A Method for Computing Vertical-Plane Coverage
Diagrams for Frequency Agile Pulse Radar Systems

W. H. HARPER AND PETER N. MARINOS
Search Radar Branch
Radar Division

COPY AVAILABLE TO DDC DOES NOT
PERMIT FULLY LEGIBLE PRODUCTION

April 1977

NAVAL RESEARCH LABORATORY
Washington, D.C.

Approved for public release; distribution unlimited.
A method for obtaining computer plots of vertical-plane coverage diagrams (i.e., range-height-angle charts) for frequency agile pulse radar systems is presented. The approach is based on well-known monochromatic radar detection calculations which are here extended to cover the frequency agile case.
CONTENTS

INTRODUCTION ............................................................. 1
MATHEMATICAL ANALYSIS ............................................... 1
PLOTTING TECHNIQUES ................................................... 3
SUMMARY ........................................................................ 4
ACKNOWLEDGMENT ......................................................... 4
REFERENCES ..................................................................... 5
APPENDIX A — Fortran Listing for Program LOBMUF .......... 9
APPENDIX B — Input Data Card Format for LOBMUF
(For ASC — Jan. 1977) ....................................................... 48
A METHOD FOR COMPUTING VERTICAL-PLANE COVERAGE
DIAGRAMS FOR FREQUENCY AGILE PULSE RADAR SYSTEMS

INTRODUCTION

In a comprehensive report on maximum radar range computations, Blake [1] provides all the necessary tools for obtaining coverage plots (i.e., range-height-angle charts) of monochromatic pulse radar systems. This paper provides a method for extending the range computations applicable to the monochromatic case to pulse radar systems featuring frequency agility. The number of different frequencies employed, as well as the number of consecutive pulses to be transmitted at the same frequency, are both arbitrary.

MATHEMATICAL ANALYSIS

The detection scheme assumed in the analysis which follows utilizes a square-law detector followed by a linear integrator, as shown in Fig. 1. The symbols $D_0$, $X_0$, and $Y_0$ denote the SNR (i.e., signal-to-noise power ratio) at the designated points of the detector-integrator complex.

Based on approximate closed-form expressions derived by Barton [2] and later simplified by Cann [3], one may relate the input and output SNRs using the equation

$$X_0 = \frac{2D_0^2}{D_0 + 2.3} \quad (1)$$

The integrated output SNR, $Y_0$, may now be written in the form

$$Y_0 = \sum_{j=1}^{N} X_{0j} = \sum_{j=1}^{N} \frac{2D_{0j}^2}{D_{0j} + 2.3} \quad (2)$$

where $N$ denotes the number of integrated pulses and $D_{0j}$ represents the SNR at the detector input due to the $j^{th}$ pulse illuminating a target located at $R_{max}$; it is assumed that $R_{max}$ represents the maximum detection range associated with the total energy delivered on the target by the $N$ transmitted pulses.

Since our objective is to relate the integrated SNR (i.e., $Y_0$) to a maximum detection range, $R_{max}$, using the classical radar equation, an average SNR is defined at the detector output in the form

Note: Manuscript submitted March 4, 1977.
The SNR given by Eq. (3) may be thought of as the SNR necessary to achieve a specified probability of detection, \( P_d \), and false alarm, \( P_{fa} \), for a specified number of pulses and a postulated type of target; normally, it is the corresponding SNR, \( \hat{D}_0 \), at the input of the square-law detector rather than \( \hat{X}_0 \) that one associates with \( P_d \) and \( P_{fa} \). Given, however, \( \hat{D}_0 \), the corresponding value of \( \hat{X}_0 \) is obtained using Eq. (1). For example, given \( N, P_d, \) and \( P_{fa} \), and a target described by Swerling's Case 2, one may utilize published results [1] to determine \( \hat{D}_0 \) and subsequently \( \hat{X}_0 \).

Once \( \hat{X}_0 \) is obtained, the problem becomes one of relating \( \hat{X}_0 \) to a corresponding maximum detection range, \( R_{max} \), in a way which takes advantage of the computational aids developed by Blake [1,4]. This is accomplished by writing the radar equation in the form,

\[
\hat{X}_0 = \frac{1}{N} \sum_{j=1}^{N} D_{0j} = \frac{1}{N} \sum_{j=1}^{N} \frac{2D_{0j}^2}{D_{0j} + 2.3}
\]

where \( D_{0j} \) represents the SNR at the detector input due to the \( j^{th} \) pulse return from a target positioned at \( R_{max} \); \( \hat{D}_0 \) is the (averaged) minimum SNR based on \( N \) pulses; \( P_{noise} \) is the pattern-propagation factor; and the remaining symbols in Eq. (4) represent well-known radar parameters [5]. The bracketed portion of Eq. (4) represents the free-space range, \( R_{fs} \), corresponding to \( N \) pulses, but as if they all had been transmitted at frequency \( f_j \) (i.e., wavelength \( \lambda_j \)) and each resulted in a SNR, \( \hat{D}_0 \), at the detector input. In view of the above interpretation, one may rewrite Eq. (4) in the form,

\[
D_{0j} = \hat{D}_0 \left( \frac{R_{xj}}{R_{max}} \right)^4
\]

where \( R_{xj} = R_{fs} F_j \), and \( j = 1,2,...,N \).

Since

\[
R_{fs} = \frac{P_t A^2 \sigma}{4\pi \lambda^2 \hat{D}_0 P_{noise}}
\]

varying only the frequency results in a relationship of the form

\[
R_{fs} = R_{ifs} \left( \frac{\lambda_i}{\lambda_j} \right)^2 = R_{ifs} \left( \frac{f_j}{f_i} \right)^2
\]

2
which provides the free-space range at frequency \( f_j \) in terms of the free-space range at another frequency, \( f_1 \), while all other parameters in Eq. (6) remain fixed.

In some instances the maximum free-space range, \( R_{j \text{ max}} \), based on the number of pulses at each frequency, is known. In this case, there is a \( \hat{b}_{oj} \) given by

\[
\hat{b}_{oj} = \frac{P_t A_j^2 CF_j}{4 \pi \lambda_j^2 P_{\text{noise}} R_{j \text{ max}}} \ldots (8)
\]

The free-space range at frequency \( f_j \) is then given by

\[
R_{jfs}^4 = R_{j \text{ max}}^4 \left( \frac{\hat{b}_{oj}}{\hat{b}_o} \right) \ldots (9)
\]

Substituting next Eqs. (3) and (5) into Eq. (1), that is, into expression

\[
\frac{2 \hat{b}_o}{1 + 2.3 \left( \frac{1}{\hat{b}_o} \right)} \ldots (10)
\]

one obtains

\[
\frac{1}{1 + \alpha} = \frac{1}{N} \sum_{j=1}^{N} \left[ \left( \frac{R_{xj}}{R_{\text{max}}} \right)^4 \right] \ldots (11)
\]

where \( \alpha = (2.3/\hat{b}_o) \).

**PLOTTING TECHNIQUES**

Solution of Eq. (11) for \( R_{\text{max}} \) provides the information necessary to produce the so-called radar vertical-plane coverage diagrams.

Blake [4] has developed a computer program for presenting the radar interference lobing phenomena on range-height-angle plots. This program was named LOBEPLOT and was written in Fortran for use on a CDC-3800 computer.

Relatively simple modifications to LOBEPLOT have been made which have resulted in the solution of Eq. (11) and enabled the computer (Calcomp) plotting of the lobing charts for frequency agile radars. The original modifications were made, and the program was debugged on a CDC-3800 computer at NRL. More
recently, the program has been changed to allow running on the new Texas Instruments Advanced Scientific Computer (ASC) at NRL. This new version has been named LOBMUF for "lobes, multi-frequency." The complete program is listed in Appendix A; the input data card formats are given in Appendix B.

In LOBMUF, P_f, P_s, the total number of pulses integrated, the Swerling fluctuation case, and the number of radars or radar frequencies are specified on input data cards. In addition, for each radar or radar frequency, the free-space range, beamwidth, number of pulses at that frequency, sidelobe level, and a parameter FREF are required. FREF is zero if the free-space range was calculated at each frequency as indicated in connection with Eqs. (8) and (9); otherwise, FREF is the midband or average frequency and the free-space range corresponding to this reference frequency. Other inputs to the program are identical to those for LOBEPLT. They concern the dimensions of the range-height chart, polarization radiated, antenna height and tilt, etc.

Using subroutines from another of Blake's computer programs [5], RGCALC, the signal-to-noise ratio, D_0, and the parameter, A, are computed. If FREF = 0, D_0 is calculated for each frequency is calculated. Next, the pattern propagation factor, F_j, is calculated exactly as in Blake's LOBEPLT. Finally, a simple search routine is used to converge on the value of R_{max} in Eq. (11). The search is terminated when the two sides of Eq. (11) differ by less than 0.01.

Figure 2 gives a typical coverage diagram generated by the computer program LOBMUF. The inputs involved a hypothetical radar radiating two pulses at each of 21 frequencies from 1350 MHz to 1650 MHz in 15 MHz steps. 1500 MHz was chosen as the reference frequency, and the free-space range was chosen to be 100 n. mi.

Figure 3 shows the lobing plot for a single frequency radar at 1500 MHz. A comparison of Figs. 2 and 3 shows that the incoherent integration in a frequency agile radar does much to fill in the nulls of the lobing pattern and give a solid elevation coverage.

SUMMARY

A computer program has been developed to plot range-height-angle lobing charts for frequency agile radars. This program is quite flexible. It may be used to plot the lobing charts for several radars or a single radar including several frequencies. This program, LOBMUF, builds on Blake's range-height charts and lobing plots. The program assumes incoherent integration and a closed form expression for a square-law detector.

ACKNOWLEDGMENT

Thanks are due to Dr. W. M. Waters of NRL for many helpful discussions concerning this problem.
REFERENCES


Fig. 1 — Radar detection and integration diagram
Fig. 2 — Lobing chart for hypothetical radar radiating two pulses at each of 21 frequencies from 1350 MHz to 1650 MHz in 15 MHz steps. Wave height of 4 feet, free-space range of 100 n. mi., and vertical beamwidth of 4.0° were assumed.
Fig. 3 — Lobing chart for single frequency radar at 1500 MHz with free-space range of 100 n. mi. A wave height of 4 feet and vertical beamwidth of 4.0° were assumed.
APPENDIX A: Fortran Listing for Program LOBMUF

The basic modifications to Blake's LOBE PLOT for program LOBMUF occur in Subroutine LOBES, as described in the Plotting Techniques section of this report. In addition, Subroutines PDSN, PD, INVERS, WARR, DGAM, DEVAL, GAM, EVAL, and SUMLOG are incorporated directly from RGCALC.
STATEMENT

PROGRAM L4M44F
REAL*8 ILABL(10)
COMMON XMAX/YMAX/G1,G2,G3,G4,G5,G6,ERROR
COMMON/ADLTS/1ACD
COMMON/ASC/ICSC
COMMON/LT/LTN
DIMENSION A(1500)
DATA LT,N/1/
CALL RSTCP
READ 1,ANTHT
ERRORN=001
CALL PLTS (A,1500,0,5)
PRINT 30
30 FORMAT (' THE PEN TURRET POSITIONS ARE USED, 110 AND 111. PLOT ST
ARTS WITH PEN TURRET POSITION 110. CAUSER PENG IS 111, I/)!
2 READ (5,5,END=100) ILABL
200 READ 3, XMAX,YMAX,RMAX,WMAX,THMIN,THMAX,WHT,RDR
NDR=NRD
IF (NDR ,LT ,1 ) GOTO 110
SFAC = XMAX*125
52 M = .175*SFAC
  Y = 1.5*SFAC
IF (LTN*,EQ.,18) GO TO 61
IF ((YMAX+Y),.LE.,9.5) GO TO 61
60 SFAC = .5/(YMAX+Y)
  XMAX = YMAX*SFAC
  XMAX = XMAX*SFAC
  Y = Y*SFAC
  H = H*SFAC
31 X= Y+3,H
CALL ORIGIN (X,Y)
PRINT 10, ILABL
CALL PHACT (XMAX,YMAX,RMAX,WMAX,THCENTGT)
CALL LABES (XMAX,YMAX,RMAX,WMAX,WHT,THMIN,THMAX,NDR)
CALL LETTER (N,=Y,H,ILABL,0,0,B0)
CALL ORIGIN (XMAX+3,=Y,=Y)
CALL REZEP
GO TO 2
100 CALL ENPLT
GOTO 130
110 FORMAT(120,NDR
120 FORMAT ('/// PROGRAM TERMINATION DUE TO ILLEGAL DATA ENTERED FOR NU
MBER OF RACAPS. . . . . . NUMBER ENTERED IS 1,IS)
130 CONTINUE
1 FORMAT (F10.0)
3 FORMAT (F10.0)
5 FORMAT (10A8)
10 FORMAT (3X,10A8//)
END
STATEMENT

SUBROUTINE LOBES (XMAX, YMAX, RMAX, HMIX, AHFT, THMIN, THMAX, NCRD)
THIS VERSION COMPLETED JUNE 1976 FOR MULTIFREQUENCY CASE WITH INCOHERENT INTEGRATION.
COMMON /CSC/ ICSC
COMMON /TFEE/ X2, Y2, X11, Y11, X4, Y4, ERR4R
COMMON /XX1(181),YY1(181),CT1(181),SN1(181),DEL1
COMMON /NUM/ PFF (2000)
DIMENSION RX4(50), NUM(50), SNR(50)
DIMENSION RFS(50), FMAX(50), BCD(50), SLDB(50)
DATA IIMAX/2000/
DATA ICSC / /0/;
DATA P1/3, 1, 41592654/
DATA R2/4, 2831, 5307/
DATA R0V/2, 1, 7453, 2, 9252/
DATA CNV/1, 6457, 8533, 5, 6104/
DATA AE / 2, 7, 865, 26, 947775/
R0SC#R0SC
R0SC#FLAT(R0SC)
READ R0SC, P0T, PFA, PULS, CASE, AHFT, TILT, P0L, CSC
PRINT R0SC, P0T, PFA, PULS, CASE, AHFT, TILT, P0L, CSC
G05 FORMAT (X, 2X, WM0T#, FS, 2X, WM0FA#, FS, 2X, WM0PULS#, FS, 1, 2X, S0H#CASE= 
X, FS, 2X, 5X, AHFT#, FS, 1, 2X, RHTILT#, FS, 2X, WM0PULS#, FS, 1, 2X, WM0SC#, FU, 
1, 3)

DATA R0SC#PULS
DATA R0SC#CASE
DATA R0SNC (POT, PFA, N0PLS, CASE, SDR)
SNN#10, ** (SDR/10.)
GAMMA#2, 3, SNN
BETA#1, (1, +GAMMA)

G04 FORMAT (BF12, 0)
PRINT G04
500 FORMAT (THE CALLING PARAMETERS SENT TO SUBROUTINE L0BES ARE AS FOLLOWS):
PRINT G05, YMAX, YMAX, RMAX, HMIX, AHFT, THMIN, THMAX, NCRD
591 FORMAT (X, 2X, WM0MAX#, FU, 1, 2X, WM0MAX#, FU, 1, 2X, WM0MAX#, 
RFS, 1, 2X, WM0MAX#, FU, 1, 2X, WM0MAX#, FU, 1, 2X, WM0MAX#, 
1, 3)
58 84 1 I#1, IIMAX
PFF (II# 0, 0
E = YMAX#MAX#/ (XMAX#MAX#CNV)
EX=MAX#/MAX# Z=AHFT#, 3535354
Y#1, E=45
Y#1, E=45
DM 801 JJ#1, NR0SC
READ (5, 802, END#807) RANGE, FREQ, BEAM, S0L#B, PULNUM, FREF
803 ICSC#CSC
IP#1#4, 1
NUM(JJ)=PULNUM
RFS(JJ)=RANGE
FMAX(JJ)=RANGE
FMAX(JJ)=RANGE
STATEMENT

FIND(JJ)=FIND
SDB(JJ)=SDADB
R2 FORMAT (514,9)
PRINT 99,9,9
Q9 FORMAT (23,3,21),"THE RADAR PARAMETERS INPUT FOR RADAR NUMBER I,J3I"
R1 ARE AT F U L L M A S T E R"
PRINT 592,REFS(JJ),AZME,FLWZ(JJ),B=0(JJ),SCLB(JJ),TILT,POLE,CSC,POLE
EUP,REF
592 FORMAT (7,2X,4HRFSZ,F6,1,2X,5AHFTZ,F6,1,2X,5HMSZ,F7,1,2X,4HMBCZ,
*F5,1,2X,5HSLDH,F5,1,2X,5HTILT,F5,1,2X,4HPOLZ,F2,0,2X,4HCSCZ,F2,0
*F2,0,2X,5HREFZ,F7,1/)
IF (REFS,EE,0.) GO TO 514
CALL POSN((POF,PF,WEU,WEU(JJ),KSE,SDR)
SNR(JJ)=10.*((SDR/10.)
GO TO 515
514 SNR(JJ)=SNR
REFS(JJ)=REFS(JJ)*SGRT(SGRT(F'WZ(JJ)/FREF))
515 CONTINUE
REFS(JJ)=REFS(JJ)*(SGRT(SGRT(SNR(JJ)/SNR)))
RDR=SNR=1
801 CONTINUE
TILTR=TILT+RON
TMIN=TMIN+RON
TMAX=TMAX+RON
DEL1 = (TMAX-TMIN)/IIMAX
PI23DE = 3.0*DEL1 + PI2
T=2*TMIN=DEL1
N = 0
INDEX = 0
F=1.5
IDAS=1
CALL DASN
AH=AHFT*AHFT
HAE = 2.*AHFT/(3.5*AE)
AEH = AE + (AE+AHFT)
PRAW=SGRT((AE/HAE))
ON 810 IIM=1.IIMAX
PRAW=0.
THT2=THT2+DEL1
IF (THT2.GF.P123DE ) GOTO 40
T2=TAN(THT2)
S2=SGRT(THT2)
S3 = S2
PSI=THT2
ON 805 RAS=1.RAS
RAS=RAS(M)*RAS
B=1+RAS
C=SQR( .7071 + SIN(B-2*TILT))
IF (RAD(M),LE,45.) GO TO 251
250 PRINT=""
STATEMENT

IF (RANGE(4),EQ.,360.) IBEAM = 1
GO TO 252
251 IBEAM = 1
CONST = 1.39157/SIN(RH=2)
ALL = RSI/(PI/CONST)
CPAT1 = 10.**(((11.2*SLDR(1))*0.05)
252 = = 183.573/FM2**2
END = 0.25 =
**LIM = 0.01 =
FAC = PI2/1
PD = 2., AHFT = S2
IF (INDEX, LE, 1) GO TO 77
T/2 = T2/3.
GAM = HAE/SQRT(T2**2+HAE)+T2)
PSI = THET2 + GAM
S3 = SIN(PSI)
ZETA = PIHAR*T2
DI = .577355/SQRT(1.+2.*ZETA/SQRT(ZETA*ZETA*3))
POI = SQRT(AM2*AFH*GAM*GAM)+2.*S3*S3
IF (ABS(DP) >=LIM) GO TO 79
79 IF (C1, LT, 0.999) GO TO 79
780 DI = 1.
INDEX = 1
790 PD = POI
IF (PO, LT, 4.0, THET2, LT, 0.00873) GO TO 300
GO TO 301
301 IF (IDASH, NE, 1) GO TO 300
302 CALL DASH/NE
IDASH = 0
304 CONTINUE
77 CALL SKEFF(FMUZ(M),PSI,IP,L,RH,M,PHI)
PI2 = PI*Z&S3
PI2 = PI*PI2
RUF = EXP(-6.*PI2)
ANG = THET2 = TILTH
IF (IBEAM) 260, 261, 262
260 PAT = 1.
GO TO 24
261 PAT = COS(CONST*ANG)
GO TO 24
262 IF (ICSC, NE, 1) GO TO 61
30 IF (ANG, LE, 2.) G* TO 61
60 PAT = CSCI/S2
GO TO 24
61 IF (ANG, NE, 0.) GO TO 23
22 PAT = 1.
GO TO 24
23 LU = COS * SIN (ANG)
INT = sinP/12
DIFF = LU = INT * PI2
IF (ABS(ANG), LE, ALL) GO TO 161
STATEMENT

160 CPAT = CPAT1
   GO TO 162
161 CPAT = 1.
162 PAT = CPAT*SIN(DIFF)/UU
24 ANGR = THE2 + TILTH + 2. * GAM
   IF (IEHAV) 270, 271, 272
270 PATR = 1.
   GO TO 27
271 PATR = COS(CONST*ANGR)
   GO TO 27
272 IF (ANGR, NE, 0.1) G0 TO 26
25 PATR = 1.
   GO TO 27
26 UUR = CONST * SIN(ANGR)
   INTR = UUR / PI2
   DIFF = UUR - INTR * PI2
   IF (ABS(ANGR), LE, 0.0) G0 TO 66
65 CPAT = CPAT1
   GO TO 67
66 CPAT = 1.
67 PAT = CPAT*SIN(DIFF)/UUR
27 IF (ABS(PAT), GE, 1.0 = 44) G0 TO 29
28 IF (PAT, GE, 0.0) GO TO 281
280 D = UHR*UUF*PATR1, EUS = D1
   GO TO 30
281 D = D1*PHI*UUF*PATR1, EUS = D1
   GO TO 30
29 D = D1*PHI*PATR/PAT * UUF
30 ALPHA = FAC * PD * PHI
   INTR = ALPHA / PI2
   DIFF = ALPHA - INTR * PI2
   F = ABS(PAT) * SQRT((1.0 + D*D + 2.0*D*COS(DIFF)))
   RXN4(M) = RFS(M) + F**4
   RFX4 = SQRT(SQRT(RXN4(M)))/RNX
805 CONTINUE
   RNX = RXN
   RNX2 = RXN2
   RNX2 = RXN2
   FACMIN = 0.
997 SUMEN = 9.
998 MM = 1.
      RPRATX = RNX(M) / RNX
      RNX = FLAT1(UUM(M))
      SUMEN = (BNCPULS) * (RRATX / (1.0 + GAMMA/MM)) * SUMEN
998 CONTINUE
   FACM = FACMIN.
   FACMIN = SUMEN + BETA
   SUM = FACM + FACM
   IF (RNG, GE, 0.0) GO TO 993
   IF (ARIS (FACMIN), LT, 0.01) G0 TO 993
   IF (FACM, GE, 1.) GO TO 902

14
<table>
<thead>
<tr>
<th>i</th>
<th>j</th>
<th>k</th>
<th>l</th>
<th>m</th>
<th>n</th>
<th>o</th>
<th>p</th>
<th>q</th>
<th>r</th>
<th>s</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
STATEMENT

Y11 ZY
GOT 800
440 CALL MINTAP (X,Y)
620 XP=X
YP=Y
IF (THET2.GE.THMAX) GOTO 40
GOT A 800
602 IF (N.EQ.1) GOTO 620
CALL DASH5.
IDASH50
IF (1AX1S.EQ.1) CALL PLOT(XMAX,YMAX,2)
CALL INTRST (0., YMAX,XMAX, X,Y, XP, YP, X0, Y0)
621 CALL PLOT(X0,Y0, 2)
CALL DASH5.
XZ=XY
Y2=YO
X11=X
XP=X
Y11=XY
Y11=XY
YP=YN
GOT A 800
603 IF (N.EQ.1) GOTO 620
CALL DASH5.
IDASH50
CALL INTRST (XMAX,0., XMAX, YMAX, X,Y, XP, YP, X0, Y0)
GOTO 621
605 IF (N.EQ.1) GOTO 620
CALL MINTAP (XP,YP)
CALL PLOT(XP,YP,2)
CALL INTRST (0., YMAX,XMAX, X,Y, XP, YP, X0, Y0)
IAX15=2
CALL PLOT(X0,Y0,2)
GOTO 620
609 IF (N.EQ.1) GOTO 620
CALL MINTAP (XP,YP)
CALL PLOT(XP,YP,2)
CALL INTRST (XMAX,0., XMAX, YMAX, X,Y, XP, YP, X0, Y0)
IAX15=1
CALL PLOT(X0,Y0,2)
GOTO 620
40 IF (ISUM.NE.1) GOTO 651
650 CALL PLOT(X,Y,2)
RETURN
651 IF (X.GT.XMAX.AND.XP.GT.XMAX) GOTO 652
GOTO 653
652 CALL INTRST (XMAX,0., XMAX, YMAX, N,.EQ., X,Y, X0, Y0)
CALL DASH5.
IF (YP.LE.YMAX) GOTO 655
54 CALL PLT(XMAX,YMAX,2)
      CALL INTHST(Y,0,XMIN,YMIN,X,Y,0)
55 CALL PLT(X0,Y0,2)
      CALL CASME
      RETURN
56 IF(YGT,YMAX,X0,Y0,GT,YMAX) G0 TO 656
      GO TO 657
57 CALL CASME
      CALL INTHST(0,YMAX,XMAX,YMAX,0,0,0,0,X,Y,X0,Y0)
      IF(X0,LE,XMAX) G0 TO 659
58 CALL INTHST(XMAX,0,XMAX,YMAX,0,0,0,0,X,Y,X0,Y0)
      CALL PLT(X0,Y0,2)
      CALL CASME
      RETURN
59 IF(IAXIS,LE,1) CALL PLT(XMAX,YMAX,2)
      GO TO 660
60 CONTINUE
   DATA 802
61 PRINT 802
62 FORMAT(1X,'END OF FILE WAS READ WHERE DATA CARD SHOULD BE')
63 END
STATEMENT

SUBROUTINE PULS(PDT, PFL, PULS, KASE, SDB)
EXTERNAL PD
DIMENSION BO(5), SL*PE(5), PDFAC(5)
COMMON /PO5/ PFLAST, SLAST, KLAST
DATA PFLAST/0/
DATA SLAST/0/
DATA KLAST/0/
DATA CR0/2, 5, 14, 14, 13, 2, 13, 2/
DATA SL*PE/6, 7, 8, 7, 8/5
DATA PDFAC/4, 8, 20, 20, 13, 13, 20/
DATA CRIN/-30/7
DATA CRMAX/50/7
DATA CR1/1/7
DATA CR2/0/7
DATA PDLAST/C/
IF (PDT .Eq. PDLAST) Goto 20
1 IF (PFL .Lt. PFLAST) Goto 20
2 IF (PULS .Lt. SLAST) Goto 20
3 IF (KASE .Eq. KLAST) RETURN
20 PDLAST = PDT
PFLAST = PFL
SLAST = SL*PE
KLAST = KASE
KASE = 1
PULS = PULS
CR1 = CR1 + K * SL*PE(K) * AL*G10(PULS) * (PDT .Eq. 5) * PDFAC(K) * (PFL .Eq. 8) , 4 = 1.
CR2 = CR1 + 2.
CALL INVERS(CMI, CRMAX, CR1, CR2, 4, 15, AUX, SCR, PDI, PDT, PD)
END
STATEMENT

FUNCTION FTC(SHOR)
COMMON PDS/FA,KASE
IP=""FAN#FA
KASE=KASE
CALL MERSAR (SHOR,MP,FAN,KAS,P01)
P0=P01
END

BEST AVAILABLE COPY
STATEMENT

SLDRELTIE [X MTX, X MAX, XL, X = I, "SIG, LI", "F1, F2, FT, F"
TEST = 1. **(N SIG)
FD = FT
IF (FT = EB, 0.) FD = 1.
OM = 1
DELTA = X = I = XL
X 1 = XL
X 2 = X I
F 1 = F (X 1)
F 2 = F (X 2)
SLOPE = (F 2 - F 1) / (X 2 - X 1)
IF (SLOPE, F 0.) GO TO 21
1) F = X MAX (X MAX)
F = F MIN (X MIN)
SLOPE = (F MAX - F MIN) / (X MAX - X MIN)
21 IF (SLOPE + GE 0.) GO TO 23
22 X 1 = X 2
F 1 = F 2
X 1 = X 2 + DELTA
IF (X 2 = GT, X MAX) X 2 = X MAX
F 2 = F (X 2)
GO TO 21
23 IF (SLOPE + GE 0.) GO TO 25
24 X 2 = X 1
X 1 = X 2 + DELTA
IF (X 1 = LT, X MIN) X 1 = X MIN
F 2 = F 1
F 1 = F (X 1)
GO TO 23
25 X = X 1
X 1 = X 2 + DELTA
IF (ABS (F 2 - F 1) + GE, ABS (F 1 - F 2)) GO TO 6
7 F 2 = F 2
F 2 = F 1
F 1 = F 2
X 2 = X 1
X 1 = X 2
GO TO 6
1 F 1 = F (X)
X = X 1
TEST2 = ABS (ABS (F 1 - FT) / FD)
IF (TEST2 = GT, TEST) GO TO 6
RETURN
6 IF (N XI, LT, LI) GO TO 13
12 PRINT 40
PRINT 41, LI
PRINT 42, X MIN, X MAX, XL, X = I, "SIG, LI", "F1, F2, FT"
RETURN
13 IF (F 1, NE, F 2) GO TO 10
STATEMENT

15 IA (F1, F2, F3) GA T* 14
17 X4 = X1
    X2 = XA
    GA T* 19
19 X4 = X1
    X2 = XA
10 X = (X4 + X3) * 5
    GA T* 1
15 X = (F1 + F2) * (F1 + F2) + X2
    IF (X .LT. X4) X = XA
    IF (X .GT. X8) X = X8
    TI = ""I + 1
X2 = X1
GA T* 1
40 FORMAT (""MESSAGE FROM SUBROUTINE INVER = ""/")
41 FORMAT (""FUNCTION INVERSION NOT ACCOMPLISHED WITHIN SPECIFIED ",
        * ""I3, ""ITERATIONS,""/")
42 FORMAT (""INVER PARAMETERS WERE "", E10.3, E10.3, E10.3,
        * 2(E10.3, E10.3) ")
END
STATEMENT

COMMON P, KAPPA (SNDR, FA, KASE, FR)
DOUBLE PRECISION EPSR, YHPR, GAMPR, VPR, Y, EN, Y1, E1, STEP, VR
COMMON J, IN1, QAM, DEVAL, SUMLT, SUML, FA, FA
SNDR = 100.*1.0D0*1.0D0
DEVAL = 1
IF (MORE) 600, 600, 910
900 FAN = DLOG10(DLOG((1.500)/DLOG(1.0/(10.000)*=FA))))
GO TO 905
600 FAN = FA
905 IF (N) 90, 90, 9
2 IF (FA) 99, 99, 3
3 IF (KASE) 99, 99, 4
4 IF (KASE = 1) 5, 5, 99
5 ENPR = 0
6 ENPR = FAN
EN = K
VPRR = 0
7 IF (VPR > EQ. 1.0, AND. FAPREV, EQ. FA) GO TO 777
IF (N = 12) 17, 7, 8
7 VPRR = EM = (1.0 - 1.3 + *ENPR/EN**((2.000/3.000) + 0.15 + ENPR))
GO TO 11
8 VPRR = EM = (1.0 - 1.3 + *ENPR/EN**((4.000/11.000) + ENPR))
11 ENPR = 10.000
GAMPR = (GAMPR + VPRR, VPRR = 1)
PYR = EM = (5.000/ENPR)
SUML = SUMLT = (1.0 + ENPR)
IF (GAMPR = PYR) 10, 12, 12
10 H = 0
GO TO 14
12 H = = 0
14 V = = VPRR
EN = DEVAL(Y0, N = 1.5, SUML)
16 Y1 = Y + H
E1 = DEVAL(Y1, N = 1, SUML)
STEP = GAMPR + H*(E + E1) / 2,
IF ((CSSING1, DO, STEP + PYR) = CSSING1, DO, H)) GO TO 20
19 Y0 = Y1
E0 = E1
GAMPR = STEP
GO TO 13
20 IF (N) 22, 24, 24
22 YB = Y1 = + (PYR + STEP)/(GAMPR + STEP)
GO TO 30
24 YB = Y0 = - (PYR + GAMPR)/(STEP + GAMPR)
30 SIAS = YB
777 YH = SIAS
VPRR = FAM
FA = 5.0
K = KASE = 1
GO TO (100, 200, 300, 400, 500)
STATEMENT

100 SL" = 0,
P = E"+1
IF(VA=2/3) 150,172,102
102 KS = ((E-1)/2, + SQRT(((E-1)/2)*2+PYRA)
KS = "AK"(KS,)
GS = 1, + GA"(YB, KS, = 1, TH)
TS = EVAL (P, KS) * GS
G = GS
k = KS
TERM = TS
TL = TL
110 TEMP = SL"+TERM
IF(TEMP = 112, 115, 116
112 SL" = TEMP
IF(x) 115, 116, 114
114 TERM = TERM*FLOAT (K)*(G=TL)/(P*G)
G = G=TL
k = k+1
TL = TL*FLOAT (K+1)/YB
G = G, 110
116 TL = TL*YR/FLAT (KS+i)
G = GS*TL
TERM = TS*E*G/(GS*FLAT (K))
120 TEMP = SL"+TERM
IF(TEMP = 122, 190, 190
122 SL" = TEMP
G = G*TL
G = YR, 140
150 KS = 1, = E/Y2, + NSQAT(E)*2/4, + PayB)
KS = "AK"(KS,0)
GS = GA"(YB, KS = 1, TH)
IF(GS) 174, 174, 155
155 TS = EVAL (P, KS) * GS
G = GS
TERM = TS
k = KS
TL = TL
160 TEMP = SL"+TERM
IF(TEMP = 152, 155, 156
162 SL" = TEMP
IF(x) 155, 156, 164
164 TERM = TERM*FLOAT (K)*(G=TL)/(P*G)
G = G=TL
TL = TL*FLOAT (K+1)/YB
k = k+1
G = TS, 150
160 TL = TL*YR/FLAT (KS+i)

23
STATEMENT

100 X = X*5 + 1
110 G = G*TL
120 TERM = T*S*P*G/(G*FLAT (K))
130 TEMP = SUM + TERM
140 IF (SUM = TEMP) 172, 174, 176
172 SUM = TEMP
180 TL = TL*H/(HFLAT (K+1))
190 TERM = TERM*H/(G*FLAT (K+1))
200 G = G*TL
210 X = X + 1
220 SUM = 0, SUM
230 PN = SUM
240 GO TO 90
250 IF (N = 1) 210, 210
260 PN = DEXP (YB/(1. * X))
270 GO TO 90
280 TEMP = 1 + EXP (1EN)**X
290 PN = 1 + GYH/(1. + X)**YB/(1. + EN)**X
295 * G = GYH/TERM
300 SUM = 1
310 J = N
320 TEMP = SUM + TERM
330 IF (SUM = TEMP) 340, 350, 360
340 SUM = TEMP
350 TERM = TERM*YB*FLAT (J)
360 J = J + 1
370 GO TO 442
440 PN = 1 = GYH/(1. + X)**YB*EVAL (YB, N = 2)
445 + EVAL (YB, N = 1)*CAYR = (EN = 2.)*C/D)*SUM
450 PN = 1 + GYH/(1. + X)**YB*EVAL (YB, N = 3)
455 + EVAL (CAYR = (EN = 2.)*ALG (C)) = (1 + CAYR = (EN = 2.)*C/D)
460 PN = 1 = GYH/(1. + X)**YB**EVAL (YB, N = 3)
470 SUM = 0
STATEMENT

\[ C = \frac{2}{(P \cdot X)} \]
\[ D = \frac{I}{C} \]
\[ G = C \cdot D \]
\[ P = C \cdot V \]
\[ KS = \frac{(3 \cdot E \cdot (X \cdot D))}{2 \cdot \sqrt{((EN+1) \cdot (YD+1))}} \cdot \frac{2}{4} \cdot (YD+1) \]
\[ KS = \frac{M}{Y} (KS, n) \]
\[ KS = \frac{M}{Y} (KS, n) \]
\[ K = KS \]
\[ J = 1 \cdot KS \]
\[ FKS = KS \]
\[ IF(YD \cdot F = (1, 0)) 550, 501, 501 \]
\[ GS = 1 \cdot G \cdot (P, 2, 1, 1) = KS, T) \]
\[ IF(GS) 520, 520, 592 \]
\[ 502 \quad TS = C \cdot EXP(FKS, ALNG(C), (E \cdot FKS) \cdot ALNG(D) \cdot SUMLOG(N) = SUMLOG(KS) \]
\[ = SUMLOG(J) \cdot ALNG(GS) \]
\[ G = GS \]
\[ TERM = TS \]
\[ TL = T, I \]
\[ 510 \quad TEMP = SUM + TERM \]
\[ IF(SUM = TEMP) 512, 510, 516 \]
\[ 512 \quad SUM = TEMP \]
\[ IF(KS) 516, 516, 516 \]
\[ 514 \quad TL = TL + FLMAT (2 \cdot N \cdot K) \]
\[ TERM = TERM + FLMAT (K) \cdot (G + TL) / (FLMAT(K + N + 1) \cdot G) \]
\[ G = G + TL \]
\[ K = KS, 1 \]
\[ GS, T = 510 \]
\[ 516 \quad IF(KS + N) 516, 520, 520 \]
\[ 518 \quad TERM = TS + FLMAT (K = KS) \cdot (GS + TN) / (FLMAT(KS + 1) \cdot GS) \]
\[ G = GS + T \]
\[ TL = TN + FLMAT (2 \cdot N + 1 \cdot K) \]
\[ K = KS + 1 \]
\[ 520 \quad TEMP = SUM + TERM \]
\[ IF(SUM = TEMP) 522, 520, 520 \]
\[ 522 \quad SUM = TEMP \]
\[ IF(KS) 524, 520, 520 \]
\[ 524 \quad TERM = TERM + FLMAT (K = K) \cdot (G + TL) / (FLMAT(K + 1) \cdot G) \]
\[ G = G + TL \]
\[ TL = TL + FLMAT (2 \cdot K \cdot K) \]
\[ K = KS + 1 \]
\[ GS, T = 520 \]
\[ 526 \quad PN = SUM \]
\[ PN = G \]
\[ 550 \quad GS = G \cdot (P, 2, 1, 1) = KS, T) \]
\[ IF(GS) 570, 570, 552 \]
\[ 552 \quad TS = C \cdot EXP(FKS, ALNG(C), (E \cdot FKS) \cdot ALNG(D) \cdot SUMLOG(N) = SUMLOG(KS) \]
\[ = SUMLOG(J) \cdot ALNG(GS) \]
\[ G = GS \]
\[ TERM = TS \]
\[ TL = T, I \]

25
STATEMENT

500 TEMP = SUM*TERM
  IF(SUM=TEMP) 562,566,570
502 SUM = TEMP
  IF(K) 569,570,574
504 TL = TL*P/FLMAT (Z*1) 
  TERM = TERM*FLMAT (K)*(G=TL)/(G*FLMAT (K+1)*G)
  G = G+TL
  K = K+1
506 IF(K=5) 569,570,576
508 TERM = TERM*FLMAT (K)*(G=TL)/(G*FLMAT (K+1)*G)
  G = G+TL
  TL = TL*FLMAT (Z*1=1=K)/P
  K = K+1
510 TEMP = SUM*TERM
  IF(SUM=TEMP) 572,576,576
512 SUM = TEMP
  IF(K) 574,576,576
514 TERM = TERM*FLMAT (K)*(G=TL)/(G*FLMAT (K+1)*G)
  G = G+TL
  TL = TL*FLMAT (Z*1=1=K)/P
  K = K+1
516 TO 570
518 FN = 1.*SUM
  50 TO 9
90 IF(FN) 91,94,92
91 FN = 0.
  50 TO 94
92 IF(FN=1.) 94,94,93
93 FN = 1.
94 RETURN
99 WRITE (0,9) "F1,SNR,KASE"
9 FFORMAT (1=0)/SUM=UNREAUSABLE CALL SEQUENCE TO MRCUM, ZERO RESULT
  "F1S GIVEN //SUM = \10,5X,5FPA = E10,5X,5FPA = "
  E10,5X,5FPA = KASE = 0"
  FN = 0.
  HTO = 0.
RETURN
END
STATEMENT

FUNCTION DGAM(P, "N")
DOUBLE PRECISION: SUM, TERM, TEMP, FJ, DGAM, DEVAL, F, SUML, SUMLG
SUM = 0
K = 1
IF(K <= 1) GO TO 200, 200
J = J + 1
SUML = SUMLG(J)
TERM = DEVAL(B, J, SUML)
10 TEMP = SUM * TERM
IF(SUM * TEMP) 15, 20, 20
15 SUM = TEMP
J = J + 1
FJ = J
TERM = TERM / FJ
GM TA 10
20 DGAM = SUM
RETURN
200 J = N
SUML = SUMLG(J)
TERM = DEVAL(B, J, SUML)
30 TEMP = SUM * TERM
IF(SUM * TEMP) 35, 40, 40
35 SUM = TEMP
IF(J = 1) GO TO 40, 36, 36
36 FJ = J
TERM = TERM / FJ
J = J + 1
GO TO 30
40 DGAM = 1 + SUM
RETURN
END
STATEMENT

FUNCTION DEVAL(Y,N,SUML)
DOUBLE PRECISION XPN,EN,DEVAL, Y,SUML
XPN = Y
IF(N) 20, 20, 10
10 EN = N
XPN = XPN*EN*DLNG(Y)*SUML
20 DEVAL = DEXP(XPN)
RETURN
END
STATEMENT

FUNCTION GAM(R, N, TN)  
SL = 0  
K = 0  
IF(K<>2) 100, 200, 200  
100 J = ' ' + 1  
TERM = EVAL(R, J)  
TN = TERM*FLAT(J/8)  
10 TEMP = SL + TERM  
IF(SUM=TEMP) 15, 20, 20  
15 SL = TEMP  
J = J + 1  
FJ = J  
TERM = TERM*F/J  
GA = 1  
20 GAM = SL  
RETURN  
200 J = 1  
TERM = EVAL(R, J)  
TN = TERM  
30 TEMP = SUM + TERM  
IF(SUM=TEMP) 35, 40, 40  
35 SUM = TEMP  
IF(J=1) 40, 50, 50  
36 FJ = J  
TERM = TERM*F/J  
J = J + 1  
GO TO 31  
40 GAM = 1 + SUM  
RETURN  
END
STATEMENT

FUNCTION EVAL(y,n)
  XPRN = y
  IF(n) 2 2n+10
  FA = n
  XPRM = XPRM + EVALLog(y)+SUMLog(n)
  EVAL = EXP(XPRN)
  RETURN
END
STATEMENT

FUNCTION SIMLOG(N)
DOUBLE PRECISION A, B, SIMLOG
DIMENSION A(NMAX)
DATA CUM / 0.0/
DATA COMA / 0.0/
DATA NMAX/100/,
NMAX=100,
IF (CUM=0.0) 20, 10, 20
10 CUM = 1.
COMA = 0.
LAST = 1
A(1) = J,
20 NN = IABS (N)
IF (NN=1) 30, 50, 40
30 SUMLOG = 0
RETURN
40 IF (NN=LAST) 50, 50, 0
50 SUMLOG = A(NN)
RETURN
60 K = LAST+1
IF (NN=MAX) 70, 70, 80
70 DO 72 IMAX=1
72 A(IM) = A(IM-1) + DLOG(DFLAT(IM))
LAST = IM
GO TO 50
80 IF (LAST=MAX) H2, 90, 90
90 DO 44 IMAX=1
44 A(IM) = A(IM-1) + DLOG(DFLAT(IM))
LAST = IMAX
45 B = A(1)
K = NMAX+1
72 92 IMAX=1
92 B = B + DLOG(DFLAT(IM))
SUMLOG = B
RETURN
END
STATEMENT

SUBROUTINE MAXC(T,MAX,MAX,MAX,MAX,ANTHTG)
EXTERNAL F;
DIMENSION SX(I81),HG(I81),TV(I81),JIA(9)
DIMENSION XZ(I80),YZ(I80)
DIMENSION IFAC(5),IFAC(9)
COMMON/SH/REF,GRAD,RAH,COST,U(162)
COMMON/TYPE/X22,Y22,X11,Y11,XAA,YAA,EXEI
DIMENSION RANGI(9),RANG(9)
COMMON/AX/XX(I81),YY(I81),CTI(I81),SN1(I81),DEL
DATA EPS=""/.001/
DATA REF/000114/
DATA GRAD/0000384#22/
DATA RANG/1,.2,3,4,5,6,7,11,12,15,1,172,181/
DATA AANG/0,.5,1,.3,5,.1,10,.3,0,.69,.90,/
DATA IMPAC/6,119,500,1000,5000,10000,50000,100000/
DATA IMPAC/1,2,1,10,10,10,500/
DATA JAI/11,1,71,121,151,172/
CALL PREC=""(9)
DEL=MAX*01
IF (YMAX,LT,XMAX) DEL=MAXX,01
FA = 6078.1155PA*KXM*/(MAX*MAX)
F2 = FA*FA
FREC = "".0001
CONST = 2.532/1852,0
RAD=246%195,13*ANTHTG
AR = 1.0+REF
AR2 = AR*AR
CD = 2.0+REF+REF*REF
FLEV = "".002""
II=11
DA 29 JI=10
GA TO (190,191,192,193,194,195), JI
490 ADEL =.02
MM=2b
GA TO 51
491 ADEL =.1
MM=45
GA TO 51
492 ADEL=.5
MM=20
GA TO 51
493 ADEL =1.0
MM=25
GA TO 51
494 ADEL=2.5
MM=12
GA TO 51
495 ADEL=.9
MM=3
51 DA 29 IN = 1,IM
II=II+1
IF (x,LE,1E-11) GO TO 47
46 CALL PLOT (1,0,0,3)
   XX(1)=X
   YY(1)=Y
   X=0,
   Y=0
   Z2=D0,
   Y2=EB
   GO TO 304
47 XLAST=X
   YLAST=Y
   XMAX=XMAX
   YMAX=YMAX
   IF (-2*VE,MAX) GO TO 68
67 XX(V)=X
   YY(V)=Y
68 IF (K,LE,2) GO TO 781
780 X11=V
   Y11=V
   XMAX=X
   YMAX=Y
   GO TO 304
781 IF (X,LE,(XZ(N)+0.001)) GO TO 783
782 CALL INPOST (XLAST,YLAST,X,Y,XZ(N),YZ(N),XZ(N+1),YZ(N+1),X0,Y0)
   CALL MINTAP (X0,Y0)
   CALL PLOT (X0,Y0,2)
   IF (2*E2,MAX) GO TO 785
   GO TO 304
783 IF (K,LE,1.1) GO TO 788
787 CALL MINTAP (X,Y)
   CALL PLOT (X,Y,2)
   XCNR=X
   YCNR=Y
   GO TO 304
788 CALL MINTAP (X,Y)
   CONTINUE
304 CONTINUE
305 M=M+2
30 CONTINUE
   GO TO 789
785 CALL PLOT (XZ(1),0,3)
   GO 786
   X=X(1)
   YY(N)=Y
   YY(N)=Y
   CALL PLOT (X,Y,2)
   CALL PLOT (X0,Y0,2)
   XCNR=X0
   YCNR=Y0
789 X=0,
   Y=YMAX
   CALL PLOT (X,Y,3)
   Y=0,3
STATEMENT

CALL PI insufficient
CALL PI insufficient
CALL REASON(10)

IF (MAX, MAX, 100.0) INT = 5
IF (MAX, GT, 300.0) INT = 25

IF 31 XA = INT

G = XA
A = RG * XA
B = BAG

IF XA < 1, 101
X = 1

IF (KC, NE, 181) G = 10 TO 72

91 CALL R (0, 1, 2)

G = TO 31

92 XA = XE(KC) / XA
IF (X = GT, VX(1)) G = 31

VX = VX(1) / XA

73 IF (Y, LE, (YH(KC) * 1000)) G = TO 72

74 X1 = XE

75 Y1 = YK

G = TO 732

731 X1 = XE(KA = 1)

732 X2 = XE(KA = 1)

Y2 = YHE

733 XA = X

YV = Y

CALL INRST(X1, Y1, X2, Y2, XA, Y1, XB, Y0, Y0)

CALL PLT(X0, Y0, 2)
G = TO 31

72 XA = X

75 G = TO 80

85 CALL PLT(X0, Y0, 3)

86 IF (KC, NE, 2) G = TO 801

861 X1 = X

Y1 = Y

X1 = X

Y1 = Y
G = TO 36

781 CALL R (X, Y)

94 CONTINUE

31 CONTINUE
STATEMENT

710 CALL ATICK(1,26,5,9,5)
CALL ATICK(1,21,1,9,10)
CALL ATICK(1,21,2,9,5)
CALL ATICK(1,21,1,9,5)
CALL ATICK(1,21,2,9,2)
CALL ATICK(1,21,2,9,2)
CALL ATICK(1,21,1,9,2)
DO 34 KF = 1, 6
  IF(JA = J) THEN
    CALL PLOT(X,Y,Z)
    X = XX(NF)
    Y = YY(NF)
  CALL PLOT(X,Y,2)
  CONTINUE
  KT = 9
  CM 364 KU = 1, H00
  KT = KT + 1
  IF(KT,N=10) GO TO 369
368 FAC = 2,0
  KT = 0
  GO TO 370
369 FAC = 1,0
370 P2 = XU = Y
  IF(X GT (XMAX + 0.001)) GO TO 365
  Y = 0,0
  CALL PLOT(X,Y,Z)
  Y = FAC + DEL
  CALL PLOT(X,Y,2)
384 CONTINUE
385 KS = 9
  KJM = HMAX / HINT + 1,001
  CM 37 KJ = 1, HJM
  KS = KS + 1
  IF(KS,N=10) GO TO 374
375 FAC = 2,0
  KS = 0
  GO TO 377
376 FAC = 1,0
377 H = HINT * (KJ + 1)
  Y = HMAX/HMAX
  X = 0,0
  CALL PLOT(X,Y,Z)
  X = FAC + DEL
  CALL PLOT(X,Y,2)
37 CONTINUE
CALL RCHG(11)
IF(IXMAX=YMAX) 460,460,461
460 SFAC = MAX * .125
STATEMENT

60 T0 102
40 T0 SFACEV=2*x,125
50 x=175*SFAC
40 100 I=-1,5
50 N=IPTRAX / I RFAC (15)
60 IF (NR GT 10) G0 TO 100
101 I UNIT = IRFAC (1H)
60 TO 102

101 CONTINUE
20 I UNIT = IRFAC (5)
30 10 10 I=-1,5
20 N=IPTRAX / IRFAC (1H)
30 IF (NR GT 10) G0 TO 100
103 I UNIT = 1 - FAC (1H)
60 TO 128

100 I UNIT = 1 - FAC (6)
128 = 505*SFAC
10 Y=505*SFAC
20 CALL I UNP (X, Y, H, M, 0, 0)
30 = I UNIT
120 IG D ITS=ALMG10 (FLAT (N)) + 1,000001
20 CALL SENTER (N, IG DITS, IG DITS, BIAS)
30 X = (N, IPTRAX) = XMAX = BIAS
40 IF (X + BIAS, ST, XMAX) G0 TO A01
20 CALL I UNP (X, Y, H, M, 0, 0)
30 = N = I UNIT
60 TO 120

201 Y=1,0*SFAC
10 CALL SENTER (N, 21, 21, BIAS)
20 X=0,5 + XMAX = BIAS
30 CALL SENTER (X, Y, H, 21 RANGE, NAUTICAL MILES, 0, 0, 21)
40 X=0*SFAC
50 Y=0*SFAC
60 CALL I UNP (X, Y, H, M, 0, 0)
70 = I UNIT
121 IG D ITS=ALMG10 (FLAT (N)) + 3,000001
20 X=0,15*IG DITS*SFAC
30 Y= (Y, IPTRAX) = YMAX = 0,75*SFAC
40 IF (Y, ST, YMAX) G0 TO A03
20 CALL I UNP (X, Y, H, M, 0, 0)
30 = N = I UNIT
60 TO 121

A03 X = -1,0*SFAC
10 CALL SENTER (N, 33, 33, BIAS)
20 Y=0,5 + YMAX = BIAS
30 CALL SENTER (X, Y, H, 33 TARGET HEIGHT ABOVE ANTE NNA, FEET, 90, 33)
40 X=3,0
50 CALL SENTER (X, 26, 26, BIAS)
60 CALL SENTER (X, Y, H, 26 ANTE NNA HEIGHT FEET, 90, 26)
70 Y=14,5714*H
80
CALL NUMBER (X1, Y2, M, A, ANG, GT, 90, 1)

38
STATEMENT

FUNCTION F1(Y)
C=V*HEF, GRAD, END, CONST, U(152), N
HEF=EXP(-GRAD*X)
C=V/34
V=2.0*HEF
=2.0*CC+CC
FX = SQRT (U(N) + V + A + V*A)
F1 = CONST * (1.0 *V)/(1.0 + CC)/FX
END
SUBROUTINE SIMCO(X1, XEND, TEST, LIM, AREA, NCI, V, F)

IMPLICIT none

*DO=0,0
INT=1
V=1,0
EVEN=0,0
AREA=1,0
10 ENDS#F(X1)+F(XEND)
2 HM=(XEND-X1)/V
ODD=EVEN+MOD
X=(1+H)/2,
EVEN=0,0
10 3 I=1,INT
21 EVEN=EVEN#F(X)
X=X+H
3 CONTINUE
31 AREA=(ENDS+4.0#EVEN+2.0#ODD)*H/6.0
NCI=NCI+1
34 R=ABS((AREA-AREA)/AREA)
IF(R=TEST)341,35,35
341 IF(R=TEST)35,35,4
35 RETURN
4 AREA=AREA
46 INT=2*INT
V=2.0#V
GO TO 2
END

40
STATEMENT

SUBROUTINE ATICM (IA,JA,KA,NI,MC)
COMMON /X/ XX(1A1), YY(1A1), CT1(1E1), S1(1A1), DEL

MA = MR
GO TO 1
MA = MA + 1
IF (MA .GE. MC) GO TO 3
2 FAC = 3.0
MA = 0
GO TO 0
3 FAC = 1.0
4 X = XX(MA)
Y = YY(X)
CALL PLOT(X,Y,3)
X = X + DEL*FAC*CT1(Y)
Y = Y + DEL*FAC*S1(X)
CALL PLOT(X,Y,2)
1 CONTINUE
END
SUBROUTINE SENTER (\( m \), \( v \), \( l \), \( A^{\prime} \))
\[ C^{\prime} = 0.257143 \times \]
\[ A^{\prime} = \left( 3 \times (i-1) \right) \times C^{\prime} \]
\[ A^{\prime} = \left( 3 \times (i) \right) \times C^{\prime} \]
\[ A^{\prime} = 0.5 \times \left( 3 \times (i) \right) \]
END
SUBROUTINE MIXTAP (X,Y)
COMMON /FRED/ X2, Y2, X1, Y1, XA, YA, ERROR
COMMON /FPAD/ XA
DATA X/0/ DATA Y/0/
DATA U, V
DX(U1,X1,U2,Y2) = SQRT((U2-U1)**2 + (V2-V1)**2)
D1 = DX(X1,Y1,X2,Y2)
IF (D1 .EQ. 0.) GOTO 11
U2 = DX(X,Y,X1,Y1)
IF (D2 .EQ. 0.) GOTO 2
D3 = DX(X,Y,X2,Y2)
D1 = DX(X1,Y1,X2,Y2)
IF (D3 .LT. D1) GOTO 1
C*S** = (C1*C2-C1*D2+C1*D1)/(2.*C1*D2)
IF (C*S** .GT. 1. .OR. C*S** .LT. -1.) C*S** = 1.
SINE = SQRT(1. - C*S***C*S**)
DEVN = D2*SINE
IF (DEVN .LE. ERROR) GOTO 2
1 CALL PLOT (XA, YA, Z)

11 X1 = X
Y1 = Y
2 XA = X
YA = Y
Z = N+1
END
FORMULATE SHARE
(*HZ, PSI, IPOL, RHO, PHI)

COMPLEX EPSIL, GAM, SGTERM, TERM

DATA FLAST, 0, /.

SINPSI = SIN (PSI)

CSPSI = COS (PSI)

IF (FHMZ * EPSIL) GT 200

FLAST = FHMZ

+ = 299.793 / FHMZ

IF (FHMZ, 1500,) GT TO 151

150 SIG = 0.3

EPSIL = 90

GO TO 155

151 SIG = 0.3 * (FHMZ = 1500,) * .00148

IF (FHMZ, 1500,) GT TO 154

153 EPSIL = 90 = (FHMZ = 1500,) * .00733

GO TO 155

154 EPSIL = 90, = (FHMZ = 3000,) * .002429

SIG = 0.52 = (FHMZ = 3000,) * .001314

155 EPSIL = COMPLEX (EPSIL, 60, **SIG)

200 SGTERM = CSC (EPSIL, CSPSI)

IF (IPOL, NE, 1,) GO TO 161

160 TERM = EPSIL * SINPSI

GAM = (TERM - SGTERM) / (TERM + SGTERM)

GO TO 160

161 GAM = (SINPSI - SGTERM) / (SINPSI + SGTERM)

170 GAM = CABS (GAM)

PHI = ATAN2 (AIMAG (GAM), REAL (GAM))

RETURN

END
SUBROUTINE DEGREE(X,Y,Z)
DIMENSION DELX(8), DELY(8)
DATA DELX/0,7071,1,0,0,7071,0,0,7071,1,0,0,7071,0,0/
DATA DELY/0,7071,0,0,0,7071,1,0,0,7071,0,0,0,7071,1,0/
C = M +1.42114
X = X + S*M
Y = Y + S*O
CALL PLOT(X,Y,3)
* 1 II = 1, 8
X = X + DELX(II)*D
Y = Y + DELY(II)*D
1 CALL PLOT(X,Y,2)
END
SUBROUTINE INTERSECT (X1, Y1, X2, Y2, XA, YA, XB, YB, X0, Y0)

1. IF (ABS(X2-X1), LT, 10, E=78) G0 TO 2
2. IF (ABS(XB-XA), LT, 10, E=78) G0 TO 99

X0 = X1
S = (YL-YA)/(XB-XA)
Y0 = S*(X0-XA) + YA
RETURN

2. IF (ABS(XA-XB), LT, 10, E=78) G0 TO 4

S = (Y2-Y1)/(X2-X1)
Y0 = S*(X0-X1) + Y1
RETURN

4. S1 = (YB-YA)/(XB-XA)
S2 = (Y2-Y1)/(X2-X1)

IF (S1 < S2, 0.) G0 TO 5

IF (ABS(S2-S1), LT, 10, E=78) G0 TO 99

XT = S2/S1
Y0 = (Y2+S2*(X4-X2)-YA+TH0)/(1,=TH0)
X0 = (Y0-YA)/S1 + XA
RETURN

5. IF (S2 .EQ., 0.) G0 TO 99

X0 = (Y1-Y2)/S2 + X2
RETURN

RETURN

PRINT 105, X1, Y1
PRINT 101, X2, Y2, XA, YA, XB, YB
X0 = 0.
Y0 = 0.

100 FORMAT (3X, 6H1, CALL TO INTERSECT ADDED, LINES PARALLEL, NO INTERSECT)

*CT10X, X1 = ,E10,2, Y1 = ,E10,2
101 FORMAT (4X, 6H2, X2 = ,E10,2, Y2 = ,E10,2,
* 6H3, XA = ,E10,2, YA = ,E10,2, XB = ,E10,2, YB = ,E10,2 //)
END
SUBRUTINE ARRA (X1, Y1, X2, Y2, TIPL)
COMMON /ANG/ ANG
DATA ANG / °35/
YA = Y2 = Y1
XA = X2 = X1
A = ATAN2(YA,XA)
A3 = A + ANG
A4 = A + ANG
X3 = X2 - TIPL * COS(A3)
Y3 = Y2 - TIPL * SIN(A3)
X4 = X2 - TIPL * COS(A4)
Y4 = Y2 - TIPL * SIN(A4)
CALL PLOT (X1, Y1, 3)
CALL PLOT (X2, Y2, 2)
CALL PLOT (X3, Y3, 2)
CALL PLOT (X4, Y4, 3)
CALL PLOT (X2, Y2, 2)
END

SUBRUTINE REZERO
ENTRY PEING
ENTRY DASHN
ENTRY DASHF
END
APPENDIX B: Input Data Card Formats for LOBMUF (For ASC - Jan 1977)

Program LOBMUF will plot the envelope of N radars using the pattern calculations contained in the original program described in NRL Report 7098 by L. Blake. Incoherent integration is assumed. An antenna height must be specified in subroutine RHACHT. The parameter name for this antenna height is ANTHITE and is used to draw the range-height grid. The actual antenna heights for the individual radars are input separately and used for the multipath calculations. ANTHITE is not very important except for the labeling of the height axis of the grid.

Input Card Sequence

1. ANTHITE in F10.0 field
2. Label Card
3. Grid Parameters in F10.0 fields:
   a. XMAX 1-10
   b. YMAX 11-20
   c. RMAX 21-30
   d. HMAX 31-40
   e. THMIN 41-50
   f. THMAX 51-60
   g. WHFT 61-70
   h. RDR 71-80

XMAX = maximum X-dimension of chart in inches
YMAX = maximum Y-dimension of chart in inches
RMAX = maximum range of chart in nautical miles
HMAX = maximum height on chart in feet
THMIN = minimum elevation angle for the plot
THMAX = maximum elevation angle for the plot
WHFT = sea wave height in feet
RDR = number of radars for which an envelope is to be plotted.

4. Common Parameters in F10.0 fields:
   a. PDT 1-10
   b. PFA 11-20
   c. PULS (total number of pulses) 21-30
   d. CASE 31-40
   e. AHFT 41-50
   f. TILT 51-60
   g. POL 61-70
   h. CSC 71-80

PDT = probability of detection
PFA = false alarm exponent, i.e., the positive value of the exponent (power of ten); for \(10^{-6}\), enter the number 6.0, etc.
PULS = total number of pulses

48
CASE = Swerling case number
AHFT = antenna height in feet
TILT = tilt angle of the antenna beam maximum with respect to the horizon in degrees.

For reference, POL is polarization as follows:

POL = 1. vertical
POL = 2. horizontal

CSC is for pencil or cosecant squared antenna pattern:

CSC = 0. pencil beam
CSC = 1. cosecant squared beam

5. Radar Parameters in FL0.0 fields:
   a. RFS = calculated or assumed free-space range of the radar on the specified target
   b. FMHZ = radar frequency in megahertz
   c. BWD = antenna half-power beamwidth in degrees
   d. SLDB = first elevation sidelobe level relative to the main lobe
   e. PULNUM (number pulses at FMHZ) = number of pulses at each specified frequency
   f. FREF = frequency used to calculate RFS if RFS was calculated for the total number of pulses (PULS)
      FREF = 0, if RFS was calculated at each frequency using PULNUM.

Repeat card 5 for N repetitions.

RFS = calculated or assumed free-space range of the radar on the specified target
FMHZ = radar frequency in megahertz
BWD = antenna half-power beamwidth in degrees
SLDB = first elevation sidelobe level relative to the main lobe
PULNUM = number of pulses at each specified frequency
FREF = frequency used to calculate RFS if RFS was calculated for the total number of pulses (PULS)
FREF = 0, if RFS was calculated at each frequency using PULNUM.

After cards 1-5, the sequence can be repeated as many times as desired, starting with card 2.

A note of caution is that the elevation plot angle increment is determined by

\[
\frac{\theta_{\text{MAX}} - \theta_{\text{MIN}}}{2000}
\]

The value has proven adequate for S-band radars where 10° and 0° were the values for \( \theta_{\text{MAX}} \) and \( \theta_{\text{MIN}} \). It is recommended that an envelope for a single radar (RDR = 1) be run when there is doubt that this angle increment is small enough.