGV,DEGH))
NCURVES=MAX1F(DEGMX/DEGPDIV,1.)
N=50
PI=3.14159265359
RADIAN=PI/180.
RO=DYPROW/(UNITVO*RADIAN)
BNDPY=(NHR-1)*UNITVR*RADIAN*RO
BNDMY=(NHR-NHR1)*UNITVR*RADIAN*RO
BNDMX=CENTER-NMXR*UNITHR*RADIAN*RO
BNDPX=CENTER+NPXR*UNITHR*RADIAN*RO
IF(BNDMY.EQ.0.) CALL LINE(0.,0.,1.,0.,1,0)
CALL LINE(BNDMX,BNDMY,BNDMX,BNDPY,1,682)
CALL LINE(BNDPX,BNDMY,BNDPX,BNDPY,1,682)
CALL LINE(BNDMX,BNDPY,BNDPX,BNDPY,1,682)
CALL LINE(BNDMX,BNDMY,BNDPX,BNDPY,1,682)
Y90=RO*90.*RADIAN
IF(Y90.LE.BNDPY) CALL LINE(0.,Y90,1.,Y90,1,0)
CALL SETCRT(0.,0.,1,1023)
PNC=NCURVES
PN=N
N1=N+1
DELTA=DEGMX/PN
THETA = (NHR-NHR1)*UNITVR
IB = THETA/DEGPDIV
IDRCT=1
GO TO 31
30 IDRCT=2
31 CONTINUE
DO 10 I=1,NCURVES
GO TO(11,12),IDRCT
11 CONTINUE
THETA = DEGPDIV*(I +IB)
PHI=-DELTA
GO TO 13
12 CONTINUE
PHI=DEGPDIV*I
THETA = -DELTA+(NHR-NHR1)*UNITVR
13 CONTINUE

```

```

DO 20 K=1,N1
GO TO(21,22),IDRCT
21 CONTINUE
PHI=PHI+DELTA
GO TO 23
22 CONTINUE
THETA=THETA+DELTA
23 CONTINUE
A=SINF(PHI*RADIAN)
B=SINF(THETA*RADIAN)
C=COSF(THETA*RADIAN)
THBAR=ASIN(A*C)
PHIBAR=ASIN(B/SQRTF(1.-(A**2)*(C**2)))
X(K)=CENTER-RO*THBAR
Y(K)=RO*PHIBAR
20 CONTINUE
CALL TRACE(X,Y,N1,1,1)
DO 50 K=1,N1
X(K)=-X(K)+2.*CENTER
50 CONTINUE
CALL TRACE(X,Y,N1,1,1)
IF((BNDPY/(RADIAN*RO)).LT.90.) GO TO 10
DO 51 K=1,N1
Y(K)=-Y(K)+2.*Y90
51 CONTINUE
CALL TRACE(X,Y,N1,1,1)
DO 52 K=1,N1
X(K)=-X(K)+2.*CENTER
52 CONTINUE
CALL TRACE(X,Y,N1,1,1)
10 CONTINUE
GO TO(30,32),IDRCT
32 CONTINUE
BNDMY=(NHR-NHR1)*UNITVR
CALL SETCH(1.,1.,1,0,1,0,0)
WRITE(100,102)BNDMY,DEGPDIV
102 FORMAT(1X,F12.5,F72.5)
CALL SETCHWO
RETURN
END

```

Appendix 4

DDTV Writeup

This routine, unique to the LLL computer system, is included for the sake of completeness for LLL users.

UTILITY ROUTINE

M-020

DDTV
UR-412
Page 1

DDTV

April 20, 1971

IDENTIFICATION

Purpose of Routine:	To selectively scan frames of dd80 files via TMDS.
Available on Machines:	CDC 6600, CDC 7600
Programmed by:	Peter R. Keller
Documented by:	Peter R. Keller
Memory Requirements:	CDC 6600 field length: $22074_8 (9276_{10})$ CDC 7600 FLS: $22014_8 (9228_{10})$ FLL: $22014_8 (9228_{10})$
References:	<ol style="list-style-type: none">1. CIC Report N3.8-001, <i>Introduction to the TMDS</i>, December 1970.2. Utility routine <i>FILES</i> (UR-307).3. Utility routine <i>SWITCH</i> (UR-301).4. Utility routine <i>GIVE</i> (UR-309).
Replacement Notice:	This document replaces Octopus Communique No. 296, <i>Controllee 'DDTV' is Now in File Public</i> , August 6, 1970.

TABLE OF CONTENTS

General Description	4
Options Available	4
Viewing Options	4
Print Option	4
Comparison Option	4
Definition of Terms	4
Summary of Usage Forms	5
Detailed Information	5
Gaining Access to dd80 Files	5
Instructions to DDTV	6
Comparing Frames from Different dd80 Files	7
Viewing a Picture which is on Two Frames of Two dd80 Files	7
How to Give a dd80 File for Printing	7
Restrictions	8
Family of dd80 Files	8
Proper 10-Character Dnames	8
Help Facilities	8
Error Indications	8
Examples of Usage	10
Instructions for the Casual User	10
To Compare Frames from Different dd80 Files	11
To View dd80 Files above Access Level P	12
Message Formats	13
TMDS Error Messages Sent by DDTV	13
Messages Originating in DDTV	14
Definitions of Special Characters	14

GENERAL DESCRIPTION

DDTV allows you to view a dd80 file one frame at a time and to compare frames from different dd80 files via the television monitor display system (TMDS). The dd80 files may also be output to the dd80 microfilmer and subsequently to the Xerox Copy-Flo for printing. This routine is especially helpful in debugging a dd80 program without having to wait for a printed copy of the output.

DDTV was written using the rules outlined in Ref. 1.

OPTIONS AVAILABLE

Viewing Options

There are two viewing options: white (green) picture on a black background or black on a white (green) background. (See Instructions to DDTV.)

Print Option

After viewing the dd80 file, you may print it by using the instruction GIVE.

Comparison Option

A frame from one dd80 file may be easily compared with a frame from another dd80 file. (See Detailed Information.)

DEFINITION OF TERMS

<i>monitor</i>	TMDS monitor number.
<i>filename</i>	Any name of a dd80 file.
<i>Dname</i>	Name of a dd80 file whose first character is the letter D, e.g., D1 and DXAAAAAAAA.
<i>. (period)</i>	Response by DDTV indicating it is waiting for a user instruction.
<i>instr</i>	Any valid user instruction to DDTV.
<i>resp</i>	DDTV's response to user's instruction.
<i>n</i>	Any integer.
<i>linefeed</i>	Depress the LINEFEED key on the TTY.

SUMMARY OF USAGE FORMS

User: DDTV monitor / t v
Routine: TYPE DD80 FILENAME
 • filename (User types on same line as period, but for clarity here, the user
User: instruction is shown on the next line, but indented appropriately.)
Routine: •

User: instr
Routine: resp
 •
 •
 •

User: instr
Routine: resp

User: END
Routine: ALL DONE

Note: A variation of the usage form is as follows:

User: DDTV / t v
Routine: MONITOR NO
 •

User: monitor
Routine: TYPE DD80 FILENAME
 •
 •
 •

(Then continue as above.)

DETAILED INFORMATION

Gaining Access to dd80 Files

Normally the user is not aware of any dd80 files; his dd80 program is executed and shortly thereafter the dd80 printout is available in the printout distribution room. Actually, when the dd80 program is executed, a dd80 file is created; this file contains the data necessary to print the output. The system then takes this file and prints it.

To gain access to your dd80 file, you must stop the system from printing it. To do this, insert the subroutine call CALL KEEP80(1) immediately after the call to CRTID in your dd80 program. For example:

```
CALL CRTID(2HXX,1,1)
CALL KEEP80(1)
```

Then compile and execute your program as usual. During execution, the system assigns *Dnames* as filenames for the dd80 files. (See Restrictions.)

To execute DDTV, you must know these *Dnames*. Use utility routine FILES (UR-307) to obtain them and then execute DDTV as shown in Examples of Usage.

Instructions to DDTV

INSTRUCTION ACTION

<i>linefeed</i>	Display the next frame. (The first frame is either the ID frame or blank; DDTV identifies a blank frame by displaying the word blank.)
END	End DDTV and release the monitor(s).
S <i>n</i>	Skip <i>n</i> frames and display the next frame.
S	Skip. DDTV then asks for <i>n</i> by typing: TYPE NO. OF FRAMES TO BE SKIPPED.
- <i>n</i>	Backup <i>n</i> frames. <i>n</i> must be space delimited from the minus sign.
GIVE	Give the dd80 file to the system to be printed. The <i>filename</i> must be in the proper format. (See Restrictions.)
• (period)	Send the same frame again. (Useful when a TMDS error resulted on the first try.)
ERASE or E	Erase the frame. However, the frame still remains in the buffer and may be sent again by typing a period.
ADD	Superimpose the next frame to be displayed on the frame currently displayed. You may superimpose as many frames as you like, but you must type ADD each time.
P	Change screen polarity. Reverses the screen background color of this frame and the following frames.
STATUS	Display the TMDS status frame. To return to the dd80 file frame, type STATUS again.
LINK <i>n</i>	Simultaneously displays your frames on another TMDS monitor = monitor <i>n</i> .
UNLINK <i>n</i>	Erase and release monitor <i>n</i> .
<i>Dname</i>	The name of a dd80 file beginning with D. This <i>instr</i> allows you to begin scanning a second dd80 file without having scanned the entire first dd80 file.
SRD	Display a SRD reminder on this frame and all following frames. This does not flag the file or cause any label to appear on the hardcopy. The SRD labels for the hardcopy are generated by the subroutine call CRTID in the dd80 program.
CONFIRM <i>nnnn</i>	<i>nnnn</i> is the four-digit confirm number displayed on the monitor you reserved. The confirm feature is intended for those users operating at an access level greater than P. If you do not confirm within 1 minute, your monitor will be released.
HELP	Display a help message.

Comparing Frames from Different dd80 Files

By using the *Dname* instruction, you can compare a frame from one dd80 file with a frame(s) from another dd80 file. It is not necessary to scan the entire dd80 file before viewing frames from the second dd80 file. To initiate the comparison, type the *Dname* of the second dd80 file whenever DDTV requests an instruction. (See Examples of Usage.)

If the filenames are not in a *Dname* format or the filenames are *Dnames* but awkward to type, you can first use utility routine SWITCH (UR-301) to change the filenames to short *Dnames*. However, if you plan to print the dd80 files, you must change their short *Dnames* to the proper 10-character *Dname* format. (See Restrictions.)

Viewing a Picture Which is on Two Frames of Two dd80 Files

A picture may begin in a frame of one dd80 file and be completed in a frame of the next dd80 file in the same family of files. At the completion of displaying the first part of the picture, type ADD; then type the *Dname* of the next file in the family, and then skip over the ID pages and blank frames to complete the picture on the monitor.

How to Give a dd80 File for Printing

By DDTV

You may give a dd80 file for printing any time after you have it open (i.e., viewing it), but before opening a second file. Just type GIVE, and DDTV sends only the file you have currently open. Note: the filename must be a proper 10-character *Dname* or the system will not accept it. (See Restrictions.)

Once a file has been given for printing, it cannot be viewed again without rerunning the dd80 program. Therefore, if you want to see the files after they have been given to the system, save a copy of them with utility routine COPY (UR-201) before using DDTV.

By Utility Routine GIVE

If your dd80 file was not printed while using DDTV, you can also give it for printing by using utility routine GIVE (UR-309). If the dd80 filename is not a proper *Dname*, you must switch the name before giving it. For example, suppose your dd80 filename is DD3D:

```
User:      SWITCH DD3D DXAAAAAAAA / 1 1
Routine:    ALL DONE

User:      GIVE DXAAAAAAAA 999999 / 1 1
Routine:    OK

User:      END
Routine:    ALL DONE
```

RESTRICTIONS

Family of dd80 Files

DD80 output is written in a family of *Dname* files: D0_____ is the first file in a family; D1_____ is the second file; etc. DX_____ is the last file in a family. If there is only one file in a family, its *Dname* is DX_____.

Proper 10-Character *Dnames*

Normally, when executing a dd80 program, proper 10-character *Dnames* are automatically assigned to dd80 files; then these files are accepted by the system (user-1) for printing. However, with CALL KEEP80(1) inserted in the dd80 program, the system does not assign proper 10-character *Dnames* if the program is run under Monitor or compiled with CHIP. Under monitor, 9-character *Dnames* are assigned; under CHIP, 7-character *Dnames*.

Note

The system will not accept any dd80 file for printing unless it has a proper 10-character *Dname*. Therefore you must switch an improper *filename* or *Dname* to a proper 10-character *Dname* before giving the file to the system for printing.

HELP FACILITIES

For assistance, type HELP any time DDTV expects an instruction i.e., after DDTV types a period. DDTV displays the help package on the TMDS monitor.

ERROR INDICATIONS

In addition to sending TMDS error messages (which include file transport error messages since TMDS uses file transport), DDTV also sends the following error indications:

CANT GIVE

DDTV is unable to give the dd80 file to the system to be printed; this usually means that the filename is in the wrong format. (See Restrictions.)

AGAIN..

DDTV receives an instruction from the user while waiting for a message from the PDP-6. The user instruction is ignored. You must wait until DDTV sends a period before sending instructions. If after a reasonable time no period is received, type ABORT. DDTV responds with TMDS ERROR 70 DID NOT GET MSG FROM PDP-6 and then a period. This returns control to you and you may continue.

CANNOT BACKUP

DDTV can not respond to the skip backward command; this usually means that there are not enough frames in the files to skip backward over.

(ding) **WHAT?**

DDTV does not recognize your instruction. Type another instruction.

ILLEGAL DD80 COMMAND

DDTV recognizes bad dd80 data; it sends this error message, and displays on the monitor the data it has converted up to this point.

EXAMPLES OF USAGE

Instructions for the Casual User

Comment

(1) Gain access to dd80 file by inserting subroutine call CALL KEEP80(1) immediately after CALL CRTID in your dd80 program.

(2) Execute utility routine FILES to determine *Dnames* of your dd80 files.

User: FILES / 1 1

Routine: 141520 D0BFOMFFLC
103100 DXBFOMFFLC
ALL DONE

If *Dnames* are not in 10-character format, see Restrictions.

(3) Execute DDTV.

User: DDTV 11 / 1 1

Routine: TYPE DD80 FILENAME

User ID No. appears on monitor No. 11.

User: D0BFOMFFLC

Routine: .

DDTV opens file D0BFOMFFLC.

User: *linefeed*

Routine: .

First frame of file appears, and DDTV ready for next instruction.

User: *linefeed*

Routine: .

The next frame appears. Continue typing *linefeed* to scan frames sequentially.

.

.

.

User: *linefeed*

Routine: .

User: GIVE

Routine: GAVE

File D0BFOMFFLC is given for printing.

Routine: TYPE DD80 FILENAME

.

DDTV ready for next file.

User: DXBFOMFFLC

Routine: .

DDTV opens file DXBFOMFFLC.

User: S 5

Routine: .

Five frames are skipped and frame 6 is displayed.

User: *linefeed*

Routine: .

Frame 7 is displayed. Continue typing *linefeed* to scan frames sequentially.

.

.

.

User: *linefeed*

Routine: .

User: GIVE

Routine: GAVE

File DXBFOMFFLC is given for printing.

TYPE DD80 FILENAME

.

User: END

Routine: ALL DONE

To Compare Frames from Different dd80 Files

- (1) For ease in typing, you may first change the *Dnames* to shorter *Dnames* with utility routine SWITCH before using DDTV. In this example, DØBFOMFFLC is changed to D1 and DXBFOMFFLC to D2.

- (2) Problem: Compare frame 5 of file D1 with frame 42 of file D2. The sequence is as follows:

User: DDTV 11 / 1 1

Routine: TYPE dd80 FILENAME User ID No. appears on monitor No. 11.

.

User: D1

Open file D1.

Routine: .

File D1 is open.

User: S 4

Skip 4 frames.

Routine: .

Frame 5 of file D1 appears.

User: D 2

Close file D1 and open file D2.

Routine: .

File D2 is open, but frame 5 of

D1 is still visible.

User: S 41

Skip 41 frames in file D2.

Routine: .

Frame 42 of file D2 appears.

User: D1

Close file D2 and open file D1.

Routine: .

File D1 is open, but frame 42

of file D2 is still visible.

User: S 4

Skip 4 frames in file D1.

Routine: .

Frame 5 of file D1 appears.

User: END

Routine: ALL DONE

- (3) Files are still in your list of files. To have them printed, first switch their names back to proper 10-character *Dnames* and then use utility routine GIVE to give them to user-1 (999999) for printing.

To View dd80 Files above Access Level P

User:	DDTV 23 / 1 1	
Routine:	TYPE DD80 FILENAME	User ID No. and confirm No. appear on monitor No. 23.
	.	
User:	DXBZHQMNA	
Routine:	.	
User:	CONFIRM 7123	You must respond with the confirm number within 1 minute or your monitor will be released.
Routine:	FOR SRD LABEL, TYPE:SRD	
	.	
User:	SRD	
Routine:	.	Flag is set to label all the frames that follow.
User:	S 3	User knows that first three frames are blank or ID and skips them. First picture in file appears.
Routine:	.	
User:	P	Change polarity of screen for next and all following frames.
Routine:	.	
User:	.	Send same frame. (Screen polarity is reversed.)
Routine:	.	
User:	GIVE	
Routine:	GAVE	File DXBZHQMNA is given for printing.
	TYPE DD80 FILENAME	
	.	
User:	END	
Routine:	ALL DONE	

TMDS Error Messages Sent by DDTV

```

TMDS ^ ERROR ^ 1 ^ NO ^ BUFFER ^ AVAILABLE ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 2 ^ TMDS ^ IS ^ OFF-LINE ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 3 ^ NO ^ CHANNEL ^ AVAILABLE ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 4 ^ TMDS ^ IS ^ HUNG ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 5 ^ SECURITY ^ LEVEL ^ PARITY ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 10 ^ BAD ^ BIT ^ COUNT ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 11 ^ (H) ^ WORKER ^ INTERFACE ^ STOPPED ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 14 ^ NO ^ SUCH ^ FILE ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 15 ^ (X) ^ ZERO ^ COUNT ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 16 ^ ILLEGAL ^ WORKER ^ FWA ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 17 ^ (X) ^ COUNT ^ CK ^ ERROR ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 21 ^ (H) ^ WORKER ^ DISK ^ TRANS ^ ERROR ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 24 ^ (H) ^ WORKER ^ INTERFACE ^ IS ^ DOWN ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 25 ^ (H) ^ TRANSPORT ^ INCOMPLETE ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 26 ^ (X) ^ INSUFFICIENT ^ FUNDS ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 27 ^ (X) ^ FILE ^ INFO ^ ERROR ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 30 ^ DECLASSIFICATION ^ NOT ^ PERMITTED ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 32 ^ (X) ^ FALSE ^ ERROR ^ NUMBER ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 70 ^ DID ^ NOT ^ GET ^ MSG ^ FROM ^ PDP-6 ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 71 ^ CANT ^ OPEN ^ USER ^ TMDS ^ FILE ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 72 ^ CANT ^ CLOSE ^ FILE ^ JUST ^ OPENED ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 73 ^ CANT ^ MAKE ^ TMDS ^ BLOCK ^ FILE ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 74 ^ CANT ^ CLOSE ^ FILE ^ TO ^ SEND ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 75 ^ CANT ^ OPEN ^ FILE ^ TO ^ DESTROY ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 76 ^ CANT ^ DESTROY ^ FILE ^ OPENED ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 77 ^ ERROR ^ SENDING ^ MSG ^ TO ^ PDP-6 ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 100 ^ CAN ^ NOT ^ RESERVE ^ DISPLAY ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 101 ^ CAN ^ NOT ^ LINK ^ DISPLAY ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 102 ^ CAN ^ NOT ^ UNLINK ^ DISPLAY ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 200 ^ TOO ^ MUCH ^ DATA ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 201 ^ ILLEGAL ^ MON ^ # ^ IN ^ COMMAND ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 202 ^ ILLEGAL ^ MESSAGE ^ HEADER ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 203 ^ INCOMPLETE ^ MESSAGE ^ GROUP ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 204 ^ TEXT ^ WORD ^ COUNT ^ IS ^ ZERO ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 205 ^ ILLEGAL ^ COMMAND ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 206 ^ NO ^ RESERVE ^ COMMAND ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 207 ^ BAD ^ LIST ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 300 ^ SECURITY ^ LEVEL ^ TOO ^ HIGH ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 301 ^ CANT ^ RAISE ^ SECURITY ^ LEVEL ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ 302 ^ NO ^ CONFIRMATION ^ 7 -> 7 Δ$
TMDS ^ ERROR ^ UNKNOWN ^ 7 -> 7 Δ$

```

Messages Originating in DDTV

MONITOR ^ NO ^→→ . ^Δ \$
 TYPE ^ DD80 ^ FILENAME ^→→ . ^Δ \$
 TYPE ^ NO ^ OF ^ FRAMES ^ TO ^ BE ^ SKIPPED ^→→ . ^Δ \$
 . ^Δ \$
 GAVE ^→→ ^Δ \$
 CANT ^ GIVE ^→→ ^Δ \$
 CANNOT ^ BACKUP ^→→ ^Δ \$
 FOR ^ SRD ^ LABEL, ^ TYPE: ^ SRD ^→→ ^Δ \$
 ILLEGAL ^ DD80 ^ COMMAND ^→→ ^Δ \$
 ^Δ 'WHAT? ^→→ ^Δ \$

Definitions of Special Characters

The special characters used above are defined as follows:

<u>Character</u>	<u>ASCII code</u>	<u>Card punch combination</u>
^ (space bar)	00	Blank
^	40	0,6,8
→	77	12,5,8
Δ	02	11,7,8
\$	04	11,3,8
'	07	4,8

Appendix 5

PONDER Code Refractive Option (Listing)

Note that the subroutines in Appendices 1, 5, and 6 have the same names. Subtle differences may exist in some of the routines; therefore, these routines should not be interchanged but should be used only with the PONDER options indicated.

PONDER

CONA = mode switch
GLATØ = latitude of origin
GLONØ = longitude of origin
HHR = altitude of origin
XRØ = (east) x coordinate of radar with respect to origin
YRØ = (north) y coordinate of radar with respect to origin
ZRØ = (vert) z coordinate of radar with respect to origin
ARØ = azimuth coordinate of radar with respect to origin
AROSP = radar heading
ITOT = total number of targets used to produce PONDER picture

Listing

```
PROGRAM PONDER(INPUT,TAPE5=INPUT,COMMENT,TAPE6=COMMENT,TRIG,
1TAPE7=TRIG,PLOTTA,TAPE8=PLOTTA,PLOTEL,TAPE9=PLOTEL,PLOTAZ,
2TAPE10=PLOTAZ,PLOTSN,TAPE11=PLOTSN,PLOTRE,TAPE12=PLOTRE,PLOTRM,
3TAPE13=PLOTRM)
CALL CHANGE(TH+PONDER)
ILEN=10000
CALL CREATE(7HCOMMENT,ILEN,M1WD)
ILEN=10000
CALL CREATE(4HTRIG,ILEN,M1WD)
ILEN=24000
CALL CREATE(6HPLOTTA,ILEN,M1WD)
ILEN=14000
CALL CREATE(6HPLOTEL,ILEN,M1WD)
ILEN=14000
CALL CREATE(6HPLOTAZ,ILEN,M1WD)
ILEN=14000
CALL CREATE(6HPLOTSN,ILEN,M1WD)
ILEN=14000
CALL CREATE(6HPLOTRE,ILEN,M1WD)
ILEN=14000
CALL CREATE(6HPLOTRM,ILEN,M1WD)
COMMON /EXEC/CLTR(50),SRCL(50),DSRCL(50),VC(50),DVC(50),
1SIGMAC(50),ETA(50),FBNE(50),FBD(50),FBRHD(50),
2KEYC(50),TANT(50),AZFB(50),ELFB(50),DBTDFB(50),
3DBS(50),OMEGAB(50),KEYN(50),ACLTR,SRAC,DSRAC,VAC,
4DVAC,ASIGMA,ETAA,ENEA,ADBPKM,KEYA,AZAC,ELAC,OBAC,
5TASUM
COMMON /BURST/NB,W(50),TSX(50),FF(50),HB(50),XBR(50),YBR(50),
1ZBR(50),TB(50),COSCLB(50),RDZERO(50),RHZERO(50),
2RUZERO(50),LHVB(50),RHO(50),HSB(50),PKTERM(50),
3RLK(50),RTK(50),DISTK(50),XKR(50),YKR(50),ZKR(50),
4TILTK(50),WUV(3,50),HR,COSCLR,SINCLR,KALCH,KALCN,
5KALCR,DAY,CIRAX,TEM150,NSW1,NSW2,NSW3,NSW4,NSW5,NSW6
COMMON/REGION/NF,INDXF(50),FLABL(50),KINDF(50),RFD(50),AFB(50),
1HF(50),XFR(50),YFR(50),ZFR(50),RLF(50),RTF(50),
2TILTF(50),HMINF(50),HMAXF(50),TFO(50),TF(50),
3RHDF(50),HSNF(50),HDDT(50),RDOT(50)
COMMON /RADAR/FREQ,POWER,DELTAR,DELTAV,TAG,TBW,TSL,SNM0B,TSYS,BWN,
```

```

1      PCOMP,RAG,RBW,RSL,KSPEC,SCAN,EFACE,AFACEM(4),NFACE
COMMON/TARGET/RCS,STR,ATR,ETR,TSEC(1),SIGNAL,SNCTRC,DBTAR,AER,EER,
1RER,DISP,KEYT,RMISS,EAR,KOUT,LFOUT
COMMON/RNCIN/PLDEN,RDEN,NMXR,NPXR,NHR1,NHR,UNITH,UNITV,AZRF,RTG,
1MAXNMX,NCRCT,VARR,SBOCRIT,DSNBOCR
COMMON/CONST/RE,P1,RADIAN
COMMON/RNC/ICONF
DIMENSION SNPRES(500),SNPREV(500),EARS(500),EERS(500),AERS(500),
1RERS(500),RMIS(500),KYT(500),S(150)
ICONF=1
GO TO 3
1 ICCNF=2
3 CALL RANC
ICONF=3
IAREA=0
AREA1=0.
AREA0=0.
MXTANT=0.
MXDBTF=0.
MXDBS=0.
MXTF=0.
MXFBR=0.
MXFBE=0.
MXFBD=0.
BOCR=.0005
SNBOCR=SBOCRIT+DSNBOCR
AZREF=AZRF-VARR
UNITV=PLDEN*UNITV
UNITH=PLDEN*UNITH
UNITVR=RDEN*UNITV0
UNITHR=RDEN*UNITH0
WRITE(8,3003)UNITHR,UNITVR,UNITH0,UNITV0,UNITH,UNITV,NMXR,NPXR,
1NHR1,NHR,MAXNMX,AZREF,SBOCRIT,DSNBOCR
IPLDEN=PLDEN
IRDEN=RDEN
IRP=IRDEN*IPLDEN
NMXIRD=NMXR*IRDEN
NPXIRD=NPXR*IRDEN
MAXNRV=(NMXR+NPXR)*IRP+1
AXNHR=INTF(85./(2.*RDEN))-1.
NHRM0=(AXNHR-NHR1)*RDEN
IF(NHRM0.LE.0) GO TO 8401
S(1)=1H
DO 8402 I1=8,13
DO 8400 I=1,NHRM0
WRITE(11,3001)S(1)
8400 CONTINUE
8402 CONTINUE
8401 CONTINUE
DO 8100 I=1,MAXNRV
SNPRES(I)=0.
8100 SNPREV(I)=0.
IPL0T=2
IRUN=2
NPRES=0
NLPRES=0
NRPRES=0
ELREF=NHR*UNITVR
DO 6000 I1=1,NHR1
GO TO(7001,7002),IRUN
7002 GO TO(7701,7702),IPL0T
7702 CONTINUE
IPL0T=1

```

```

IRUN=1
NMX=NMXR
NPX=NPXR
UNH=UNITHR
7001 CONTINUE
ELREF=ELREF-UNITVR
IO=1
GO TO 7003
7701 CONTINUE
ELR=ELREF
UNH=UNITH
IO=IRP
7003 DO 6003 I2=1, IO
GO TO(7006,7005), IRUN
7005 ELREF=ELR+(IRP-I2)*UNITV
ILNCT=0
IRNCT=0
7006 CONTINUE
IL=0
IR=0
ILEFT=1
NR=NMX+1
NRV=NMX+NPX+1
NRVPREV=0
7703 CONTINUE
DO 309 ITRGT=1, NRV
GO TO(5001,5002,5003), ILEFT
5001 CONTINUE
NR=NR-1
GO TO 5004
5002 CONTINUE
NR=-NPX
ILEFT=3
GO TO 5004
5003 CONTINUE
NR=NR+1
5004 CONTINUE
CALL SPRAY(RTG, UNH, NR, ELREF, AZREF, ATR, STR, ETR)
CALL RANC
DO 8800 INB=1, NF
MXTANT=MAX1F(MXTANT, TANT(INB))
MXDBTF=MAX1F(MXDBTF, DBTOFB(INB))
MXDBS=MAX1F(MXDBS, DBS(INB))
MXTF=MAX1F(MXTF, TF(INB))
MXFBR=MAX1F(MXFBR, FBRHO(INB))
MXFBE=MAX1F(MXFBE, FBNE(INB))
MXFBD=MAX1F(MXFBD, FBDB(INB))
8800 CONTINUE
GO TO(5005,308), IRUN
5005 CONTINUE
IF((RMIS-80CR).GE.0.) GO TO 5006
IF((SIGNAL-SN80CR).GE.0.) GO TO 309
5006 GO TO(5007,5008,5008), ILEFT
5007 ILEFT=2
IL=ITRGT
GO TO 309
5008 ITRGT=NRV
IR=NPX+NR+1
GO TO 309
308 CONTINUE
SNPRES(ITRGT+NRVPREV)=SIGNAL
EARS(ITRGT+NRVPREV)=EAR

```



```

EERS(ITRGT+NRVPREV)=EER
AERS(ITRGT+NRVPREV)=AER
RERS(ITRGT+NRVPREV)=RER
RMIS(ITRGT+NRVPREV)=RMIS
KYT(ITRGT+NRVPREV)=KEYT
309 CONTINUE
GO TO(7007,7008),IRUN
7007 CONTINUE
IF((IL.GT.0).AND.(IR.EQ.0)) IR=NRV+1-IL
CALL FIT(NPRES,NLPRES,NKPRES,IRUN,NMXR,NPXR,NMXO,NPXO,NMX,NPX,
IPLDEN,IRDEN,MAXNMX,IL,IR)
GO TO 6003
7008 CONTINUE
IF(NCRCT.EQ.0) GO TO 9003
IF(IRNCT.EQ.1) GO TO 7801
IF(ILNCT.EQ.1) GO TO 7802
ILCT=0
IF((RMIS(1)-BOCR).GE.0.) ILCT=1
IF((SNPRES(1)-SNBOCR).LT.0.) ILCT=1
IRCT=0
IF((RMIS(NRV)-BOCR).GE.0.) IRCT=1
IF((SNPRES(NRV)-SNBOCR).LT.0.) IRCT=1
IF((ILCT.EQ.0).AND.(IRCT.EQ.0)) GO TO 9003
CALL CRCT(NMXIRD,NPXIRD,NMXO,NPXO,NMX,NPX,NPXRI,IPLDEN,NCRCT,ILCT,
IIRCT,NRVL,NRVR,NRL,NRR)
ILNCT=1
IF(NRVL.GT.0) GO TO 7803
7804 CONTINUE
IRNCT=1
IF(NRVR.GT.0) GO TO 7805
7806 CONTINUE
ILCT=0
IF((RMIS(1)-BOCR).GE.0.) ILCT=1
IF((SNPRES(1)-SNBOCR).LT.0.) ILCT=1
IRCT=0
IF((RMIS(NRV)-BOCR).GE.0.) IRCT=1
IF((SNPRES(NRV)-SNBOCR).LT.0.) IRCT=1
CALL ANALYZE(NRVL,NRVR,ILCT,IRCT,ELREF)
GO TO 9003
7801 CONTINUE
NRV=NMX+NPX+1
GO TO 7806
7802 CONTINUE
NRV=NMX+NPX+1
GO TO 7804
7803 CONTINUE
NRVPREV=NRV
NRV=NRVL
NR=NRL
DO 7900 I=1,NRVPREV
SNPRES(NRVPREV+NRV+1-I)=SNPRES(NRVPREV+1-I)
EARS(NRVPREV+NRV+1-I)=EARS(NRVPREV+1-I)
EERS(NRVPREV+NRV+1-I)=EERS(NRVPREV+1-I)
AERS(NRVPREV+NRV+1-I)=AERS(NRVPREV+1-I)
RERS(NRVPREV+NRV+1-I)=RERS(NRVPREV+1-I)
RMIS(NRVPREV+NRV+1-I)=RMIS(NRVPREV+1-I)
KYT(NRVPREV+NRV+1-I)=KYT(NRVPREV+1-I)
7900 CONTINUE
NRVPREV=0
GO TO 7703
7805 CONTINUE
NRVPREV=NRV
NRV=NRVR

```

```

NR=NRN
NPX=NPX#I
GO TO 7703
9003 I3=MOD(I2,IPLDEN)
IF(I3)8001,8000,8001
8000 CONTINUE
KTAPE=8
CALL PLOT(EARS,KYT,.01,NPXO,NMXO,0,0,MAXNMX,IPLDEN,KTAPE)
KTAPE=9
CALL PLOT(EERS,KYT,.01,NPXO,NMXO,0,0,MAXNMX,IPLDEN,KTAPE)
KTAPE=10
CALL PLOT(AERS,KYT,.01,NPXO,NMXO,0,0,MAXNMX,IPLDEN,KTAPE)
CALL PLOTSNISNPRES,KYT,SNBOCR,CSNBOCR,NPXO,NMXO,0,0,MAXNMX,IPLDEN)
KTAPE=12
CALL PLOT(RERS,KYT,NPXO,NMXO,0,0,MAXNMX,IPLDEN,KTAPE,STR)
KTAPE=13
CALL PLOT(RMIS,KYT,NPXO,NMXO,0,0,MAXNMX,IPLDEN,KTAPE,STR)
IPLDT=2
8001 CALL AREA(SNPRES,SNPREV,SBOCRIT,UNITH,UNITV,NMX,NPX,NRV,MAXNRV,
IAREA,AREA1,AREAD)
6003 CONTINUE
IF((IRUN.EQ.2).AND.(IPLDT.EQ.1)) GO TO 7701
6000 CONTINUE
JO=MAXNMX+1
DO 9100 J=1,JO
9100 S(J)=IH
J1=JO+1
S(J1)=IH+
WRITE(8,3001) (S(J),J=1,J1)
WRITE(8,3010)
WRITE(9,3001) (S(J),J=1,J1)
WRITE(9,3011)
WRITE(10,3001) (S(J),J=1,J1)
WRITE(10,3012)
WRITE(11,3001) (S(J),J=1,J1)
WRITE(11,3013)
WRITE(12,3001) (S(J),J=1,J1)
WRITE(12,3014)
WRITE(13,3001) (S(J),J=1,J1)
WRITE(13,3015)
AREAAB=(AREA1+AREAD)/2.
AA=AREAAB*(CT**2)
IF(AREA1)3501,3502,3501
3502 APPERR=100.
GO TO 3507
3501 APPERR=(AREAD/AREAAB-1.)*100.
3507 AI=AREA1*(STR**2)
AD=AREAD*(STR**2)
GCDH=STR*UNITHO*RADIAN
WRITE OUTPUT TAPE 8,3005,TSEC(1)
DO 3503 I=1,NB
IF(TB(I)-TSEC(1))3504,3504,3503
3504 CONTINUE
CALL CORDX(0.,2,XBR(I),YBR(I),ZBR(I),X1,ABR,X2,X3,X4,X5,
ISBR,EBR)
WRITE OUTPUT TAPE 8,3006,TB(1),W(1),FF(1),TSX(1),HB(1),SBR,EBR,ABR
3503 CONTINUE
WRITE OUTPUT TAPE 8,3007,STR,VTR,SBOCRIT,RCS,FREQ
WRITE OUTPUT TAPE 8,3008,AA,AREAAB,APPERR
TANT(1)=MXTANT
DBTOFB(1)=MXDBTF
DBS(1)=MXDBS

```

```

TF(1)=MXTF
FBRHO(1)=MXFBR
FBNE(1)=MXFBE
FBOB(1)=MXFBOB
CALL RNCOT
DO 8403 I=8,13
WRITE(I,3009)
8403 CONTINUE
GO TO 1
3001 FORMAT(120A1)
3003 FORMAT(31/2F20.10),/5110,/F20.3,/2F20.10)
3005 FORMAT(1H1,F51.3)
3006 FORMAT(F52.3,/F57.5,/F60.3,/F56.3,/F57.3,/F57.3,/F52.3,/F55.3)
3007 FORMAT(F57.3,/F53.3,/F57.3,/F46.3,/F56.3)
3008 FORMAT(F54.3,/F54.6,/F41.3)
3009 FORMAT(3HEND)
3010 FORMAT(40X,8HTANG(1V))
3011 FORMAT(40X,7HEER(1V))
3012 FORMAT(40X,7HAER(1V))
3013 FORMAT(41X,6HSN(1V))
3014 FORMAT(40X,7HRER(1V))
3015 FORMAT(39X,9HRMISS(1V))
END

```

RANC

This is an excerpt from the RANC routine, showing the modifications needed to run PONDER.

Listing

```
SUBROUTINE RANC
COMMON /EXEC/CLTR(50),SRCL(50),DSRCL(50),VC(50),DVC(50),
1      SIGMAC(50),ETA(50),FBNE(50),FBD(50),FBRHO(50),
2      KEYC(50),TANT(50),AZFB(50),ELFD(50),DBTOFB(50),
3      DBS(50),OMEGAB(50),KEYN(50),ACLTR,SRAC,DSRAC,VAC,
4      DVAC,ASIGMA,ETAA,ENEA,ADBPKM,KEYA,AZAC,ELAC,DBAC,
5      TASUM
COMMON /RADAR/FREQ,POWER,DELTAR,DELTAV,TAG,TBW,TSL,SNMDB,TSYS,BWN,
1      PCOMP,RAG,RBW,RSL,KSPEC,SCAN,EFACE,AFACEM(4),NFACE
COMMON/TARGET/RCS,STR(1),ATR(1),ETR(1),TSEC(1),SIGNAL(1),SNCTRC(1)
1,DBTAR(1),AER(1),EER(1),RER(1),DISP(1),KEYT(1),RMISS(1),EAR(1),
2,KQOUT,LQOUT
COMMON /BURST/NB,W(50),TSX(50),FF(50),HB(50),XBR(50),YBR(50),
1      ZBR(50),TB(50),CUSCLB(50),RDZERO(50),RHZERO(50),
2      RUZERO(50),LHVB(50),RHOB(50),HSB(50),PKTERM(50),
3      RLK(50),RTK(50),DISTK(50),XKR(50),YKR(50),ZKR(50),
4      TILTK(50),WUV(3,50),HR,COSCLR,SINCLR,KALCH,KALCN,
5      KALCR,DAY,CIRAX,TEM150,NSW1,NSW2,NSW3,NSW4,NSW5,NSW6
COMMON/REGION/NF,INDXF(50),FLABL(50),KINDF(50),RFB(50),AFB(50),
1      HF(50),XFR(50),YFR(50),ZFR(50),RLF(50),RTF(50),
2      TILTF(50),HMINF(50),HMAXF(50),TF(50),TF(50),
3      RHOF(50),HSNF(50),HDOF(50),RDOF(50)
COMMON/DEBRIS/ND,INDXD(50),DLABL(50),RDB(50),ADB(50),HD(50),
1      XDR(50),YDR(50),ZDR(50),RLO(50),RTO(50),CUSCLD(50),
2      DISTD(50),WD(50),PD(50),RHOD(50),HSD(50),HBETA(50)
COMMON /PARAM/ALPHAR,GAMMA,UKCEA,UN,UBETA,UGAMMA,N1,N2,DMAX,HBMAX,
1      DHSTEP,RSTOP,DBSTOP,DEL,EDMIN,ENMINF,KINT,DENRAT,
2      FTRAIN,HTRAIN
COMMON/CNEXEC/GXMTR,BXMTR,SMXTR,KXMTR,GRCVR,BRCVR,SRGVR,KRCVR,
1      GTM,GRM,GM,COSCAN,PHIB,TAMB,NFB,BEAM,SNTERM
COMMON /CONST/RE,PI,RADIAN
COMMON/RNC/ICONF
DIMENSION INEXT(50)
J=1
GO TO (B1,1,80),ICONF
81 CONTINUE
RE = 6370.
PI = 3.14159265
RADIAN = 1.74532925E-2
ALPHAR = 1.E-12
GAMMA = 1.5
UKCEA = 1.E4
UN = 0.015
UBETA = 2.0
UGAMMA = 0.025
N1 = 3
N2 = 3
DMAX = 100.
HBMAX = 200.
DHSTEP = 2.5
RSTOP=0.01
```

```

DBSTOP=0.1
DEL = 0.001
EDMING = 5.4E-9
ENMINF = 5.E4
KINT=1
DENRAT=2.
FTRAIN=1.
HTRAIN=1.
KASE=0
KSPEC=0
NFACE=0
NB=0
BOLTZ=1.3805E-20
C=2.998E+9
DBCON=4.34294482
FOURPI=4.*PI
TAMB=250.
HACHIN=90.
HACMAX=130.
RHOAC=1.E-10
1 CALL RANCIN(KASE)
  IF(POWER)2,730,2
2  SNMIN=10.**((SNMDB/10.))
  GXMTR=10.**((TAG/10.))
  BXMTR=TEW*RACIAN
  SXMTR=10.**((-TSL/10.))
  KXMTR=MCD(KSPEC,100)/10
  GRCVR=10.**((RAG/10.))
  BRCVR=REW*RACIAN
  SRCVR=10.**((-RSL/10.))
  KRCVR=MCD(KSPEC,10)
  DELRC=1.E-3*DELTAR
  DELRU=DELRC*PCOMP
  SNTERM=POWER*(C/FREQ)**2*PCOMP/(FOURPI)**3
  IF(NB)11,20,11
11 DO 12 I=1,NB
12 COSCLR(I)=(COSCLR*(RE+HR+ZBR(I))+SINCLR*YBR(I))/(RE+HB(I))
20 CONTINUE
  JTIME=1
  TIME=TSEC(JTIME)
  CALL PHENOM(TSEC,JTIME)
  RETURN
80 CONTINUE
  AER(J)=1000./RADIAN
  EER(J)=1000./RACIAN
  FAR(J)=1000./RACIAN
  REK(J)=10000./STR(J)
  RMIS(J)=10000./STR(J)
  TASUM=TSYS
  AZBEAM=ATR(J)
  ELBEAM=ETR(J)

```

GRID

This subroutine is used as an automatic searching routine. When PONDER is run in its auto mode ($UNITH \leq 0$), Grid will compute the limits of the search grid based on the fireball and debris dimension, which is used by PONDER to produce a picture enclosing this total region. To perform this function, Grid normally need only be presented PLDEN and RDEN plus the auto run control values from the number 9 card. All other variables related to PONDER picture production are internally generated.

Listing

For this listing, see Appendix 1.

SPRAY

This subroutine produces the target matrix point by point, left to right, row by row. Each target point produced by SPRAY is then checked by RANC to find its S/N.

RTG = target shell radius
UNH = grid horizontal increment either
NR = position along horizontal line at which target is now located
ELREF = elevation angle (degrees cona = 0, km cona = 2) set to topmost row of
PCONDER picture and decremented by the vertical print grid increment until
desired region of sky plotted
AZREF = azimuth from radar to grid center column (for case cona = 0)
VARR = magnetic variation
ATR = azimuth radar to target
STR = slant range radar to target
ETR = elevation angle to target relative to radar

Listing

```
SUBROUTINE SPRAY(RTG,UNH,NR,ELREF,AZREF,ATR,STR,ETR)
PI=3.14159265359
RADIAN=PI/180.
RN=NR
PHIO=AZREF
THETA0=ELREF
B=SINF(THETA0*RADIAN)
IF(THETA0-90.)/1,2,3
1 THEBAR=RN*UNH
5 A=COSE(THEBAR*RADIAN)
C=SINF(THEBAR*RADIAN)
THETA=ASINF(A*B)/RADIAN
ALPHA=ASINFC/SQRTF(1.-(A**2)*(B**2))/RADIAN
PHI=PHIO-ALPHA
GO TO 4
2 THEBAR=RN*UNH
THETA=90.+SIGNF(-1.,RN)*THEBAR
PHI=PHIO+SIGNF(1.,RN)*90.
GO TO 4
3 PHIO=PHIO+180.
THEBAR=-RN*UNH
GO TO 5
4 ETR=THETA
ATR=PHI
STR=RTG
RETURN
END
```

RANCIN

This subroutine reads the input for PONDER calculations.

Listing

```
SUBROUTINE RANCIN(KASE)
COMMON /RADAR/FREQ,POWER,DELTAR,DELTAV,TAG,TBW,TSL,SNMDB,TSYS,BWN,
1      PCOMP,RAG,RBW,RSL,KSPEC,SCAN,EFACE,AFACEM(4),NFACE
COMMON/TARGET/RCS,STR(1),ATR(1),ETR(1),TSEC(1),SIGNAL(1),SNCTRC(1)
1,DBTAR(1),AER(1),EER(1),RER(1),DISP(1),KEYT(1),RMISS(1),EAR(1),
2KOUT,LFCUT
COMMON /BURST/NB,M(50),TSX(50),FF(50),HB(50),XBR(50),YBR(50),
1      ZBR(50),TB(50),COSCLB(50),RDZERO(50),RHZERO(50),
2      RUZERO(50),LHVB(50),RHUB(50),HSB(50),PKTERM(50),
3      RLK(50),RTK(50),LISTK(50),XKR(50),YKR(50),ZKR(50),
4      TILTK(50),WUV(3,50),HR,COSCLR,SINCLR,KALCH,KALCN,
5      KALCR,DAY,CIRAX,TEM150,NSW1,NSW2,NSW3,NSW4,NSW5,NSW6
COMMON /CONST/RE,PI,RADIAN
COMMON/RNCIN/PLDEN,RDEN,NMXR,NPXR,NHR1,NHR,UNITH,UNITV,AZKF,RTG,
1MAXNMX,NCRCT,VARR,SBOCRIT,DSNBOCR
DIMENSION X(7),AFACE(4)
DIMENSION RBD(50),ABD(50),XBD(50),YBD(50),ZBD(50),SBD(50),EBD(50),
1GLATB(50),GLONB(50),KLB(50),ALPHA6(50)
DIMENSION RYD(1),ATO(1),HT(1),XTO(1),YTO(1),ZTO(1),STO(1),ETO(1),
1GLATT(1),GLONT(1),KLT(1),ALPHA7(1)
LOGICAL ENFIL
LFIN=5
LFOUT=6
GO=980.0
RGAS=8.3143E+7
NHLP=0
IN6=0
NRV=0
NT=1
1 READ(LFIN,1001) NCARD,(X(I),I=1,7),KODE,ALPHA
IF(EOF,5) 4,5
4 CALL EXIT
5 IF(NCARD)2,101,2
2 GO TO (11,21,31,41,51,61,71,81,91),NCARD
11 FREQ=X(1)
POWER=X(2)
DELTAR=X(3)
DELTAV=X(4)
TAG=X(5)
TBW=X(6)
TSL=X(7)
ALPHA1=ALPHA
IF(KODE)12,1,12
12 CALL DATAIN(KODE,AFACE)
ALPHA2=ALPHA
GO TO 1
21 SNMDB=X(1)
TSYS=AMAX1(X(2),1.)
BWN=AMAX1(X(3),1.E-3)
PCOMP=X(4)
RAG=X(5)
RBW=X(6)
```



```

RSL=X(7)
ALPHA2=ALPHA
GO TO 1
31 KSPEC=X(1)
SCAN=X(2)
EFACE=X(3)
AFACE(1)=X(4)
NFACE=1
DO 33 I=5,7
IF(X(1))32,33,32
32 NFACE=NFACE+1
AFACE(NFACE)=X(1)
33 CONTINUE
IF(SCAN)34,34,1
34 NFACE=0
GO TO 1
41 GLATO=X(1)
GLGNO=X(2)
DAY=X(3)
KALC=X(4)
KALCH=MAXO(1-MOD(KALC,1000)/100,0)
KALCN=MAXO(1-MOD(KALC,100)/10,0)
KALCR=MAXO(1-MOD(KALC,10),0)
KOUT=X(5)
KTRACK=MIN1(ABS(X(6)),10.)
KODE=X(7)
NSW1=MOD(KODE,1000000)/100000
NSW2=MOD(KODE,100000)/10000
NSW3=MOD(KODE,10000)/1000
NSW4=MOD(KODE,1000)/100
NSW5=MOD(KODE,100)/10
NSW6=MOD(KODE,10)
IF(NSW6.EQ.1) NSW5=1
IF(NSW5.EQ.1) NSW4=1
IF(NSW4.EQ.1) NSW3=1
GO TO 1
51 IF(KODE)52,53,52
52 CALL CDCRDX(0.,KODE,X(1),X(2),X(3),RRQ,ARQ,HR,XRO,YRO,ZRO,SRO,EKO)
GO TO 54
53 GLATR=X(1)
GLCNR=X(2)
HR=X(3)
54 KLR=KODE
DIPR=X(4)
VARR=X(5)
ALPHA5=ALPHA
GO TO 1
61 IF(IN6)62,65,62
62 IF(IN6-50)63,69,63
63 IF(X(4)-TB(NB))64,65,65
64 NHELP=63
65 IN6=IN6+1
NB=IN6
IF(KODE)66,67,66
66 CALL CDCRDX(0.,KODE,X(1),X(2),X(3),RBO(NB),ABO(NB),H8(NB),XBO(NB),
1YBC(NB),ZBO(NB),SBO(NB),EBO(NB))
GO TO 68
67 GLATB(NB)=X(1)
GLONB(NB)=X(2)
HB(NB)=X(3)
68 KLB(NB)=KODE
TB(NB)=X(4)
W(NB)=X(5)

```

```

FF(NB)=X(6)
TSX(NB)=X(7)
ALPHA6(NB)=ALPHA
GO TO 1
69 NHLP=62
GO TO 1
71 TSEC(NT)=X(1)
AXNMX=X(2)
NCRC=X(3)
GO TO 1
81 AZRF=X(1)
RTG=X(2)
RCS=X(3)
SBOCRIT=X(4)
DSNBOCR=X(5)
GO TO 1
91 PLDEN=X(1)
RDEN=X(2)
UNITHO=X(3)
UNITVO=X(4)
NMXO=X(5)
NPXO=X(6)
NHR=X(7)
UNITH=UNITHO*PLDEN
UNITV=UNITVO*PLDEN
NMXR=NMXO/RDEN
NPXR=NPXO/RDEN
NHR=NHR/RDEN
PLDEN=1./PLDEN
NHR1=NHR
IF(NHR.LT.100000) GO TO 1
NHR1=NHR/100000
NHR=NHR-NHR1*100000
GO TO 1
101 CONTINUE
WRITE(6,4014)
IF(NCRCT.EQ.0) WRITE OUTPUT TAPE 6,4015
WRITE(7,4016)
IF(KLR)102,103,102
102 CALL LATLON(GLATO,GLONO,0.,XRO,YRO,ZRO,GLATR,GLONR,X(1))
GO TO 104
103 CALL XYZ(GLATO,GLONO,0.,GLATR,GLONR,HR,XRO,YRO,ZRO)
CALL COORDX(0.,2,XRO,YRO,ZRO,RRO,ARO,X(1),XRO,YRO,ZRO,SRO,ERO)
104 IF(NB)105,110,105
105 DO 108 I=1,NB
IF(KLB(I))106,107,106
106 CALL LATLON(GLATO,GLONO,0.,XBO(I),YBO(I),ZBO(I),GLATB(I),GLONB(I),
IX(I))
GO TO 108
107 CALL XYZ(GLATO,GLONO,0.,GLATB(I),GLONB(I),HB(I),XBO(I),YBO(I),
IZBO(I))
CALL COORDX(0.,2,XBO(I),YBO(I),ZBO(I),RBO(I),ABO(I),X(1),XBO(I),
IYBO(I),ZBO(I),SBO(I),EBO(I))
108 CONTINUE
110 IF(NRV)111,115,111
111 DO 114 I=1,NRV
IF(KLT(I))112,113,112
112 CALL LATLON(GLATO,GLONO,0.,XTO(I),YTO(I),ZTO(I),GLATT(I),GLONT(I),
IX(I))
GO TO 114
113 CALL XYZ(GLATO,GLONO,0.,GLATT(I),GLONT(I),HT(I),XTO(I),YTO(I),
IZTO(I))
CALL COORDX(0.,2,XTO(I),YTO(I),ZTO(I),RTO(I),ATO(I),X(1),XTO(I),

```

```

      IYTO(I),ZTO(I),STO(I),ETO(I))
114 CONTINUE
115 CONTINUE
      CLO=(90.-GLATO)*RADIAN
      COSCLO=COS(CLO)
      SINCL0=SIN(CLO)
      IF(NB)303,311,303
303 CALL ASPECT(COSCLO,SINCL0,0.,XRO,YRO,ZRO,NB,XBO,YBO,ZBO,XBR,YBR,
      1ZBR)
      DO 308 I=1,NB
      CALL COORDX(HR,2,XBR(I),YBR(I),ZBR(I),RBR,ABR,X(1),X(2),X(3),X(4),
      1SBR,EBR)
308 CONTINUE
311 IF(NRV)312,400,312
312 DO 317 I=1,NRV
      CALL ASPECT(COSCLO,SINCL0,0.,XRO,YRO,ZRO,1,XTO(I),YTO(I),ZTO(I),
      1XTR,YTR,ZTR)
      CALL COORDX(HR,2,XTR,YTR,ZTR,RTR,ATR(1),X(1),X(2),X(3),X(4),STR(1)
      1,ETR(1))
317 CONTINUE
400 CONTINUE
402 CALL ATMOSD(-LFOUT,0.,X(1),X(2),X(3),X(4),X(5),X(6),X(7))
      CALL CIRAG5(-LFOUT,150.,X(1),X(2),TEM150,X(3),X(4),X(5),EM150)
      G150=GO*(RE/(RE+150.))**2
      CIRAX=1.+150.E+5*G150*EM150/RGAS/TEM150
      X(1)=SIN(DIPK*RADIAN)
      COSCLR=X(1)/SQRT(4.-3.*X(1)**2)
      SINCLR=SQRT(1.-COSCLR**2)
      VMR=VARR*RADIAN
      COSVMR=COS(VMR)
      SINVMR=SIN(VMR)
      IF(NB)702,711,702
702 DO 703 I=1,NB
      X(1)=XBR(I)*COSVMR-YBR(I)*SINVMR
      YBR(1)=XBR(1)*SINVMR+YBR(I)*COSVMR
703 XBR(I)=X(1)
711 IF(NRV)712,721,712
712 DO 713 I=1,NRV
713 ATR(I)=ATR(I)-VARR
721 IF(NFACE)722,724,722
722 DO 723 I=1,NFACE
723 AFACEM(I)=AFACE(I)-VARR
724 CONTINUE
      IF(NHELP.NE.0) CALL HELP(6HRANCIN,NHELP,1)
      RETURN
1001 FORMAT(11,E9.0,6E10.0,1X11,2XA6)
4014 FORMAT(13H1RUN COMMENTS)
4015 FORMAT(14HNO FINE CORRECTIONS MADE TO TAILORED GRID)
4016 FORMAT(17H1TRIG CORRECTIONS)
      END

```

FIT

This subroutine plots blank horizontal lines for the PONDER picture, for cases in which no blackout was found at that elevation. This subroutine defines the boundaries of the print and area grids using the search grid calculations. The flag IRUN that switches the executive routine from search to area grid calculations and vice versa is set in this routine.

IRUN = Grid Selection Flag (IRUN = 1:search grid, IRUN = 2:area grid)
NMXØ = number of targets on print grid row to the left of center column
NPXØ = number of targets on print grid row to the right of center column
NMX = number of targets on area grid row to the left of center column
NPX = number of targets on area grid to the right of center column
NMXR = number of targets on search grid row to the left of center column
NPXR = number of targets on search grid row to the right of center column
UNITHØ = print grid horizontal increment
UNITVØ = print grid vertical increment
UNITH = area grid horizontal increment
UNITV = area grid vertical increment
UNITHR = search grid horizontal increment
UNITVR = search grid vertical increment

Listing

```
SUBROUTINE FIT(NPRES,NLPRES,NRPRES,IRUN,NMXR,NPXR,NMXØ,NPXØ,NMX,
1NPX,IPLDEN,IRDEN,MAXNMX,IL,IR)
  DIMENSION S(150)
  NPREV=NPRES
  NLPREV=NLPRES
  NRPREV=NRPRES
  IF(IL)3,4,3
4  NPRES=1
  IF(NPREV-2)5,6,5
6  NMXØ=NLPREV*IRDEN
  NPXØ=NRPREV*IRDEN
  NMX=NMXØ*IPLDEN
  NPX=NPXØ*IPLDEN
  IRUN=2
  RETURN
5  DO 13 I=1,IRDEN
  IO=MOD(I,IRDEN)
  IF(IO)14,15,14
14 S(I)=IH
  WRITE OUTPUT TAPE 8,3001,S(I)
  WRITE OUTPUT TAPE 9,3001,S(I)
  WRITE OUTPUT TAPE 10,3001,S(I)
  WRITE OUTPUT TAPE 11,3001,S(I)
  WRITE OUTPUT TAPE 12,3001,S(I)
  WRITE OUTPUT TAPE 13,3001,S(I)
  GO TO 13
15 JO=MAXNMX+1
```

```

DO 16 J=1,J0
16 S(J)=1H
   J1=J0+1
   S(J1)=1H-
   WRITE OUTPUT TAPE 8,3001,(S(I),I=1,J1)
   WRITE OUTPUT TAPE 11,3001,(S(I),I=1,J1)
   WRITE OUTPUT TAPE 13,3001,(S(I),I=1,J1)
   S(J1)=1H+
   WRITE OUTPUT TAPE 9,3001,(S(I),I=1,J1)
   WRITE OUTPUT TAPE 10,3001,(S(I),I=1,J1)
   WRITE OUTPUT TAPE 12,3001,(S(I),I=1,J1)
13 CONTINUE
3001 FORMAT(120A1)
   NMX=NMXR
   NPX=NPXR
   IRUN=1
   RETURN
3 IF(NPRES)11,12,11
12 WRITE OUTPUT TAPE 6,3009
3009 FORMAT(28H VERTICAL ESTIMATE TOO SMALL)
11 NPRES=2
   NL=NMXR+2-1L
   NR=NPXR+2-1R
   NLPRES=MINOF(NL,NMXR)
   NRPRES=MINOF(NR,NPXR)
   IF(NPREV-2)9,10,9
9  NMXO=NLPRES*IRDEN
   NPXO=NRPRES*IRDEN
   NMX=NMXO*1PLDEN
   NPX=NPXO*1PLDEN
   IRUN=2
   RETURN
10 NMXO=MAXOF(NLPREV,NLPRES)*IRDEN
   NPXO=MAXOF(NRPREV,NRPRES)*IRDEN
   NMX=NMXO*1PLDEN
   NPX=NPXO*1PLDEN
   IRUN=2
   RETURN
END

```

CRCT

This subroutine performs horizontal grid corrections, as specified by NCRCT. When the blackout region extends past the PONDER picture limits, the subroutine CRCT sets in and extends the bounds by "NCRCT" number of grids to see if it can reach the outer limit of the blackout region.

NMX = (NMXR) number of targets per row to the left of center column

NPX = (NPXR) number of targets per row to the right of center column

NCRCT = grid correction factor (in units of print grid increments)

Listing

For this listing, see Appendix 1.

ANALYZE

This subroutine checks for the horizontal corrections made by subroutine CRCT and prints out the appropriate information concerning each correction.

ELREF = increment in the vertical direction. ELREF set to topmost point to be observed and decremented to lowest point to be observed.

Listing

For this listing, see Appendix 1.

PLOTSN

This is the plotting subroutine. It takes the S/N calculations produced elsewhere, codes them alpha numerically point by point, and stores them in a row matrix left to right for one row. The coded line is next written into the output file called PLOTSN.

SIGNAL = number generated by RANC for target presently being observed (this is the completed signal/noise ratio)

KEYT = number generated by RANC for target presently being observed (target calculation code)

DSNBOCR = signal/noise increment between output contours

NPX = number of target per row in unrotated system to the right of center column

NMX = number of targets per row in unrotated system to the left of center column

NPY = number of rows in unrotated system above the row containing reference point

NMY = number of rows in unrotated system below the row containing reference point

Listing

For this listing, see Appendix 1.

AREA1

This subroutine calculates the area of radar blackout on the 1KM target shell for the spherical mode; otherwise, it calculates the blackout on the target surface.

Listing

For this listing, see Appendix 1.

COORDX

This subroutine accepts coordinates of a point in any of four systems and transforms them to all of the other systems.

Inputs

HO

MODE =	1	2	3	4	5
RXSE =	GND RGE(km)	X(EAST, km)	SLNT RGE(km)	ELEV.(deg)	SLNT RGE(km)
AYAA =	AZIMUTH(deg)	Y(NORTH, km)	AZIMUTH(deg)	AZIMUTH(deg)	AZIMUTH(deg)
HZEH =	ALTITUDE(km)	Z(UP, km)	ELEV.(deg)	ALTITUDE(km)	ALTITUDE(km)

Outputs

R, A, H = surface range, azimuth, altitude coordinates of point

X, Y, Z = rectangular tangent-plane coordinates of point

S, E = slant range, elevation coordinates of point

Listing

For this listing, see Appendix 1.

RNCOT

This subroutine is listed and described in Refs. 4 and 5.

Listing

```
SUBROUTINE RNCOT  
END
```

SLCT

This subroutine reduces the number KEYT to a 1 digit representing its most significant digit. It then returns this digit to the plotting subroutine for appropriate action.

KEYT = number generated by RANC for target presently being observed (target calculation code)

Listing

For this listing, see Appendix 1.

PLOT

This is the plotting subroutine. It takes the S/N calculations produced elsewhere, codes them alpha numerically point by point, and stores them in a row matrix left to right for one row. The coded line is next written into the output file called PLOTTA, PLOTEL, PLOTAZ.

EER = number generated by RANC for target presently being observed (this is the completed temperature/noise ratio)
KEYT = number generated by RANC for target presently being observed (target calculation code)
BOCRIT = temperature/noise increment between output contours
NPX = number of targets per row in unrotated system to the right of center column
NMX = number of targets per row in unrotated system to the left of center column
NPY = number of rows in unrotated system above the row containing reference point
NMY = number of rows in unrotated system below the row containing reference point

Listing

```
SUBROUTINE PLOT(EER,KEYT,BOCRIT,NPX,NMX,NPY,NMY,MAXNMX,IPLDEN,KTAP
1E)
  DIMENSION EER(2),KEYT(2),R(150)
  PI=3.14159265359
  RADIAN=PI/180.
  NO=MAXNMX-NMX+1
  N1=MAXOF(1,NO)
  J1=NPY+NMY+1
  K1=NPX+NMX+1
  K2=N1+K1
  I=1
  DO 1016 J=1,J1
  DO 1017 N=1,N1
1017 R(N)=IH
  DO 1015 K=1,K1
  N=N1+K
  EERAD=ABSF(EER(I)*RADIAN)
  IF(EERAC-500.)1018,1018,1019
1019 R(N)=IH/
  GO TO 1015
1018 CONTINUE
  E=EERAD-BOCRIT
  KK=MOD(KEYT(I),10)
  IF((KK.EQ.2).OR.(KK.EQ.4)) GO TO 6001
  IF(E)1002,1001,1001
1001 CONTINUE
  KEY=KEYT(I)
  CALL SLCT(KEY)
  KEY=KEY+1
  GO TO(2001,2002,2004,2005,2007,2008),KEY
2001 R(N)=IH0
  GO TO 1015
```

```

2002 R(N)=1H1
GO TO 1015
2004 R(N)=1H2
GO TO 1015
2005 R(N)=1H3
GO TO 1015
2007 R(N)=1H4
GO TO 1015
2008 R(N)=1H5
GO TO 1015
1002 IE=EERAC*1000.+1.
GO TO(1,2,3,4,5,6,7,8,9,10),IE
  1 R(N)=1H
GO TO 1015
  2 R(N)=1H1
GO TO 1015
  3 R(N)=1HH
GO TO 1015
  4 R(N)=1HG
GO TO 1015
  5 R(N)=1HF
GO TO 1015
  6 R(N)=1HE
GO TO 1015
  7 R(N)=1HD
GO TO 1015
  8 R(N)=1HC
GO TO 1015
  9 R(N)=1HB
GO TO 1015
 10 R(N)=1HA
GO TO 1015
6001 CONTINUE
IF(E)5002,5001,5001
5001 CONTINUE
KEY=KEYT(I)
CALL SLCT(KEY)
KEY=KEY+1
GO TO(4001,4002,4004,4005,4007,4008),KEY
4001 R(N)=1H*
GO TO 1015
4002 R(N)=1H$
GO TO 1015
4004 R(N)=1H
GO TO 1015
4005 R(N)=1H
GO TO 1015
4007 R(N)=1H(
GO TO 1015
4008 R(N)=1H)
GO TO 1015
5002 CONTINUE
IE=EERAC*1000.+1.
GO TO(11,12,13,14,15,16,17,18,19,20),IE
 11 R(N)=1H-
GO TO 1015
 12 R(N)=1HZ
GO TO 1015
 13 R(N)=1HY
GO TO 1015
 14 R(N)=1HX
GO TO 1015

```

```
15 R(N)=1HW  
GO TO 1015  
16 R(N)=1HV  
GO TO 1015  
17 R(N)=1HU  
GO TO 1015  
18 R(N)=1HT  
GO TO 1015  
19 R(N)=1HS  
GO TO 1015  
20 R(N)=1HR  
1015 I=I+1PLDEN  
1016 WRITE OUTPUT TAPE KTAPE,3001,(R(K),K=1,K2)  
RETURN  
3001 FORMAT(120A1)  
END
```

PLOTR

This is the plotting subroutine. It takes the refractive/noise calculations produced elsewhere, codes them alpha numerically point by point, and stores them in a row matrix left to right for one row. The coded line is next written into the output file called PLOTRE, PLOTMR.

RER = number generated by RANC for target presently being observed (this is the completed refractive/noise ratio)
KEYT = number generated by RANC for target presently being observed (target calculation code)
NPX = number of targets per row in unrotated system to the right of center column
NMX = number of targets per row in unrotated system to the left of center column
NPY = number of rows in unrotated system above the row containing reference point
NMY = number of rows in unrotated system below the row containing reference point

Listing

```
SUBROUTINE PLOTR(RER,KEYT,NPX,NMX,NPY,NMY,MAXNMX,IPLOEN,KTAPE,
1STR)
  DIMENSION RER(2),KEYT(2),R(150)
  NO=MAXNMX-NMX+1
  N1=MAXOF(1,NO)
  J1=NPY+NMY+1
  K1=NPX+NMX+1
  K2=N1+K1
  I=1
  DO 1016 J=1,J1
  DO 1017 N=1,N1
1017 R(N)=1H
  DO 1015 K=1,K1
  N=N1+K
  RR=ABSF(RER(1))
  IF(RR-5000./STR)1018,1018,1019
1019 R(N)=1H/
  GO TO 1015
1018 CONTINUE
  KK=MOD(KEYT(1),10)
  IF((KK.EQ.2).OR.(KK.EQ.4)) GO TO 6001
  IF(RR-.01)1,1001,1001
1001 CONTINUE
  KEY=KEYT(1)
  CALL SLCT(KEY)
  KEY=KEY+1
  GO TO(2001,2002,2004,2005,2007,2008),KEY
2001 R(N)=1H0
  GO TO 1015
2002 R(N)=1H1
  GO TO 1015
2004 R(N)=1H2
  GO TO 1015
2005 R(N)=1H3
```



```

GO TO 1015
2007 R(N)=1H4
GO TO 1015
2008 R(N)=1H5
GO TO 1015
1 IF(RR-.005)3,4,4
4 R(N)=1H4
GO TO 1015
3 IF(RR-.001)5,6,6
6 R(N)=1H8
GO TO 1015
5 IF(RR-.0005)7,8,8
8 R(N)=1H4
GO TO 1015
7 R(N)=1H
GO TO 1015
6001 CONTINUE
IF(RR-.01)2,5001,5001
5001 CONTINUE
KEY=KEYT(1)
CALL SLCT(KEY)
KEY=KEY+1
GO TO(4001,4002,4004,4005,4007,4008),KEY
4001 R(N)=1H*
GO TO 1015
4002 R(N)=1H5
GO TO 1015
4004 R(N)=1H
GO TO 1015
4005 R(N)=1H
GO TO 1015
4007 R(N)=1H(
GO TO 1015
4008 R(N)=1H)
GO TO 1015
2 CONTINUE
IF(RR-.005)13,14,14
14 R(N)=1HX
GO TO 1015
13 IF(RR-.001)15,16,16
16 R(N)=1HY
GO TO 1015
15 IF(RR-.0005)17,18,18
18 R(N)=1HZ
GO TO 1015
17 R(N)=1H-
1015 I=1+IPLDEN
1016 WRITE OUTPUT TAPE KTAPE,30C1,(R(K),K=1,K2)
RETURN
3001 FORMAT(120A1)
END

```

Appendix 6

PONDER Code CLUTTER Option (Listing)

Note that the subroutines in Appendices 1, 5, and 6 have the same names. Subtle differences may exist in some of the routines; therefore, these routines should not be interchanged but should be used only with the PONDER options indicated.

PONDER

CONA = mode switch
GLATØ = latitude of origin
GLONØ = longitude of origin
HHR = altitude of origin
XRØ = (east) x coordinate of radar with respect to origin
YRØ = (north) y coordinate of radar with respect to origin
ZRØ = (vert) z coordinate of radar with respect to origin
ARØ = azimuth coordinate of radar with respect to origin
AROSP = radar heading
ITOT = total number of targets used to produce PONDER picture

Listing

```
PROGRAM PONDER(INPUT,TAPE5=INPUT,OUTPUT,TAPE6=OUTPUT,DOOT,  
1TAPE7=DOOT,PLOTSN,TAPE8=PLOTSN,PLOTCL,TAPE9=PLOTCL,PLOTDF,  
2TAPE10=PLOTDF,PLOTTPC,TAPE11=PLOTTPC)  
CALL CHANGE(7H+PONDER)  
ILEN=10000  
CALL CREATE(6HOUTPUT,ILEN,M1WD)  
ILEN=10000  
CALL CREATE(4HDOOT,ILEN,M1WD)  
ILEN=40000  
CALL CREATE(6HPLOTSN,ILEN,M1WD)  
ILEN=40000  
CALL CREATE(6HPLOTCL,ILEN,M1WD)  
ILEN=40000  
CALL CREATE(6HPLOTDF,ILEN,M1WD)  
ILEN=40000  
CALL CREATE(6HPLOTTPC,ILEN,M1WD)  
COMMON /EXEC/CLTR(50),SRCL(50),DSRCL(50),VC(50),DVC(50),  
1 SIGMAC(50),ETA(50),FBNE(50),FBDB(50),FBRHO(50),  
2 KEYC(50),TANT(50),AZFB(50),ELFB(50),DBTOFB(50),  
3 DBS(50),OMEGAB(50),KEYN(50),ACLTR,SRAC,DSRAC,VAC,  
4 DVAC,ASIGMA,ETAA,ENEA,ADBPXM,KEYA,AZAC,ELAC,DBAC,  
5 TASUM  
COMMON /BURST/NB,W(50),TSX(50),FF(50),HB(50),XBR(50),YBR(50),  
1 ZBR(50),TB(50),CUSCLB(50),RDZERO(50),RHZERO(50),  
2 RUZERO(50),LHVB(50),RHDB(50),HSB(50),PKTERM(50),  
3 RLK(50),RTK(50),DISTK(50),XKR(50),YKR(50),ZKR(50),  
4 TILTK(50),WUV(3,50),HR,CUSCLR,SINCLR,KALCH,KALCN,  
5 KALCR,DAY,CIRAX,TEM150,NSW1,NSW2,NSW3,NSW4,NSW5,NSW6  
COMMON/REGION/NF,INDEX(50),FLABL(50),KINDF(50),RFB(50),AFB(50),  
1 HF(50),XFR(50),YFR(50),ZFR(50),RLF(50),RTF(50),  
2 TILTF(50),HM1NF(50),HMAXF(50),TF0(50),TF(50),  
3 RHUF(50),HSNF(50),HDOT(50),RDOT(50)  
COMMON /RADAR/FREQ,POWER,DELTAR,DELTAV,TAG,TBW,TSL,SNMDB,TSYS,BWN,  
1 PCOMP,RAG,RBW,RSL,KSPEC,SCAN,EFACE,AFACEM(4),NFACE  
COMMON/TARGET/RCS,STR,ATK,ETR,VTR,TSEC(1),SIGNAL,SNCTRC,DBTAR,AER,  
1EER,RER,DISP,KEYT,KOUT,LFOUT  
COMMON/RNCIN/PLOEN,RDEN,NMXX,NPXR,NHR1,NHR,UNITH,UNITV,UNITHO,  
1UNITVO,UNITHR,UNITVR,AZREF,AZRF,RTG,MAXNMX,NCRC,T,VARR,SBOCRIT,  
2DSNBOCR,AREAD,THCLTR,DTHCLTR,THD1FF,0THD1FF
```

```

COMMON/CONST/RE,PI,RADIAN
COMMON/RNC/ICONF
COMMON/CONAL/CONA,GLATO,GLONO,HHR
COMMON/CMSPRAY/COSPCLO,S INCLQ,XRO,YRO,ZRO,ARO,AROSP,ISPF
COMMON/TARCNT/ITOT
DIMENSION SNPRES(500),CLPRES(500),DFPRES(500),KYT(500)
1,PCPRES(500),S(150)
NDOUT=7
NPLTSN=8
NPLTCL=9
NPLTDF=10
NPLTPC=11
ITOT=0
AREAO=1
ICONF=1
GO TO 3
1 ICONF=2
3 CALL RANC
ITOT=1
ICONF=3
IAREA=0
AREAI=0.
AREAO=0.
MXTANT=0.
MXDBTF=0.
MXDBS=0.
MXTF=0.
MXFBR=0.
MXFBE=0.
MXFBD=0.
ICONA=CONA+1.
SNBOCR=SBOCRIT+DSNBOCR
CLBOCR=THCLTR-DTHCLTR
DFBOCR=THDIFF+DTHDIFF
IF(ICONA.EQ.1.) GO TO 20
AZREF=AZRF-VARR
GO TO 21
20 CONTINUE
AZREF=AZRF
21 CONTINUE
CALL GRID(AXNHR)
WRITE(NDOUT,3003) UNITHR,UNITVR,UNITHO,UNITVD,UNITH,UNITV,NMXR,
INPXR,NHR1,NHR,MAXNMX,CONA,AZREF,SBOCRIT,OSNBOCR,THCLTR,DTHCLTR,
2THDIFF,DTHDIFF
IPLDEN=PLDEN
IRDEN=RDEN
IRP=IRGEN*IPLDEN
NMXIRD=NMXR*IRDEN
NPXIRD=NPXR*IRDEN
MAXNRV=(NMXR+NPXR)*IRP+1
AXNHR=INTF(85./(2.*RDEN))-1.
NHRMO=(AXNHR-NHR1)*RDEN
IF(NHRMO.LE.0) GO TO 8401
S(1)=1H
DO 8400 I=1,NHRMO
WRITE(NPLTSN,3001) S(1)
WRITE(NPLTCL,3001) S(1)
WRITE(NPLTDF,3001) S(1)
WRITE(NPLTPC,3001) S(1)
8400 CONTINUE
8401 CONTINUE
DO 8100 I=1,MAXNRV
PCPRES(I)=0.

```

```

CLPRES(I)=0.
DFPRES(I)=0.
SNPRES(I)=0.
8100 SNPREV(I)=0.
    IF(CONA.NE.2.) GO TO 8101
    IF((SQRTF(XRO**2+YRO**2+ZRO**2).LT..1).OR.(ABSF(ARO+180.-AZRF).LT.
1.1).OR.(ABSF(ARO-180.-AZRF).LT..1)) GO TO 8102
    IF(ABSF(ARO-AZRF).LT..1) GO TO 8103
    IF(YRO)8104,8105,8106
8104 CONTINUE
    AROSP=ATANF(XRO/YRO)/RADIAN
    ISPF=3
    GO TO 8101
8106 CONTINUE
    AROSP=ATANF(XRO/YRO)/RADIAN
8107 CONTINUE
    ISPF=4
    GO TO 8101
8105 CONTINUE
    IF(XRO)8108,8101,8110
8108 CONTINUE
    AROSP=-90.
    GO TO 8107
8110 CONTINUE
    AROSP=90.
    GO TO 8107
8102 CONTINUE
    ISPF=1
    GO TO 8101
8103 CONTINUE
    ISPF=2
8101 CONTINUE
    IPLOT=2
    IRUN=2
    NPRES=0
    NLPRES=0
    NRPRES=0
    ELREF=NHR*UNITVR
    DO 6000 I1=1,NHRI
    GO TO(7001,7002),IRUN
7002 GO TO(7701,7702),IPLOT
7702 CONTINUE
    IPLOT=1
    IRUN=1
    NMX=NMXR
    NPX=NPXR
    UNH=UNITHR
7001 CONTINUE
    ELREF=ELREF-UNITVR
    IO=1
    GO TO 7003
7701 CONTINUE
    ELR=ELREF
    UNH=UNITH
    IO=IRP
7003 DO 6003 I2=1,IO
    GO TO(7006,7005),IRUN
7005 ELREF=ELR+(IRP-I2)*UNITV
    ILNCT=0
    IRNCT=0
7006 CONTINUE
    IL=0

```

```

IR=0
ILEFT=1
NR=NMX+1
NRV=NMX+NPX+1
NRVPREV=0
7703 CONTINUE
DO 309 ITRGT=1,NRV
GO TO(5001,5002,5003), ILEFT
5001 CONTINUE
NR=NR-1
GO TO 5004
5002 CONTINUE
NR=-NPX
ILEFT=3
GO TO 5004
5003 CONTINUE
NR=NR+1
5004 CONTINUE
CALL SPRAY(RTG,UNH,NR,ELREF,AZREF,VARR,ATR,STR,ETR)
CALL RANC
DIFF=SIGNAL-SNCTRC
SNPC=10.**(-DIFF/10.)+10.**(-SIGNAL/10.)
SNPC=-10.*LOG10F(SNPC)
DO 8800 INB=1,NF
MXTANT=MAX1F(MXTANT,TANT(INB))
MXDBTF=MAX1F(MXDBTF,DBTOFB(INB))
MXDBS=MAX1F(MXDBS,DBS(INB))
MXTF=MAX1F(MXTF,TF(INB))
MXFBR=MAX1F(MXFBR,FBRHD(INB))
MXFBE=MAX1F(MXFBE,FBNE(INB))
MXFBD=MAX1F(MXFBD,FBDB(INB))
8800 CONTINUE
GO TO(5005,308), IRUN
5005 CONTINUE
IF(((SIGNAL-SNBOCR).GE.0.).AND.((DIFF-DFBOCR).GE.0.).AND.((SNCTRC-
1CLBOCR).LT.0.)) GO TO 309
5006 GO TO(5007,5008,5008), ILEFT
5007 ILEFT=2
IL=ITRGT
GO TO 309
5008 ITRGT=NRV
IR=NPX+NR+1
GO TO 309
308 CONTINUE
PCPRES(ITRGT+NRVPREV)=SNPC
SNPRES(ITRGT+NRVPREV)=SIGNAL
CLPRES(ITRGT+NRVPREV)=SNCTRC
DFPRES(ITRGT+NRVPREV)=DIFF
KYT(ITRGT+NRVPREV)=KEYT
309 CONTINUE
GO TO(7007,7008), IRUN
7007 CONTINUE
IF((IL.GT.0.).AND.(IR.EQ.0)) IR=NRV+1-IL
CALL FIT(NPRES,NLPRES,NRPRES,IRUN,NMXR,NPXR,NMXO,NPXO,NMX,NPX,
1IPLDEN,IRDEN,MAXNMX,IL,IR)
GO TO 6003
7008 CONTINUE
IF(NCRCT.EQ.0) GO TO 9003
IF(1RNCT.EQ.1) GO TO 7801
IF(1LNCT.EQ.1) GO TO 7802
ILCT=0
IF(((SNPRES(1)-SNBOCR).LT.0.).OR.((CLPRES(1)-CLBOCR).GE.0.).OR.
1((DFPRES(1)-DFBOCR).LT.0.)) ILCT=1

```

```

IRCT=0
IF(((SNPRES(NRV)-SNBOCR).LT.0.).OR.((CLPRES(NRV)-CLBOCR).GE.0.)
1.OR.((DFPRES(NRV)-DFBOCR).LT.0.)) IRCT=1
IF((ILCT.EQ.0).AND.(IRCT.EQ.0)) GO TO 9003
CALL CRCT(NMXIRD,NPXIRD,NMXO,NPXO,NMX,NPX,NPXRI,IPLDEN,NCRCT,ILCT,
1IRCT,NRVL,NRVR,NRL,NRR)
ILNCT=1
IF(NRVL.GT.0) GO TO 7803
7804 CONTINUE
IRNCT=1
IF(NRVR.GT.0) GO TO 7805
7806 CONTINUE
ILCT=0
IF(((SNPRES(1)-SNBOCR).LT.0.).OR.((CLPRES(1)-CLBOCR).GE.0.).OR.
1((DFPRES(1)-DFBOCR).LT.0.)) ILCT=1
IRCT=0
IF(((SNPRES(NRV)-SNBOCR).LT.0.).OR.((CLPRES(NRV)-CLBOCR).GE.0.)
1.OR.((DFPRES(NRV)-DFBOCR).LT.0.)) IRCT=1
CALL ANALYZE(NRVL,NRVR,ILCT,IRCT,ELREF)
GO TO 9003
7801 CONTINUE
NRV=NMX+NPX+1
GO TO 7806
7802 CONTINUE
NRV=NMX+NPX+1
GO TO 7804
7803 CONTINUE
NRVPREV=NRV
NRV=NRVL
NR=NRL
DO 7900 I=1,NRVPREV
PCPRES(NRVPREV+NRV+1-I)=PCPRES(NRVPREV+1-I)
SNPRES(NRVPREV+NRV+1-I)=SNPRES(NRVPREV+1-I)
CLPRES(NRVPREV+NRV+1-I)=CLPRES(NRVPREV+1-I)
DFPRES(NRVPREV+NRV+1-I)=DFPRES(NRVPREV+1-I)
KYT(NRVPREV+NRV+1-I)=KYT(NRVPREV+1-I)
7900 CONTINUE
NRVPREV=0
GO TO 7703
7805 CONTINUE
NRVPREV=NRV
NRV=NRVR
NR=NRR
NPX=NPXRI
GO TO 7703
9003 I3=MOD(12,IPLDEN)
IF(I3/8001,8000,8001)
8000 CALL PLOTSN(SNPRES,KYT,SNBOCR,DSNBOCR,NPXO,NMXO,0,0,MAXNMX,IPLDEN,
INPLTSN)
CALL PLOTSN(PCPRES,KYT,SNBOCR,DSNBOCR,NPXO,NMXO,0,0,MAXNMX,IPLDEN,
INPLTPC)
CALL PLOTCL(CLPRES,KYT,CLBOCR,DTHCLTR,NPXO,NMXO,0,0,MAXNMX,IPLDEN,
INPLTCL)
CALL PLOTSN(DFPRES,KYT,DFBOCR,DTHDIFF,NPXO,NMXO,0,0,MAXNMX,IPLDEN,
INPLTOF)
IPLDT=2
8001 CALL AREA(SNPRES,SNPREV,SBOCRIT,UNITH,UNITV,NMX,NPX,NRV,MAXNRV,
1IAREA,AREA1,AREAO,ICONA)
6003 CONTINUE
IF((IRUN.EQ.2).AND.(IPLDT.EQ.1)) GO TO 7701
6000 CONTINUE
JO=MAXNMX+1
DO 9100 J=1,JO

```

```

9100 S(J)=1H
      J1=J0+1
      S(J1)=1H+
      WRITE(NPLTSN,3001) (S(J),J=1,J1)
      WRITE(NPLTSN,3013)
      WRITE(NPLTCL,3001) (S(J),J=1,J1)
      WRITE(NPLTCL,3016)
      WRITE(NPLTDF,3001) (S(J),J=1,J1)
      WRITE(NPLTDF,3017)
      WRITE(NPLTPC,3001) (S(J),J=1,J1)
      WRITE(NPLTPC,3018)
      AREAAV=(AREAI+AREAO)/2.
      AA=AREAAV*(STR**2)
      IF(AREAI)3501,3502,3501
3502 APPERR=100.
      GO TO 3507
3501 APPERR=(AREAO/AREAAV-1.)*100.
3507 AI=AREAI*(STR**2)
      AO=AREAO*(STR**2)
      GCDH=STR*UNITHO*RADIAN
      WRITE(NDOUT,3005) TSEC(1)
      DO 3503 I=1,NB
      IF(TB(I)-TSEC(1))3504,3504,3503
3504 CONTINUE
      CALL COORDX(0.,2,XBR(I),YBR(I),ZBR(I),X1,ABR,X2,X3,X4,X5,
1SBR,EBR)
      WRITE(NDOUT,3006)TB(I),W(I),FF(I),TSX(I),HB(I),SBR,EBR,ABR
3503 CONTINUE
      WRITE(NDOUT,3007)STR,VTR,SBOCKIT,RCS,FREQ
      WRITE(NDOUT,3008)AA,AREAAV,APPERR
      TANT(1)=MXTANT
      DBTOFB(1)=MXDBTF
      DBS(1)=MXDBS
      TF(1)=MXTF
      FBRHO(1)=MXFBR
      FBNE(1)=MXFBE
      FBDB(1)=MXFBD
      CALL RNCOT
      WRITE(NDOUT,3009)
      WRITE(NPLTSN,3009)
      WRITE(NPLTCL,3009)
      WRITE(NPLTDF,3009)
      WRITE(NPLTPC,3009)
      GO TO 1
3001 FORMAT(120A1)
3003 FORMAT(3(/2F20.101,/5I10,2X,F2.0,/F20.3,3(/2F20.1C))
3005 FORMAT(1H1,F51.10)
3006 FORMAT(F52.3,/F57.5,/F60.3,/F56.3,/F57.3,/F57.3,/F52.3,/F55.3)
3007 FORMAT(F57.3,/F53.3,/F57.3,/F46.3,/F56.3)
3008 FORMAT(F54.3,/F54.6,/F41.3)
3009 FORMAT(3HEND)
3013 FORMAT(39X,6HSN(IV))
3016 FORMAT(38X,8HCLTR(IV))
3017 FORMAT(39X,6HSC(IV))
3018 FORMAT(38X,8HNPC(IV))
      END

```


RANC

This is an excerpt from the RANC routine, showing the modifications needed to run PONDER.

Listing

```
SUBROUTINE RANC
COMMON /EXEC/CLTR(50),SRCL(50),DSRCL(50),VC(50),DVC(50),
1 SIGMAC(50),ETA(50),FBNE(50),FBD(50),FBRHO(50),
2 KEYC(50),TANT(50),AZFB(50),ELFB(50),DBTOFB(50),
3 DBS(50),OMEGAB(50),KEYN(50),ACLTR,SRAC,DSRAC,VAC,
4 DVAC,ASIGMA,ETAA,ENEA,ADBPKM,KEYA,AZAC,ELAC,DBAC,
5 TASUM
COMMON /RADAR/FREQ,POWER,DELTAR,DELTAV,TAG,TBW,TSI,SNMDB,TSYS,BWN,
1 PCOMP,RAG,RBW,RSL,KSPEC,SCAN,EFACE,AFACEM(4),NFACE
COMMON/TARGET/RCS,STR(1),ATR(1),ETR(1),TSEC(1),SIGNAL(1),SNCTR(1)
1,DBTAR(1),AER(1),EER(1),RER(1),DISP(1),KEYT(1),KOUT,LFOUT
COMMON /BURST/NB,W(50),TSX(50),FF(50),HB(50),XBR(50),YBR(50),
1 ZBR(50),TB(50),CUSCLB(50),RDZERO(50),RHZERO(50),
2 RUZERO(50),LHVB(50),RHOB(50),HSB(50),PKTERM(50),
3 RLK(50),RTK(50),DISTK(50),XKR(50),YKR(50),ZKR(50),
4 TILTR(50),WUV(3,50),HR,COSCLR,SINCLR,KALCH,KALCN,
5 KALCR,DAY,CIRAX,TEM150,NSW1,NSW2,NSW3,NSW4,NSW5,NSW6
COMMON/REGION/NF,INDXF(50),FLABL(50),KINDF(50),RFB(50),AFB(50),
1 HF(50),XFR(50),YFR(50),ZFR(50),RLF(50),RTF(50),
2 TILTF(50),HMINF(50),HMAXF(50),TF(50),TF(50),
3 RHOF(50),HSNF(50),HDOF(50),RDOF(50)
COMMON/DEBRIS/ND,INDXD(50),DLABL(50),RDB(50),ADB(50),HD(50),
1 XDR(50),YDR(50),ZDR(50),RLD(50),RTD(50),COSCLD(50),
2 CISTD(50),WD(50),PD(50),RHOD(50),HSD(50),HBETA(50)
COMMON /PARAM/ALPHAR,GAMMA,UKOEA,UN,UBETA,UGAMMA,N1,N2,DMAX,HBMAX,
1 CHSTEP,RSTOP,DBSTOP,DEL,EDMIND,ENMINF,KINT,DENRAT,
2 FTRAIN,HTRAIN
COMMON/CNEXEC/GXMTR,BXMTR,SMXTR,KXMTR,GRCVR,BRCVR,SRGVR,KRCVR,
1 GTM,GRM,GM,COSCAN,PHIB,TAMB,NFB,BEAM,SNTERM
COMMON /CONST/RE,PI,RADIAN
COMMON/RNC/ICGNF
DIMENSION INDEXT(50)
J=1
GO TO (81,1,80),ICGNF
81 CONTINUE
RE = 6370.
PI = 3.14159265
RADIAN = 1.74532925E-2
ALPHAR = 1.E-12
GAMMA = 1.5
UKOEA = 1.E4
UN = 0.015
UBETA = 2.0
UGAMMA = 0.025
N1 = 3
N2 = 3
DMAX = 100.
HBMAX = 200.
DHSTEP = 2.5
RSTOP=0.01
DBSTOP=0.1
```

```

DEL = 0.001
EDMIND = 5.4E-9
ENMINF = 5.E4
KINT=1
DENRAT=2.
FTRAIN=1.
HTRAIN=1.
KASE=0
KSPEC=0
NFACE=0
NB=0
BOLTZ=1.3805E-20
C=2.998E+5
DBCON=4.34294482
FOURP1=4.*PI
TAMB=250.
HACMIN=90.
HACMAX=130.
RHOAC=1.E-10
1 CALL RANCIN(KASE)
  IF(POWER)2,730,2
2  SNMIN=10.**((SNMDB/10.))
  GXMTR=10.**((TAG/10.))
  BXMTR=TEN*RADIAN
  SXMTR=10.**((-TSL/10.))
  KXMTR=MCD(KSPEC,100)/10
  GRCVR=10.**((RAG/10.))
  BRCVR=REN*RADIAN
  SRCVR=10.**((-RSL/10.))
  KRCVR=MCD(KSPEC,10)
  DELRC=1.E-3*DELTAR
  DELRU=DELRC*PCOMP
  SNTERM=POWER*(C/FREQ)**2*PCOMP/(FOURP1)**3
  IF(NB)11,20,11
11 DO 12 I=1,NB
12 COSCLB(I)=(COSCLR*(RE+HR+ZBR(I))+SINCLR*YBR(I))/(RE+HB(I))
20 CONTINUE
  JTIME=1
  TIME=TSEC(JTIME)
  CALL PHENOM(TSEC,JTIME)
  RETURN
80 CONTINUE
  TASUM=TSYS
  AZBEAM=ATR(J)
  ELBEAM=ETR(J)

```

GRID

This subroutine is used as an automatic searching routine. When PONDER is run in its auto mode ($UNITH \leq 0$), Grid will compute the limits of the search grid based on the fireball and debris dimension, which is used by PONDER to produce a picture enclosing this total region. To perform this function, Grid normally need only be presented PLDEN and RDEN plus the auto run control values from the number 9 card. All other variables related to PONDER picture production are internally generated.

Listing

For this listing, see Appendix 1.

SPRAY

This subroutine produces the target matrix point by point, left to right, row by row. Each target point produced by SPRAY is then checked by RANC to find its S/N.

RTG = target shell radius
UNH = grid horizontal increment either
NR = position along horizontal line at which target is now located
ELREF = elevation angle (degrees cona = 0, km cona = 2) set to topmost row of
PONDER picture and decremented by the vertical print grid increment until
desired region of sky plotted
AZREF = azimuth from radar to grid center column (for case cona = 0)
VARR = magnetic variation
ATR = azimuth radar to target
STR = slant range radar to target
ETR = elevation angle to target relative to radar

Listing

For this listing, see Appendix 1.

RANCIN

This subroutine reads the input for PONDER calculations.

Listing

```
SUBROUTINE RANCIN(KASE)
COMMON /RADAR/FREQ,POWER,DELTAR,DELTAV,TAG,TBW,TSL,SNMDB,TSYS,BWN,
1      PCOMP,RAG,RBW,RSL,KSPEC,SCAN,EFACE,AFACE(4),NFACE
COMMON/TARGET/RCS,STR(1),ATR(1),ETR(1),VTR,TSEC(1),SIGNAL(1),
1SNCTRC(1),DBTAR(1),AER(1),EER(1),RER(1),DISP(1),KEYT(1),KOUT,LFOUT
COMMON /BURST/NB,W(50),TSX(50),FF(50),HB(50),XBR(50),YBR(50),
1      ZBR(50),TB(50),COSCLB(50),RDZERO(50),RHZERO(50),
2      RUZERO(50),LHVB(50),RHOB(50),HSB(50),PKTERM(50),
4      TILTK(50),WUV(3,50),HR,COSCLR,SINCLR,KALCH,KALCN,
3      RLK(50),RTK(50),DISTK(50),XKR(50),YKR(50),ZKR(50),
5      KALCR,DAY,CIRAX,TEM150,NSW1,NSW2,NSW3,NSW4,NSW5,NSW6
COMMON /CONST/RE,PI,RADIAN
COMMON/RNCIN/PLDEN,KDEN,NMXR,NPXR,NHR1,NHR,UNITH,UNITV,UNITHO,
1UNITVO,UNITHR,UNITVR,AZREF,AZRF,RTG,MAXNMX,NCRC,T,VARR,SBOCRIT,
2DSNBCCR,AREAD,THCLTR,DTHCLTR,THDIFF,DTHDIFF
COMMON/CONAL/CONA,GLATO,GLONG,HHR
COMMON/CMEPRAY/COSCLU,SINCLC,XRO,YRO,ZRO,ARO,AROSP,ISPF
DIMENSION X(7),AFACE(4)
DIMENSION RBO(50),ABO(50),XBO(50),YBO(50),ZBO(50),SBO(50),EBO(50),
1GLATB(50),GLONB(50),KLB(50),ALPHA6(50)
DIMENSION RTD(1),ATO(1),HT(1),XTO(1),YTO(1),ZTO(1),STO(1),ETO(1),
1GLATT(1),GLONT(1),KLT(1),ALPHA7(1)
GO TO 80
8700 AREAD = 1
GO TO 1
80 CONTINUE
LOGICAL ENFIL
LFIN=5
LFOUT=6
GO=980.0
RGAS=8.3143E+7
NHLP=0
IN6=0
NRV=0
NT=1
1 READ(LFIN,1001) NCARD,(X(I),I=1,7),KODE,ALPHA
IF(EOF,5) 4,5
4 CONTINUE
CALL PLQTE
CALL EXIT
5 IF(NCARD.EQ.0.AND.KODE.EQ.9) NCARD=10
IF(AREAD)8701,8702,8701
8702 IF(NCARD.EQ.10)8700,1
8701 IF(NCARD)2,101,2
2 GO TO (11,21,31,41,51,61,71,81,91,1)NCARD
11 FREQ=X(1)
POWER=X(2)
DELTAR=X(3)
DELTAV=X(4)
TAG=X(5)
TBW=X(6)
```

```

TSL=X(7)
ALPHA1=ALPHA
IF(KODE)12,1,12
12 CALL DATIN(KODE,AFACE)
ALPHA2=ALPHA
GO TO 1
21 SNMDB=X(1)
TSYS=AMAX1(X(2),1.)
BWN=AMAX1(X(3),1.E-3)
PCOMP=X(4)
RAG=X(5)
RBW=X(6)
RSL=X(7)
ALPHA2=ALPHA
GO TO 1
31 KSPEC=X(1)
SCAN=X(2)
EFACE=X(3)
AFACE(1)=X(4)
NFACE=1
DO 33 I=5,7
IF(X(I))32,33,32
32 NFACE=NFACE+1
AFACE(NFACE)=X(I)
33 CONTINUE
IF(SCAN)34,34,1
34 NFACE=0
GO TO 1
41 GLATO=X(1)
GLONO=X(2)
DAY=X(3)
KALC=X(4)
KALCH=MAX(1-MOD(KALC,1000)/100,0)
KALCN=MAX(1-MOD(KALC,100)/10,0)
KALCR=MAX(1-MOD(KALC,10),0)
KOUT=X(5)
KTRACK=MIN1(ABS(X(6)),10.)
KODE=X(7)
NSW1=MOD(KODE,1000000)/100000
NSW2=MOD(KODE,100000)/10000
NSW3=MOD(KODE,10000)/1000
NSW4=MOD(KODE,1000)/100
NSW5=MOD(KODE,100)/10
NSW6=MOD(KODE,10)
IF(NSW6.EQ.1) NSW5=1
IF(NSW5.EQ.1) NSW4=1
IF(NSW4.EQ.1) NSW3=1
GO TO 1
51 IF(KODE)52,53,52
52 CALL CQORDX(0.,KODE,X(1),X(2),X(3),RRQ,ARO,HR,XRD,YRD,ZRD,SRQ,ERQ)
GO TO 54
53 GLATR=X(1)
GLONR=X(2)
HR=X(3)
54 KLR=KODE
DIPR=X(4)
VARR=X(5)
ALPHA5=ALPHA
HHR=HR
GO TO 1
61 IF(IN6)62,65,62
62 IF(IN6-50)63,69,63
63 IF(X(4)-TB(NB))64,65,65

```

```

64 NHLP=63
65 IN6=IN6+1
   NB=IN6
   IF(KODE)66,67,66
66 CALL COORDX(0.,KODE,X(1),X(2),X(3),RBD(NB),ABD(NB),HB(NB),XBD(NB),
   1YBD(NB),ZBD(NB),SBD(NB),EBD(NB))
   GO TO 68
67 GLATB(NB)=X(1)
   GLONB(NB)=X(2)
   HB(NB)=X(3)
68 KLB(NB)=KODE
   TB(NB)=X(4)
   W(NB)=X(5)
   FF(NB)=X(6)
   TSX(NB)=X(7)
   ALPHA6(NB)=ALPHA
   GO TO 1
69 NHLP=62
   GO TO 1
71 TSEC(NT)=X(1)
   MAXNMX=X(2)
   NCRCT=X(3)
   AZRF=X(4)
   RTG=X(5)
   RCS=X(6)
   VTR=X(7)
   GO TO 1
81 SBOCRIT=X(1)
   DSNBOCR=X(2)
   THCLTR=X(3)
   DTHCLTR=X(4)
   THDIFF=X(5)
   DTHDIFF=X(6)
   CONA=KODE
   GO TO 1
91 PLDEN = X(1)
   RDEN =X(2)
   UNITH =X(3)
   UNITH0=X(3)
   UNITV0=X(4)
   NMX0=X(5)
   NPX0=X(6)
   NHR=X(7)
   UNITH=UNITH0*PLDEN
   UNITV=UNITV0*PLDEN
   NMXR=NMX0/RDEN
   NPXR=NPX0/RDEN
   NHR=NHR/RDEN
   PLDEN=1./PLDEN
   NHR1=NHR
   IF(NHR.LT.100000) GO TO 1
   NHR1=NHR/100000
   NHR=NHR-NHR1*100000
   GO TO 1
101 CONTINUE
   WRITE OUTPUT TAPE 6,4009
   IF(NCRCT.EQ.0) WRITE OUTPUT TAPE 6,4015
   IF(KLR)102,103,102
102 CALL LATLON(GLAT0,GLON0,0.,XRD,YRD,ZRD,GLATR,GLONR,X(1))
   GO TO 104
103 CALL XYZ(GLAT0,GLON0,0.,GLATR,GLONR,HR,XRD,YRD,ZRD)
   CALL COORDX(0.,2,XRD,YRD,ZRD,ARD,X(1),XRD,YRD,ZRD,SRD,ERD)

```

```

104 IF(NB)105,110,105
105 DO 108 I=1,NB
    IF(KLB(I))106,107,106
106 CALL LATLON(GLATO,GLONO,0.,XBO(I),YBO(I),ZBO(I),GLATB(I),GLONB(I),
    IX(I))
    GO TO 108
107 CALL XYZ(GLATO,GLONO,0.,GLATB(I),GLONB(I),HB(I),XBO(I),YBO(I),
    IZBO(I))
    CALL COGRDX(0.,2,XBO(I),YBO(I),ZBO(I),RBO(I),ABO(I),X(1),XBO(I),
    IYBO(I),ZBO(I),SBO(I),EBO(I))
108 CONTINUE
110 IF(NRV)111,115,111
111 DO 114 I=1,NRV
    IF(KLT(I))112,113,112
112 CALL LATLON(GLATO,GLONO,0.,XTO(I),YTO(I),ZTO(I),GLATT(I),GLONT(I),
    IX(I))
    GO TO 114
113 CALL XYZ(GLATO,GLONO,0.,GLATT(I),GLONT(I),HT(I),XTO(I),YTO(I),
    IZTO(I))
    CALL COGRDX(0.,2,XTO(I),YTO(I),ZTO(I),RTO(I),ATC(I),X(1),XTC(I),
    IYTO(I),ZTO(I),STO(I),ETO(I))
114 CONTINUE
115 CONTINUE
    CLO=(90.-GLATO)*RADIAN
    COSCLO=COS(CLO)
    SINCLC=SIN(CLO)
    IF(NB)303,311,303
303 CALL ASPECT(COSCLO,SINCLC,0.,XRO,YRO,ZRO,NB,XBO,YBO,ZBO,XBR,YBR,
    IZBR)
    DO 308 I=1,NB
    CALL COGRDX(HR,2,XBR(I),YBR(I),ZBR(I),RBR,ABR,X(1),X(2),X(3),X(4),
    ISBR,EBR)
308 CONTINUE
311 IF(NRV)312,400,312
312 DO 317 I=1,NRV
    CALL ASPECT(COSCLO,SINCLC,0.,XRO,YRO,ZRO,1,XTO(I),YTO(I),ZTO(I),
    IXTR,YTR,ZTR)
    CALL COGRDX(HR,2,XTR,YTR,ZTR,RTR,ATR(I),X(1),X(2),X(3),X(4),STR(I),
    I,ETR(I))
317 CONTINUE
400 CONTINUE
402 CALL ATMOSD(-LFOUT,0.,X(1),X(2),X(3),X(4),X(5),X(6),X(7))
    CALL CIRA65(-LFOUT,150.,X(1),X(2),TEM150,X(3),X(4),X(5),EM150)
    G150=G0*(RE/(RE+150.))**2
    CIRAX=1.+150.E+5*G150*EM150/RGAS/TEM150
    X(1)=SIN(DIPR*RADIAN)
    COSCLR=X(1)/SQRT(4.-3.*X(1)**2)
    SINCLR=SQRT(1.-COSCLR**2)
    VMR=VARR*RADIAN
    COSVMR=COS(VMR)
    SINVMR=SIN(VMR)
    IF(NB)702,711,702
702 DO 703 I=1,NB
    X(1)=XBR(I)*COSVMR-YBR(I)*SINVMR
    YBR(I)=XBR(I)*SINVMR+YBR(I)*COSVMR
703 XBR(I)=X(1)
711 IF(NRV)712,721,712
712 DO 713 I=1,NRV
713 ATR(I)=ATR(I)-VARR
721 IF(NFACE)722,724,722
722 DO 723 I=1,NFACE
723 AFACEM(I)=AFACE(I)-VARR

```



```
724 CONTINUE
    IF(NHELP.NE.0) CALL HELP(6HRANCIN,NHELP,1)
    RETURN
1001 FORMAT(11,E9.0,6E10.0,1X11,2XA6)
4009 FORMAT(13H1RUN COMMENTS)
4015 FORMAT(41HNO FINE CORRECTIONS MADE TO TAILORED GRID)
    END
```

FIT

This subroutine plots blank horizontal lines for the PONDER picture, for cases in which no blackout was found at that elevation. This subroutine defines the boundaries of the print and area grids using the search grid calculations. The flag IRUN that switches the executive routine from search to area grid calculations and vice versa is set in this routine.

IRUN = Grid Selection Flag (IRUN = 1:search grid, IRUN = 2:area grid)
NMXØ = number of targets on print grid row to the left of center column
NPXØ = number of targets on print grid row to the right of center column
NMX = number of targets on area grid row to the left of center column
NPX = number of targets on area grid to the right of center column
NMXR = number of targets on search grid row to the left of center column
NPXR = number of targets on search grid row to the right of center column
UNITHØ = print grid horizontal increment
UNITVØ = print grid vertical increment
UNITH = area grid horizontal increment
UNITV = area grid vertical increment
UNITHR = search grid horizontal increment
UNITVR = search grid vertical increment

Listing

```
SUBROUTINE FIT(NPRES,NLPRES,NRPRES,IRUN,NM XR,NP XR,NMXØ,NPXØ,NMX,
1NPX,IPLDEN,IRDEN,MAXNMX,IL,IR)
  DIMENSION S(150)
  NPREV=NPRES
  NLPREV=NLPRES
  NRPREV=NRPRES
  IF(IL)3,4,3
4  NPRES=1
  IF(NPREV-2)5,6,5
6  NMXØ=NLPREV*IRDEN
  NPXØ=NRPREV*IRDEN
  NMX=NMXØ*IPLDEN
  NPX=NPXØ*IPLDEN
  IRUN=2
  RETURN
5  DO 13 I=1,IRDEN
  IO=MOD(I,IRDEN)
  IF(IO)14,15,14
14 S(I)=1H
  WRITE(8,3001) S(I)
  WRITE(9,3001) S(I)
  WRITE(10,3001) S(I)
  WRITE(11,3001) S(I)
  GO TO 13
15 JO=MAXNMX+1
  DO 16 J=1,JO
```

```

16 S(J)=1H
   J1=J0+1
   S(J1)=1H-
   WRITE(8,3001) (S(I),I=1,J1)
   WRITE(9,3001) (S(I),I=1,J1)
   WRITE(10,3001) (S(I),I=1,J1)
   WRITE(11,3001) (S(I),I=1,J1)
13 CONTINUE
3001 FORMAT(120A1)
   NMX=NMXR
   NPX=NPXR
   IRUN=1
   RETURN
3 IF(NPRES)11,12,11
12 WRITE OUTPUT TAPE 6,3009
3009 FORMAT(28H VERTICAL ESTIMATE TOO SMALL)
11 NPRES=2
   NL=NMXR+2-1L
   NR=NPXR+2-1R
   NLPRES=MINOF(NL,NMXR)
   NRPRES=MINOF(NR,NPXR)
   IF(NPREV-2)9,10,9
9 NMXO=NLPRES*IRDEN
  NPXO=NRPRES*IRDEN
  NMX=NMXO*IPLDEN
  NPX=NPXO*IPLDEN
  IRUN=2
  RETURN
10 NMXO=MAXOF(NLPREV,NLPRES)*IRDEN
  NPXO=MAXOF(NRPREV,NRPRES)*IRDEN
  NMX=NMXO*IPLDEN
  NPX=NPXO*IPLDEN
  IRUN=2
  RETURN
END

```

CRCT

This subroutine performs horizontal grid corrections, as specified by NCRCT. When the blackout region extends past the PONDER picture limits, the subroutine CRCT sets in and extends the bounds by "NCRCT" number of grids to see if it can reach the outer limit of the blackout region.

NMX = (NMXR) number of targets per row to the left of center column
NPX = (NPXR) number of targets per row to the right of center column
NCRCT = grid correction factor (in units of print grid increments)

Listing

For this listing, see Appendix 1.

ANALYZE

This subroutine checks for the horizontal corrections made by subroutine CRCT and prints out the appropriate information concerning each correction.

ELREF = increment in the vertical direction. ELREF set to topmost point to be observed and decremented to lowest point to be observed.

Listing

For this listing, see Appendix 1.

PLOTSN

This is the plotting subroutine. It takes the S/N calculations produced elsewhere, codes them alpha numerically point by point, and stores them in a row matrix left to right for one row. The coded line is next written into the output file called PLOTSN.

SIGNAL = number generated by RANC for target presently being observed (this is the completed signal/noise ratio)

KEYT = number generated by RANC for target presently being observed (target calculation code)

DSNBOCR = signal/noise increment between output contours

NPX = number of target per row in unrotated system to the right of center column

NMX = number of targets per row in unrotated system to the left of center column

NPY = number of rows in unrotated system above the row containing reference point

NMY = number of rows in unrotated system below the row containing reference point

Listing

For this listing, see Appendix 1.

AREA

This subroutine calculates the area of radar blackout on the 1KM target shell for the spherical mode; otherwise, it calculates the blackout on the target surface.

Listing

```
SUBROUTINE AREA(SIGNAL,SSIGNAL,BOCRIT,UNITH,UNITV,NMX,NPX,NT,MAXNT
1,IAREA,AREA1,AREAD,ICONA)
DIMENSION SIGNAL(2),SSIGNAL(2)
PI=3.14159265359
RADIAN=PI/180.
NTP1=NT+1
MXNTP1=MAXNT+1
DO 100 I=1,NT
IF(SIGNAL(I)-BOCRIT)101,102,102
101 SIGNAL(I)=1.
GO TO 100
102 SIGNAL(I)=0.
100 CONTINUE
DO 103 I=NTP1,MXNTP1
103 SIGNAL(I)=0.
IF(IAREA)1,2,1
2 NMXPREV=NMX
NPXPREV=NPX
1 IGAP=NMX-NMXPREV
IF(IGAP)10,13,12
12 DO 20 I=1,NTPREV
20 SSIGNAL(NTPREV+IGAP+1-I)=SSIGNAL(NTPREV+1-I)
DO 30 I=1,IGAP
30 SSIGNAL(I)=0.
13 ISHIFT=0
GO TO 11
10 IGAP=-IGAP
DO 40 I=1,NT
40 SIGNAL(NT+IGAP+1-I)=SIGNAL(NT+1-I)
DO 50 I=1,IGAP
50 SIGNAL(I)=0.
ISHIFT=IGAP
11 NM=MAXOF(NMX,NMXPREV)+MAXOF(NPX,NPXPREV)
NMP2=NM+2
DO 120 I=1,NM
NO=SIGNAL(I)+SIGNAL(I+1)+SSIGNAL(I)+SSIGNAL(I+1)+1.
GO TO(120,121,121,121,122),NO
121 IFLAG=1
GO TO 123
122 IFLAG=2
123 CONTINUE
GO TO (301,302,303),ICONA
301 CONTINUE
MXNMX=MAXOF(NMX,NMXPREV)
IF(1-(MXNMX+1))124,125,125
124 THETA=(MXNMX-1)*UNITH
GO TO 126
125 THETA=(1-(MXNMX+1))*UNITH
126 A=UNITV*RADIAN*(SINF((THETA+UNITH)*RADIAN)-SINF(THETA*RADIAN))
202 CONTINUE
```

```

      GO TO(127,128),IFLAG
128 AREA1=AREA1+A
127 AREA0=AREA0+A
      GO TO 120
302 CONTINUE
      A=0.
      GO TO 202
303 CONTINUE
      A=UNITV*UNITH
      GO TO 202
120 CONTINUE
      NTPREV=NT
      NMXPREV=NMX
      NPXPREV=NPX
      IAREA=1
      DO 60 I=1,NT
60  SSIGNAL(I)=SIGNAL(ISHIFT+I)
      DO 70 I=NTP1,NMP2
70  SSIGNAL(I)=0.
      RETURN
      END
      FUNCTION ASIN(X)
      PI=3.14159265359
      IF(X)1,2,3
1  Z=X+1.0
      IF(Z)4,5,6
4  WRITE(6,1001) Z
      ASIN=-PI/2.
      GO TO 7
3  Z=1.0-X
      IF(Z)8,9,6
8  WRITE(6,1002) Z
      ASIN=PI/2.
      GO TO 7
2  ASIN=0.
      GO TO 7
6  Y=SQRT(1.0-X*X)
      ASIN=ATANF(X/Y)
7  RETURN
1001 FORMAT(28HARGUMENT OF ARCSINE PLUS ONE,1PE20.12)
1002 FORMAT(29HONE MINUS ARGUMENT OF ARCSINE,1PE20.12)
      END
      FUNCTION ACOS(X)
      PI=3.14159265359
      ACOS=PI/2.-ASINF(X)
      RETURN
      END

```


COORDX

This subroutine accepts coordinates of a point in any of four systems and transforms them to all of the other systems.

Inputs

HO = altitude of origin of coordinate systems (km)

MODE = 1 2 3 4 5

RXSE = GND RGE(km) X(EAST, km) SLNT RGE(km) ELEV.(deg) SLNT RGE(km)

AYAA = AZIMUTH(deg) Y(NORTH, km) AZIMUTH(deg) AZIMUTH(deg) AZIMUTH(deg)

HZEH = ALTITUDE(km) Z(UP, km) ELEV.(deg) ALTITUDE(km) ALTITUDE(km)

Outputs

R, A, H = surface range, azimuth, altitude coordinates of point

X, Y, Z = rectangular tangent-plane coordinates of point

S, E = slant range, elevation coordinates of point

Listing

For this listing, see Appendix 1.

RNCOT

This subroutine is listed and described in Refs. 4 and 5.

Listing

```
SUBROUTINE RNCOT  
END
```

SLCT

This subroutine reduces the number KEYT to a 1 digit representing its most significant digit. It then returns this digit to the plotting subroutine for appropriate action.

KEYT = number generated by RANC for target presently being observed (target calculation code)

Listing

For this listing, see Appendix 1.

PLOT C

This is the plotting subroutine. It takes the clutter/noise calculations produced elsewhere, codes them alpha numerically point by point, and stores them in a row matrix left to right for one row. The coded line is next written into the output file called PLOT C.

CLTR = number generated by RANC for target presently being observed (this is the completed clutter/noise ratio)
KEYT = number generated by RANC for target presently being observed (target calculation code)
DSNBOCR = clutter/noise increment between output contours
NPX = number of targets per row in unrotated system to the right of center column
NMX = number of targets per row in unrotated system to the left of center column
NPY = number of rows in unrotated system above the row containing reference point
NMY = number of rows in unrotated system below the row containing reference point

Listing

```
SUBROUTINE PLOT C (CLTR, KEYT, BOCR, DBOCR, NPX, NMX, NPY, NMY, MAXNMX,
1 I P DEN, KTAPE)
  DIMENSION CLTR(2), KEYT(2), R(150)
  NO = MAXNMX - NMX + 1
  N1 = MAXOF(1, NO)
  J1 = NPY + NMY + 1
  K1 = NPX + NMX + 1
  K2 = N1 + K1
  I = 1
  DO 1016 J = 1, J1
  DO 1017 N = 1, N1
1017 R(N) = 1H
  DO 1015 K = 1, K1
  N = N1 + K
  CL = CLTR(1) - BOCR
  IF (CL .GE. 0.) GO TO 1001
  R(N) = 1H
  GO TO 1015
1001 IF ((CL - DBOCR) .GE. 0.) GO TO 1002
  R(N) = 1H.
  GO TO 1015
1002 IF ((CL - 2.*DBOCR) .GE. 0.) GO TO 1003
  R(N) = 1HH
  GO TO 1015
1003 IF ((CL - 3.*DBOCR) .GE. 0.) GO TO 1004
  R(N) = 1HHH
  GO TO 1015
1004 IF ((CL - 4.*DBOCR) .GE. 0.) GO TO 1005
  R(N) = 1HGH
  GO TO 1015
1005 IF ((CL - 5.*DBOCR) .GE. 0.) GO TO 1006
  R(N) = 1HFF
```

```

GO TO 1015
1006 IF((CL-6.*DBOCR).GE.0.) GO TO 1007
R(N)=1HE
GO TO 1015
1007 IF((CL-7.*DBOCR).GE.0.) GO TO 1008
R(N)=1HC
GO TO 1015
1008 IF((CL-8.*DBOCR).GE.0.) GO TO 1009
R(N)=1HC
GO TO 1015
1009 IF((CL-9.*DBOCR).GE.0.) GO TO 1010
R(N)=1HE
GO TO 1015
1010 IF((CL-10.*DBOCR).GE.0.) GO TO 1011
R(N)=1HA
GO TO 1015
1011 CONTINUE
KEY=KEYT(I)
CALL SLCT(KEY)
KEY=KEY+3
GO TO(2001,2002,2003,2004,2005,2006,2007,2008,2009),KEY
2001 CONTINUE
R(N)=1H8
GO TO 1015
2002 CONTINUE
R(N)=1H7
GO TO 1015
2003 CONTINUE
R(N)=1H0
GO TO 1015
2004 CONTINUE
R(N)=1H1
GO TO 1015
2005 CONTINUE
R(N)=1H2
GO TO 1015
2006 CONTINUE
R(N)=1H3
GO TO 1015
2007 CONTINUE
R(N)=1H4
GO TO 1015
2008 CONTINUE
R(N)=1H5
GO TO 1015
2009 CONTINUE
R(N)=1H6
1015 I=I+1PLCEN
1016 WRITE(KTape,3001) (R(K),K=1,K2)
RETURN
3001 FORMAT(120A1)
END

```