PROJECT APOLLO
TASK MSC/TRW E78

VHF ANTENNA SIMULATION PROGRAM (VHFASP)
HV014B USER'S MANUAL

HCC REPORT NO. 3145.30-012 1 JULY 68

Prepared By ____________________
Carl A. Argila
Communications Systems
Programming Group

Approved By ____________________
D. A. Schnebly, Project Leader
Communications Systems
Programming Group

Approved By ____________________
W. P. Bennett, Head
Subsystems Analysis Section

ABSTRACT

The VHF Antenna Simulation Program (VHFASP) simulates the CSM-LM communication link for all Apollo missions. In the standard configuration VHFASP prints values for CSM and LM antenna gains, polarization losses, CSM-LM range and gain product as functions of time. In an optional mode, every variable involved in the simulation may be printed out. A plot tape is generated for use with the TRWPLT general plotting program. VHFASP is written in FORTRAN V for use with the SRU 1108 system.

PREFACE

The VHF Antenna Simulation Program presents a classic example of how the logic of a computer program may be altered so as to swap memory size for run time and vice versa. The basic problem in simulating the CSM-LM VHF antenna system is this: A portion of a trajectory tape must be processed four times for each simulation. Each time this tape is processed, random reference must be made to one of two sets of CSM antenna patterns and one of two sets of LM antenna patterns. Each set of antenna patterns consists of six 90-row by 180-column matrices of integers. Now, from the point of view of minimum run time, we might attempt to store all simulation data into memory at one time. This would required a memory size in excess of 24 x 90 x 180 = 388,800 cells. On the other hand, from the point of view of minimum memory size we might attempt to store all simulation data on tape and then scan these tapes each time data is required. This method would increase run time to the order of several minutes per time-point.

Under a single processor system, program efficiency is primarily determined by run time (since run time is the prime cost factor). Hence we have attempted to minimize run time while remaining within the practical limitations of memory size. To solve this problem we observed that each 90 x 180 antenna pattern matrix consisted of integers less than 63 (77_8). Hence a single 36-bit word could store six such integers provided they were "packed" into six-bit intervals. A 90 x 180 matrix could then store an entire set of antenna pattern data. Now it is quite practical to store two 90 x 180 matrices (32,400 words) into memory at one time. Also, it is practical to alter the logic of the program so as to "juggle" four sets of antenna patterns so that only two need be in memory at any one time.

This combination of tape manipulation with "packed" memory storage has enabled us to minimize run time (to approximately two seconds per time-point) while remaining within the memory size limitations of the SRU 1108 system.

I wish to acknowledge the assistance of Mr. R. E. Meeker without whose help it would have been impossible to complete the VHF antenna simulation program in this form.

CONTENTS

1. PROBLEM DESCRIPTION

Both the LM (Lunar Module) and the CSM (Command Service Module) have two VHF antennae each of which may be used at any time to establish a communication link between the two vehicles. The radiation patterns generated by these four antennae depend upon the frequency of the communication link ("High Frequency," 296.8 MHz and "Low Frequency," 259.7 MHz). Since the geometry of the LM is different in the ascent and descent cases, the LM antenna patterns also depend upon the ascent/descent status of the vehicle.

We wish to develop a computer program which will simulate the CSM-LM VHF communication links for all Apollo missions given the appropriate CSM and LM antenna pattern data and the CSM and LM trajectory data. When the CSM and/or LM are "close" to the moon we wish to consider the multipath effects caused by lunar reflection, otherwise we wish to neglect the lunar reflection problem.

The CSM and LM antenna gains, polarization gains, CSM-LM range and gain product should be printed out as functions of time. The program should have the option of printing out every variable involved in the simulation, at each time-point. A plot tape for use with the TRWPLT general plotting program should be generated.

2. INPUT DATA

The data for this program is entered via the FORTRAN "namelist" $INPUT. The symbols "$INPUT" must appear in columns two thru seven of the first data card. Input parameters may then be entered in any order beginning in any column (except column one) in the form PARAM1=xxxxx, PARAM2=xxxxx,... The only restrictions are that no punch ever appear in column one and that no number be split between two data cards. After all the desired input parameters have been entered (not every parameter need be entered) a card containing the symbols "$END" in columns two thru five must appear immediately after the last data card; this signifies the end of input data. Following the "$END" card up to five Comment/Description cards may appear. These cards will be printed as headings to the output listing for this particular data case. After the Comment/Description cards another "$INPUT" card may appear for a second data case. Any number of data cases may be entered in this manner.

2.1 DETAILED DESCRIPTION OF INPUT PARAMETERS

There are nine types of input parameters; seven of these are specified by a single value and two are specified by an array of up to four values. Each parameter has the value(s) entered for it; if no value is entered a certain value is assumed for the parameter as indicated below.

Mnemonic(s)	Description
FREQCY	Frequency flag (-1,1) -1 indicates low frequency case (259.7 MHz). 1 indicates high frequency case (296.8 MHz). FREQCY initialized to -1.
ADFLAG	Ascent/Descent flag (-1,1) -1 indicates descent case (LM descending). 1 indicates ascent case (LM ascending). ADFLAG initialized to -1.
MULTIP	Multipath flag (0,1) 0 indicates no multipath calculations desired. 1 indicates multipath calculations are to be executed. MULTIP initialized to 1.
PRINT	Print flag (0,1) 0 indicates only standard print output is desired 1 indicates extra print output is desired. PRINT initialized to 0.

Mnemonic(s)	Description
PLOT	Plot flag (0,1) 0 indicates no plot tape is to be generated. 1 indicates plot tape is to be generated. PLOT initialized to 1.
COMENT	Comment/Description cards (1,...,5). Number of Comment/ Description cards to follow "$END" card. COMENT initialized to 0.
TMIN(1),...,TMIN(4)	Minimum time-point on trajectory tape to be considered, in days, hours, minutes and seconds, respectively. TMIN(1),...,TMIN(4) initialized to 0, 0, 0, 0.
TMAX(1),...,TMAX(4)	Maximum time-point on trajectory tape to be considered, in days, hours, minutes and seconds, respectively. TMAX(1),...,TMAX(4) initialized to 0, 0, 0, 0.
DELTAT	Minimum time increment (in seconds) between time-points. DELTAT initialized to 0.

2.2 INPUT TAPES

The antenna patterns tape is imbeded as a second file on the program PCF tape and for all pratical purposes may be ignored by the user. Care must be taken, however, to follow the CUR operations given in section 5 precisely so that this tape is properly positioned for program execution.

The trajectory tape, whose format is specified in Appendix B, is mounted on unit "E" in the 800 BPI density mode.

2.3 RESTRICTIONS

A maximum of 200 time-points are allowed. This means that between a specified TMIN (in days, hours, minutes and seconds) and a specified TMAX with a specified DELTAT, there can exist at most 200 trajectory time-points. Larger numbers of time-points may be processed by multiple, 200 time-point, data cases.

In the event that the user attempts to process more than 200 time-points a message will be printed out and only the first 200 time-points will be processed.

3. OUTPUT DATA

Data output for this program is in the form of standard and optional printed outputs and plot tape output.

3.1 STANDARD PRINTED OUTPUT

For each data case processed an input parameter listing is printed giving the values of each of the eight input parameters.

For each data case processed an output listing is printed giving the range between the CSM and LM (in nautical miles), the net gain product (in db), the CSM antenna gain (both direct and specular), the LM antenna gain (both direct and specular) and the polarization gain (both direct and specular) as functions of time. Four successive print outputs are given, one for each of the four possible communication links which exist at a given frequency and at a given LM ascent/descent status.

Each of the above listings is headed by the Comment/Description cards entered for that particular data case.

3.2 OPTIONAL PRINTED OUTPUT

When the print flag is on (i.e. PRINT=1) each of the intermediate variables involved in the VHF antenna simulation will be printed out at each time-point. These variables are described below.

Mnemonic(s)	Description
TIME	Time, in seconds, of data case.
ROCB	Position vector of CSM in basic coordinates.
ROLB	Position vector of LM in basic coordinates.
ACB	Direction cosine matrix of CSM.
ALB	Direction cosine matrix of LM.
IFLAG	Flag indicating data case classification. 1 indicates a "bounce-point" exists, but only direct path calculations are to be executed. 2 indicates that the LM and CSM are on opposite sides of the Lunar horizon. No link exists. 3 indicates that the LM and CSM are on diametrically opposite sides of the Moon. No link exists. 4 indicates that the LM and CSM lie along a lunar radius on the same side of the moon. Only direct path calculations are executed.

Mnemonic(s)	Description
	5 indicates "bounce-point" was not determined to the desired accuracy, however only direct path calculations were desired. 6 indicates "bounce-point" exists and multipath calculations are to be executed. 7 indicates that the LM and CSM are on opposite sides of the Lunar horizon. No link exists. 8 indicates that the LM and CSM are on diametrically opposite sides of the Moon. No link exists. 9 indicates that the LM and CSM lie along a lunar radius on the same side of the moon. Only direct path calculations are executed even though multipath calculations are desired. 10 indicates that the "bounce-point" was not determined to the desired accuracy. No calculations are executed. In general, IFLAG=1,2,3,4,5 indicates that the program is in the direct path mode whereas IFLAG=6,7,8,9,10 indicates that the program is in the multipath mode.
ROSB	Position vector of "bounce-point" in basic coordinates.
THTAC	Theta look angle of CSM along direct path.*
PHIC	Phi look angle of CSM along direct path.
THTACS	Theta look angle of CSM along specular path.
PHICS	Phi look angle of CSM along specular path.
THTAL	Theta look angle of LM along direct path.
PHIL	Phi look angle of LM along direct path.
THTALS	Theta look angle of LM along specular path.
PHILS	Phi look angle of LM along specular path.
PATERN	4 x 6 matrix representing the antenna pattern values for this time-point. Rows one and two represent pattern values for the CSM antenna along the direct and specular paths, respectively. Rows three and four represent pattern values for the LM antenna along the direct and specular paths, respectively. Columns one through six represent the P_θ, P_ϕ, $P_{\theta+45^o}$, $P_{\phi+45^o}$, P_{LCP}, P_{RCP} pattern values, respectively.
R	Range (in nautical miles) between the CSM and LM.
RHO	Polarization angle along the direct path.
TAU1S	Polarization ellipse tilt angle at CSM along direct path.
TAU2S	Polarization ellipse tilt angle at LM along direct path.
BETA	Coordinate system misalignment along the direct path.
G1	CSM antenna gain along the direct path.**

* all angles expressed in radians
** all gains expressed as a ratio

Mnemonic(s)	Description
G2	LM antenna gain along the direct path.
RR1	Axial ratio at CSM along the direct path.
RR2	Axial ratio at LM along the direct path.
LP	Polarization gain along the direct path.
R1S	Distance between CSM and "bounce-point."
R2S	Distance between LM and "bounce-point."
RS	= R1S + R2S.
DELTAR	= RS - R.
PSI	Grazing angle.
RHO1S	Polarization angle along the specular path CSM to LM.
RHO2S	Polarization angle along the specular path LM to CSM.
TAU1S(1), TAU1S(2)	Polarization ellipse tilt angle of transmitting antenna along the specular paths CSM-LM and LM-CSM, respectively.
TAU2S(1), TAU2S(2)	Polarization ellipse tilt angle of receiving antenna along the specular paths CSM-LM and LM-CSM, respectively.
RA	Reflection coefficient for the electric field component parallel to the plane of incidence. RA is complex.
RB	Reflection coefficient for the electric field component perpendicular to the plane of incidence. RB is complex.
G1S	CSM antenna gain along the specular path.
G2S	LM antenna gain along the specular path.
RR1S	Axial ratio at CSM along the specular path.
RR2S	Axial ratio at LM along the specular path.
P1S(1), P1S(2)	Polarization ratio of incident wave along the specular paths CSM-LM and LM-CSM, respectively. P1S is complex.
P2S(1), P2S(2)	Polarization ratio of reflected wave along the specular paths CSM-LM and LM-CSM, respectively. P2S is complex.
Q	Depolarization factor. Q is complex.
TAUS(1), TAUS(2)	Tilt angle of reflected wave along the specular paths CSM-LM and LM-CSM, respectively.
RRS(1), RRS(2)	Axial ratio of reflected wave along the specular paths CSM-LM and LM-CSM, respectively.
BETAS(1), BETAS(2)	Coordinate system misalignment along the specular paths CSM-LM and LM-CSM, respectively.
LPS(1), LPS(2)	Polarization gain along the specular paths CSM-LM and LM-CSM, respectively.

Mnemonic(s)	Description
LR(1), LR(2)	Lunar reflection gain along the specular paths CSM-LM and LM-CSM, respectively.
D	Multipath divergence factor.
K	Multipath reflection factor.
GP	Gain product.

3.3 PLOT TAPE OUTPUT

When the plot flag is on, a plot tape will be generated on unit "F" (FORTRAN I. D. Number 8). The plot tape will consist of four files per data case. The files will be in the following order, for each data case:

File No.	Link No.	Description
1	1	CSM antenna A, LM antenna 1.
2	2	CSM antenna A, LM antenna 2.
3	3	CSM antenna B, LM antenna 1.
4	4	CSM antenna B, LM antenna 2.

Each file on the plot tape consists of one record per time-point. Each record on the plot tape consists of ninety-six words as described below. The plot tape is a FORTRAN binary tape written in the 800BPI density mode in the format required by the TRWPLT general plotting program.

Mnemonic(s)	Word(s)	Description
TIME	1	See section 3.2
ROCB	2 - 4	See section 3.2
ROLB	5 - 7	See section 3.2
ACB	8 - 16	See section 3.2
ALB	17 - 25	See section 3.2
IFLAG	26	See section 3.2
ROSB	27 - 29	See section 3.2
THTAC	30	See section 3.2
PHIC	31	See section 3.2
THTACS	32	See section 3.2
PHICS	33	See section 3.2
THTAL	34	See section 3.2
PHIL	35	See section 3.2

Mnemonic(s)	Word(s)	Description
THTALS	36	See section 3.2
PHILS	37	See section 3.2
R	38	See section 3.2
RHO	39	See section 3.2
TAU1	40	See section 3.2
TAU2	41	See section 3.2
BETA	42	See section 3.2
G1	43	See section 3.2
G1DB	44	CSM Antenna gain along the direct path in db.
G2	45	See section 3.2
G2DB	46	LM Antenna gain along the direct path in db.
RR1	47	See section 3.2
RR2	48	See section 3.2
LP	49	See section 3.2
LPDB	50	Polarization gain along the direct path in db.
R1S	51	See section 3.2
R2S	52	See section 3.2
RS	53	See section 3.2
DELTAR	54	See section 3.2
PSI	55	See section 3.2
RHO1S	56	See section 3.2
RHO2S	57	See section 3.2
TAU1S(1), TAU1S(2)	58 - 59	See section 3.2
TAU2S(1), TAU2S(2)	60 - 61	See section 3.2
RA	62 - 63	See section 3.2
RB	64 - 65	See section 3.2
G1S	66	See section 3.2
G1SDB	67	CSM antenna gain along the specular path in db.
G2S	68	See section 3.2
G2SDB	69	LM antenna gain along the specular path in db.
RR1S	70	See section 3.2
RR2S	71	See section 3.2
P1S(1), P1S(2)	72 - 75	See section 3.2

Mnemonic(s)	Word(s)	Description
P2S(1), P2S(2)	76 - 79	See section 3.2
Q	80 - 81	See section 3.2
TAUS(1), TAUS(2)	82 - 83	See section 3.2
RRS(1), RRS(2)	84 - 85	See section 3.2
BETAS(1), BETAS(2)	86 - 87	See section 3.2
LPS(1), LPS(2)	88 - 89	See section 3.2
LPSDB	90	Minimum specular polarization gain in db.
LR(1), LR(2)	91 - 92	See section 3.2
D	93	See section 3.2
K	94	See section 3.2
GP	95	See section 3.2
GPDB	96	Gain product in db.

4. SAMPLE CASE

Two data cases (one with multipath calculations and one without) are included as a sample case. In order to demonstrate usage of the TRWPLT general plot program both of these data cases have been used to generate plots.

4.1 SAMPLE CODING FORM

DATE 7-1-68
NAME ARCTILA
EXT 2503
NO OF CARDS 38

PRIORITY
PROBLEM NO H36101
SPECIAL CHARACTERS

TRW SYSTEMS
HOUSTON COMPUTING CENTER
SYMBOLIC AND FORTRAN CODING FORM

CARD STOCK: ☑ PLAIN Yellow
☐ FORTRAN SOURCE
☐ 7094 SYMBOLIC

PAGE 1 OF 2
KEYPUNCHED BY 56
VERIFIED BY 187

SYMBOL | OPERATION | ADDRESS TAG DECREMENT | COMMENTS | SEQUENCE
C NUMBER | * | FORTRAN STATEMENT

```
$INPUT ADFLAG=1,CØMENT=1,TMIN(1)=4,2,56,TMAX(1)=4,3,58,20,DELTAT=30,MULTIP=0
$END
                              TEST CASE 1
$INPUT ADFLAG=1,FREQCY=1,TMIN(1)=4,2,56,TMAX(1)=4,3,58,20,DELTAT=30,PRINT=1,
PLØT=0,CØMENT=1
$END
                              TEST CASE 2
∇ XQT TRWPLT
 NØIREC = 1
 KDAY = 1
 XLØ = 356100.0
 XHI = 359900.0
 YLØ = -60.0
 YHI = +10.0
 PLØT=1,44,ENDLST
 TITLE=ID=CSM ANTENNA GAIN
 XLABEL=ID=TIME(D.H.M.S)
 YLABEL=ID=DECIBELS
 ENDPLT
 PLØT=1,46,ENDLST
```

*FORTRAN STATEMENT CONTINUATION

TRW SYSTEMS
HOUSTON COMPUTING CENTER
SYMBOLIC AND FORTRAN CODING FORM

DATE ______ NAME ______ EXT ______ NO OF CARDS ______

PRIORITY ______ PROBLEM NO ______ SPECIAL CHARACTERS: ______

CARD STOCK: ☐ PLAIN ______ ☐ FORTRAN SOURCE ☐ 7094 SYMBOLIC

PAGE 2 OF 2 KEYPUNCHED BY ______ VERIFIED BY ______

SYMBOL / C NUMBER | * | OPERATION / FORTRAN STATEMENT | ADDRESS TAG DECREMENT | COMMENTS | SEQUENCE

```
      TITLE=ID=LM ANTENNA GAIN
      ENDPLT
      PLØT=1,50,ENDLST
      TITLE=ID=PØLARIZATIØN GAIN
      ENDPLT
      PLØT=1,96,ENDLST
      TITLE=ID=GAIN PRØDUCT
      ENDPLT
      PLØT=1,38,ENDLST
      YLØ = 0.0
      YHI = 120.0
      YLABEL=ID=NAUTICAL MILES
      TITLE=ID=RANGE BETWEEN CSM AND LM
      ENDPLT
      ENDPHA
      REPEAT
      REPEAT
      REPEAT
      ENDJØB
```

* FORTRAN STATEMENT CONTINUATION

4.2 SAMPLE PRINT OUTPUT

VHF ANTENNA SIMULATION
TEST CASE 1

PARAMETER LISTING

```
MINIMUM TIME (D.H.M.S)          =  4. 2.56.  .0
MAXIMUM TIME (D.H.M.S)          =  4. 3.58.20.0
TIME INCREMENT (SECONDS)        =          30.0
FREQUENCY FLAG (-1,1)           =            -1
ASCENT/DESCENT FLAG (-1,1)      =             1
MULTI-PATH FLAG (0,1)           =             0
PRINT FLAG (0,1)                =             0
PLOT FLAG (0,1)                 =             1
COMMENT/DESCRIPTION CARDS       =             1
```

VHF ANTENNA SIMULATION
TEST CASE 1

TIME (D.H.M.S)	RANGE (NM)	G-PRODUCT (DB)	CSM GAINS (DB) DIRECT	CSM GAINS (DB) SPECULAR	LEM GAINS (DB) DIRECT	LEM GAINS (DB) SPECULAR	POLARIZATION GAINS (DB) DIRECT	POLARIZATION GAINS (DB) SPECULAR
LINK NO. 1. CSM ANTENNA 1, LEM ANTENNA 1.								
4. 2.56.31.7	83.91	-15.95	-1.78		-10.86		-3.31	
4. 2.58.31.7	86.53	-11.42	-2.48		-8.61		-.33	
4. 3. 0.31.7	89.97	-15.15	-3.12		-9.86		-2.17	
4. 3. 2.31.7	91.16	-18.03	-2.70		-12.86		-2.47	
4. 3. 4.31.7	93.07	-16.41	-3.23		-12.94		-.23	
4. 3. 6.31.7	94.67	-13.62	-4.23		-8.94		-.45	
4. 3. 8.31.7	95.91	-13.64	-4.23		-7.43		-1.98	
4. 3.10.31.7	96.77	-11.97	-3.23		-5.89		-2.86	
4. 3.12.31.7	97.24	-10.05	-2.23		-4.76		-3.05	
4. 3.14.31.7	97.30	-5.76	-.23		-3.23		-2.30	
4. 3.16.31.7	96.94	-5.15	.77		-1.26		-4.65	
4. 3.18.31.7	96.18	-2.71	1.30		-2.19		-1.82	
4. 3.20.31.7	95.01	-4.63	2.30		-4.35		-2.59	
4. 3.22.31.7	93.46	-2.18	2.88		-2.35		-2.72	
4. 3.24.31.7	91.56	-.97	2.30		-.37		-2.90	
4. 3.26.31.7	89.32	.49	1.88		-.26		-1.13	
4. 3.28.31.7	86.79	-1.51	1.88		-1.19		-2.21	
4. 3.29.56.5	84.85	-1.70	1.30		-.35		-2.65	
4. 3.30.26.5	84.14	-3.86	1.30		-1.89		-3.27	
4. 3.30.56.5	83.41	.87	1.30		.24		-.67	
4. 3.31.26.5	82.68	-.36	1.30		-.26		-1.40	
4. 3.31.56.5	81.93	-4.04	1.30		2.39		-7.73	
4. 3.32.26.5	81.17	-.37	1.30		3.14		-4.81	
4. 3.32.56.5	80.40	2.75	1.30		2.72		-1.28	
4. 3.33.26.5	79.62	1.58	1.30		.36		-.08	
4. 3.33.56.5	78.83	1.20	1.30		1.06		-1.15	
4. 3.34.26.5	78.04	-5.70	1.30		-1.94		-5.05	
4. 3.34.56.5	77.24	-6.27	1.30		-1.61		-5.96	
4. 3.35.26.5	76.43	-.88	1.30		.06		-2.23	
4. 3.35.56.5	75.62	-.76	1.30		.06		-2.12	
4. 3.36.26.5	74.80	-.64	1.30		-.28		-1.67	
4. 3.38.43.7	71.83	-.66	1.30		-.28		-1.68	

4. 3.40.43.7	69.33	-.53	1.30	-.61	-1.22
4. 3.42.43.7	66.82	-.44	1.88	-2.21	-.11
4. 3.44.43.7	64.31	-.43	1.88	-2.21	-.11
4. 3.46.43.7	61.81	.69	2.88	-1.76	-.43
4. 3.48.43.7	59.31	1.09	2.88	-.99	-.81
4. 3.50.43.7	56.81	-.90	2.30	-1.43	-1.78
4. 3.52.43.7	54.32	-1.09	2.30	.01	-3.41
4. 3.54.43.7	51.82	-2.35	3.30	.01	-5.67
4. 3.56.43.7	49.34	-2.89	2.88	-2.13	-3.63
4. 3.57.13.7	48.72	2.94	2.88	.87	-.81
4. 3.57.43.7	48.10	1.39	2.88	-.89	-.61
4. 3.58.13.7	47.48	-.38	2.88	-.13	-3.13

LINK NO. 2. CSM ANTENNA 1, LEM ANTENNA 2.

4. 2.56.31.7	83.91	-2.52	-1.78	1.71	-2.45
4. 2.58.31.7	86.53	-4.42	-2.48	1.81	-3.76
4. 3. 0.31.7	88.97	-2.87	-3.12	3.51	-3.27
4. 3. 2.31.7	91.16	-3.33	-2.70	2.37	-3.00
4. 3. 4.31.7	93.07	-5.67	-3.23	.30	-2.75
4. 3. 6.31.7	94.67	-7.83	-4.23	-1.56	-2.04
4. 3. 8.31.7	95.91	-11.08	-4.23	-3.53	-3.32
4. 3.10.31.7	96.77	-10.46	-3.23	-3.66	-3.57
4. 3.12.31.7	97.24	-9.01	-2.23	-3.29	-3.50
4. 3.14.31.7	97.30	-5.63	-.23	-2.49	-2.91
4. 3.16.31.7	96.94	-5.53	.77	-3.63	-2.67
4. 3.18.31.7	96.18	-5.55	1.30	-3.67	-3.19
4. 3.20.31.7	95.01	-7.29	2.30	-5.53	-4.05
4. 3.22.31.7	93.46	-8.54	2.88	-7.91	-3.52
4. 3.24.31.7	91.56	-7.92	2.30	-7.73	-2.49
4. 3.26.31.7	89.32	-5.21	1.88	-5.29	-1.81
4. 3.28.31.7	86.79	-2.08	1.88	-2.19	-1.78
4. 3.29.56.5	84.85	-2.39	1.30	-1.91	-1.78
4. 3.30.26.5	84.14	.18	1.30	.41	-1.52
4. 3.30.56.5	83.41	1.11	1.30	2.31	-2.50
4. 3.31.26.5	82.68	1.65	1.30	2.33	-1.97
4. 3.31.56.5	81.93	-1.23	1.30	-.73	-1.80
4. 3.32.26.5	81.17	1.58	1.30	2.09	-1.81
4. 3.32.56.5	80.40	1.51	1.30	1.51	-1.30
4. 3.33.26.5	79.62	-.26	1.30	1.57	-3.13
4. 3.33.56.5	78.83	1.22	1.30	1.51	-1.59
4. 3.34.26.5	78.04	2.22	1.30	1.71	-.80

4.	3.34.56.5	77.24	-.36	1.30	-.49	-1.17
4.	3.35.26.5	76.43	-5.81	1.30	-6.19	-.92
4.	3.35.56.5	75.62	-5.34	1.30	-6.06	-.58
4.	3.36.26.5	74.80	-7.17	1.30	-7.91	-.56
4.	3.38.43.7	71.83	-13.12	1.30	-13.06	-1.36
4.	3.40.43.7	69.33	-20.90	1.30	-18.58	-3.62
4.	3.42.43.7	66.82	-20.12	1.88	-20.73	-1.28
4.	3.44.43.7	64.31	-11.95	1.88	-13.58	-.26
4.	3.46.43.7	61.81	-11.05	2.88	-12.24	-1.69
4.	3.48.43.7	59.31	-11.61	2.88	-11.58	-2.92
4.	3.50.43.7	56.91	-15.09	2.30	-14.69	-2.70
4.	3.52.43.7	54.32	-26.01	2.30	-20.51	-7.81
4.	3.54.43.7	51.82	-21.60	3.30	-22.49	-2.41
4.	3.56.43.7	49.34	-11.43	2.88	-13.91	-.40
4.	3.57.13.7	48.72	-10.23	2.88	-7.06	-6.05
4.	3.57.43.7	48.10	-14.27	2.88	-12.06	-5.09
4.	3.58.13.7	47.48	-3.35	2.88	-4.49	-1.74

LINK NO. 3. CSM ANTENNA 2, LEM ANTENNA 1.

4.	2.56.31.7	83.91	-15.54	.28	-10.86	-4.96
4.	2.58.31.7	86.53	-9.77	-.72	-8.61	-.44
4.	3. 0.31.7	88.97	-13.89	-.25	-9.86	-3.78
4.	3. 2.31.7	91.16	-16.17	-1.25	-12.86	-2.05
4.	3. 4.31.7	93.07	-25.27	-1.25	-12.94	-11.08
4.	3. 6.31.7	94.67	-20.09	-.72	-8.94	-10.43
4.	3. 8.31.7	95.91	-9.75	-1.25	-7.43	-1.08
4.	3.10.31.7	96.77	-8.91	-1.25	-5.89	-1.78
4.	3.12.31.7	97.24	-7.53	-1.25	-4.76	-1.52
4.	3.14.31.7	97.30	-6.11	-1.25	-3.23	-1.63
4.	3.16.31.7	96.94	-6.00	-1.25	-1.26	-3.48
4.	3.18.31.7	96.18	-6.35	-.72	-2.19	-3.44
4.	3.20.31.7	95.01	-7.90	-1.72	-4.35	-1.83
4.	3.22.31.7	93.46	-6.46	-1.14	-2.35	-2.98
4.	3.24.31.7	91.56	-4.29	-.72	-.37	-3.20
4.	3.26.31.7	89.32	-3.24	-.72	-.26	-2.25
4.	3.28.31.7	86.79	-6.33	-.72	-1.19	-4.42
4.	3.29.56.5	84.85	-4.53	-.72	-.35	-3.46
4.	3.30.26.5	84.14	-3.62	-.72	-1.89	-1.01
4.	3.30.56.5	83.41	-1.51	-.72	.24	-1.03
4.	3.31.26.5	82.68	-2.97	-.72	-.26	-1.99
4.	3.31.56.5	81.93	-2.31	-.72	2.39	-3.98

4.	3.32.26.5	81.17	-25.48	-.72	3.14	-27.90
4.	3.32.56.5	80.40	-5.24	-1.14	2.72	-6.83
4.	3.33.26.5	79.62	-1.97	-1.14	.36	-1.20
4.	3.33.56.5	78.83	-.18	-1.14	1.06	-.10
4.	3.34.26.5	78.04	-3.81	-1.14	-1.94	-.73
4.	3.34.56.5	77.24	-8.51	-1.14	-1.61	-5.77
4.	3.35.26.5	76.43	-8.96	-1.14	.06	-7.88
4.	3.35.56.5	75.62	-8.75	-1.14	.06	-7.67
4.	3.36.26.5	74.80	-9.37	-1.14	-.28	-7.96
4.	3.38.43.7	71.83	-8.83	-.72	-.28	-7.83
4.	3.40.43.7	69.33	-5.86	-.72	-.61	-4.53
4.	3.42.43.7	66.82	-4.50	-.72	-2.21	-1.58
4.	3.44.43.7	64.31	-4.65	-1.14	-2.21	-1.31
4.	3.46.43.7	61.81	-4.44	-1.14	-1.76	-1.54
4.	3.48.43.7	59.31	-8.34	-1.72	-.99	-5.63
4.	3.50.43.7	56.81	-4.24	-2.25	-1.43	-.56
4.	3.52.43.7	54.32	-4.48	-2.72	.01	-1.77
4.	3.54.43.7	51.82	-8.16	-3.72	.01	-4.45
4.	3.56.43.7	49.34	-10.35	-4.50	-2.13	-3.72
4.	3.57.13.7	48.72	-4.73	-4.80	.87	-.79
4.	3.57.43.7	48.10	-6.83	-4.80	-.89	-1.14
4.	3.58.13.7	47.48	-6.11	-4.80	-.13	-1.17

LINK NO. 4. CSM ANTENNA 2, LEM ANTENNA 2.

4.	2.56.31.7	83.91	.83	.28	1.71	-1.16
4.	2.58.31.7	86.53	-2.85	-.72	1.81	-3.94
4.	3. 0.31.7	88.97	1.19	-.25	3.51	-2.07
4.	3. 2.31.7	91.16	-2.50	-1.25	2.37	-3.62
4.	3. 4.31.7	93.07	-4.24	-1.25	.30	-3.29
4.	3. 6.31.7	94.67	-5.69	-.72	-1.56	-3.40
4.	3. 8.31.7	95.91	-8.88	-1.25	-3.53	-4.10
4.	3.10.31.7	96.77	-8.06	-1.25	-3.66	-3.15
4.	3.12.31.7	97.24	-6.54	-1.25	-3.29	-2.01
4.	3.14.31.7	97.30	-5.47	-1.25	-2.49	-1.73
4.	3.16.31.7	96.94	-6.99	-1.25	-3.63	-2.11
4.	3.18.31.7	96.18	-7.59	-.72	-3.67	-3.20
4.	3.20.31.7	95.01	-11.28	-1.72	-5.53	-4.02
4.	3.22.31.7	93.46	-15.46	-1.14	-7.91	-6.42
4.	3.24.31.7	91.56	-15.67	-.72	-7.73	-7.22
4.	3.26.31.7	89.32	-10.85	-.72	-5.29	-4.85
4.	3.28.31.7	86.79	-8.05	-.72	-2.19	-5.15

4.	3.29.56.5	84.85	-8.29	-.72	-1.91	-5.66
4.	3.30.26.5	84.14	-2.65	-.72	.41	-2.34
4.	3.30.56.5	83.41	-1.35	-.72	2.31	-2.94
4.	3.31.26.5	82.68	-.65	-.72	2.33	-2.26
4.	3.31.56.5	81.93	-1.76	-.72	-.73	-.31
4.	3.32.26.5	81.17	-4.35	-.72	2.09	-5.72
4.	3.32.56.5	80.40	-2.02	-1.14	1.51	-2.40
4.	3.33.26.5	79.62	-.60	-1.14	1.57	-1.03
4.	3.33.56.5	78.83	-.63	-1.14	1.51	-1.00
4.	3.34.26.5	78.04	-1.11	-1.14	1.71	-1.69
4.	3.34.56.5	77.24	-3.43	-1.14	-.49	-1.80
4.	3.35.26.5	76.43	-10.18	-1.14	-6.19	-2.86
4.	3.35.56.5	75.62	-9.26	-1.14	-6.06	-2.07
4.	3.36.26.5	74.80	-11.62	-1.14	-7.91	-2.58
4.	3.38.43.7	71.83	-14.35	-.72	-13.06	-.56
4.	3.40.43.7	69.33	-19.73	-.72	-18.58	-.43
4.	3.42.43.7	66.82	-25.52	-.72	-20.73	-4.07
4.	3.44.43.7	64.31	-15.15	-1.14	-13.58	-.44
4.	3.46.43.7	61.81	-13.54	-1.14	-12.24	-.16
4.	3.48.43.7	59.31	-21.32	-1.72	-11.58	-8.03
4.	3.50.43.7	56.81	-25.73	-2.25	-14.69	-8.79
4.	3.52.43.7	54.32	-32.63	-2.72	-20.51	-9.41
4.	3.54.43.7	51.82	-30.50	-3.72	-22.49	-4.23
4.	3.56.43.7	49.34	-19.82	-4.50	-13.91	-1.42
4.	3.57.13.7	48.72	-15.28	-4.80	-7.06	-3.42
4.	3.57.43.7	48.10	-21.41	-4.80	-12.06	-4.55
4.	3.58.13.7	47.48	-11.55	-4.80	-4.49	-2.25

```
                    VHF ANTENNA SIMULATION
                         TEST CASE 2

                      PARAMETER LISTING
         MINIMUM TIME (D.H.M.S)          =   4. 2.56.   .0
         MAXIMUM TIME (D.H.M.S)          =   4. 3.58.20.0
         TIME INCREMENT (SECONDS)        =            30.0
         FREQUENCY FLAG (-1,1)           =               1
         ASCENT/DESCENT FLAG (-1,1)      =               1
         MULTI-PATH FLAG (0,1)           =               1
         PRINT FLAG (0,1)                =               1
         PLOT FLAG (0,1)                 =               0
         COMMENT/DESCRIPTION CARDS       =               1
```

VHF ANTENNA SIMULATION
TEST CASE 2

TIME (D.H.M.S)	RANGE (NM)	G-PRODUCT (DB)	CSM GAINS (DB) DIRECT	CSM GAINS (DB) SPECULAR	LEM GAINS (DB) DIRECT	LEM GAINS (DB) SPECULAR	POLARIZATION GAINS (DB) DIRECT	POLARIZATION GAINS (DB) SPECULAR
LINK NO. 1. CSM ANTENNA 1, LEM ANTENNA 1.								
4. 2.56.31.7	83.91	-15.86	-2.61	-4.94	-12.33	-1.00	-.93	-1.76

```
TIME   =  3.562+05
ROCB   =  3.148+03, -6.765+02, -1.545+03
ROLB   =  3.127+03, -7.521+02, -1.575+03
ACB    =  2.586-01,  9.000-01,  3.509-01,  9.046-01, -3.531-01,  2.389-01,  3.389-01,  2.557-01, -9.054-01
ALB    = -5.924-01, -7.882-01, -1.666-01,  3.245-01, -4.227-01,  8.462-01, -7.374-01,  4.472-01,  5.062-01
IFLAG  =  6.000+00
ROSB   =  8.234+02, -1.875+02, -4.094+02
THTAC  =  3.123+00
PHIC   =  3.074+00
THTACS =  1.485+00
PHICS  =  5.431+00
THTAL  =  1.689+00
PHIL   =  4.365-01
THTALS =  1.307+00
PHILS  =  5.319+00
PATERN = -7.810+00, -4.810+00, -1.871+01, -2.710+00, -4.730+00, -6.730+00,
         -4.810+00, -1.981+01, -7.710+00, -7.710+00, -5.730+00, -1.273+01,
         -1.249+01, -2.049+01, -1.631+01, -1.331+01, -1.709+01, -1.409+01,
         -3.490+00, -5.490+00, -4.310+00, -6.310+00, -1.809+01, -1.090+00
R      =  8.391+01
RHO    =  2.072+00
TAU1   = -9.382-01
TAU2   = -2.281-01
BETA   = -3.594-01
G1     =  5.488-01
G2     =  5.854-02
RR1    = -8.724+00
RR2    =  5.848+00
LP     =  8.081-01
R1S    =  2.633+03
```

```
R2S     =  2.643+03
RS      =  5.275+03
DELTAR  =  5.192+03
PSI     =  1.555+00
RHO1S   =  1.005+00
RHO2S   =  2.509+00
TAU1S   =  2.136+00,  9.781-01
TAU2S   =  0.000   ,  3.458-01
RA      =(-2.529-01) + J( 2.023-02)
RB      =( 2.511-01) + J( 2.025-02)
G1S     =  3.206-01
G2S     =  7.936-01
RR1S    = -2.615+00
RR2S    =  1.329+00
P1S     =(-9.867-01) + J(-9.770-01), ( 2.865-01) + J( 1.073+00)
P2S     =( 1.138+00) + J( 8.128-01), (-4.573-01) + J(-1.020+00)
Q       =(-9.943-01) + J( 1.608-01)
TAUS    = -5.866-01,  6.518-01
RRS     =  3.337+00, -1.567+00
BETAS   = -1.219+00,  1.311+00
LPS     =  6.760-01,  6.663-01
LR      =  6.405-02,  6.395-02
D       =  4.898-03
K       =  3.614-05
GP      =  2.596-02

 4. 2.58.31.7        86.53        -19.55        -4.72       -4.76        -9.92       -.92        -4.91       -1.86

TIME    =  3.563+05
ROCR    =  3.289+03, -2.254+02, -1.374+03
ROLR    =  3.278+03, -3.038+02, -1.409+03
ACR     =  1.283-01,  9.058-01,  4.039-01,  9.253-01, -2.559-01,  2.798-01,  3.568-01,  3.378-01, -8.710-01
ALR     = -5.924-01, -7.882-01, -1.466-01,  3.245-01, -4.227-01,  8.462-01, -7.374-01,  4.472-01,  5.062-01
IFLAG   =  6.000+00
ROSR    =  8.617+02, -6.943+01, -3.651+02
THTAC   =  3.090+00
PHIC    =  3.598+00
THTACS  =  1.498+00
PHICS   =  5.476+00
THTAL   =  1.652+00
PHIL    =  3.105-01
```

```
THTALS =  1.288+00
PHILS  =  5.174+00
PATERN = -1.181+01, -3.810+00, -9.710+00, -3.710+00, -7.730+00, -7.730+00,
         -4.810+00, -1.581+01, -8.710+00, -6.710+00, -5.730+00, -1.173+01,
         -1.149+01, -1.349+01, -1.331+01, -1.131+01, -2.409+01, -1.009+01,
         -3.490+00, -4.490+00, -2.310+00, -5.310+00, -1.509+01, -1.090+00
R      =  8.653+01
RHO    =  2.543+00
TAU1   = -1.201+00
TAU2   = -4.031-01
BETA   =  1.995-01
G1     =  3.373-01
G2     =  1.018-01
RR1    =  0.000
RR2    =  1.499+00
LP     =  3.232-01
R1S    =  2.633+03
R2S    =  2.642+03
RS     =  5.275+03
DELTAR =  5.189+03
PSI    =  1.554+00
RHO1S  =  9.341-01
RHO2S  =  2.548+00
TAU1S  =  2.081+00,  1.227+00
TAU2S  = -1.266-01,  6.332-01
RA     =(-2.529-01) + J( 2.023-02)
RB     =( 2.511-01) + J( 2.025-02)
G1S    =  3.344-01
G2S    =  8.090-01
RR1S   = -3.010+00
RR2S   =  1.499+00
P1S    =(-1.175+00) + J(-1.031+00), ( 3.461-01) + J( 1.313+00)
P2S    =( 1.335+00) + J( 8.357-01), (-5.552-01) + J(-1.250+00)
Q      =(-9.943-01) + J( 1.608-01)
TAUS   = -5.324-01,  4.530-01
RRS    =  3.908+00, -1.713+00
BETAS  = -9.998-01,  7.539-01
LPS    =  6.518-01,  7.730-01
LR     =  6.410-02,  6.404-02
D      =  4.898-03
K      =  5.761-05
```

```
GP       =  1.110-02

 4. 3. 0.31.7        88.97        -17.81       -3.27     -3.94       -9.88        .12        -4.66      -1.23

TIME     =  3.564+05
ROCB     =  3.365+03,  2.302+02, -1.175+03
ROLB     =  3.364+03,  1.505+02, -1.214+03
ACB      = -4.533-03,  8.936-01,  4.489-01,  9.319-01, -1.590-01,  3.260-01,  3.627-01,  4.198-01, -8.320-01
ALB      = -5.924-01, -7.882-01, -1.666-01,  3.245-01, -4.227-01,  8.462-01, -7.374-01,  4.472-01,  5.062-01
IFLAG    =  6.000+00
ROSB     =  8.831+02,  4.997+01, -3.136+02
THTAC    =  3.050+00
PHIC     =  3.746+00
THTACS   =  1.512+00
PHICS    =  5.514+00
THTAL    =  1.614+00
PHIL     =  1.867-01
THTALS   =  1.274+00
PHILS    =  5.028+00
PATERN   = -1.981+01, -2.810+00, -6.710+00, -4.710+00, -4.730+00, -8.730+00,
           -4.810+00, -1.381+01, -9.710+00, -5.710+00, -4.730+00, -1.173+01,
           -1.149+01, -2.549+01, -1.731+01, -1.231+01, -2.309+01, -1.009+01,
           -1.490+00, -1.490+00, -3.100-01, -5.310+00, -1.309+01, -9.000-02
R        =  9.897+01
RHO      =  2.646+00
TAU1     = -1.452+00
TAU2     = -2.664-01
BETA     =  6.993-01
G1       =  4.705-01
G2       =  1.029-01
RR1      = -4.419+00
RR2      =  1.577+00
LP       =  3.420-01
R1S      =  2.633+03
R2S      =  2.641+03
RS       =  5.275+03
DELTAR   =  5.186+03
PSI      =  1.554+00
RHO1S    =  8.576-01
RHO2S    =  2.589+00
TAU1S    =  2.029+00,  1.338+00
```

```
TAU2S   = -2.551-01,  7.854-01
RA      =(-2.529-01) + J( 2.023-02)
RB      =( 2.511-01) + J( 2.025-02)
G1S     =  4.037-01
G2S     =  1.029+00
RR1S    = -2.615+00
RR2S    =  1.577+00
P1S     =(-1.081+00) + J(-1.221+00), ( 3.088-01) + J( 1.462+00)
P2S     =( 1.271+00) + J( 1.040+00), (-5.421-01) + J(-1.404+00)
Q       =(-9.943-01) + J( 1.608-01)
TAUS    = -4.910-01,  3.544-01
RRS     =  3.247+00, -1.756+00
BETAS   = -7.889-01,  4.266-01
LPS     =  7.532-01,  9.119-01
LR      =  6.411-02,  6.409-02
D       =  4.899-03
K       =  7.418-05
GP      =  1.655-02

 4. 3. 2.31.7        91.16        -17.61       -3.54      -4.27       -7.88       1.12       -6.20      -2.98

TIME    =  3.566+05
ROCP    =  3.374+03,  6.812+02, -9.527+02
ROLB    =  3.384+03,  6.019+02, -9.965+02
ACB     = -1.374-01,  8.637-01,  4.849-01,  9.241-01, -6.444-02,  3.766-01,  3.565-01,  4.999-01, -7.893-01
ALB     = -5.924-01, -7.882-01, -1.666-01,  3.245-01, -4.227-01,  8.462-01, -7.374-01,  4.472-01,  5.062-01
IFLAG   =  6.000+00
ROSB    =  8.871+02,  1.684+02, -2.558+02
THTAC   =  3.005+00
PHIC    =  3.826+00
THTACS  =  1.527+00
PHICS   =  5.547+00
THTAL   =  1.575+00
PHIL    =  6.431-02
THTALS  =  1.266+00
PHILS   =  4.882+00
PATERN  = -2.981+01, -3.810+00, -5.710+00, -6.710+00, -4.730+00, -9.730+00,
          -4.810+00, -1.381+01, -1.071+01, -5.710+00, -5.730+00, -9.730+00,
          -1.049+01, -1.849+01, -1.831+01, -1.131+01, -2.109+01, -8.090+00,
          -4.900-01, -4.490+00, -1.310+00, -5.310+00, -1.209+01,  9.100-01
R       =  9.116+01
```

TWO HUNDRED AND FORTY PAGES OF OUTPUT DELETED AT THIS POINT

```
ROCB    = -2.271+03, -2.711+03, -4.947+02
ROLB    = -2.296+03, -2.679+03, -4.693+02
ACB     =  7.587-01, -4.185-01, -4.993-01, -2.236-01, -8.871-01,  4.037-01, -6.119-01, -1.947-01, -7.666-01
ALB     =  9.683-01,  1.719-01, -1.810-01,  2.019-01, -9.658-01,  1.629-01, -1.468-01, -1.943-01, -9.699-01
IFLAG   =  6.000+00
ROSB    = -6.011+02, -7.094+02, -1.269+02
THTAC   =  2.616+00
PHIC    =  4.476+00
THTACS  =  1.343+00
PHICS   =  4.833+00
THTAL   =  1.119+00
PHIL    =  5.071+00
THTALS  =  7.095-01
PHILS   =  1.758+00
PATERN  = -1.990+01, -4.900+00, -6.880+00, -9.880+00, -7.750+00, -7.750+00,
          -2.290+01, -8.900+00, -1.088+01, -1.388+01, -1.075+01, -1.275+01,
          -6.940+00, -5.940+00, -7.760+00, -2.276+01, -1.781+01, -1.810+00,
          -1.094+01, -3.940+00, -4.760+00, -5.760+00, -6.810+00, -2.810+00
R       =  4.748+01
RHO     =  2.666+00
TAU1    =  1.413+00
TAU2    =  9.416-01
BETA    = -9.466-01
G1      =  3.358-01
G2      =  6.757-01
RR1     =  0.000
RR2     =  1.377+00
LP      =  5.490-01
R1S     =  2.633+03
R2S     =  2.621+03
RS      =  5.254+03
DELTAR  =  5.206+03
PSI     =  1.562+00
RHO1S   =  9.647-01
RHO2S   = -1.882+00
TAU1S   =  4.470-01,  3.348+00
TAU2S   =  1.412+00,  1.466+00
RA      =(-2.529-01) + J( 2.023-02)
RB      =( 2.511-01) + J( 2.025-02)
G1S     =  1.372-01
G2S     =  7.320-01
```

```
RR1S    = -8.724+00
RR2S    =  4.419+00
P1S     =( 4.717-01) + J(-1.405-01), ( 1.985-01) + J( 2.357-01)
P2S     =(-4.463-01) + J( 2.155-01), (-2.352-01) + J(-2.024-01)
Q       =(-9.941-01) + J( 1.607-01)
TAUS    = -4.346-01, -2.400-01
RRS     =  5.601+00, -5.225+00
BETAS   = -3.729+00, -7.412-01
LPS     =  7.381-01,  5.817-01
LR      =  6.363-02,  6.353-02
D       =  4.929-03
K       =  1.290-06
GP      =  1.246-01
```

```
XECUTION TERMINATED BY AN ATTEMPT TO READ THRU AN END OF FILE
  I/O CALLED  AT  SEQUENCE  NUMBER  00203  OF  MAIN PROGRAM
```

12821637

@XQT TRWPLT

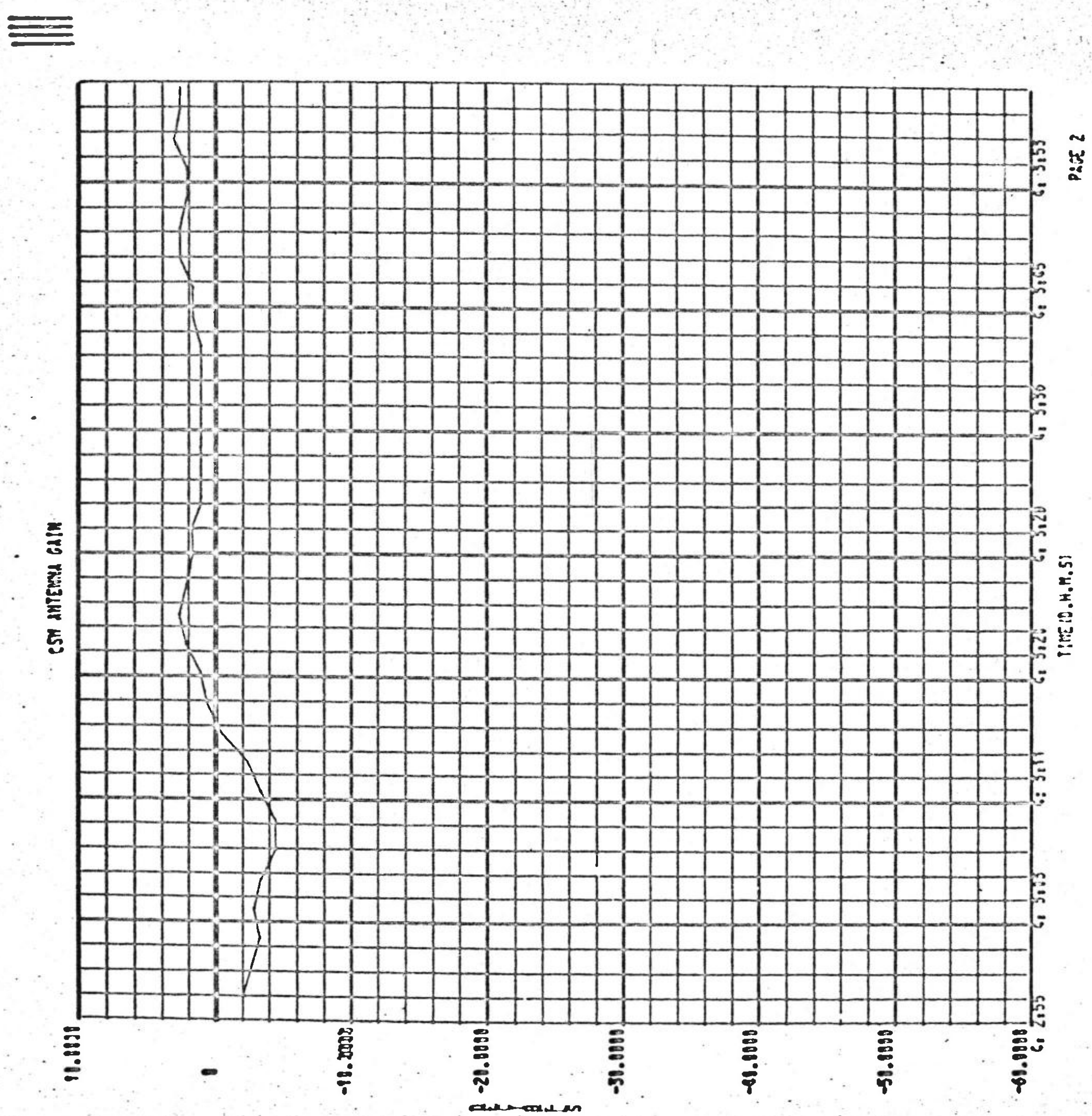
```
NOIREC = 1
KDAY = 1
XLO = 356100.0
XHI = 359900.0
YLO = -60.0
YHI = +10.0
PLOT=1,44,ENDLST
TITLE=ID=CSM ANTENNA GAIN
XLABEL=ID=TIME(D.H.M.S)
YLABEL=ID=DECIBELS
ENDPLT
PLOT=1,46,ENDLST
TITLE=ID=LM ANTENNA GAIN
ENDPLT
PLOT=1,50,ENDLST
TITLE=ID=POLARIZATION GAIN
ENDPLT
PLOT=1,96,ENDLST
TITLE=ID=GAIN PRODUCT
ENDPLT
PLOT=1,38,ENDLST
YLO = 0.0
YHI = 120.0
YLABEL=ID=NAUTICAL MILES
TITLE=ID=RANGE BETWEEN CSM AND LM
ENDPLT
ENDPHA

DD80 PLOT COMPLETED

DD80 PLOT COMPLETED

DD80 PLOT COMPLETED

DD80 PLOT COMPLETED
```

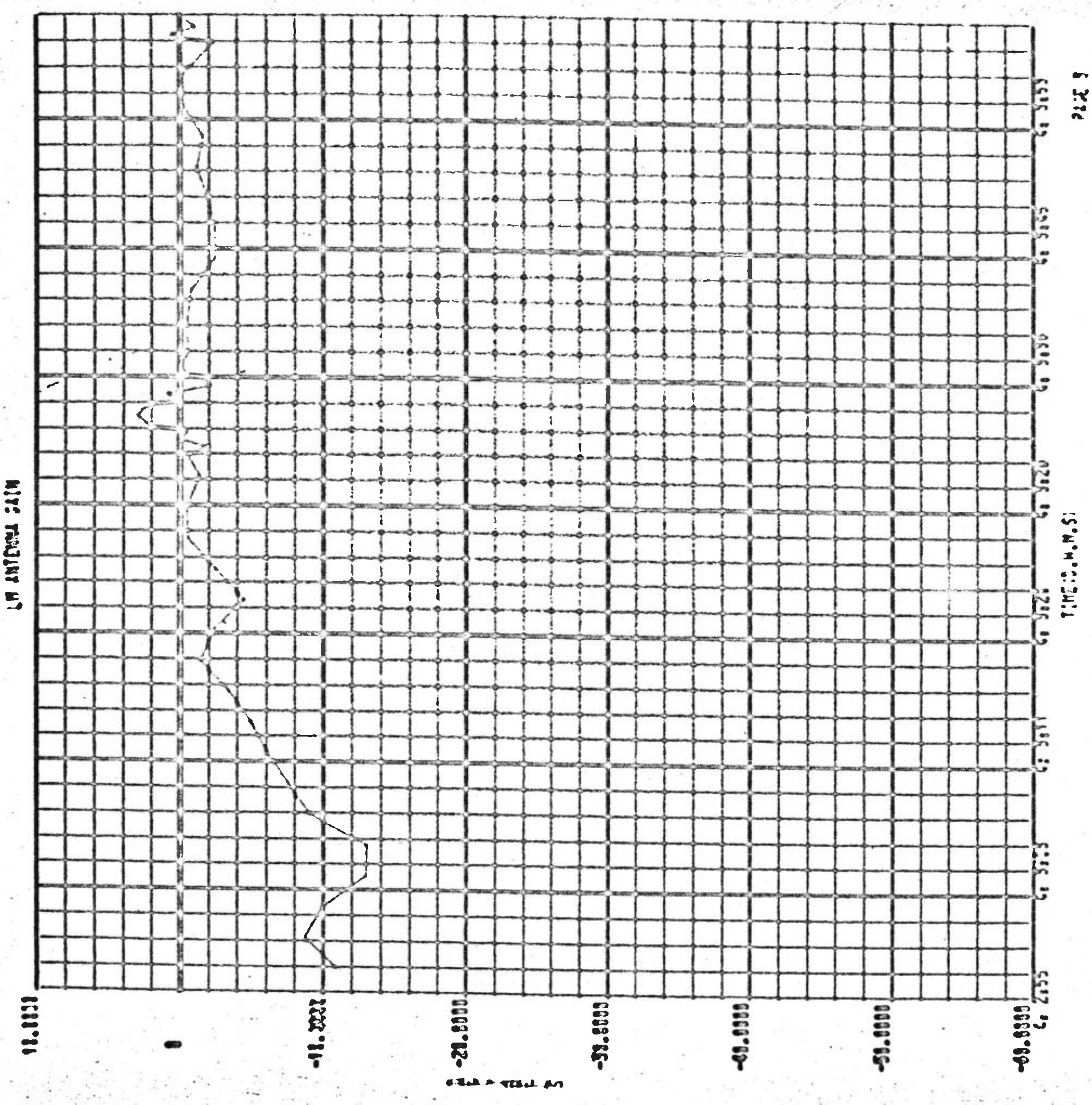
```
DD80  PLCT  COMPLETED
                REPEAT

DD80  PLCT  COMPLETED

DD80  PLCT  COMPLETED

DD80  PLCT  COMPLETED

DD80  PLCT  COMPLETED

DD80  PLCT  COMPLETED
                REPEAT

DD80  PLCT  COMPLETED

DD80  PLCT  COMPLETED

DD80  PLCT  COMPLETED

DD80  PLCT  COMPLETED

DD80  PLCT  COMPLETED
                REPEAT

DD80  PLOT  COMPLETED

DD80  PLCT  COMPLETED
```

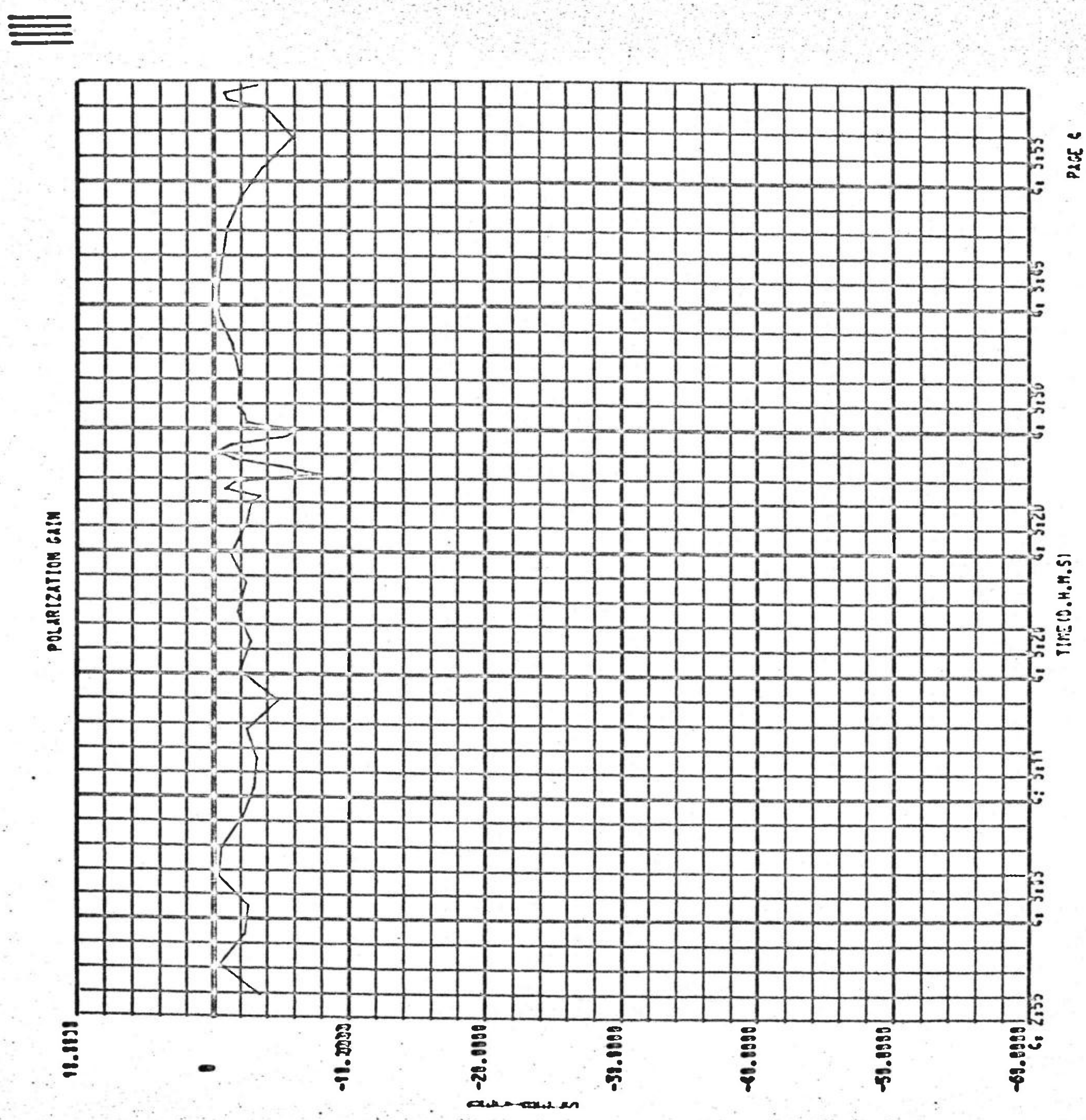
```
DD80 PLOT COMPLETED

DD80 PLOT COMPLETED

DD80 PLOT COMPLETED
               ENDJOB
```

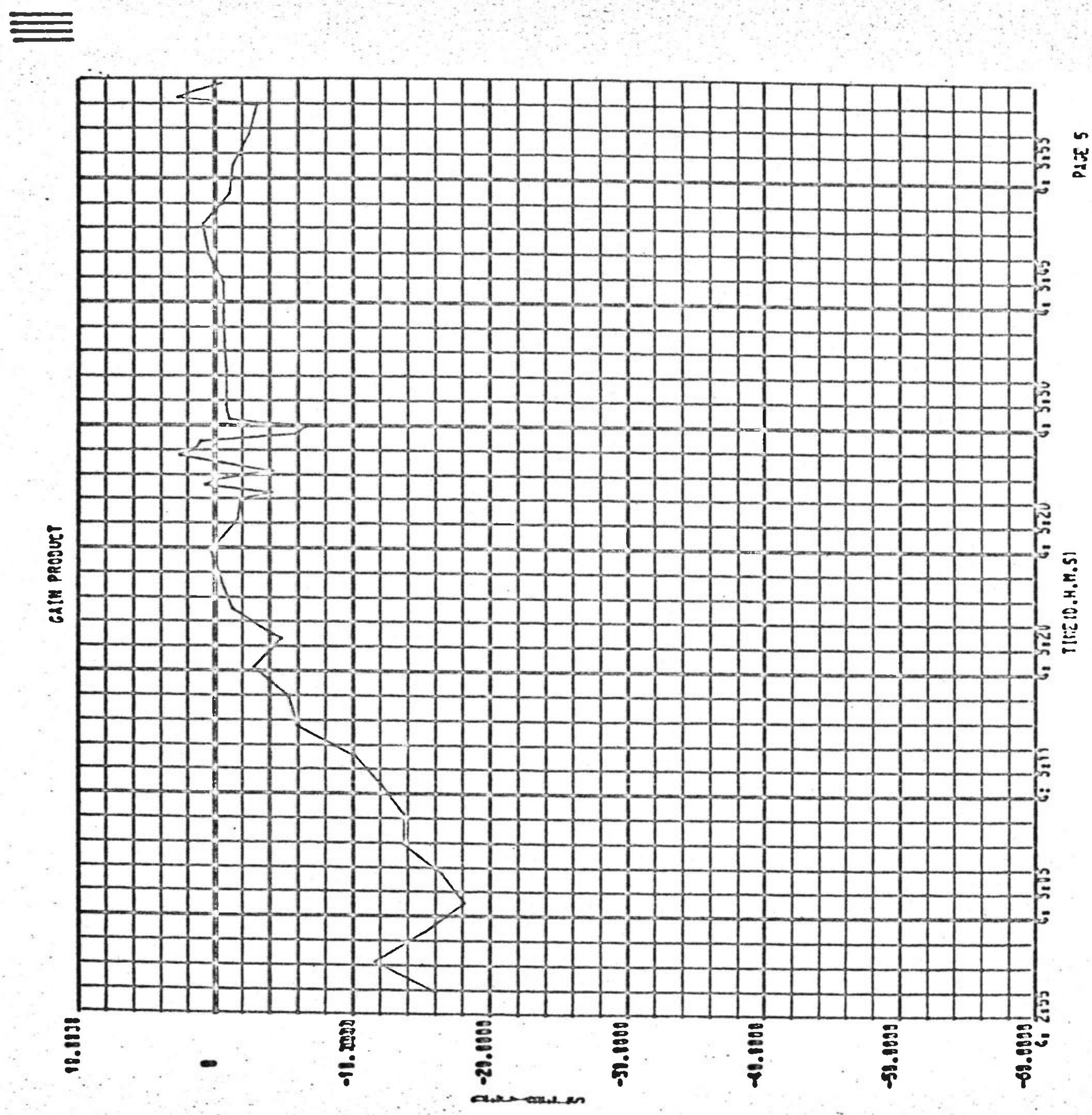
& EOF EOF & 12&21&58

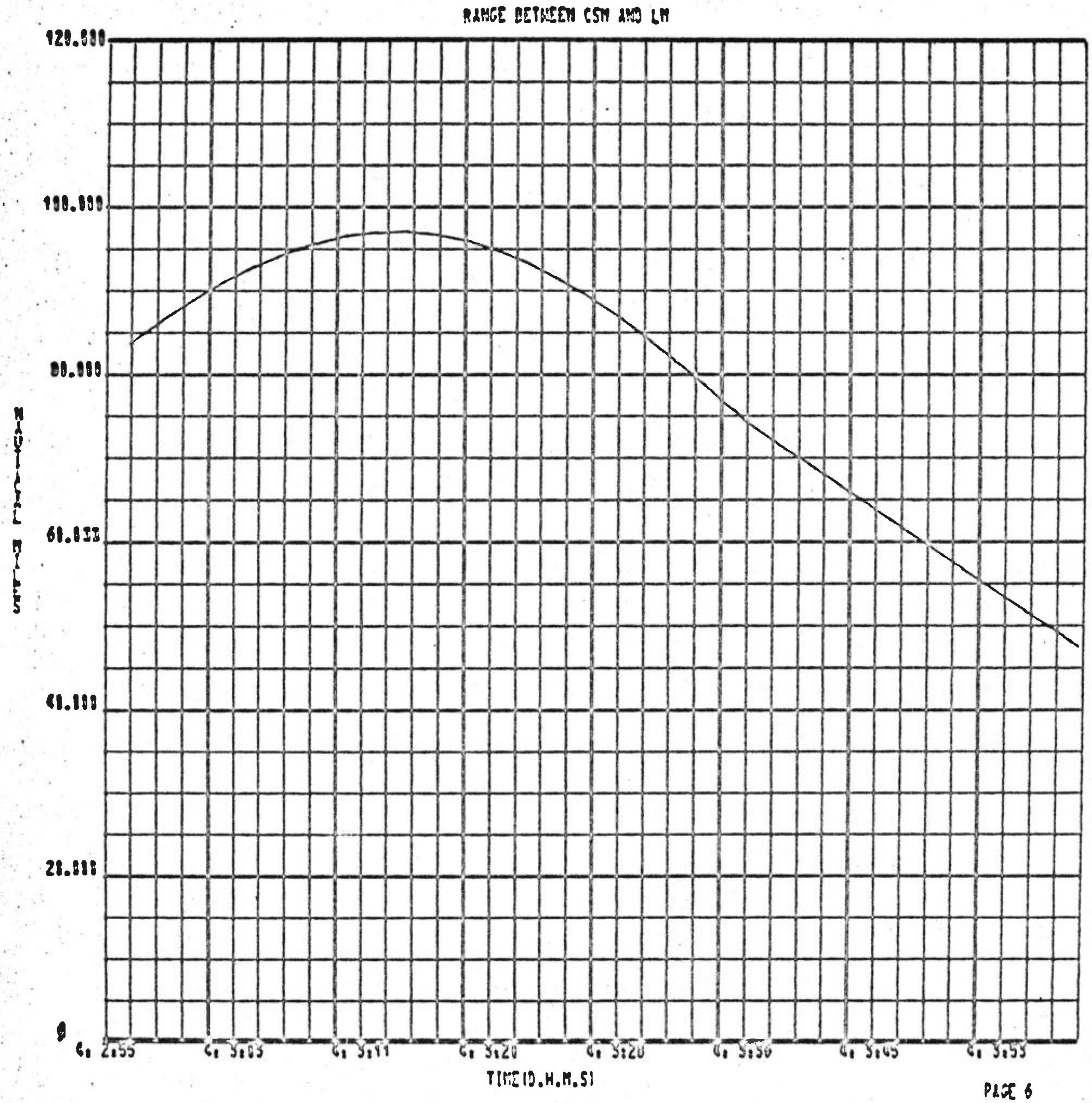
4.3 SAMPLE PLOT OUTPUT

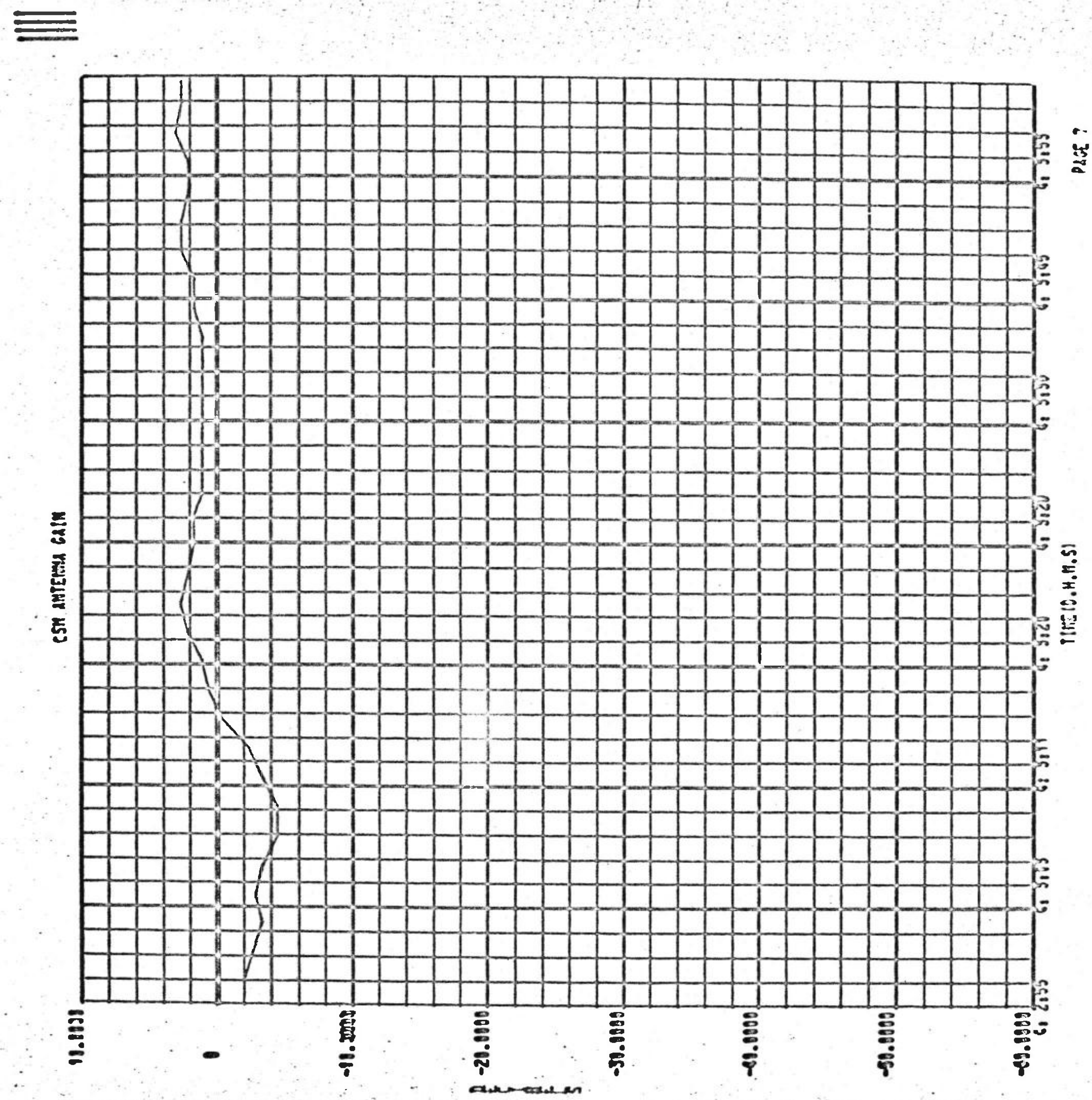


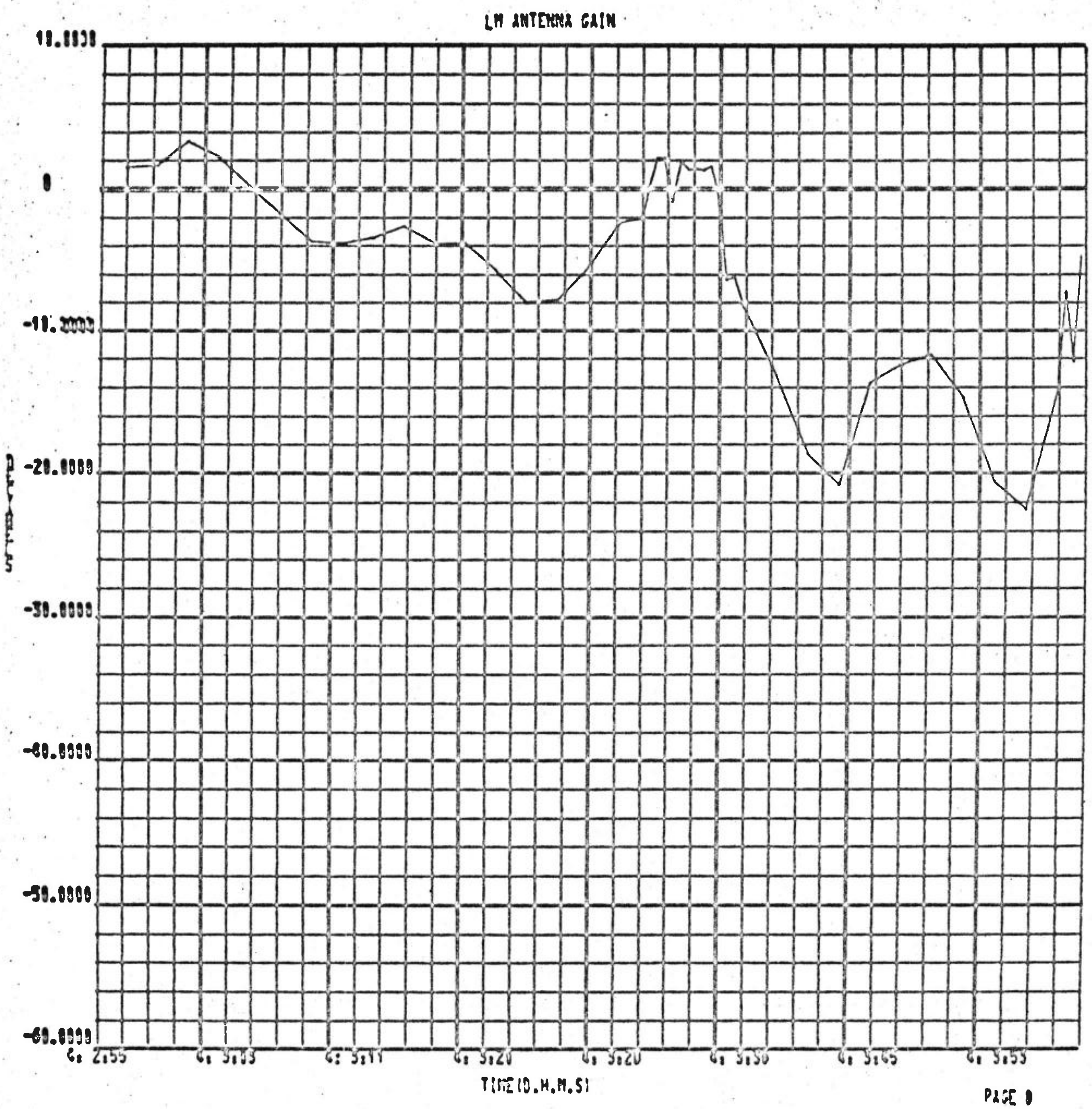

LW ANTENNA GAIN

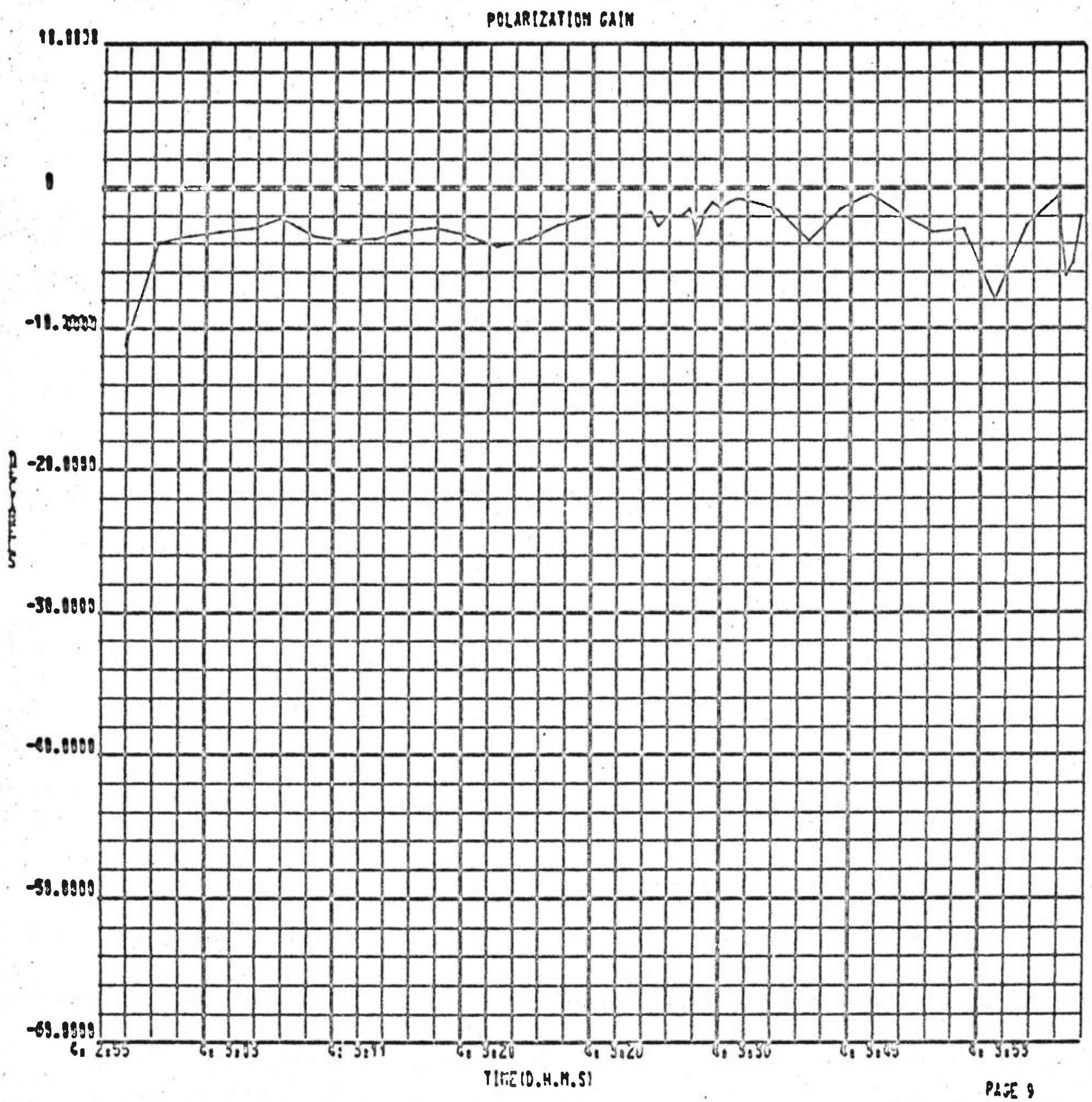

POLARIZATION GAIN
TIME(D.H.M.S)
PAGE 6
10.0000
0
-10.0000
-20.0000
-30.0000
-40.0000
-50.0000
-60.0000

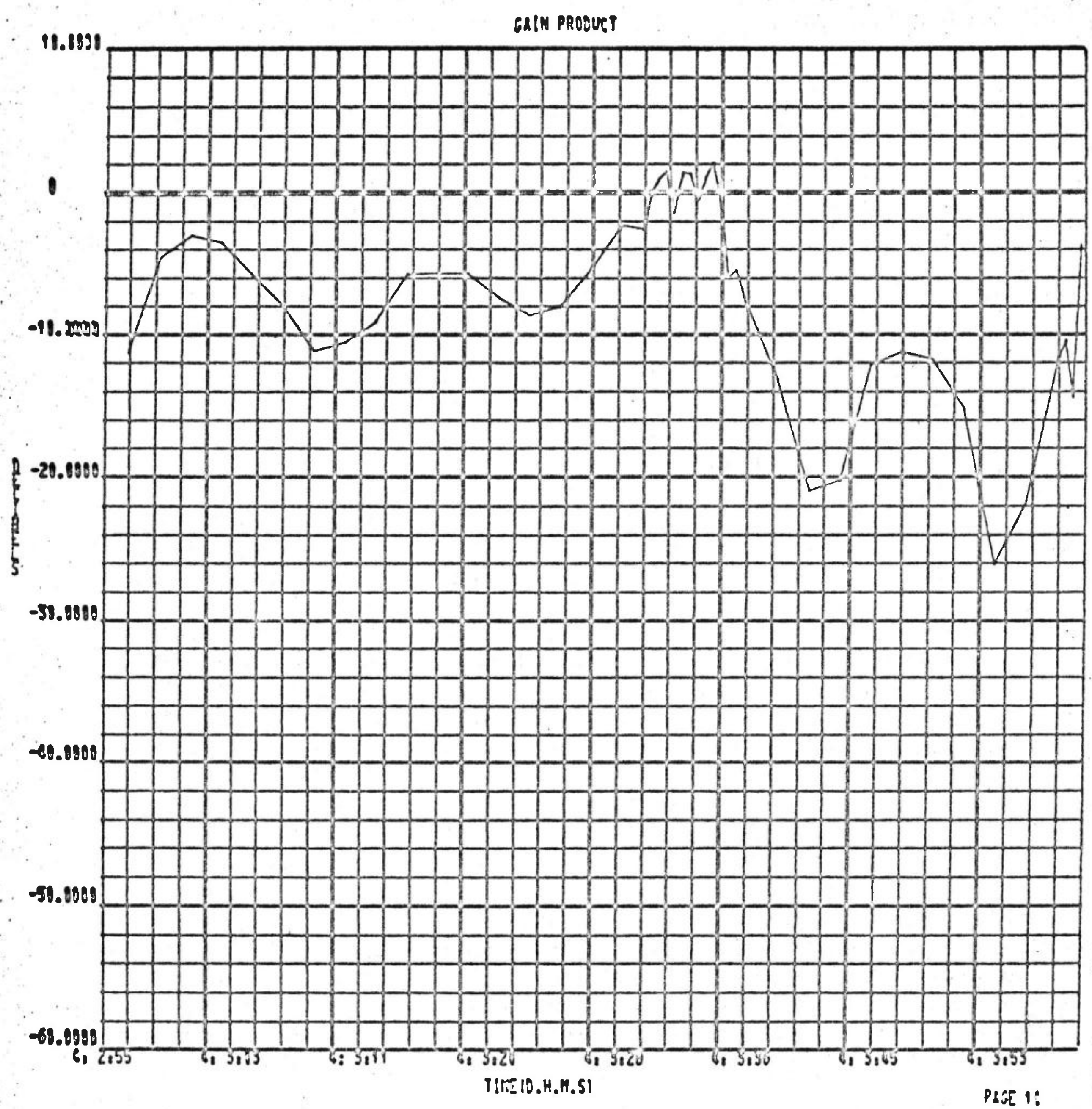

GAIN PRODUCT
0
-20.0000
-30.0000
-40.0000
-60.0000
TIME(D.H.M.S)
PAGE 5


RANGE BETWEEN CSM AND LM
120.000
100.000
80.000
60.000
40.000
20.000
0
NAUTICAL MILES
TIME(D.H.M.S)
PAGE 6

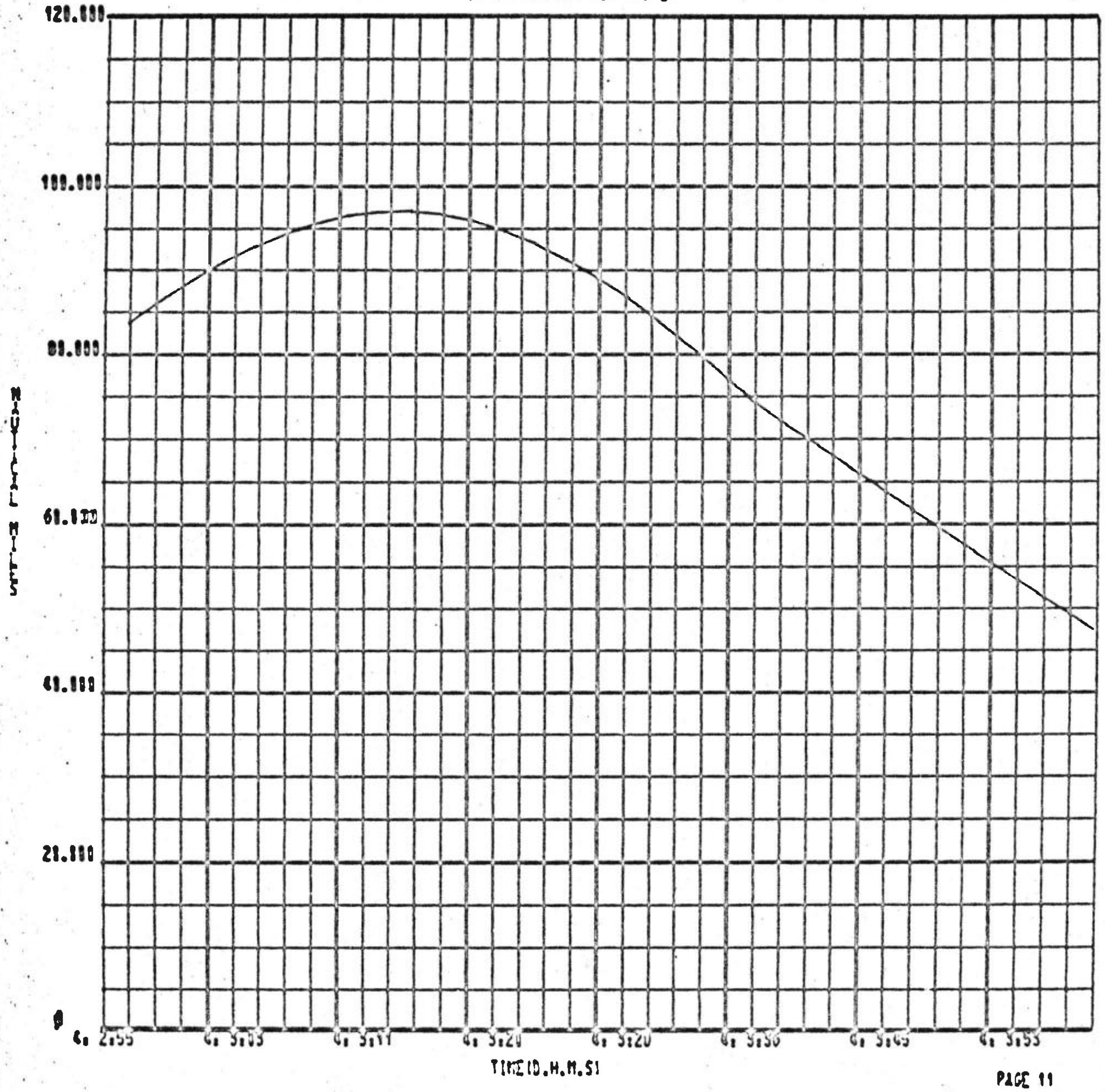

CSM ANTENNA GAIN
TIME(D,H,M,S)
PAGE 7
10.0000
0
-10.0000
-20.0000
-30.0000
-40.0000
-50.0000
-60.0000


LM ANTENNA GAIN
10.0000
0
-10.0000
-20.0000
-30.0000
-40.0000
-50.0000
-60.0000
TIME(D.H.M.S)
PAGE 0


POLARIZATION GAIN
0
-20.0000
-30.0000
-40.0000
-50.0000
-60.0000
TIME(D.H.M.S)
PAGE 9

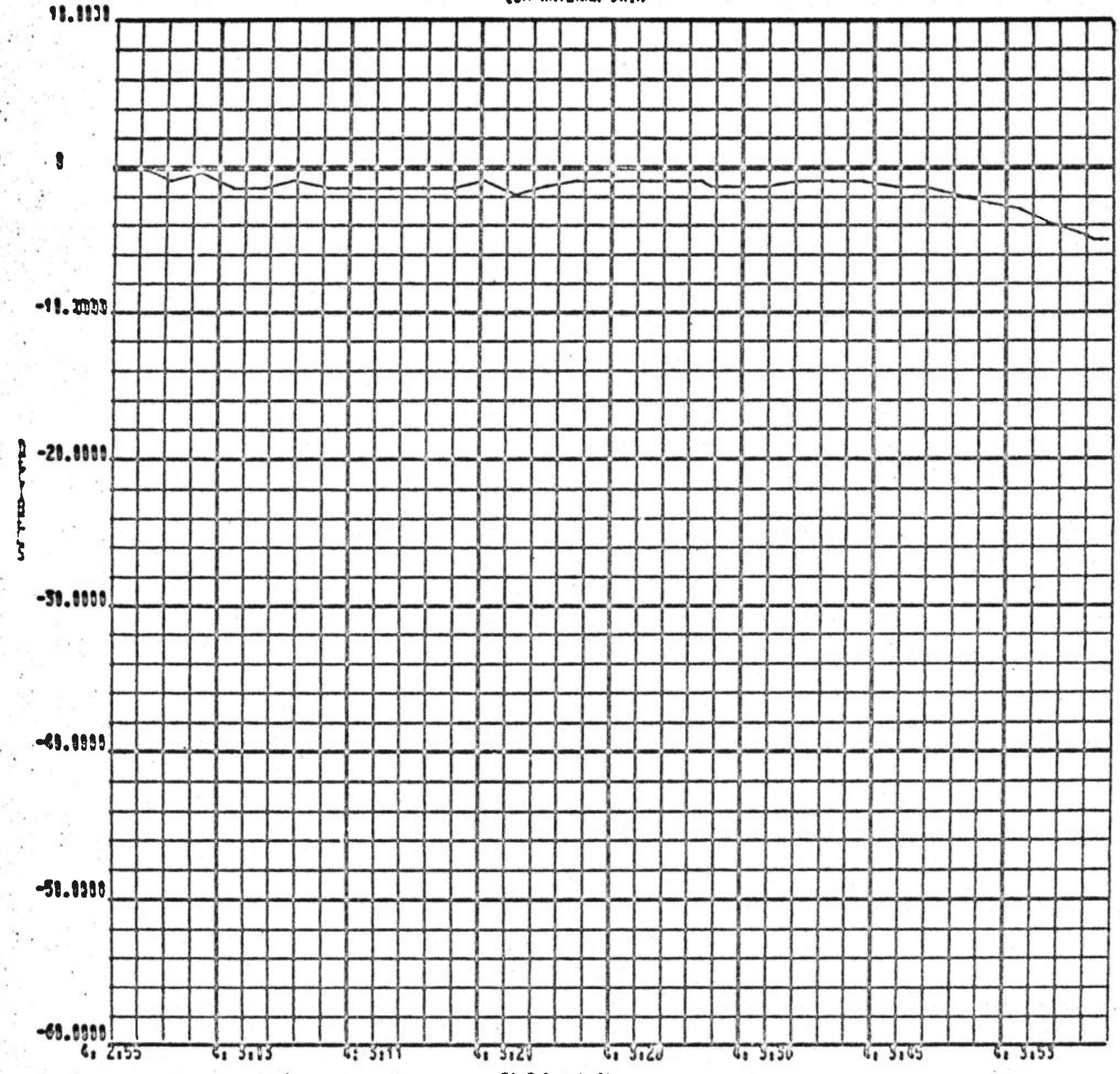

GAIN PRODUCT
10.0000
0
-10.0000
-20.0000
-30.0000
-40.0000
-50.0000
-60.0000
2:55
TIME(D.H.M.S)
PAGE 1

RANGE BETWEEN CSM AND LM



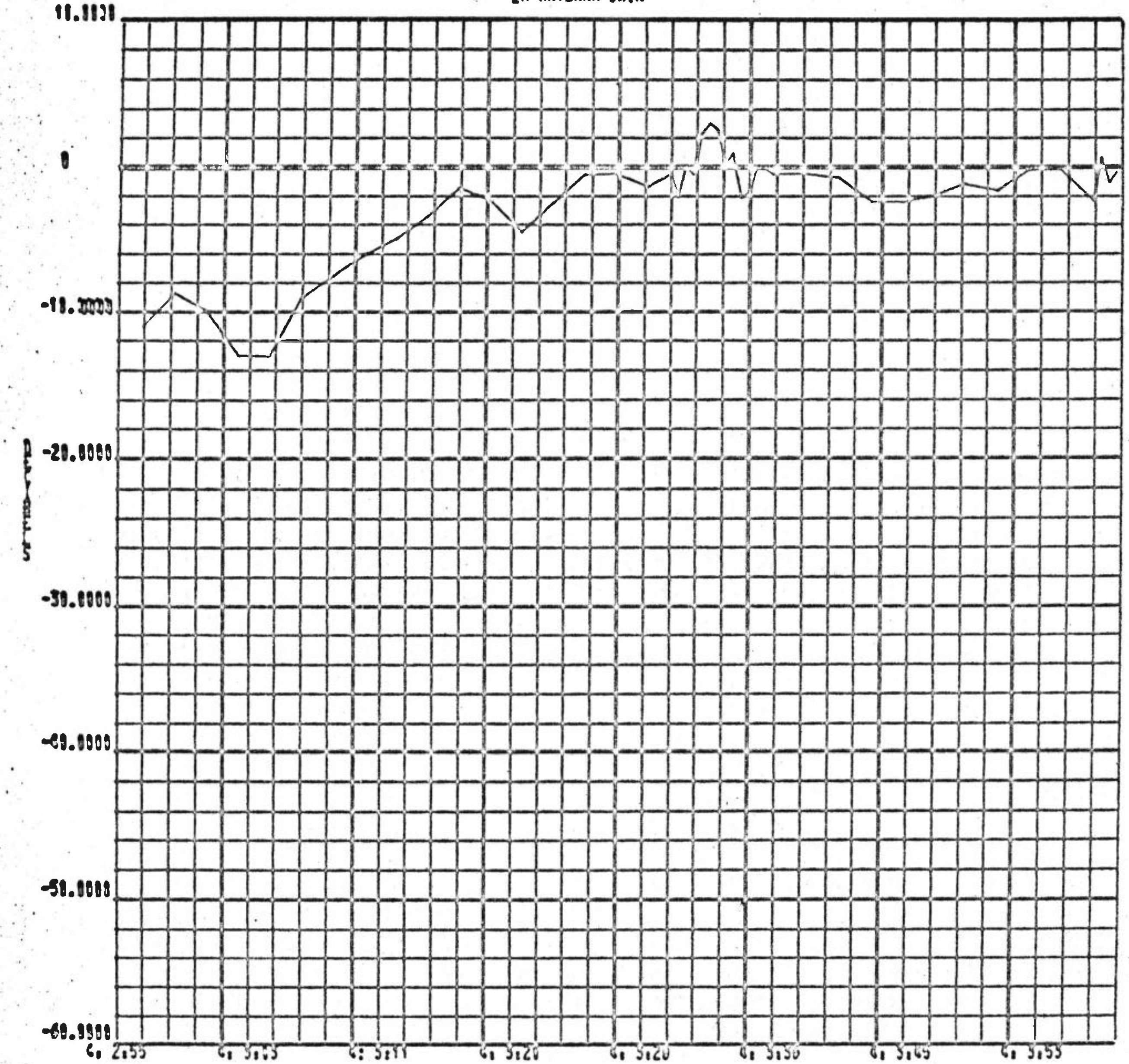
PAGE 11

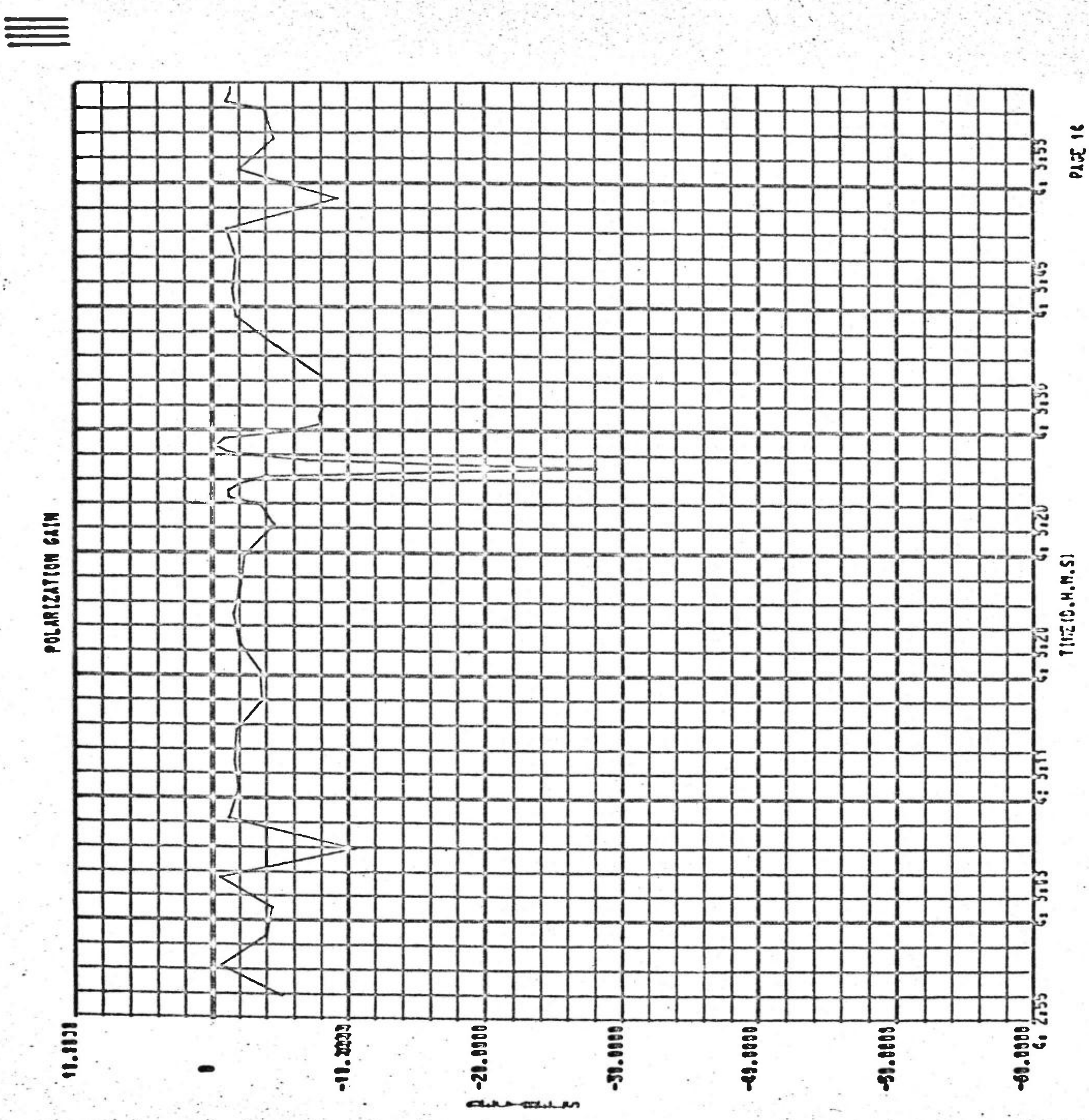
CSM ANTENNA GAIN

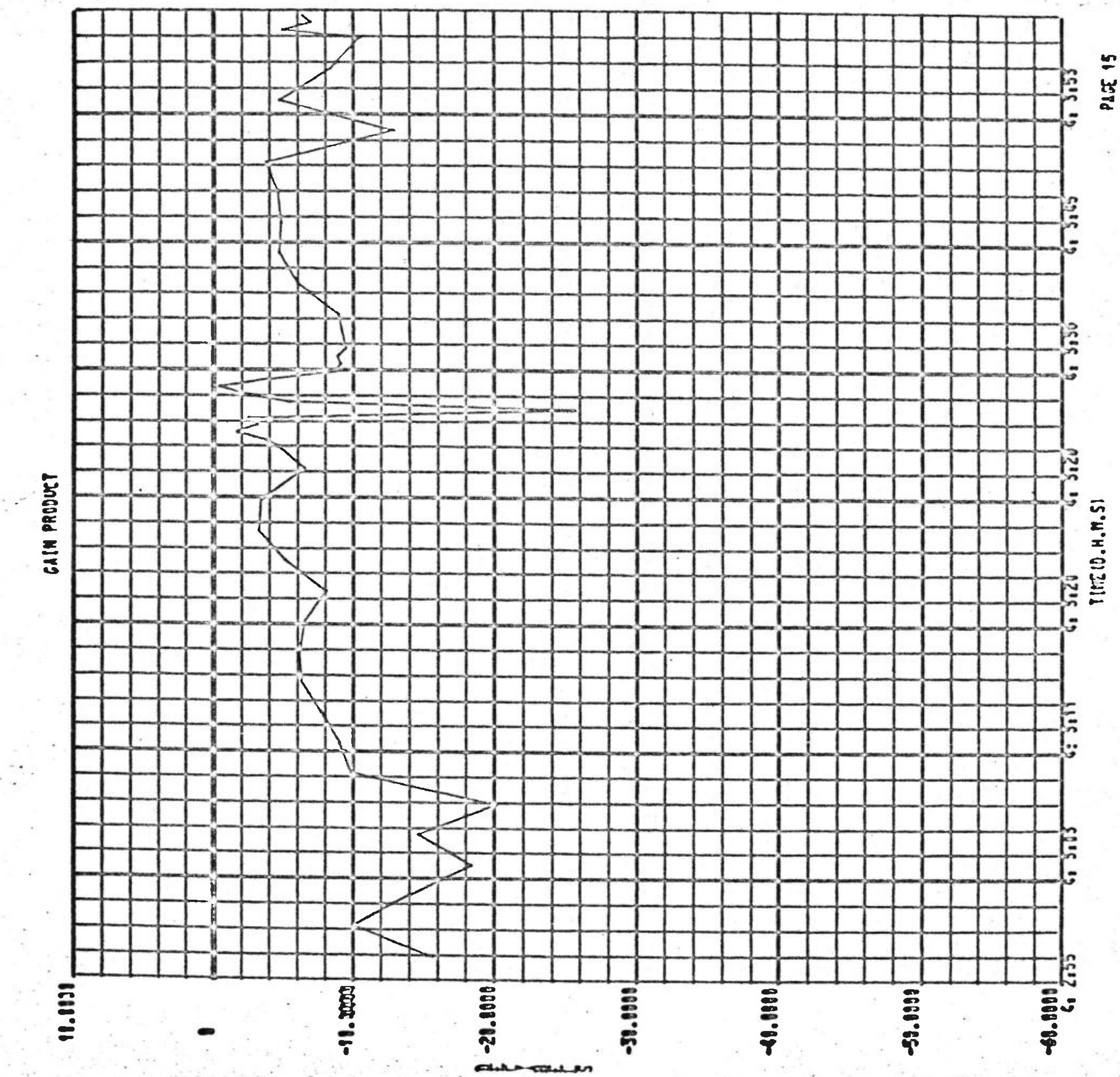


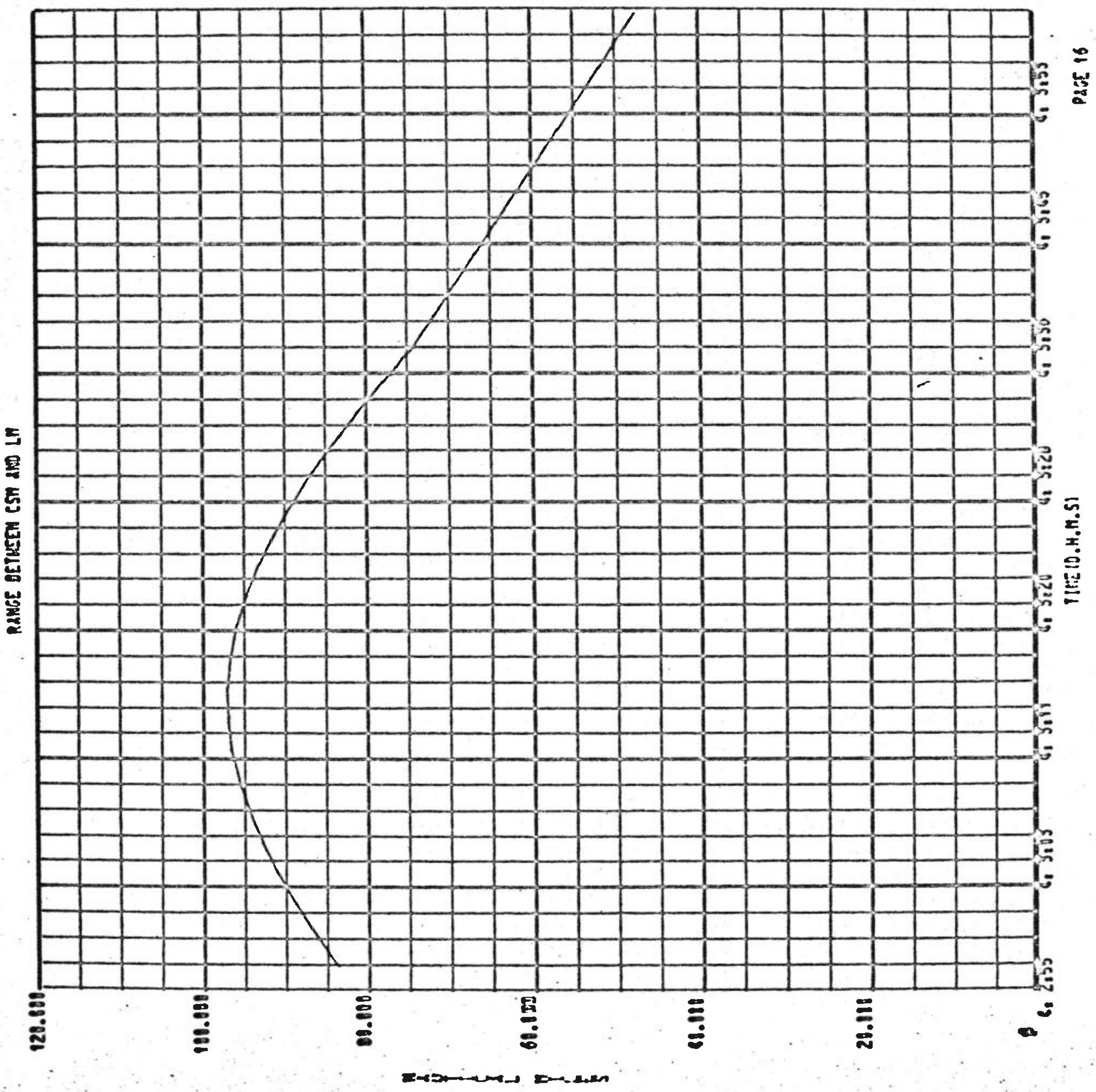
PAGE 12

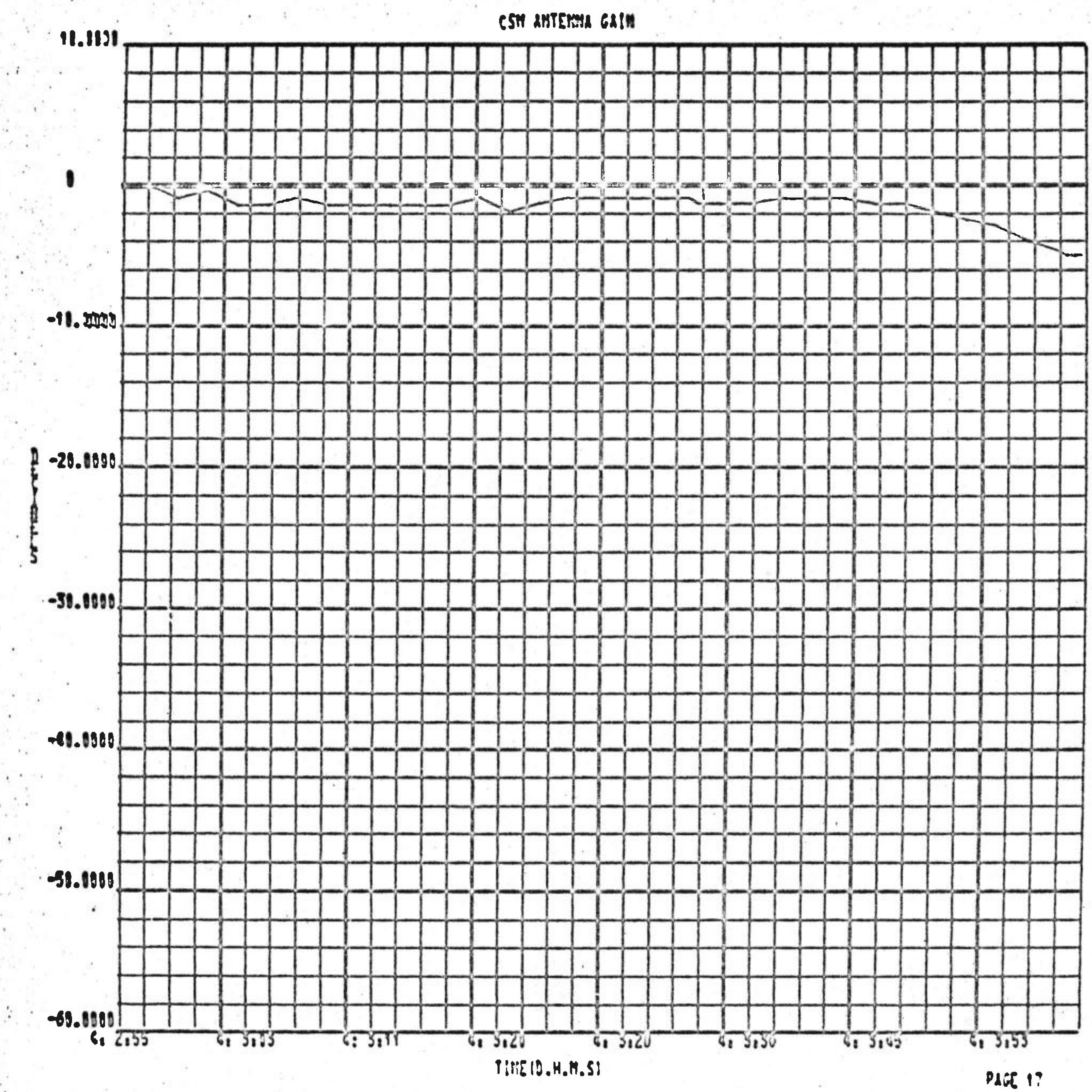
LM ANTENNA GAIN

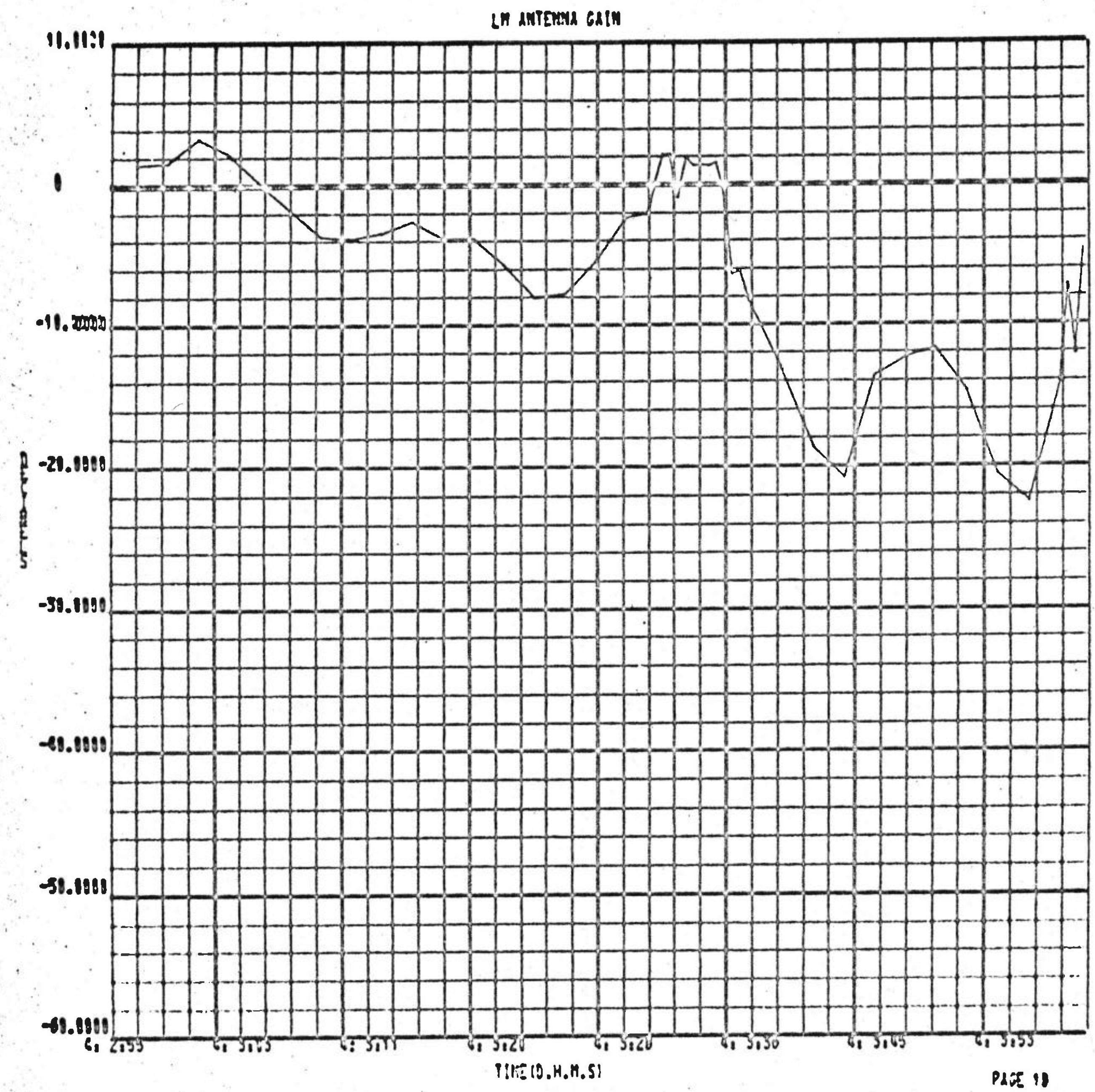



POLARIZATION GAIN
10.0000
0
-10.0000
-20.0000
-30.0000
-40.0000
-50.0000
-60.0000
TIME(D.H.M.S)
PAGE 16

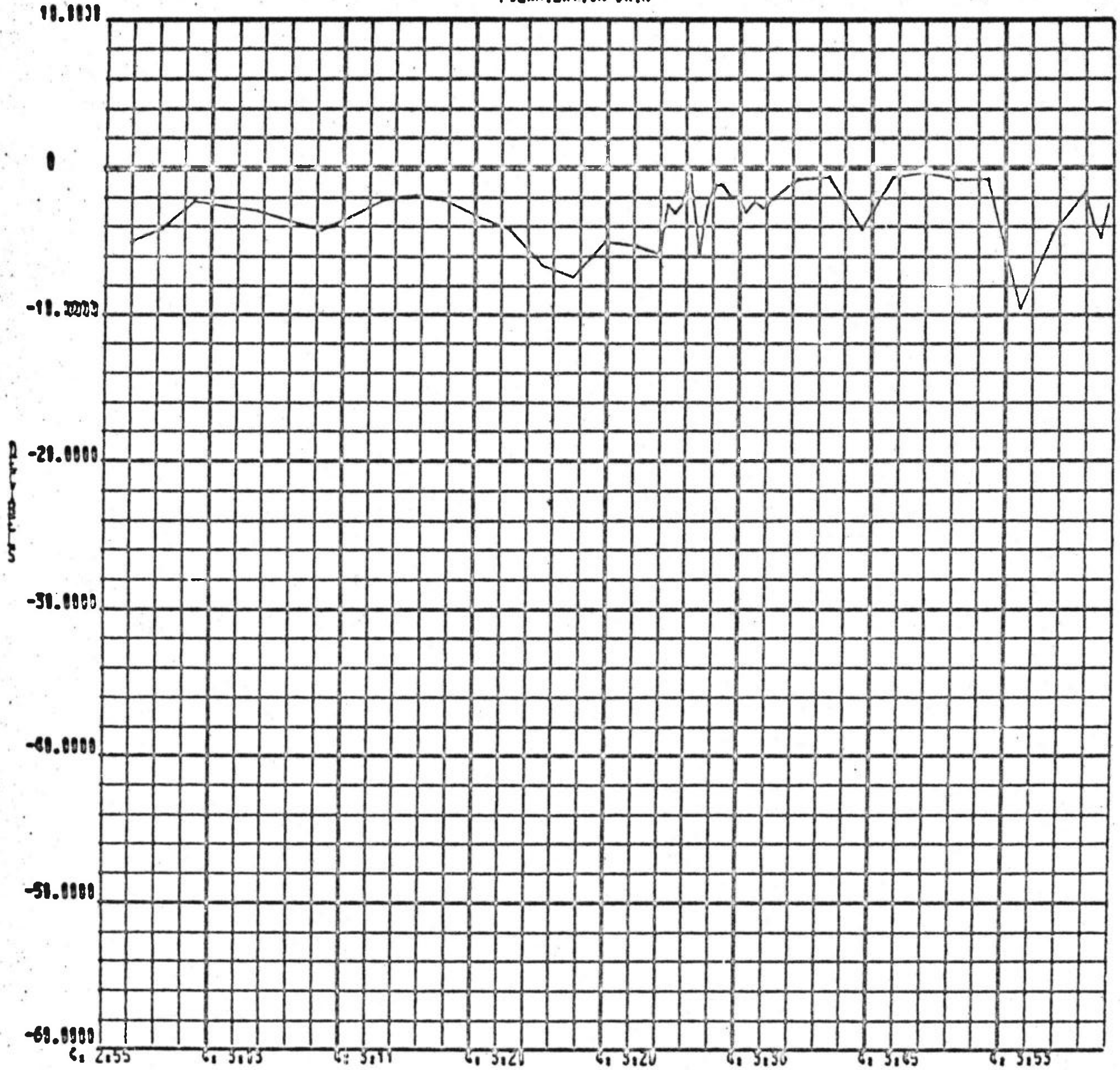

GAIN PRODUCT
10.0000
0
-10.0000
-20.0000
-30.0000
-40.0000
-50.0000
-60.0000
TIME(D.H.M.S)
PAGE 15

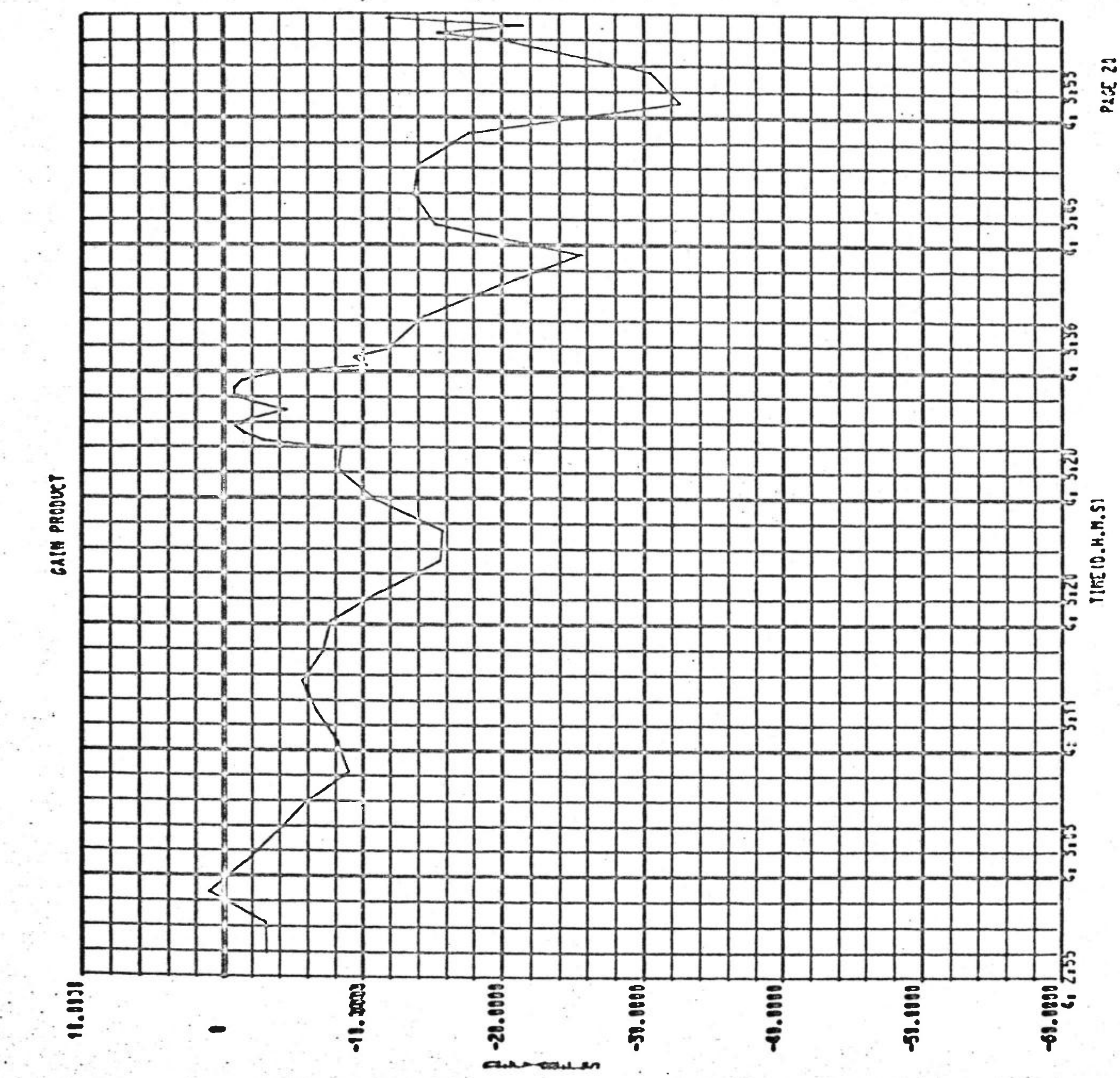

RANGE BETWEEN CSM AND LM
TIME (D.H.M.S)
PAGE 16
120.000
100.000
80.000
60.000
40.000
20.000


CSM ANTENNA GAIN
10.0000
0
-10.0000
-20.0000
-30.0000
-40.0000
-50.0000
-60.0000
TIME(D.H.M.S)
PAGE 17

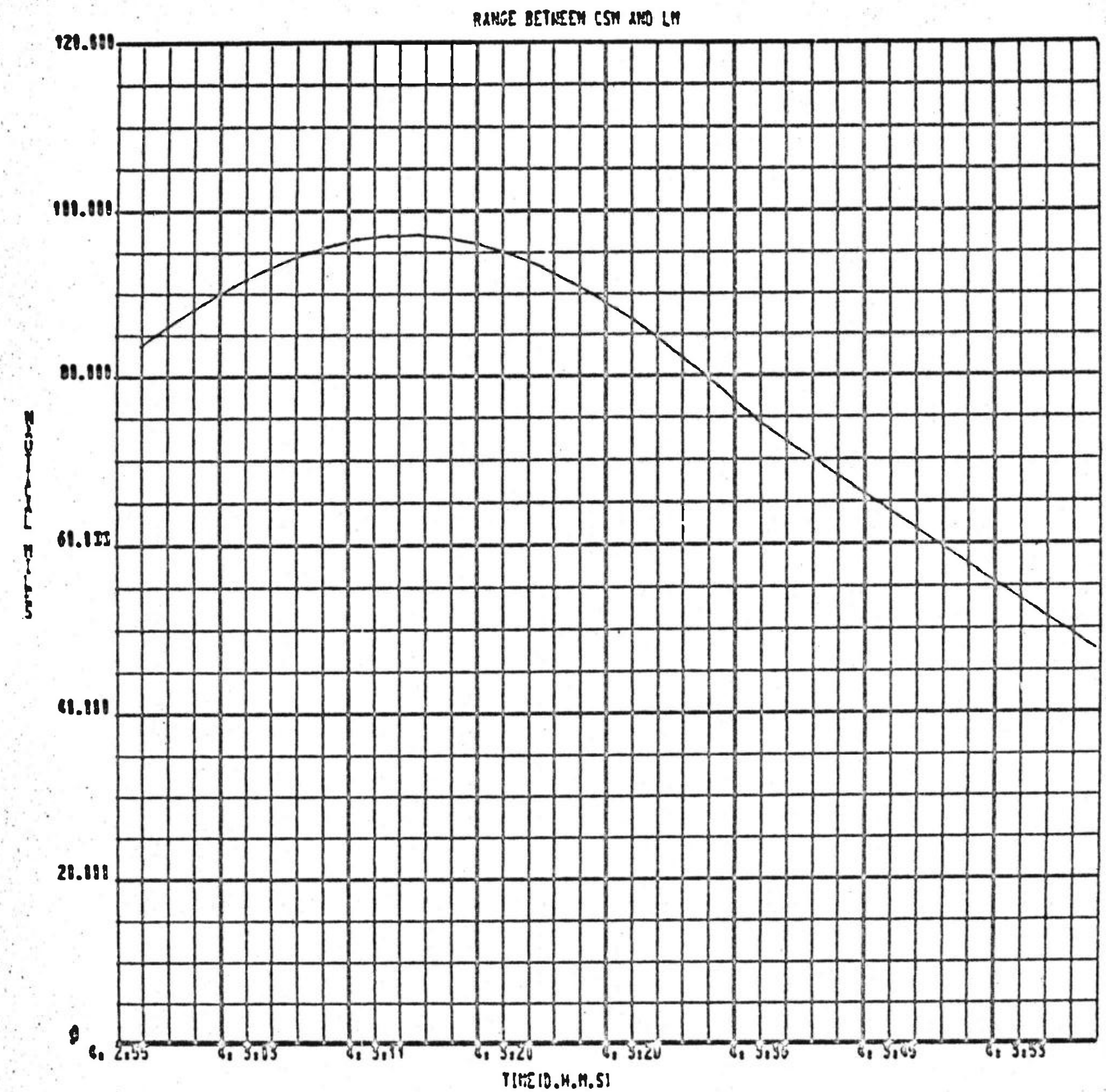

LM ANTENNA GAIN
10.0000
0
-10.0000
-20.0000
-30.0000
-40.0000
-50.0000
-60.0000
TIME(D.H.M.S)

POLARIZATION GAIN





GAIN PRODUCT
10.0000
0
-10.0000
-20.0000
-30.0000
-40.0000
-50.0000
-60.0000
TIME(D.H.M.S)
PAGE 23


RANGE BETWEEN CSM AND LM
120.000
100.000
80.000
60.000
40.000
20.000
0
TIME(D.H.M.S)

5. OPERATING PROCEDURES

This program is written in FORTRAN V for use with the SRU 1108 system. Presently this program is in production only for the SRU 1108 system. Only one input tape (the current trajectory tape) is required by this program in addition to the standard input and output tapes required by the SRU 1108 system. All antenna pattern data required by this program is included in a second file on the original PCF tape; hence, care must be taken when copying this PCF tape so as not to lose this data. Also, the PCF tape must be mounted on unit "X" (FORTRAN I. D. Number 27) and must be positioned at the beginning of the second file at the time this program is executed.

5.1 DECK SETUP -- NO PLOT OUTPUT DESIRED

```
C
C
1
$JOB etc.
∇△ASG△E=(trajectory tape number)
∇△ASG△X=(PCF tape number)
∇△XQT△CUR
△ TRW△X
△ IN△X
∇△XQT△HV014B
[program data
∇△EOF
```

where ∇ indicates a 7/8 punch and △ indicates one or more blank spaces.

5.2 DECK SETUP -- PLOT OUTPUT DESIRED

```
C
C
1
$JOB etc.
∇△ASG△E=(trajectory tape number)
∇△ASG△F (one FASTRAND unit is required)
∇△ASG△X=(PCF tape number)
∇△XQT△CUR
△ TRW△X
△ IN△X
∇△XQT△HV014B
[program data
∇△XQT△TRWPLT
[plotting data
∇△EOF
```

5.3 COMPUTER RUN REQUEST

In submitting the card deck for a computer run, the run time may be estimated at approximately two seconds per data case per time-point and the print output may be estimated at approximately four pages per thirty-five time-points for standard output and approximately eight pages per time-point for optional output. In case plots are generated, the microfilm or calcomp output estimate depends upon the number of plots generated in the plotting program.

5.3.1 Sample Run Request Form

INSTRUCTIONS FOR SCIENTIFIC COMPUTER RUNS

PRIORITY ☐

RE-RUN ☐

(DO NOT FILL IN SHADED AREAS)

TIME IN

LEFT TRW

RETURN TRW

PROGRAMMER	BADGE NO.	BOX NO.	PHONE NO.	DATE (M, D, Y)
ARGILA, C.	T57659	TRW	2503	07 01 68

DIVISION CODE	PROG. NO.	PROJ. NO.	EST. TIME (MIN)	MAX. TIME (MIN)	LINES OUTPUT	SEQ. NUMBER	TIME OUT
TRW	A025	1135P	2	3	2		

OPERATING SYSTEM

- 1108 FORTRAN V [X]
- FORTRAN FAP ☐
- 1108 FORTRAN IV ☐
- IBSYS ☐
- SYSB ☐
- 1108 COBOL ☐
- OTHER

TYPE OF RUN

- PROD. ☐
- TEST [X]
- OTHER (EXPLAIN BELOW)

COMPUTER

- 1108 [X]
- 360 ☐
- 7094 ☐
- OTHER

INPUT TAPES

UNIT	REEL	BIN NO.	DEN
E	1996		8
X	54321		8

OUTPUT TAPES

UNIT	REEL NO.	BIN NO.	DEN-SITY	SAVE	PROCESSING REQUIRED

WORKING TAPES

ACTUAL TIME USAGE

START

STOP

CHECK FOR		REEL NO.	NO. FRAMES	PROCESSING
CALCOMP ☐	4020 [X]		40	

ABNORMAL STOPS

STOP AT LOC.

SR.

LOOPING - LOC.

THRU

EXCESS OUTPUT

EXCESS TIME

PROBLEM NO.	H	3	6	1	0	1
PROGRAM NAME	H	V	0	1	4	B
TOTAL TAPES						2
INPUT (100'S CARD)						1
OUTPUT (100'S LINES)						
OUTPUT (100'S CARDS)						
RUN AT	TRW	MAC	MFD	OTHER		

PROGRAMMER'S COMMENTS:

ONE FASTRAND REQUIRED

OPERATOR'S COMMENTS:

SYSTEM OPERATOR

PERIPHERAL OPERATOR

SYSTEMS 7011-H (9-13-67)

6. REFERENCES

Argila, C. A., VHF Antenna Simulation Program for Apollo Missions D and E, HCC Report No. 3141-60.165, May 1968.

Lee, P. H., VHF Multipath Analysis, TRW Report No. 68:7251-PL-85, May 1968.

Mireles, R., Outer Reflection Geometry for a Smooth, Perfect, Spherical Mirror, April 1968.

Pool, J. W., Mireles, R. Mathematical Model for VHF Antenna Coverage Analysis of Apollo Missions D and E, TRW Report No. 05952-H431-R0-00, February 1968.

7. PROGRAM LISTING

7.1 THE MAIN DRIVER, VHFASP

```
& FOR VHFASP,VHFASP                                                    04618652
UNIVAC 1108 FORTRAN V LEVEL    2206 0016 F5016M
THIS COMPILATION WAS DONE ON 02 JUL 68 AT 04618652

  MAIN PROGRAM

  STORAGE USED (BLOCK, NAME, LENGTH)

       0001    *CODE    003467
       0000    *DATA    117145
       0002    *BLANK   000000
       0003    DATA     000031
       0004    GNREF    000044


  EXTERNAL REFERENCES (BLOCK, NAME)

       0005    INITAL
       0006    REFLKT
       0007    LOOKCM
       0010    LOOKLM
       0011    VECSUB
       0012    VECMAG
       0013    POLANG
       0014    TILT
       0015    MISALN
       0016    GAINRR
       0017    DB
       0020    POLAR
       0021    VECDOT
       0022    POLARS
       0023    NREW$
       0024    NRBU$
       0025    NIO1$
       0026    NIO2$
       0027    NWBU$
       0030    NRNL$
       0031    NWDU$
       0032    NRDU$
```

```
0033     NERR2$
0034     ACOS
0035     COS
0036     CSQRT
0037     SIN
0040     CDV$
0041     ATAN2
0042     TAN
0043     CEXP
0044     CABS
0045     ATAN
0046     ASIN
0047     SQRT
0050     NWEF$
0051     NBSP$
0052     NSTOP$
```

STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

```
0001   000113 100L     0000   116464 1000F    0000   116200 1001F    0001   002775 1002G    0000
0001   000227 1004L    0000   116211 1005F    0001   003007 1006G    0001   000311 1007L    0000
0001   002612 1009L    0000   116474 1010F    0001   002644 1011L    0001   003023 1013G    0001
0000   116176 102F     0001   003043 1023G    0001   003053 1027G    0001   000202 103L     0001
0001   003107 1046G    0001   000413 106L     0001   000415 107L     0001   003172 1076G    0001
0001   003232 1116G    0001   003244 1122G    0001   003260 1127G    0001   003270 1133G    0001
0001   003310 1143G    0001   003322 1150G    0001   000015 115G     0001   003400 1167G    0001
0001   003430 1207G    0001   000022 122G     0001   000033 130G     0001   000041 136G     0001
0001   000057 150G     0001   000065 156G     0001   000072 162G     0001   000103 170G     0001
0001   000175 223G     0001   000215 237G     0001   000222 243G     0001   000235 253G     0001
0000   116402 26F      0000   116143 28F      0001   000164 29L      0001   000277 300G     0001
0001   000535 302L     0001   000603 303L     0001   000304 304G     0001   000651 304L     0001
0001   000363 335G     0001   000374 346G     0001   001026 350L     0001   002524 356L     0001
0001   001336 400L     0001   002526 401L     0001   000704 420G     0001   000712 424G     0001
0001   000727 435G     0001   003342 450L     0001   000762 451G     0001   003407 451L     0001
0000   117013 453F     0000   117031 454F     0000   117051 455F     0001   000770 455G     0001
0001   001213 504G     0001   001516 543G     0001   001604 554G     0001   001665 566G     0001
0001   002163 616G     0001   002207 623G     0001   002363 642G     0001   002573 676G     0001
0001   002620 713G     0001   002654 733G     0001   002667 737G     0001   002741 764G     0001
0000   116505 87F      0000   116362 93F      0001   000370 94L      0001   000336 96L      0000
```

```
0001    000676 999L     0000 R 116067 A        0000 C 037525 AA       0003 R 000000 ACURCY   0000
0000 R 037547 ALPHA     0000 R 037551 ARRAY    0000 P 116102 B        0000 C 037527 BB       0000
0000 R C56121 BETAS     0000 R 116073 C        0000 I 000001 COMENT   0000 I 000002 CSMPAT   0000
0017 R 000000 DB        0000 R 116066 DELTAR   0000 P 116022 DELTAT   0003 R 000001 EPSLNR   0003
0000 R 116035 FREQ      0000 I 037512 FREQCY   0000 R 116036 FREQRD   0004 R 000000 GNREFC   0004
0000 R 116104 GP        0000 P 116105 GPDB     0000 P 116055 G1       0000 R 116061 G1DB     0000
0000 R 115100 G1SDB     0000 R 116057 G2       0000 R 116062 G2DB     0000 R 116076 G2S      0000
0003 R 000003 HALFPI    0000 I 116024 I        0000 I 116051 ICODED   0000 I 116040 IFLAG    0000
0000 I 116044 INDEX     0000 I 116050 IPHI     0000 I 116047 ITHETA   0000 I 116025 I$       0000
0000 I 116042 JJJ       0000 R 037515 K        0000 I 056123 LEMPAT   0000 I 116032 LEMTAP   0000
0000 R 037516 LP        0000 R 037517 LPDB     0000 R C37520 LPS      0000 R 037522 LPSDB    0000
0000 I 116023 MULTIP    0000 I 116027 N        0000 I 116030 NCSM     0000 I 116031 NLEM     0000
0000 R 115633 PATERN    0003 R 000004 PI       0000 I 037513 PLOT     0000 I 037514 PRINT    0000
0000 C 037531 P1S       0000 C 037535 P2S      0000 C 037541 Q        0000 R 116045 R        0000
0000 C 037545 RB        0000 R 116046 RHO      0000 R 116071 RHO1S    0000 R 116072 RHO2S    0000
0000 R 115666 RRS       0000 P 116056 RR1      0000 R 116075 RR1S     0000 R 116060 RR2      0000
0000 R 116065 RS        0000 R 115670 RSCB     0000 R 115673 RSLB     0003 R 000005 RO       0000
0000 R 116064 R2S       0003 R 000006 SIGMA    0000 R 115676 TAUS     0000 R 116052 TAU1     0000
0000 R 116053 TAU2      0000 R 115702 TAU2S    0003 R 000007 TIME     0000 R 115704 TITLE    0000
0000 R 116016 TMIN      0003 R 000013 TWOPI    0003 R 000014 TWOPIC   0003 R 000015 UNITY    0000
0000 R 116033 VMIN      0003 R 000026 ZERO
```

```
00100   C ..........................................................................
00100   C .                                                                        .
00100   C .   VHF ANTENNA SIMULATION PROGRAM (VHFASP)        C. ARGILA, 1968       .
00100   C .                                                                        .
00100   C .   DESCRIPTION-                                                         .
00100   C .      THE VHF ANTENNA SIMULATION PROGRAM SIMULATES THE CSM-LM VHF       .
00100   C .   COMMUNICATION LINK FOR ALL APOLLO MISSIONS.  IN THE STANDARD         .
00100   C .   CONFIGURATION VHFASP PRINTS VALUES OF CSM AND LM ANTENNA GAINS,      .
00100   C .   POLARIZATION LOSSES, CSM-LM RANGE AND GAIN PRODUCT AS FUNCTIONS OF   .
00100   C .   TIME.  IN AN OPTIONAL MODE EVERY VARIABLE INVOLVED IN THE SIMULATION .
00100   C .   MAY BE PRINTED OUT.  A PLOT TAPE IS GENERATED FOR USE WITH THE TRWPLT .
00100   C .   GENERAL PLOTTING PROGRAM.  THIS PROGRAM IS WRITTEN IN FORTRAN V FOR USE .
00100   C .   ON THE SPU 1108 SYSTEM.                                              .
00100   C .                                                                        .
00100   C .   INPUT-                                                               .
00100   C .     CARD INPUT- CARD INPUT IS VIA THE FORTRAN NAMELIST '$INPUT.'  INPUT .
```

```
00100   C .   PARAMETERS ARE                                                    .
00100   C .      FREQCY    FREQUENCY FLAG.  FREQCY=-1 INDICATES LOW FREQUENCY,     .
00100   C .                FREQCY=+1 INDICATES HIGH FREQUENCY.  FREQCY INITIALIZED TO -1..
00100   C .      ADFLAG    ASCENT/DESCENT FLAG.  ADFLAG=-1 INDICATES DESCENT CONDITION, .
00100   C .                ADFLAG=+1 INDICATES ASCENT CONDITION.  ADFLAG INITIALIZED  .
00100   C .                TO -1.                                                   .
00100   C .      MULTIP    MULTI-PATH FLAG.  MULTIP=0 INDICATES OMIT MULTI-PATH     .
00100   C .                CALCULATIONS, MULTIP=1 INDICATES MULTI-PATH CALCULATIONS .
00100   C .                ARE TO BE EXECUTED.  MULTIP INITIALIZED TO 1.            .
00100   C .      PRINT     PRINT FLAG.  PRINT=0 INDICATES ONLY STANDARD PRINT OUTPUT IS .
00100   C .                DESIRED.  PRINT=1 INDICATES OPTIONAL PRINT OUTPUT IS ALSO .
00100   C .                DESIRED.  PRINT INITIALIZED TO 0.                        .
00100   C .      PLOT      PLOT FLAG.  PLOT=0 INDICATES NO PLOT TAPE IS TO BE GENERATED,.
00100   C .                PLOT=1 INDICATES PLOT TAPE IS TO BE GENERATED.  PLOT INITIAL- .
00100   C .                IZED TO 1.                                               .
00100   C .      TMIN      FOUR-POINT ARRAY REPRESENTING MINIMUM TIME TO BE CONSIDERED, .
00100   C .                IN DAYS, HOURS, MINUTES AND SECONDS.  INITIALIZED TO 0,0,0,0. .
00100   C .      TMAX      FOUR-POINT ARRAY REPRESENTING MAXIMUM TIME TO BE CONSIDERED, .
00100   C .                IN DAYS, HOURS, MINUTES AND SECONDS.  INITIALIZED TO 0,0,0,0. .
00100   C .      DELTAT    MINIMUM TIME (IN SECONDS) BETWEEN TIME-POINTS.  INITIALIZED .
00100   C .                TO 0.0.                                                  .
00100   C .     TAPE INPUT- A TRAJECTORY TAPE (MOUNTED ON UNIT E) AND A 'PACKED'    .
00100   C .   ANTENNA PATTERNS TAPE ARE THE ONLY TAPE INPUTS REQUIRED.  THEESE TAPES .
00100   C .   ARE IN THE FORMAT SPECIFIED IN THE FV014B USER'S MANUAL.              .
00100   C .                                                                         .
00100   C .   OUTPUT-                                                               .
00100   C .      IN ADDITION TO THE STANDARD AND OPTIONAL PRINT OUTPUT, A PLOT TAPE  .
00100   C .   IN THE FORMAT REQUIRED BY 'TRWPLT' MAY ALSO BE GENERATED.             .
00100   C .                                                                         .
00100   C .   SUBROUTINE USAGE-                                                     .
00100   C .      INITAL                                                             .
00100   C .      REFLKT                                                             .
00100   C .         VECTOR                                                          .
00100   C .         GMPRD                                                           .
00100   C .         GMTRA                                                           .
00100   C .      LOOK                                                               .
00100   C .         VECTOR                                                          .
00100   C .         GMPRD                                                           .
00100   C .      POLANG                                                             .
00100   C .         GMTRA                                                           .
00100   C .         GMPRD                                                           .
```

```
00100      C .       TILT                                                                .
00100      C .       MISALN                                                              .
00100      C .       GAINRR                                                              .
00100      C .       DR                                                                  .
00100      C .       POLAR                                                               .
00100      C .       POLARS                                                              .
00100      C .         VECTOR                                                            .
00100      C .         GMPRD                                                             .
00100      C .       DATA (BLOCK DATA SUBROUTINE)                                        .
00100      C .                                                                           .
00100      C .    NAMED COMMON USAGE-                                                    .
00100      C .       DATA                                                                .
00100      C .       GNREF                                                               .
00100      C .                                                                           .
00100      C ............................................................................
00100      C
00101            INTEGER ADFLAG, COMENT, CSMPAT(90,180), FREQCY, PLOT, PRINT
00103            REAL K, LP, LPDB, LPS(2), LPSDB, LR(2)
00104            COMPLEX AA, BB, P1S(2), P2S(2), Q, RA, RB
00105            DIMENSION ALPHA(2), ARRAY(37,200), BETAS(2), GNREFC(6,2), GNREFL(6
00105           $,4), LEMPAT(90,180), PATERN(6,4), RLCB(3), RRS(2), RSCB(3), RSLB(3
00105           $), TAUS(2), TAU1S(2), TAU2S(2), TIME(4), TITLE(14,5), TMAX(4), TMI
00105           $N(4), UNITY(3,3), ZERO(3)
00106            COMMON /DATA/ ACURCY, EPSLNR, EPSLNC, HALFPI, PI, RO, SIGMA, TIME,
00106           $TWOPI, TWOPIC, UNITY, ZERO
00107            COMMON /GNREF/ GNREFC, GNREFL
00110            NAME LIST /INPUT/ FREQCY, ADFLAG, COMENT, PRINT, PLOT, TMIN, TMAX,
00110           $DELTAT, MULTIP
00110      C
00110      C *** INITIALIZE TAPES
00111            REWIND 1
00112            REWIND 2
00113            REWIND 3
00114            REWIND 8
00114      C
00114      C *** DUMP ANTENNA PATTERNS ONTO HIGH SPEED DRUMS
00115            DO 99 I = 1, 4
00120            READ(27) CSMPAT
00126         99 WRITE(1) CSMPAT
00135            DO 98 I = 1, 6
00140            READ(27) LEMPAT
```

```
00146        98 WRITE(2) LEMPAT
00155           DO 86 I = 1, 2
00160           READ(27) LEMPAT
00166        86 WRITE (3) LEMPAT
00175           REWIND 27
00175     C
00175     C *** REWIND TAPE DRIVES
00176       100 REWIND 1
00177           REWIND 2
00200           REWIND 3
00201           REWIND 7
00201     C
00201     C *** INITIALIZE INPUT PARAMETERS
00202           CALL INITAL(FREQCY, ADFLAG, COMENT, PRINT, PLOT, TMIN, TMAX,
00202          $DELTAT, MULTIP)
00202     C
00202     C *** READ INPUT PARAMETERS
00203           READ(5,INPUT)
00203     C
00203     C *** READ COMMENT/DESCRIPTION CARDS
00206           IF(COMENT .EQ. 0) GO TO 103
00210           IF(COMENT .LE. 5) GO TO 29
00212           WRITE(6,28)
00214        28 FORMAT(104H1INPUT CARD DECK ERROR.  NUMBER OF COMMENT/DESCRIPTION
00214          $CARDS EXCEEDS FIVE.  EXTRA CARDS WILL BE IGNORED./27HOBREVITY IS T
00214          $HE SOUL OF WIT/15X,12H-SHAKESPEARE)
00215           COMENT = 5
00216        29 DO 101 J = 1, COMENT
00221       101 READ(5,102) (TITLE(I,J),I=1,14)
00230       102 FORMAT(13A6,1A2)
00230     C
00230     C *** PRINT PARAMETER LISTING AND REPORT HEADING
00231       103 WRITE(6,1001)
00233      1001 FORMAT(1H1,43X,22HVHF ANTENNA SIMULATION)
00234           IF(COMENT .EQ. 0) GO TO 1004
00236           DO 1002 J = 1, COMENT
00241      1002 WRITE(6,1003) (TITLE(I,J),I=1,14)
00250      1003 FORMAT(15X,13A6,1A2)
00251      1004 WRITE(6,1005) TMIN, TMAX, DELTAT, FREQCY, ADFLAG, MULTIP, PRINT,
00251          $PLOT, COMENT
00272      1005 FORMAT(1H0,45X,17HPARAMETER LISTING                        /33X
```

```
00272           $,31HMINIMUM TIME (D.H.M.S)         = 3F3.0,F4.1              /33X
00272           $,31HMAXIMUM TIME (D.H.M.S)         = 3F3.0,F4.1              /33X
00272           $,31HTIME INCREMENT (SECONDS)       = F13.1                   /33X
00272           $,31HFREQUENCY FLAG (-1,1)          = I13                     /33X
00272           $,31HASCENT/DESCENT FLAG (-1,1)     = I13                     /33X
00272           $,31HMULTI-PATH FLAG (0,1)          = I13                     /33X
00272           $,31HPRINT FLAG (0,1)               = I13                     /33X
00272           $,31HPLOT FLAG (0,1)                = I13                     /33X
00272           $,31HCOMMENT/DESCRIPTION CARDS      = I13)
00273            WRITE(6,1001)
00275            IF(COMENT .EQ. 0) GO TO 1007
00277            DO 1006 J = 1, COMENT
00302       1006 WRITE(6,1003) (TITLE(I,J),I=1,14)
00311       1007 WRITE(6,1008)
00313       1008 FORMAT(1H0,109(1H-),/7X,4HTIME,10X,5HRANGE,6X,9HG-PRODUCT,5X,14HCS
00313           $M GAINS (DB),8X,14HLEM GAINS (DB),5X,23HPOLARIZATION GAINS (DB)/4X
00313           $,9H(D.H.M.S),9X,4H(NM),8X,4H(DB),7X,3(6HDIRECT,4X,8HSPECULAR,4X)/1
00313           $X,109(1H-))
00313     C
00313     C *** POSITION DRUMS FOR THIS DATA CASE
00314            N = FREQCY + 1
00315            NCSM = N / 2 + 1
00316            IF(N .EQ. 0) GO TO 96
00320            DO 97 I = 1, N
00323         97 READ(1)
00326         96 N = 2 * FREQCY + ADFLAG + 3
00327            NLEM = N / 2 + 1
00330            LEMTAP = 2 + N/6
00331            N = MOD(N,6)
00332            IF(N .EQ. 0) GO TO 94
00334            DO 95 I = 1, N
00337         95 READ(2)
00342         94 CONTINUE
00342     C
00342     C *** COMPUTE TIME LIMITS (IN SECONDS) AND SET FREQUENCY
00343            VMIN = 0.0
00344            VMAX = 0.0
00345            DO 104 I = 1, 4
00350            VMIN = VMIN + TIME(I)*TMIN(I)
00351        104 VMAX = VMAX + TIME(I)*TMAX(I)
00353            IF(FREQCY) 105,105,106
```

```
00356       105 FREQ = 259.7
00357           GO TO 107
00360       106 FREQ = 296.8
00361       107 FREQRD = FREQ * 1.0E+06 * TWOPI
00361     C
00361     C *** SCAN TRAJECTORY TAPE AND COMPUTE LOCK ANGLES
00362           NPOINT = 1
00363       300 READ(7) (ARRAY(I,NPOINT),I=1,25)
00371           IF(ARRAY(1,NPOINT) .LT. VMIN) GO TO 300
00373           IF(ARRAY(1,NPOINT) .GT. VMAX) GO TO 999
00375           CALL REFLKT(ARRAY(8,NPOINT),ARRAY(2,NPOINT),ARRAY(17,NPOINT),
00375          $ARRAY(5,NPOINT),UNITY,ZERO,RO,ACURCY,ARRAY(27,NPOINT),IFLAG)
00376           IFLAG = IFLAG + 5 * MULTIP
00377           ARRAY(26,NPOINT) = IFLAG
00400           GO TO (303,304,304,303,303,302,304,304,303,304), IFLAG
00401       302 CALL LOOKCM(ARRAY(2,NPOINT),ARRAY(27,NPOINT),ARRAY(8,NPOINT),
00401          $ARRAY(32,NPOINT),ARRAY(33,NPOINT))
00402           CALL LOOKLM(ARRAY(5,NPOINT),ARRAY(27,NPOINT),ARRAY(17,NPOINT),
00402          $ARRAY(36,NPOINT),ARRAY(37,NPOINT))
00403       303 CALL LOOKCM(ARRAY(2,NPOINT),ARRAY(5,NPOINT),ARRAY(8,NPOINT),
00403          $ARRAY(30,NPOINT),ARRAY(31,NPOINT))
00404           CALL LOOKLM(ARRAY(5,NPOINT),ARRAY(2,NPOINT),ARRAY(17,NPOINT),
00404          $ARRAY(34,NPOINT),ARRAY(35,NPOINT))
00405       304 VMIN = ARRAY(1,NPOINT) + DELTAT
00406           NPOINT = NPOINT + 1
00407           IF(VMIN .GT. VMAX) GO TO 999
00411           IF(NPOINT .LE. 200) GO TO 300
00413           WRITE(6,93)
00415        93 FORMAT(1H0,86HNUMBER OF TRAJECTORY POINTS EXCEEDS 200.  ONLY THE F
00415          $IRST 200 POINTS WILL BE PROCESSED.)
00416       999 NPOINT = NPOINT - 1
00416     C
00416     C *** SCAN ANTENNA PATTERNS AND PROCESS EACH DATA POINT
00417           DO 501 III = 1, 2
00422           READ(1) CSMPAT
00430           DO 500 JJJ = 1, 2
00433           READ(LEMTAP) LEMPAT
00433     C
00433     C *** PRINT LINK BEING SIMULATED
00441           LINK = 2*(III-1)+JJJ
00442           WRITE(6,26) LINK, III, JJJ
```

```
00447         26 FORMAT(1H0,33X,9HLINK NO. ,I1,15H.  CSM ANTENNA ,I1,14H, LEM ANTEN
00447          $NA ,I1,1H.)
00450            DO 703 INDEX = 1, NPOINT
00450      C
00450      C
00450      C *** CONVERT TIME TO (D.H.M.S)
00453            TMIN(4) = ARRAY(1,INDEX)
00454            DO 27 I = 1, 3
00457            TMIN(I) = AINT(TMIN(4)/TIME(I))
00460         27 TMIN(4) = TMIN(4) - TMIN(I)*TIME(I)
00460      C
00460      C *** TRANSFER IF NO CALCULATIONS ARE TO BE EXECUTED THIS TIME-POINT
00462            IFLAG = ARRAY(26,INDEX)
00463            GO TO (350,450,450,350,350,350,450,450,350,450), IFLAG
00463      C
00463      C *** PERFORM DIRECT PATH CALCULATIONS
00463      C *** COMPUTE THE RANGE BETWEEN THE CSM AND LEM, R
00464        350 CALL VECSUB(ARRAY(2,INDEX),ARRAY(5,INDEX),3,RLCB)
00465            CALL VECMAG(RLCB,3,R)
00465      C
00465      C *** COMPUTE POLARIZATION ANGLE ALONG DIRECT PATH, RHO
00466            CALL POLANG(ARRAY(30,INDEX),ARRAY(31,INDEX),ARRAY(34,INDEX),
00466          $ARRAY(35,INDEX),ARRAY(8,INDEX),ARRAY(17,INDEX),RHO)
00466      C
00466      C *** FIND DIRECT PATH ANTENNA PATTERN DATA
00467            ITHETA = (IFIX(ARRAY(30,INDEX) * 57.296) + 1) / 2
00470            IPHI = (IFIX(ARRAY(31,INDEX) * 57.296) + 1) / 2
00471            ICODED = CSMPAT(ITHETA,IPHI)
00472            DO 91 I = 1, 6
00475            J = 3*(III-1) + (I+1)/2
00476         91 PATERN(I,1) = GNREFC(J,NCSM) - FLD(36-6*I,6,ICODED)
00500            ITHETA = (IFIX(ARRAY(34,INDEX) * 57.296) + 1) / 2
00501            IPHI = (IFIX(ARRAY(35,INDEX) * 57.296) + 1) / 2
00502            ICODED = LEMPAT(ITHETA,IPHI)
00503            DO 90 I = 1, 6
00506            J = 3*(JJJ-1) + (I+1)/2
00507         90 PATERN(I,3) = GNREFL(J,NLEM) - FLD(36-6*I,6,ICODED)
00507      C
00507      C *** COMPUTE POLARIZATION ELLIPSE TILT ANGLES, TAU1 AND TAU2, ALONG DIRECT PATH
00511            CALL TILT(PATERN(1,1),PATERN(2,1),PATERN(3,1),PATERN(4,1),TAU1)
00512            CALL TILT(PATERN(1,3),PATERN(2,3),PATERN(3,3),PATERN(4,3),TAU2)
```

```
00512         C
00512         C *** COMPUTE COORDINATE SYSTEM MIS-ALIGNMENT ALONG DIRECT PATH, BETA
00513               CALL MISALN(TAU1,TAU2,RHO,BETA)
00513         C
00513         C *** COMPUTE ANTENNA GAINS, G1, G2, AND AXIAL RATIOS, RR1, RR2, ALONG DIRECT P
00514               CALL GAINRR(PATERN(5,1),PATERN(6,1),G1,RR1)
00515               CALL GAINRR(PATERN(5,3),PATERN(6,3),G2,RR2)
00516               G1DB = DB(G1)
00517               G2DB = DB(G2)
00517         C
00517         C *** COMPUTE POLARIZATION LOSS ALONG DIRECT PATH, LP
00520               CALL POLAR(RR1,RR2,BETA,LP)
00521               LPDB = DB(LP)
00521         C
00521         C *** TRANSFER IF MULTI-PATH CALCULATIONS ARE NOT REQUIRED
00522               IF(IFLAG .EQ. 6) GO TO 400
00524               K = 0.0
00525               GO TO 401
00525         C
00525         C *** PERFORM MULTIPATH CALCULATIONS
00525         C *** COMPUTE SPECULAR PATH LENGTHS R1S, R2S, RS, DELTAR AND THE ANGLE PSI
00526           400 CALL VECSUB(ARRAY(5,INDEX),ARRAY(27,INDEX),3,RSLB)
00527               CALL VECSUB(ARRAY(2,INDEX),ARRAY(27,INDEX),3,RSCB)
00530               CALL VECMAG(RSCB,3,R1S)
00531               CALL VECMAG(RSLB,3,R2S)
00532               RS = R1S + R2S
00533               DELTAR = RS - R
00534               CALL VECDOT(ARRAY(27,INDEX),RSLB,3,A)
00535               PSI = HALFPI - ACOS(A / (RO * R2S))
00535         C
00535         C *** COMPUTE MULTIPATH POLARIZATION ANGLES, RHO1S AND RHO2S
00536               CALL POLARS(ARRAY(2,INDEX),ARRAY(5,INDEX),ARRAY(27,INDEX),ARRAY(8,
00536              $INDEX),ARRAY(17,INDEX),PSI,ARRAY(32,INDEX),ARRAY(33,INDEX),ARRAY(3
00536              $6,INDEX),ARRAY(37,INDEX),RHO1S,RHO2S)
00536         C
00536         C *** FIND SPECULAR PATH ANTENNA PATTERN DATA
00537               ITHETA = (IFIX(ARRAY(32,INDEX) * 57.296) + 1) / 2
00540               IPHI = (IFIX(ARRAY(33,INDEX) * 57.296) + 1) / 2
00541               ICODED = CSMPAT(ITHETA,IPHI)
00542               DO 85 I = 1, 6
00545               J = 3*(III-1) + (I+1)/2
```

```
00546      89 PATERN(I,2) = GNREFC(J,NCSM) - FLD(36-6*I,6,ICODED)
00550         ITHETA = (IFIX(ARRAY(36,INDEX) * 57.296) + 1) / 2
00551         IPHI = (IFIX(ARRAY(37,INDEX) * 57.296) + 1) / 2
00552         ICODED = LEMPAT(ITHETA,IPHI)
00553         DO 88 I = 1, 6
00556         J = 3*(JJJ-1) + (I+1)/2
00557      88 PATERN(I,4) = GNREFL(J,NLEM) - FLD(36-6*I,6,ICODED)
00557   C
00557   C *** COMPUTE POLARIZATION ELLIPSE TILT ANGLE OF RECEIVING ANTENNA, TAU2S
00561         CALL TILT(PATERN(1,2),PATERN(2,2),PATERN(3,2),PATERN(4,2),TAU2S(1)
00561        $)
00562         CALL TILT(PATERN(1,4),PATERN(2,4),PATERN(3,4),PATERN(4,4),TAU2S(2)
00562        $)
00562   C
00562   C *** COMPUTE POLARIZATION ELLIPSE TILT ANGLE OF TRANSMITTING ANTENNA, TAU1S
00563         TAU1S(1) = TAU2S(1) - RHO1S
00564         TAU1S(2) = TAU2S(2) - RHO2S
00565         DO 351 I = 1, 2
00570     351 IF(TAU1S(I) .LT. 0.0) TAU1S(I) = TAU1S(I) + PI
00570   C
00570   C *** COMPUTE MULTI-PATH REFLECTION COEFFICIENTS, RA AND RB
00573         AA = CMPLX(EPSLNR,-SIGMA/(FREQRD * EPSLNO))
00574         BB = CSQRT(AA - COS(PSI)**2)
00575         C = SIN(PSI)
00576         A = AA * C
00577         RA = (C - BB) / (C + BB)
00600         RB = (A - BB) / (A + BB)
00600   C
00600   C *** COMPUTE CSM AND LEM ANTENNA GAINS AND AXIAL RATIOS ALONG SPECULAR PATH
00601         CALL GAINRR(PATERN(5,2),PATERN(6,2),G1S,RR1S)
00602         CALL GAINRR(PATERN(5,4),PATERN(6,4),G2S,RR2S)
00603         G1SDB = DB(G1S)
00604         G2SDB = DB(G2S)
00604   C
00604   C *** COMPUTE POLARIZATION RATIO OF INCIDENT WAVE, P1S
00605         ALPHA(1) = ATAN2(1.0,RR1S) * 2.0
00606         ALPHA(2) = ATAN2(1.0,RR2S) * 2.0
00607         DO 352 I = 1, 2
00612     352 P1S(I) = TAN(0.5 * ACOS(COS(ALPHA(I)) * COS(2.0 * TAU1S(I)))) *
00612        *CEXP(CMPLX(0.0,ATAN2(TAN(ALPHA(I)),SIN(2.0 * TAU1S(I)))))
00612   C
```

```
C *** COMPUTE DEPOLARIZATION FACTOR, Q, AND POLARIZATION RATIO OF REFLECTED WAVE
      Q = RA / RB
      DO 353 I = 1, 2
  353 P2S(I) = Q * P1S(I)
C
C *** COMPUTE TILT ANGLE, TAUS, AND AXIAL RATIO, RRS, OF REFLECTED WAVE
      DO 354 I = 1, 2
      A = 2.0 * ATAN(CABS(P2S(I)))
      B = ATAN2(AIMAG(P2S(I)),REAL(P2S(I)))
      RRS(I) = 1.0 / TAN(0.5 * ASIN(SIN(A) * SIN(B)))
  354 TAUS(I) = 0.5 * ATAN(TAN(A) * COS(B))
C
C *** COMPUTE THE COORDINATE SYSTEM MIS-ALIGNMENT ALONG THE SPECULAR PATH, BETAS
      CALL MISALN(TAU2S(1),TAUS(1),RHO2S,BETAS(1))
      CALL MISALN(TAU2S(2),TAUS(2),RHO1S,BETAS(2))
C
C *** COMPUTE SPECULAR POLARIZATION LOSS, LPS
      CALL POLAR(RRS(1),RR2S,BETAS(1),LPS(1))
      CALL POLAR(RRS(2),RR1S,BETAS(2),LPS(2))
      LPSDB = DB(AMIN1(LPS(1),LPS(2)))
C
C *** COMPUTE THE LUNAR REFLECTION LOSS, LR
      A = CABS(RA) **2
      B = CABS(RB) **2
      DO 355 I = 1, 2
      C = CABS(P1S(I)) **2
  355 LR(I) = (B + A * C) / (1.0 + C)
C
C *** COMPUTE MULTI-PATH DIVERGENCE FACTOR, D
      A = 2.0 * R1S * R2S
      B = SIN(PSI)
      D = 1.0 / SQRT(1.0 + A * (1.0 + B**2) / (RS * RO * B) + (A / RO)**
     $   2 / RS)
C
C *** COMPUTE MULTIPATH REFLECTION FACTOR, K
      A = F * G1S * G2S * D / (RS * G1 * G2)
      ALPHA(1) = A * LR(1) * SQRT(LPS(1) / LP)
      ALPHA(2) = A * LR(2) * SQRT(LPS(2) / LP)
      C = COS(AMOD(TWOPIC*FREQ*DELTAR,TWOPI))
      IF(ABS(ALPHA(1) + C) .LT. ABS(ALPHA(2) + C)) GO TO 356
      K = ALPHA(2)
```

```
00661            GO TO 401
00662        356 K = ALPHA(1)
00662      C
00662      C *** COMPUTE GAIN PRODUCT GP
00663        401 GP = LP * G1 * G2 * (1.0 + K * (K + 2.0 * C))
00664            GPDB = DB(GP)
00664      C
00664      C *** PRINT STANDARD OUTPUT
00665            GPDB = G1DB + G2DB + LPDB
00666            IF(IFLAG .EQ. 6) GO TO 1009
00670            IF(IFLAG .EQ. 9) WRITE(6,998)
00673        998 FORMAT(1X,107HTHE FOLLOWING TIME-POINT HAS THE LM AND CSM LYING AL
00673           $ONG A LUNAR RADIUS.  THE EXISTENCE OF A 'BOUNCE-POINT' /103H IN SU
00673           $CH A SITUATION IS MORE ACADEMIC THAN ACTUAL.  HENCE, ONLY DIRECT P
00673           $ATH CALCULATIONS WERE EXECUTED.)
00674            WRITE(6,1000) TMIN, R, GPDB, G1DB, G2DB, LPDB
00707       1000 FORMAT(1X,3F3.0,F4.1,6X,F6.2,7X,F7.2,4X,3(F7.2,15X))
00710            GO TO 1011
00711       1009 WRITE(6,1010) TMIN, R, GPDB, G1DB, G1SDB, G2DB, G2SDB, LPDB, LPSDB
00727       1010 FORMAT(1X,3F3.0,F4.1,6X,F6.2,7X,F7.2,4X,3(F7.2,3X,F7.2,5X))
00727      C
00727      C *** PRINT OPTIONAL OUTPUT
00730       1011 IF(PRINT .NE. 0) WRITE(6,87) (ARRAY(I,INDEX),I=1,37), PATERN, R,
00730           $RHO, TAU1, TAU2, BETA, G1, G2, RR1, RR2, LP, R1S, R2S, RS, DELTAR,
00730           $PSI, RHO1S, RHO2S, TAU1S, TAU2S, RA, RB, G1S, G2S, RR1S, RR2S,
00730           $P1S, P2S, Q, TAUS, RRS, BETAS, LPS, LR, D, K, GP
01042         87 FORMAT(9HOTIME    =1PE10.3/9H ROCB    =E10.3,2(1H,E10.3)/9H ROLB    =
01042           $E10.3,2(1H,E10.3)/9H ACB      =E10.3,8(1H,E10.3)/9H ALB      =E10.3,8(
01042           $1H,E10.3)/9H IFLAG   =E10.3/9H ROSP    =E10.3,2(1H,E10.3)/9H THTAC
01042           $=E10.3/9H PHIC     =E10.3/9H THTACS =E10.3/9H PHICS   =E10.3/9H THTAL
01042           $   =E10.3/9H PHIL     =E10.3/9H THTALS =E10.3/9H PHILS   =E10.3/9H PAT
01042           $ERN =3(6(E10.3,1H,)/9X)5(E10.3,1H,)E10.3/9H R         =E10.3/9H RHO
01042           $   =E10.3/9H TAU1     =E10.3/9H TAU2     =E10.3/9H BETA     =E10.3/9H G1
01042           $     =E10.3/9H G2       =E10.3/9H RR1      =E10.3/9H RR2      = E10.3/9H
01042           $LP       =E10.3/9H R1S      =E10.3/9H R2S      =E10.3/9H RS       =E10.3/9
01042           $H DELTAR =E10.3/9H PSI      =E10.3/9H RHO1S   =E10.3/9H RHO2S   =E10.3
01042           $/9H TAU1S   =E10.3,1H,E10.3/9H TAU2S   =E10.3,1H,E10.3/9H RA       =1H
01042           $(E9.3, 6H) + J(E9.3, 1H)/9H RB       =1H(E9.3, 6H) + J(E9.3, 1H)/9H
01042           $G1S      =E10.3/9H G2S      =E10.3/9H RR1S     =E10.3/9H RR2S     =E10.3/9
01042           $H P1S     =1H(E9.3, 6H) + J(E9.3, 2H),     2H (E9.3, 6H) + J(E9.3,1H
01042           $)/9H P2S      =1H(E9.3, 6H) + J(E9.3, 2H),    2H (E9.3, 6H) + J(E9.3
```

```
01042         $,1H)/9H Q       =1H(E9.3, 6H) + J(E9.3, 1H)/9H TAUS    =E10.3,1H,E10
01042         $.3/9H RRS     =E10.3,1H,E10.3/9H BETAS  =E10.3,1H,E10.3/9H LPS     =
01042         $E10.3,1H,E10.3/9H LP      =E10.3,1H,E10.3/9H D       =E10.3/9H K
01042         $   =E10.3/9H GP      =E10.3/)
01042   C
01042   C *** WRITE PLOT TAPE
01043           IF(PLOT .NE. 0) WRITE(8) (ARRAY(I,INDEX),I=1,37), R, RHO, TAU1,
01043         $TAU2, BETA, G1, G1DB, G2, G2DB, RR1, RR2, LP, LPDB, R1S, R2S, RS,
01043         $DELTAR, PSI, RHO1S, RHO2S, TAU1S, TAU2S, RA, RB, G1S, G1SDB, G2S,
01043         $G2SDB, RR1S, RR2S, P1S, P2S, Q, TAUS, RRS, BETAS, LPS, LPSDB, LR,
01043         $D, K, GP, GPDB
01160           GO TO 703
01160   C
01160   C *** NO CALCULATIONS HAVE BEEN EXECUTED - PRINT APPROPRIATE MESSAGE
01161     450 IF((IFLAG .EQ. 2) .OR. (IFLAG .EQ. 7)) GO TO 451
01163         IF((IFLAG .EQ. 3) .OR. (IFLAG .EQ. 8)) GO TO 452
01165         WRITE(6,453) TMIN
01173     453 FORMAT(1X,3F3.0,F4.1,5X,62HTHE 'BOUNCE-POINT' WAS NOT DETERMINED T
01173        $O THE DESIRED ACCURACY.)
01174         GO TO 703
01175     451 WRITE(6,454) TMIN
01203     454 FORMAT(1X,3F3.0,F4.1,5X74HLM AND CSM ARE ON OPPOSITE SIDES OF THE
01203        $LUNAR HORIZON.  TRY SMOKE SIGNALS.)
01204         GO TO 703
01205     452 WRITE(6,455) TMIN
01213     455 FORMAT(1X,3F3.0,F4.1,5X,78HLM AND CSM ARE ON DIAMETRICALLY OPPOSIT
01213        $E SIDES OF THE MOON-- SORRY 'BOUT THAT.)
01214     703 CONTINUE
01214   C
01214   C *** END-FILE PLOT TAPE
01216         IF(PLOT .NE. 0) END FILE 8
01220     500 CONTINUE
01220   C
01220   C *** RE-POSITION LEM TAPE
01222         BACK SPACE LEMTAP
01223         BACK SPACE LEMTAP
01224     501 CONTINUE
01226         GO TO 100
01227         END

END OF UNIVAC 1108 FORTRAN V COMPILATION.      0 *DIAGNOSTIC* MESSAGE(S)
```

```
& FOR INITAL,INITAL                                                          00&33&58
UNIVAC 1108 FORTRAN V LEVEL   2206 0016 F5016M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT 00&33&58

  SUBROUTINE INITAL    ENTRY POINT 000034

  STORAGE USED (BLOCK, NAME, LENGTH)

       0001   *CODE    000064
       0000   *DATA    000012
       0002   *BLANK   000000

  EXTERNAL REFERENCES (BLOCK, NAME)

       0003   NERR3$

  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)


00100      1*    C
00100      2*    C ..............................................................
00100      3*    C .                                                            .
00100      4*    C .   SUBROUTINE INITAL                       C. ARGILA, MAY '68  .
00100      5*    C .                                                            .
00100      6*    C .   DESCRIPTION-                                             .
00100      7*    C .      THIS SUBROUTINE INITIALIZES THE INPUT PARAMETERS TO THE VHF ANTENNA .
00100      8*    C .   SIMULATION PROGRAM (HV014B).                             .
00100      9*    C .                                                            .
00100     10*    C..............................................................
00100     11*    C
00101     12*          SUBROUTINE INITAL(FREQCY, ADFLAG, COMENT, PRINT, PLOT, TMIN, TMAX,
00101     13*         $DELTAT, MULTIP)
```

```
00103      14*          INTEGER FREQCY, ADFLAG, COMENT, PRINT, PLOT
00104      15*          DIMENSION TMIN(4), TMAX(4)
00105      16*          FREQCY = -1
00106      17*          ADFLAG = -1
00107      18*          COMENT = 0
00110      19*          PRINT = 0
00111      20*          PLOT = 1
00112      21*          TMIN(1) = 0.0
00113      22*          TMIN(2) = 0.0
00114      23*          TMIN(3) = 0.0
00115      24*          TMIN(4) = 0.0
00116      25*          TMAX(1) = 0.0
00117      26*          TMAX(2) = 0.0
00120      27*          TMAX(3) = 0.0
00121      28*          TMAX(4) = 0.0
00122      29*          DELTAT = 0.0
00123      30*          MULTIP = 1
00124      31*          RETURN
00125      32*          END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

7.2 SUBROUTINE REFLKT

```
& FOR REFLKT,REFLKT                                                   00&33&59
UNIVAC 1108 FORTRAN V LEVEL   2206 0016 F5016M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT CO&33&59

  SUBROUTINE REFLKT     ENTRY POINT 000735

  STORAGE USED (BLOCK, NAME, LENGTH)

        0001    *CODE   001015
        0000    *DATA   000127
        0002    *BLANK  000000
        0003    TEMPRY  000004


  EXTERNAL REFERENCES (BLOCK, NAME)

        0004    VECSUB
        0005    VECMAG
        0006    VECSCA
        0007    VECCRS
        0010    VECTST
        0011    VECDOT
        0012    GMTRA
        0013    GMPRD
        0014    VECEQU
        0015    CROSS
        0016    ETA
        0017    STRAIT
        0020    XO
        0021    YO
        0022    VECADD
        0023    ACOS
        0024    SIN
        0025    ASIN
        0026    COS
        0027    TAN
        0030    ATAN
        0031    SQRT
        0032    NERR3$
```

```
STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001     000113 10L         0001     000345 100L        0001     000666 1000L       0001     000502 150L        0001
0001     000521 170L        0001     000527 180L        0001     000537 190L        0001     000554 195L        0001
0001     000327 20L         0001     000561 200L        0001     000340 30L         0001     000565 300L        0001
0001     000661 400L        0000 R 000067 A             0000 R 000000 ABOT          0000 R 000075 ALPHA         0000
0000 R 000076 ALPHA1        0000 R 000011 A01           0000 R 000022 A01T          0000 R 000070 B             0000
0000 R 000103 CHIMAX        0000 R 000102 CHIMIN        0000 R 000077 CHI1          0000 R 000100 CHI2          0000
0015 R 000000 CROSS         0003 R 000001 DUMMY         0016 R 000000 ETA           0000 R 000074 GAMMA         0003
0000 R 000073 H2            0000 I 000066 I             0000 R 000033 RCAPC         0000 R 000036 RCAPL         0000
0000 R 000044 ROCO          0000 R 000047 ROLB          0000 R 000052 ROLO          0017 R 000000 STRAIT        0000
0000 R 000105 THETA1        0000 R 000055 U             0000 R 000060 V             0020 R 000000 XO            0003
0021 R 000000 YO            0003 R 000003 Y2            0000 R 000063 ZERO
```

```
00100   C
00100   C ...........................................................................
00100   C .                                                                         .
00100   C .   SUBROUTINE REFLKT                             C. ARGILA, APR. '68     .
00100   C .                                                                         .
00100   C .   DESCRIPTION-                                                          .
00100   C .      GIVEN A SPHERE M OF RADIUS RO AND CENTER AT O AND TWO POINTS C AND L, .
00100   C .   SUBROUTINE REFLKT DETERMINES THE UNIQUE POINT (IF IT EXISTS) ON M AT    .
00100   C .   WHICH A RAY EXTENDING FROM C OR L REFLECTS (BY SNELL'S LAW) AND MEETS   .
00100   C .   L OR C, RESPECTIVELY.  THIS PROGRAM WAS WRITTEN FROM ANALYSIS DEVELOPED .
00100   C .   BY R. MIRELES.                                                        .
00100   C .                                                                         .
00100   C .   ARGUMENTS-                                                            .
00100   C .      ABC    TRANSFORMATION MATRIX OF C-COORDINATES TO BASIC-COORDINATES   .
00100   C .      RBCB   POSITION VECTOR OF C-ORIGIN IN BASIC-COORDINATES              .
00100   C .      ABL    TRANSFORMATION MATRIX OF L-COORDINATES TO BASIC-COORDINATES   .
00100   C .      RBLB   POSITION VECTOR OF L-ORIGIN IN BASIC-COORDINATES              .
00100   C .      ABO    TRANSFORMATION MATRIX OF O-COORDINATES TO BASIC-COORDINATES   .
00100   C .      RBOB   POSITION VECTOR OF O-ORIGIN IN BASIC-COORDINATES              .
00100   C .      RO     RADIUS OF SPHERE M                                            .
00100   C .      ACURCY A MEASURE OF THE ACCURACY OF THE SOLUTION.  THE SMALLER       .
```

```
00100  C .                ACURCY, THE MORE ACCURATE THE SOLUTION WILL BE.                .
00100  C .       RBSB     POSITION VECTOR OF 'BOUNCE-POINT' IN BASIC-COORDINATES        .
00100  C .       IFLAG    = 1, SOLUTION DETERMINED TO DESIRED ACCURACY                  .
00100  C .                = 2, 'BCUNCE-PCINT' DOES NOT EXIST.  L AND C ON OPPOSITE SIDES .
00100  C .                    OF THE 'HORIZON.'                                         .
00100  C .                = 3, 'BOUNCE-POINT' DOES NOT EXIST.  THE POINTS O, C AND L ARE .
00100  C .                    CO-LINEAR AND L AND C ARE ON OPPOSITE SIDES OF M.         .
00100  C .                = 4, 'BCUNCE-POINT' DOES NOT EXIST.  THE POINTS O, C AND L ARE .
00100  C .                    CO-LINEAR AND L AND C ARE ON THE SAME SIDES OF M.         .
00100  C .                = 5, 'BOUNCE-POINT' EXISTS BUT SOLUTION WAS NOT DETERMINED    .
00100  C .                                                                              .
00100  C .    SUBROUTINE USAGE-                                                         .
00100  C .       SIN                                                                    .
00100  C .       COS                                                                    .
00100  C .       TAN                                                                    .
00100  C .       ASIN                                                                   .
00100  C .       ACOS                                                                   .
00100  C .       ATAN                                                                   .
00100  C .       SQRT                                                                   .
00100  C .       ABS                                                                    .
00100  C .       AMIN1                                                                  .
00100  C .       AMAX1                                                                  .
00100  C .       GMPRD                                                                  .
00100  C .       GMTRA                                                                  .
00100  C .       VECCRS                                                                 .
00100  C .       VECDOT                                                                 .
00100  C .       VECADD                                                                 .
00100  C .       VECSUB                                                                 .
00100  C .       VECSCA                                                                 .
00100  C .       VECMAG                                                                 .
00100  C .       VECEQU                                                                 .
00100  C .       VECTST                                                                 .
00100  C .                                                                              .
00100  C ........................................................................................
00101         SUBROUTINE REFLKT(ABC,RBCB,ABL,RBLB,ABO,RBOB,RO,ACURCY,RBSB,IFLAG)
00103         DIMENSION ABC(3,3), ABL(3,3), ABO(3,3), ABOT(3,3), AO1(3,3),
00103        $AO1T(3,3), RBCB(3), RBLB(3), RBSB(3), RBOB(3), RCAPC(3), RCAPL(3),
00103        $ROCB(3), ROCO(3), ROLB(3), ROLO(3), U(3), V(3), ZERO(3)
00104         DATA (ZERO(I),I=1,3) /3*0.0/
00104  C
00104  C *** FUNCTION DEFINITIONS
```

```
00104   C          BECAUSE OF THE INABILITY OF UNIVAC 1108 FORTRAN V COMPILERS TO
00104   C     COMPILE A NESTING OF 19 ARITHMETIC STATEMENT FUNCTIONS, THESE FUNCTIONS
00104   C     HAVE BEEN TEMPORARILY REPLACED BY 19 FUNCTION SUBROUTINES IMMEDIATELY
00104   C     FOLLOWING THIS SUBROUTINE.  THE IBM 7094 FORTRAN IV VERSION OF THIS
00104   C     SUBROUTINE USES ARITHMETIC STATEMENT FUNCTIONS.
00106         COMMON /TEMPRY/ H1, DUMMY, X2, Y2
00107         DUMMY = RO
00107   C
00107   C *** COMPUTE O-C AND O-L POSITION VECTORS
00110         CALL VECSUB(RBCB,RBOB,3,ROCB)
00111         CALL VECSUB(RBLB,RBOB,3,ROLB)
00111   C
00111   C *** DETERMINE WHETHER OR NOT A BOUNCE-POINT EXISTS AND SET IFLAG
00112         CALL VECMAG(ROCB,3,A)
00113         CALL VECSCA(ROCB,1.0/A,3,RCAPC)
00114         CALL VECMAG(ROLB,3,B)
00115         CALL VECSCA(ROLB,1.0/B,3,RCAPL)
00116         CALL VECCRS(RCAPC,RCAPL,V)
00117         CALL VECTST(V,ZERO,3,ACURCY,I)
00120         IF(I .EQ. 0) GO TO 10
00122         CALL VECDOT(RCAPC,RCAPL,3,C)
00123         IF(C) 1, 1, 2
00126       1 IFLAG = 3
00127         RETURN
00130       2 IFLAG = 4
00131         RETURN
00131   C ****** COMPUTE LOCATION OF L IN 1-COORDINATES
00132      10 H1 = A - RO
00133         ALPHAO = ACOS(RO / (RO + H1))
00134         CALL GMTRA(ABO,ABOT,3,3)
00135         CALL GMPRD(ABOT,ROCB,ROCO,3,3,1)
00136         CALL GMPRD(ABOT,ROLB,ROLO,3,3,1)
00137         CALL VECSCA(ROCO,1.0/A,3,RCAPC)
00140         CALL VECSCA(ROLO,1.0/B,3,RCAPL)
00141         CALL VECEQU(A01(1,1),RCAPC,3)
00142         CALL VECCRS(RCAPC,RCAPL,V)
00143         CALL VECMAG(V,3,A)
00144         CALL VECSCA(V,1.0/A,3,V)
00145         CALL VECEQU(A01(1,3),V,3)
00146         CALL VECCRS(V,RCAPC,U)
00147         CALL VECEQU(A01(1,2),U,3)
```

```
      CALL GMTRA(A01,A01T,3,3)
      CALL GMPRD(A01T,ROLO,V,3,3,1)
      X2 = V(1) - RO
      Y2 = V(2)
      IF(V(3)/B .GE. ACURCY) GO TO 350
C ****** COMPLETE DETERMINATION OF BOUNCE-POINT EXISTENCE
      IF(Y2 .GT. RO*SIN(ALPHAO)) GO TO 20
      IF(X2 .GT. RO*(COS(ASIN(Y2/RO)) - 1.0)) GO TO 100
      GO TO 30
   20 IF(X2 .GT. (H1 - Y2*TAN(ALPHAO))) GO TO 100
   30 IFLAG = 2
      RETURN
C
C *** A BOUNCE-POINT EXISTS.  CALCULATE REMAINING PARAMETERS.
  100 IFLAG = 1
      H2 = B - RO
      GAMMA = ACOS(RO / (RO + H2))
      CALL VECDOT(ROCB,ROLB,3,A)
      ALPHA = ACOS ( A / ((RO + H1) * (RO + H2)))
      ALPHA1 = ALPHA - GAMMA
      CHI1 = 1.5707963 - ALPHAO
      CHI2 = ATAN(RO * SIN(ALPHA) / ((RO + H1) - RO * COS(ALPHA)))
      CHI3 = ASIN(RO * SIN(ALPHA1) / SQRT(RO**2 + (RO + H1)**2 - 2.0 *
     $RO * (RO + H1) * COS(ALPHA1)))
      IF(X2 .GE. 0.0) GO TO 150
      IF(ALPHA .LT. ALPHAO) GO TO 170
      GO TO 180
  150 IF(ALPHA .GE. ALPHAO) GO TO 160
      CHI1 = AMIN1(CHI1,CHI2)
  160 CHIMIN = 0.0
      CHIMAX = CHI1
      GO TO 190
  170 CHI1 = AMIN1(CHI1,CHI2)
  180 CHIMIN = AMAX1(0.0,CHI3)
      CHIMAX = CHI1
C
C *** DETERMINE BOUNCE-POINT
  190 A = ABS(CHIMAX - CHIMIN)
      IF(A .GE. ACURCY) GO TO 195
      THETAO = 0.5 * A + CHIMIN
      GO TO 1000
```

```
00223        195 THETAO = CROSS(CHIMIN,CHIMAX)
00224        200 A = ETA(THETAO)
00225        300 IF(ABS(A) .LT. ACURCY) GO TO 1000
00227            THETA1 = AMAX1(CHIMIN,AMIN1(CHIMAX,STRAIT(THETAO)))
00230            B = ETA(THETA1)
00231            IF(A*B .GT. 0.0) GO TO 400
00233            THETAO = CROSS(AMIN1(THETAO,THETA1),AMAX1(THETAO,THETA1))
00234            IF((THETAO .GT. CHIMIN) .AND. (THETAO .LT. CHIMAX)) GO TO 200
00236        350 IFLAG = 5
00237            RETURN
00240        400 THETAO = THETA1
00241            A = B
00242            GO TO 300
00242      C
00242      C *** COMPUTE POSITION VECTOR OF BOUNCE-POINT IN BASIC COORDINATES
00243       1000 V(1) = XO(THETAO) + RO
00244            V(2) = YO(THETAO)
00245            V(3) = 0.0
00246            CALL GMPRD(AO1,V,U,3,3,1)
00247            CALL GMPRD(ABO,U,V,3,3,1)
00250            CALL VECADD(V,RBOB,3,RBSB)
00251            RETURN
00252            END

    END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

```
& FOR R1,R1                                                                            00&34&01
UNIVAC 1108 FORTRAN V LEVEL   2206 0016 F5016M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT 00&34&02

  FUNCTION R1             ENTRY POINT 000036

  STORAGE USED (BLOCK, NAME, LENGTH)

        0001    *CODE    000044
        0000    *DATA    000014
        0002    *BLANK   000000
        0003    TEMPRY   000004


  EXTERNAL REFERENCES (BLOCK, NAME)

        0004    COS
        0005    SQRT
        0006    NERR3$


  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

   0003 R 000000 H1         0003 R 000001 RO         0000 R 000000 R1         0003 R 000002 X2         0003


00101      1*          FUNCTION R1    (THETA)
00103      2*          COMMON /TEMPRY/ H1, RO, X2, Y2
00104      3*          R1         = H1 * (2.0 * RO + H1) / ((RO + H1) * COS(THETA) +
00104      4*        $             SQRT(((RO + H1) * COS(THETA))**2 - H1 * (2.0*RO + H1)))
00105      5*          RETURN
00106      6*          END

     END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

```
& FOR DR1,DR1                                                                                  00&34&03
UNIVAC 1108 FORTRAN V LEVEL   2206 0016 F5016M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT 00&34&03

  FUNCTION DR1          ENTRY POINT 000054

  STORAGE USED (BLOCK, NAME, LENGTH)

        0001    *CODE    000062
        0000    *DATA    000015
        0002    *BLANK   000000
        0003    TEMPRY   000004

  EXTERNAL REFERENCES (BLOCK, NAME)

        0004    COS
        0005    SQRT
        0006    SIN
        0007    NERR3$

  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

   0000 R 000000 DR1        0003 R 000000 H1        0003 R 000001 RO        0003 R 000002 X2        0003

00101      1*          FUNCTION DR1    (THETA)
00103      2*          COMMON /TEMPRY/ H1, RO, X2, Y2
00104      3*          DR1         = (RO + H1)**2 * SIN(2.0 * THETA) / (2.0 * SQRT(((RO +
00104      4*        $               H1) * COS(THETA))**2 - H1 * (2.0 * RO + H1)))- (RO +
00104      5*        $               H1) * SIN(THETA)
00105      6*          RETURN
00106      7*          END

     END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

```
& FOR XO,XO                                                          00&34&04
UNIVAC 1108 FORTRAN V LEVEL    2206 0016 F5016M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT 00&34&04

  FUNCTION XO            ENTRY POINT 000022

  STORAGE USED (BLOCK, NAME, LENGTH)

        0001    *CODE    000031
        0000    *DATA    000010
        0002    *BLANK   000000
        0003    TEMPRY   000004


  EXTERNAL REFERENCES (BLOCK, NAME)

        0004    R1
        0005    COS
        0006    NERR3$


  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0003 R 000000 H1         0003 R 000001 RC         0004 R 000000 R1         0000 R 000000 XO         0003
    0003 R 000003 Y2


00101       1*          FUNCTION XO     (THETA)
00103       2*          COMMON /TEMPRY/ H1, RC, X2, Y2
00104       3*          XO          = H1 - R1(THETA) * COS(THETA)
00105       4*          RETURN
00106       5*          END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

```
& FOR DXO,DXO                                                                    00&34&05
UNIVAC 1108 FORTRAN V LEVEL    2206 0016 F5016M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT 00&34&05

  FUNCTION DXO            ENTRY POINT 000033

  STORAGE USED (BLOCK, NAME, LENGTH)

        0001    *CODE    000044
        0000    *DATA    000011
        0002    *BLANK   000000
        0003    TEMPRY   000004


  EXTERNAL REFERENCES (BLOCK, NAME)

        0004    R1
        0005    DR1
        0006    COS
        0007    SIN
        0010    NERR3$


  STORAGE ASSIGNMENT FOR VARIABLES (BLCCK, TYPE, RELATIVE LOCATION, NAME)

   0005 R 000000 DR1         0000 R 000000 DXO         0003 R 000000 H1         0003 R 000001 RO         0004
   0003 R 000002 X2          0003 R 000003 Y2


00101      1*         FUNCTION DXO    (THETA)
00103      2*         COMMON /TEMPRY/ H1, RC, X2, Y2
00104      3*         DXO          = R1(THETA) * SIN(THETA) - DR1(THETA) * COS(THETA)
00105      4*         RETURN
00106      5*         END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.      0 *DIAGNOSTIC* MESSAGE(S)
```

```
& FOR YO,YO                                                                   00&34&06
UNIVAC 1108 FORTRAN V LEVEL   2206 0016 F5016M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT 00&34&06

  FUNCTION YO           ENTRY POINT 000021

  STORAGE USED (BLOCK, NAME, LENGTH)

      0001   *CODE    000030
      0000   *DATA    000010
      0002   *BLANK   000000
      0003   TEMPRY   000004

  EXTERNAL REFERENCES (BLOCK, NAME)

      0004   R1
      0005   SIN
      0006   NERR3$

  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

   0003 R 000000 H1         0003 R 000001 R0         0004 R 000000 R1         0003 R 000002 X2         0000
   0003 R 000003 Y2

00101      1*          FUNCTION YO     (THETA)
00103      2*          COMMON /TEMPRY/ H1, R0, X2, Y2
00104      3*          YO         = R1(THETA) * SIN(THETA)
00105      4*          RETURN
00106      5*          END

     END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

```
& FOR DYO,DYO                                                              00&34&07
UNIVAC 1108 FORTRAN V LEVEL   2206 0016 F5C16M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT 00&34&07

  FUNCTION DYO          ENTRY POINT 000033

  STORAGE USED (BLOCK, NAME, LENGTH)

       0001    *CODE    000044
       0000    *DATA    000011
       0002    *BLANK   000000
       0003    TEMPRY   000004

  EXTERNAL REFERENCES (BLOCK, NAME)

       0004    R1
       0005    DR1
       0006    SIN
       0007    COS
       0010    NERR3$

  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

   0005 R 000000 DR1       0000 R 000000 DYO       0003 R 000000 H1       0003 R 000001 R0       0004
   0003 R 000002 X2        0003 R 000003 Y2

00101      1*          FUNCTION DYO    (THETA)
00103      2*          COMMON /TEMPRY/ H1, R0, X2, Y2
00104      3*          DYO        = R1(THETA) * COS(THETA) + DR1(THETA) * SIN(THETA)
00105      4*          RETURN
00106      5*          END

     END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

```
& FOR R2,R2                                                                 00&34&08
UNIVAC 1108 FORTRAN V LEVEL   2206 0016 F5C16M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT C0&34&08

  FUNCTION R2            ENTRY POINT 000032

  STORAGE USED (BLOCK, NAME, LENGTH)

        0001   *CODE    000041
        0000   *DATA    000010
        0002   *BLANK   000000
        0003   TEMPRY   000004

  EXTERNAL REFERENCES (BLOCK, NAME)

        0004   XO
        0005   YO
        0006   SQRT
        0007   NERR3$

  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

   0003 R 000000 H1        0003 R 000001 RO        0000 R 000000 R2        0004 R 000000 XO        0003
   0005 R 000000 YO        0003 R 000003 Y2

00101      1*          FUNCTION R2    (THETA)
00103      2*          CCMMON /TEMPRY/ H1, RO, X2, Y2
00104      3*          R2         = SQRT((X2 - XO(THETA))**2 + (Y2 - YO(THETA))**2)
00105      4*          RETURN
00106      5*          END

     END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

```
& FOR DR2,DR2                                                                      00&34&09
UNIVAC 1108 FORTRAN V LEVEL   2206 0016 F5C16M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT C0&34&09

  FUNCTION DR2           ENTRY POINT 000C43

  STORAGE USED (BLOCK, NAME, LENGTH)

        0001    *CODE    000055
        0000    *DATA    000011
        0002    *BLANK   000000
        0003    TEMPRY   000004

  EXTERNAL REFERENCES (BLOCK, NAME)

        0004    X0
        0005    DX0
        0006    Y0
        0007    DY0
        0010    R2
        0011    NERR3$

  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

   0000 R 000000 DR2          0005 R 00000C DX0          0007 R 000000 DY0          0003 R 000000 H1          0003
   0010 R 000000 R2           0004 R 000000 X0           0003 R 000002 X2           0006 R 000000 Y0          0003

00101      1*          FUNCTION DR2    (THETA)
00103      2*          CCMMON /TEMPRY/ H1, R0, X2, Y2
00104      3*          DR2          =-((X2 - X0(THETA)) * DX0(THETA) + (Y2 - Y0(THETA)) *
00104      4*         $              DY0(THETA)) / R2(THETA)
00105      5*          RETURN
00106      6*          END
```

```
END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

```
& FOR PHI,PHI                                                                              00&34&10
UNIVAC 1108 FORTRAN V LEVEL   2206 0016 F5016M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT 00&34&10

  FUNCTION PHI           ENTRY POINT 000C26

  STORAGE USED (BLOCK, NAME, LENGTH)

        0001   *CODE    000035
        0000   *DATA    000010
        0002   *BLANK   000000
        0003   TEMPRY   000004


  EXTERNAL REFERENCES (BLOCK, NAME)

        0004   R1
        0005   SIN
        0006   ASIN
        0007   NERR3$


  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

   0003 R 000000 H1          0000 R 000000 PHI          0003 R 000001 RO          0004 R 000000 R1          0003
   0003 R 000003 Y2


00101      1*          FUNCTION PHI   (THETA)
00103      2*          COMMON /TEMPRY/ H1, RO, X2, Y2
00104      3*          PHI        = ASIN(R1(THETA) * SIN(THETA) / RO)
00105      4*          RETURN
00106      5*          END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

```
& FOR DPHI,DPHI                                                                  00&34&11
UNIVAC 1108 FORTRAN V LEVEL   2206 0016 F5016M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT 00&34&11

  FUNCTION DPHI          ENTRY POINT 000055

  STORAGE USED (BLOCK, NAME, LENGTH)

        0001   *CODE    000070
        0000   *DATA    000012
        0002   *BLANK   000000
        0003   TEMPRY   000004


  EXTERNAL REFERENCES (BLOCK, NAME)

        0004   R1
        0005   DR1
        0006   SIN
        0007   SQRT
        0010   COS
        0011   NERR3$


  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

   0000 R 000000 DPHI      0005 R 000000 DR1      0003 R 000000 H1      0003 R 000001 R0      0004
   0003 R 000002 X2        0003 R 000003 Y2


00101     1*          FUNCTION DPHI  (THETA)
00103     2*          COMMON /TEMPRY/ H1, R0, X2, Y2
00104     3*          DPHI        = (R1(THETA) * COS(THETA) + DR1(THETA) * SIN(THETA)) /
00104     4*         $              SQRT((R0 + R1(THETA) * SIN(THETA)) * (R0 - R1(THETA)
00104     5*         $              * SIN(THETA)))
00105     6*          RETURN
```

```
00106      7*          END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

```
& FOR ZETA,ZETA                                                                              00&34&12
UNIVAC 1108 FORTRAN V LEVEL    2206 0016 F5016M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT 00&34&12

  FUNCTION ZETA           ENTRY POINT 000026

  STORAGE USED (BLOCK, NAME, LENGTH)

        0001   *CODE    000035
        0000   *DATA    000010
        0002   *BLANK   000000
        0003   TEMPRY   000004


  EXTERNAL REFERENCES (BLOCK, NAME)

        0004   X0
        0005   R2
        0006   ACOS
        0007   NERR3$


  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

   0003 R 000000 H1         0003 R 000001 R0         0005 R 000000 R2         0004 R 000000 X0         0003
   0003 R 000003 Y2         0000 R 000000 ZETA


00101      1*         FUNCTION ZETA  (THETA)
00103      2*         COMMON /TEMPRY/ H1, R0, X2, Y2
00104      3*         ZETA        = ACOS((X2 - X0(THETA))/R2(THETA))
00105      4*         RETURN
00106      5*         END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

```
& FOR DZETA,DZETA                                                                   00&34&13
UNIVAC 1108 FORTRAN V LEVEL   2206 0016 F5016M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT 00&34&13

  FUNCTION DZETA          ENTRY POINT 000060

  STORAGE USED (BLOCK, NAME, LENGTH)

        0001    *CODE   000073
        0000    *DATA   000012
        0002    *BLANK  000000
        0003    TEMPRY  000004

  EXTERNAL REFERENCES (BLOCK, NAME)

        0004    DXO
        0005    XO
        0006    DR2
        0007    R2
        0010    SQRT
        0011    NERR3$

  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

   0006 R 000000 DR2      0004 R 000000 DXO      0000 R 000000 DZETA      0003 R 000000 H1      0003
   0007 R 000000 R2       0005 R 000000 XO       0003 R 000002 X2         0003 R 000003 Y2

00101      1*          FUNCTION DZETA (THETA)
00103      2*          COMMON /TEMPRY/ H1, RO, X2, Y2
00104      3*          DZETA        =  (DXO(THETA) + (X2 - XO(THETA)) * (DR2(THETA) /
00104      4*     $                   R2(THETA))) / SQRT(R2(THETA)**2 - (X2 - XO(THETA))
00104      5*     $                   **2)
```

```
00105        6*          RETURN
```

```
00106      7*          END
      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

```
& FOR ETA1,ETA1                                                                                00&34&14
UNIVAC 1108 FORTRAN V LEVEL    2206 0016 F5016M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT 00&34&14

  FUNCTION ETA1          ENTRY POINT 000015

  STORAGE USED (BLOCK, NAME, LENGTH)

        0001   *CODE    000024
        0000   *DATA    000007
        0002   *BLANK   000000
        0003   TEMPRY   000004

  EXTERNAL REFERENCES (BLOCK, NAME)

        0004   PHI
        0005   NERR3$

  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000 R 000000 ETA1        0003 R 000000 H1          0004 R 000000 PHI         0003 R 000001 RO          0003
    0003 R 000003 Y2

00101      1*          FUNCTION ETA1  (THETA)
00103      2*          COMMON /TEMPRY/ H1, RC, X2, Y2
00104      3*          ETA1          = THETA + PHI(THETA)
00105      4*          RETURN
00106      5*          END

       END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

```
& FOR DETA1,DETA1                                                                              00&34&15
UNIVAC 1108 FORTRAN V LEVEL    2206 0016 F5016M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT 00&34&15

  FUNCTION DETA1          ENTRY POINT 000015

  STORAGE USED (BLOCK, NAME, LENGTH)

       0001    *CODE    000023
       0000    *DATA    000010
       0002    *BLANK   000000
       0003    TEMPRY   000004

  EXTERNAL REFERENCES (BLOCK, NAME)

       0004    DPHI
       0005    NERR3$

  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

   0000 R 000000 DETA1       0004 R 000000 DPHI       0003 R 000000 H1        0003 R 000001 R0        0003
   0003 R 000003 Y2

00101      1*          FUNCTION DETA1 (THETA)
00103      2*          COMMON /TEMPRY/ H1, R0, X2, Y2
00104      3*          DETA1          = 1.0 + DPHI(THETA)
00105      4*          RETURN
00106      5*          END

     END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

```
& FOR ETA2,ETA2                                                                      00&34&16
UNIVAC 1108 FORTRAN V LEVEL   2206 0016 F5016M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT 00&34&16

   FUNCTION ETA2          ENTRY POINT 000C21

   STORAGE USED (BLOCK, NAME, LENGTH)

         0001    *CODE    000030
         0000    *DATA    00001C
         0002    *BLANK   000000
         0003    TEMPRY   000004


   EXTERNAL REFERENCES (BLOCK, NAME)

         0004    ZETA
         0005    PHI
         0006    NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000 R 000000 ETA2       0003 R 000000 HI       0005 R 000000 PHI       0003 R 000001 RO       0003
    0003 R 000003 Y2         0004 R 000000 ZETA


00101      1*          FUNCTION ETA2  (THETA)
00103      2*          COMMON /TEMPRY/ H1, RO, X2, Y2
00104      3*          ETA2         = ZETA(THETA) - PHI(THETA)
00105      4*          RETURN
00106      5*          END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

```
& FOR DETA2,DETA2                                                                          00&34&17
UNIVAC 1108 FORTRAN V LEVEL   2206 0016 F5016M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT 00&34&17

  FUNCTION DETA2          ENTRY POINT 000021

  STORAGE USED (BLOCK, NAME, LENGTH)

        0001   *CODE    000030
        0000   *DATA    000010
        0002   *BLANK   000000
        0003   TEMPRY   000004

  EXTERNAL REFERENCES (BLOCK, NAME)

        0004   DZETA
        0005   DPHI
        0006   NERR3$

  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000 R 000000 DETA2       0005 R 000000 DPHI        0004 R 000000 DZETA       0003 R 000000 H1          0003
    0003 R 000002 X2          0003 R 000003 Y2

00101      1*          FUNCTION DETA2 (THETA)
00103      2*          COMMON /TEMPRY/ H1, RC, X2, Y2
00104      3*          DETA2         = DZETA(THETA) - DPHI(THETA)
00105      4*          RETURN
00106      5*          END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

```
& FOR ETA,ETA                                                              00&34&18
UNIVAC 1108 FORTRAN V LEVEL   2206 0016 F5016M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT 00&34&18

  FUNCTION ETA          ENTRY POINT 000021

  STORAGE USED (BLOCK, NAME, LENGTH)

        0001    *CODE    000030
        0000    *DATA    000010
        0002    *BLANK   000000
        0003    TEMPRY   000004


  EXTERNAL REFERENCES (BLOCK, NAME)

        0004    ETA2
        0005    ETA1
        0006    NERR3$


  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000 R 000000 ETA        0005 R 000000 ETA1       0004 R 000000 ETA2       0003 R 000000 H1         0003
    0003 R 000002 X2         0003 R 000003 Y2


00101      1*           FUNCTION ETA   (THETA)
00103      2*           COMMON /TEMPRY/ H1, RO, X2, Y2
00104      3*           ETA         = ETA2(THETA) - ETA1(THETA)
00105      4*           RETURN
00106      5*           END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

```
& FOR STRAIT,STRAIT                                                    00&34&19
UNIVAC 1108 FORTRAN V LEVEL   2206 0016 F5C16M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT 00&34&19

  FUNCTION STRAIT       ENTRY POINT 000034

  STORAGE USED (BLOCK, NAME, LENGTH)

       0001    *CODE    000046
       0000    *DATA    000011
       0002    *BLANK   000000
       0003    TEMPRY   000004


  EXTERNAL REFERENCES (BLOCK, NAME)

       0004    ETA2
       0005    ETA1
       0006    DETA1
       0007    DETA2
       0010    NERR3$


  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

   0006 R 000000 DETA1      0007 R 000000 DETA2      0005 R 000000 ETA1       0004 R 000000 ETA2       000
   0003 R 000001 R0         0000 R 000000 STRAIT     0003 R 000002 X2         0003 R 000003 Y2


00101      1*         FUNCTION STRAIT(THETA)
00103      2*         COMMON /TEMPRY/ H1, R0, X2, Y2
00104      3*         STRAIT         = (ETA2(THETA) - ETA1(THETA)) / (DETA1(THETA) -
00104      4*        $                 DETA2(THETA)) + THETA
00105      5*         RETURN
00106      6*         END
```

```
END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

```
& FOR CROSS,CROSS                                                                    00&34&20
UNIVAC 1108 FORTRAN V LEVEL   2206 0016 F5016M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT 00&34&21

  FUNCTION CROSS        ENTRY POINT 000C40

  STORAGE USED (BLOCK, NAME, LENGTH)

        0001   *CODE    000055
        0000   *DATA    000012
        0002   *BLANK   000000
        0003   TEMPRY   000004


  EXTERNAL REFERENCES (BLOCK, NAME)

        0004   ETA1
        0005   ETA2
        0006   NERR3$


  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000 R 000000 CROSS     0004 R 00C000 ETA1      0005 R 000000 ETA2      0003 R 000000 H1       0003
    0003 R 000002 X2        0003 R 000003 Y2


00101      1*          FUNCTION CROSS(THETA0,THETA1)
00103      2*          COMMON /TEMPRY/ H1, RC, X2, Y2
00104      3*          CROSS                  = (THETA1 - THETA0) / (1.0 + (ETA1(THETA1) -
00104      4*         $                       ETA2(THETA1))/(ETA2(THETA0) - ETA1(THETA0)))
00104      5*         $ + THETA0
00105      6*          RETURN
00106      7*          END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

7.3 SUBROUTINE VECTOR

```
& FOR VECTOR,VECTOR                                                      00&34&21
UNIVAC 1108 FORTRAN V LEVEL   2206 0016 F5016M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT 00&34&22

  SUBROUTINE VECTOR     ENTRY POINT 000320

             VECCRS     ENTRY POINT 000323

             VECDOT     ENTRY POINT 000334

             VECADD     ENTRY POINT 000357

             VECSUB     ENTRY POINT 000372

             VECSCA     ENTRY POINT 000405

             VECMAG     ENTRY POINT 000430

             VECEQU     ENTRY POINT 000451


             VECTST     ENTRY POINT 000462

  STORAGE USED (BLOCK, NAME, LENGTH)

        0001   *CODE    000500
        0000   *DATA    000025
        0002   *BLANK   000000


  EXTERNAL REFERENCES (BLOCK, NAME)

        0003   SQRT
        0004   NERR3$


  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

  0001   000046 116G        0001   000100 126G        0001   000131 136G        0001   000155 146G        0001
```

0001 000226 170G 0001 000250 200G 0001 000271 71L 0000 I 000000 I

```
C
C ..................................................................
C .                                                                .
C .   SUBROUTINE VECTOR                         C. ARGILA, APR, '68 .
C .                                                                .
C .   DESCRIPTION-                                                 .
C .     THIS PROGRAM IS A MULTIPLE ENTRY POINT SUBROUTINE WHICH PERFORMS .
C .   N-DIMENSIONAL VECTOR ARITHMETIC OPERATIONS.                  .
C .                                                                .
C .   ARGUMENTS-                                                   .
C .     VECCRS V1   VECTOR                                         .
C .            V2   VECTOR                                         .
C .            V3   CROSS PRODUCT V1 X V2 (V3 DISTINCT FROM V1 AND V2) .
C .     VECDOT V1   VECTOR                                         .
C .            V2   VECTOR                                         .
C .            N    DIMENSION                                      .
C .            A    DOT PRODUCT V1 . V2                            .
C .     VECADD V1   VECTOR                                         .
C .            V2   VECTOR                                         .
C .            N    DIMENSION                                      .
C .            V3   VECTOR SUM V1 + V2                             .
C .     VECSUB V1   VECTOR                                         .
C .            V2   VECTOR                                         .
C .            N    DIMENSION                                      .
C .            V3   VECTOR DIFFERENCE V1 - V2                      .
C .     VECSCA V1   VECTOR                                         .
C .            A    SCALAR                                         .
C .            N    DIMENSION                                      .
C .            V2   VECTOR  =  A V1                                .
C .     VECMAG V1   VECTOR                                         .
C .            N    DIMENSION                                      .
C .            A    MAGNITUDE OF VECTOR V1                         .
C .     VECEQU V1   VECTOR                                         .
C .            V2   VECTOR                                         .
C .            N    DIMENSION                                      .
C .                 PERFORMS THE VECTOR REPLACEMENT V1 = V2.       .
```

```
00100   C .      VECTST V1   VECTOR                                                     .
00100   C .             V2   VECTOR                                                     .
00100   C .             N    DIMENSION                                                  .
00100   C .             ACURCY   ACCURACY TO WHICH TEST IS TO BE APPLIED                .
00100   C .             IFLAG = 0 IFF V1 .NE. V2                                        .
00100   C .                   = 1 IFF V1 .EQ. V2                                        .
00100   C .                                                                             .
00100   C .   SUBROUTINE USAGE-                                                         .
00100   C .      ABS                                                                    .
00100   C .                                                                             .
00100   C .   RESTRICTIONS-                                                             .
00100   C .      THE VECTOR CROSS PRODUCT IS DEFINED ONLY FOR THREE-DIMENSIONAL         .
00100   C .   VECTORS.  VECTORS ARE ASSUMED TO BE RECTANGULAR.                          .
00100   C ...............................................................................
00100   C
00101         SUBROUTINE VECTOR
00103         ENTRY VECCRS(V1,V2,V3)
00105         DIMENSION V1(1), V2(1), V3(1)
00106         V3(1) = V1(2)*V2(3) - V2(2)*V1(3)
00107         V3(2) = V1(3)*V2(1) - V2(3)*V1(1)
00110         V3(3) = V1(1)*V2(2) - V2(1)*V1(2)
00111         RETURN
00112         ENTRY VECDOT(V1,V2,N,A)
00114         A = 0.0
00115         DO 10 I = 1, N
00120      10 A = A + V1(I)*V2(I)
00122         RETURN
00123         ENTRY VECADD(V1,V2,N,V3)
00125         DO 20 I = 1, N
00130      20 V3(I) = V1(I) + V2(I)
00132         RETURN
00133         ENTRY VECSUB(V1,V2,N,V3)
00135         DO 30 I = 1, N
00140      30 V3(I) = V1(I) - V2(I)
00142         RETURN
00143         ENTRY VECSCA(V1,A,N,V2)
00145         DO 40 I = 1, N
00150      40 V2(I) = A*V1(I)
00152         RETURN
00153         ENTRY VECMAG(V1,N,A)
00155         A = 0.0
```

```
00156      78*           DO 50 I = 1, N
00161      79*        50 A = A + V1(I)**2
00163      80*           A = SQRT(A)
00164      81*           RETURN
00165      82*           ENTRY VECEQU(V1,V2,N)
00167      83*           DO 60 I = 1, N
00172      84*        60 V1(I) = V2(I)
00174      85*           RETURN
00175      86*           ENTRY VECTST(V1,V2,N,ACURCY,IFLAG)
00177      87*           DO 70 I = 1, N
00202      88*        70 IF(ABS(V1(I) - V2(I)) .GE. ACURCY) GO TO 71
00205      89*           IFLAG = 1
00206      90*           RETURN
00207      91*        71 IFLAG = 0
00210      92*           RETURN
00211      93*           END

     END OF UNIVAC 1108 FORTRAN V COMPILATION.      0 *DIAGNOSTIC* MESSAGE(S)
```

7.4 SUBROUTINE GMPRD

```
; FOR GMPRD,GMPRD                                                      00&34&23
JNIVAC 1108 FORTRAN V LEVEL   2206 0016 F5016M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT 00&34&23

  SUBROUTINE GMPRD      ENTRY POINT 000113

  STORAGE USED (BLOCK, NAME, LENGTH)

        0001   *CODE    000145
        0000   *DATA    000047
        0002   *BLANK   000000

  EXTERNAL REFERENCES (BLOCK, NAME)

        0003   NERR3$

  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

   0001   000026 107G       0001   000033 113G       0001   000050 122G       0000 I 000006 I       0000
   0000 I 000001 IK         0000 I 000000 IR         0000 I 000003 J         0000 I 000004 JI      0000

00100      1*   C                                                                    GMPRD001
00100      2*   C     ..................................................................GMPRD002
00100      3*   C                                                                    GMPRD003
00100      4*   C        SUBROUTINE GMPRD                                            GMPRD004
00100      5*   C                                                                    GMPRD005
00100      6*   C        PURPOSE                                                     GMPRD006
00100      7*   C           MULTIPLY TWO GENERAL MATRICES TO FORM A RESULTANT GENERAL GMPRD007
00100      8*   C           MATRIX                                                   GMPRD008
00100      9*   C                                                                    GMPRD009
00100     10*   C        USAGE                                                       GMPRD010
00100     11*   C           CALL GMPRD(A,B,R,N,M,L)                                  GMPRD011
00100     12*   C                                                                    GMPRD012
```

```
00100   C        DESCRIPTION OF PARAMETERS                                      GMPRD013
00100   C           A - NAME OF FIRST INPUT MATRIX                              GMPRD014
00100   C           B - NAME OF SECOND INPUT MATRIX                             GMPRD015
00100   C           R - NAME OF OUTPUT MATRIX                                   GMPRD016
00100   C           N - NUMBER OF ROWS IN A                                     GMPRD017
00100   C           M - NUMBER OF COLUMNS IN A AND ROWS IN B                    GMPRD018
00100   C           L - NUMBER OF COLUMNS IN B                                  GMPRD019
00100   C                                                                       GMPRD020
00100   C        REMARKS                                                        GMPRD021
00100   C           ALL MATRICES MUST BE STORED AS GENERAL MATRICES             GMPRD022
00100   C           MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRIX A         GMPRD023
00100   C           MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRIX B         GMPRD024
00100   C           NUMBER OF COLUMNS OF MATRIX A MUST BE EQUAL TO NUMBER OF ROWGMPRD025
00100   C           OF MATRIX B                                                 GMPRD026
00100   C                                                                       GMPRD027
00100   C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                  GMPRD028
00100   C           NONE                                                        GMPRD029
00100   C                                                                       GMPRD030
00100   C        METHOD                                                         GMPRD031
00100   C           THE M BY L MATRIX B IS PREMULTIPLIED BY THE N BY M MATRIX A GMPRD032
00100   C           AND THE RESULT IS STORED IN THE N BY L MATRIX R.            GMPRD033
00100   C                                                                       GMPRD034
00100   C     ..................................................................GMPRD035
00100   C                                                                       GMPRD036
00101         SUBROUTINE GMPRD(A,B,R,N,M,L)                                     GMPRD037
00103         DIMENSION A(1),B(1),R(1)                                          GMPRD038
00103   C                                                                       GMPRD039
00104         IR=0                                                              GMPRD040
00105         IK=-M                                                             GMPRD041
00106         DO 10 K=1,L                                                       GMPRD042
00111         IK=IK+M                                                           GMPRD043
00112         DO 10 J=1,N                                                       GMPRD044
00115         IR=IR+1                                                           GMPRD045
00116         JI=J-N                                                            GMPRD046
00117         IB=IK                                                             GMPRD047
00120         R(IR)=0                                                           GMPRD048
00121         DC 10 I=1,M                                                       GMPRD049
00124         JI=JI+N                                                           GMPRD050
00125         IB=IB+1                                                           GMPRD051
00126      10 R(IR)=R(IR)+A(JI)*B(IB)                                           GMPRD052
00132         RETURN                                                            GMPRD053
```

```
00133     54*          END                                                  GMPRD054

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

7.5 SUBROUTINE GMTRA

```
& FOR GMTRA,GMTRA                                                                   00&34&24
UNIVAC 1108 FORTRAN V LEVEL   2206 0016 F5016M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT 00&34&25

  SUBROUTINE GMTRA      ENTRY POINT 000060

  STORAGE USED (BLOCK, NAME, LENGTH)

        0001    *CODE     000102
        0000    *DATA     000032
        0002    *BLANK    000000

  EXTERNAL REFERENCES (BLOCK, NAME)

        0003    NERR3$

  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

   0001   000017 106G       0001   000024 112G       0000 I 000001 I          0000 I 000002 IJ        0000
   0000 I 000003 J

00100      1*   C                                                                       GMTRA001
00100      2*   C      .................................................................GMTRA002
00100      3*   C                                                                       GMTRA003
00100      4*   C         SUBROUTINE GMTRA                                              GMTRA004
00100      5*   C                                                                       GMTRA005
00100      6*   C         PURPOSE                                                       GMTRA006
00100      7*   C            TRANSPOSE A GENERAL MATRIX                                 GMTRA007
00100      8*   C                                                                       GMTRA008
00100      9*   C         USAGE                                                         GMTRA009
00100     10*   C            CALL GMTRA(A,R,N,M)                                        GMTRA010
00100     11*   C                                                                       GMTRA011
00100     12*   C         DESCRIPTION OF PARAMETERS                                     GMTRA012
```

```
00100     13*    C          A - NAME OF MATRIX TO BE TRANSPOSED                                  GMTRA013
00100     14*    C          R - NAME OF RESULTANT MATRIX                                         GMTRA014
00100     15*    C          N - NUMBER OF ROWS OF A AND COLUMNS OF R                             GMTRA015
00100     16*    C          M - NUMBER OF COLUMNS OF A AND ROWS OF R                             GMTRA016
00100     17*    C                                                                               GMTRA017
00100     18*    C       REMARKS                                                                 GMTRA018
00100     19*    C          MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRIX A                  GMTRA019
00100     20*    C          MATRICES A AND R MUST BE STORED AS GENERAL MATRICES                  GMTRA020
00100     21*    C                                                                               GMTRA021
00100     22*    C       SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                           GMTRA022
00100     23*    C          NONE                                                                 GMTRA023
00100     24*    C                                                                               GMTRA024
00100     25*    C       METHOD                                                                  GMTRA025
00100     26*    C          TRANSPOSE N BY M MATRIX A TO FORM M BY N MATRIX R                    GMTRA026
00100     27*    C                                                                               GMTRA027
00100     28*    C    ..................................................................GMTRA028
00100     29*    C                                                                               GMTRA029
00101     30*           SUBROUTINE GMTRA(A,R,N,M)                                                GMTRA030
00103     31*           DIMENSION A(1),R(1)                                                      GMTRA031
00103     32*    C                                                                               GMTRA032
00104     33*           IR=0                                                                     GMTRA033
00105     34*           DO 10 I=1,N                                                              GMTRA034
00110     35*           IJ=I-N                                                                   GMTRA035
00111     36*           DO 10 J=1,M                                                              GMTRA036
00114     37*           IJ=IJ+N                                                                  GMTRA037
00115     38*           IR=IR+1                                                                  GMTRA038
00116     39*        10 R(IR)=A(IJ)                                                              GMTRA039
00121     40*           RETURN                                                                   GMTRA040
00122     41*           END                                                                      GMTRA041

     END OF UNIVAC 1108 FORTRAN V COMPILATION.      0 *DIAGNOSTIC* MESSAGE(S)
```

```
. FOR LOCK,LOOK                                                                   00&34&26
UNIVAC 1108 FORTRAN V LEVEL   2206 0016 F5016M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT 00&34&26

 SUBROUTINE LOOK      ENTRY POINT 000105

            LOOKCM    ENTRY POINT 000110

            LOOKLM    ENTRY POINT 000131

 STORAGE USED (BLOCK, NAME, LENGTH)

      0001   *CODE    000152
      0000   *DATA    000024
      0002   *BLANK   000000
      0003   DATA     000031

 EXTERNAL REFERENCES (BLOCK, NAME)

      0004   VECSUB
      0005   GMPRD
      0006   SQRT
      0007   ATAN2
      0010   NERR3$

 STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

  0001     000007 1L       0003 R 000000 ACURCY    0003 R 000001 EPSLNR    0003 R 000002 EPSLNO    0003
  0003 R 000004 PI         0003 R 000005 RO        0003 R 000006 SIGMA     0000 R 000000 T         0003
  0003 R 000013 TWOPI      0003 R 000014 TWOPIC    0000 R 000003 U         0003 R 000015 UNITY     0000
  0000 R 000007 Y          0003 R 000026 ZERO

00100      1*    C ..........................................................................
```

```
C .    SUBROUTINE LOOK                                         C. ARGILA, JUNE '68     .
C .                                                                                    .
C .    DESCRIPTION-                                                                    .
C .       COMPUTES THE LOOK ANGLES FOR THE CSM ANTENNA (ENTRY POINT LOOKCM) AND        .
C .    THE LEM ANTENNA (ENTRY POINT LOOKLM) FOR THE VHFASP PROGRAM.                    .
C .                                                                                    .
C .    ARGUMENTS-                                                                      .
C .       ENTRY POINT LOOKCM-                                                          .
C .          ROCB    POSITION VECTOR OF CSM                                            .
C .          RO2B    POSITION VECTOR OF REMOTE POINT                                   .
C .          ACB     CSM MATRIX                                                        .
C .          THETA   LOOK ANGLE                                                        .
C .          PHI     LOOK ANGLE                                                        .
C .       ENTRY POINT LOOKLM-                                                          .
C .          ROLB    POSITION VECTOR OF LM                                             .
C .          RO1B    POSITION VECTOR OF REMOTE POINT                                   .
C .          ALB     LM MATRIX                                                         .
C .          THETA   LOOK ANGLE                                                        .
C .          PHI     LOOK ANGLE                                                        .
C ......................................................................................
C
       SUBROUTINE LOOK
       DIMENSION R(3), S(3), T(3), U(3), A(3,3)
       DIMENSION TIME(4), UNITY(3,3), ZERO(3)
       COMMON /DATA/ ACURCY, EPSLNR, EPSLNO, HALFPI, PI, RO, SIGMA, TIME,
      $TWOPI, TWOPIC, UNITY, ZERO
       ENTRY LOOKCM(R,S,A,THETA,PHI)
       X = 1.0
       Y = -1.0
       GO TO 1
       ENTRY LOOKLM(R,S,A,THETA,PHI)
       X = -1.0
       Y = 1.0
     1 CALL VECSUB(S,R,3,T)
       CALL GMPRD(A,T,U,3,3,1)
       THETA = ATAN2(SQRT(U(2)**2 + U(3)**2),U(1))
       IF(THETA .LT. 0.0) THETA = THETA + TWOPI
       PHI = ATAN2(X*U(2),Y*U(3))
       IF(PHI .LT. 0.0) PHI = PHI + TWOPI
       RETURN
       END
```

```
END OF UNIVAC 1108 FORTRAN V COMPILATION.      0 *DIAGNOSTIC* MESSAGE(S)
```

7.7 SUBROUTINE POLANG

```
& FOR POLANG,POLANG                                                    00&34&27
UNIVAC 1108 FORTRAN V LEVEL   2206 0016 F5C16M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT C0&34&27

  SUBROUTINE POLANG    ENTRY POINT 000161

  STORAGE USED (BLOCK, NAME, LENGTH)

      0001   *CODE    000206
      0000   *DATA    C00075
      0002   *BLANK   000000


  EXTERNAL REFERENCES (BLOCK, NAME)

      0003   GMTRA
      0004   GMPRD
      0005   SIN
      0006   COS
      0007   ATAN2
      0010   NERR3$


  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

   0000 R 000033 AAA      0000 R 000C00 AAL      0000 R 000022 ABC      0000 R 000011 ACA      0000
   0000 R 000053 CP2      0000 R 000045 CT1      0000 R 000051 CT2      0000 R 000046 SP1      0000
   0000 R 000044 ST1      0000 R 000050 ST2


00100     C ..........................................................................
00100     C .   SUBROUTINE POLANG                           C. ARGILA, JUNE '68        .
00100     C .                                                                          .
00100     C .   DESCRIPTION-                                                           .
00100     C .      COMPUTES THE POLARIZATICN ANGLE ALONG THE DIRECT PATH FOR THE VHFASP .
00100     C .   PROGRAM.                                                               .
```

```
C .                                                                    .
C .   ARGUMENTS-                                                       .
C .      THETAC  LOOK ANGLE CSM                                        .
C .      PHIC    LOOK ANGLE CSM                                        .
C .      THETAL  LOOK ANGLE LEM                                        .
C .      PHIL    LOOK ANGLE LEM                                        .
C .      ACB     CSM MATRIX                                            .
C .      ALB     LEM MATRIX                                            .
C .      RHO     POLARIZATION ANGLE                                    .
C .                                                                    .
C ......................................................................
C
      SUBROUTINE POLANG(THETAC,PHIC,THETAL,PHIL,ACB,ALB,RHO)
      DIMENSION ACB(3,3), ALB(3,3), AAL(3,3), ACA(3,3), ABC(3,3), AAA(3,
     $3)
      ST1 = SIN(THETAC)
      CT1 = COS(THETAC)
      SP1 = SIN(PHIC)
      CP1 = COS(PHIC)
      ST2 = SIN(THETAL)
      CT2 = COS(THETAL)
      SP2 = SIN(PHIL)
      CP2 = COS(PHIL)
      ACA(1,1) = ST1
      ACA(2,1) = -CT1 * SP1
      ACA(3,1) = CT1 * CP1
      ACA(1,2) = 0.0
      ACA(2,2) = CP1
      ACA(3,2) = SP1
      ACA(1,3) = -CT1
      ACA(2,3) = -ST1 * SP1
      ACA(3,3) = ST1 * CP1
      AAL(1,1) = -ST2
      AAL(2,1) = 0.0
      AAL(3,1) = CT2
      AAL(1,2) = -CT2 * SP2
      AAL(2,2) = -CP2
      AAL(3,2) = SP2 * ST2
      AAL(1,3) = CT2 * CP2
      AAL(2,3) = -SP2
      AAL(3,3) = ST2 * CP2
```

```
00136          CALL GMTRA(ACB,ABC,3,3)
00137          CALL GMPRD(AAL,ALB,AAA,3,3,3)
00140          CALL GMPRD(AAA,ABC,AAL,3,3,3)
00141          CALL GMPRD(AAL,ACA,AAA,3,3,3)
00142          RHO = ATAN2(AAA(2,1),AAA(1,1))
00143          RETURN
00144          END

     END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

7.8 SUBROUTINE TILT

```
& FOR TILT,TILT                                                              07&47&15
UNIVAC 1108 FORTRAN V LEVEL   2206 0016 F5016M
THIS COMPILATION WAS DONE ON 30 JUN 68 AT 07&47&15

  SUBROUTINE TILT       ENTRY POINT 000065

  STORAGE USED (BLOCK, NAME, LENGTH)

        0001   *CODE    000104
        0000   *DATA    000020
        0002   *BLANK   000000

  EXTERNAL REFERENCES (BLOCK, NAME)

        0003   NEXP6$
        0004   ATAN2
        0005   NERR3$

  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

   0001   000055 2L        0000 R 000000 A

00100      1*    C .....................................................................
00100      2*    C .   SUBROUTINE TILT                              C. ARGILA, JUNE '68   .
00100      3*    C .                                                                      .
00100      4*    C .   DESCRIPTION-                                                       .
00100      5*    C .      COMPUTES POLARIZATICN ELLIPSE TILT ANGLE FOR THE VHFASP PROGRAM.  .
00100      6*    C .                                                                      .
00100      7*    C .   ARGUMENTS-                                                         .
00100      8*    C .      P1     THETA PATTERN DATA (DB)                                  .
00100      9*    C .      P2     PHI PATTERN CATA (DB)                                    .
00100     10*    C .      P3     THETA + 45 PATTERN DATA (DB)                             .
00100     11*    C .      P4     PHI + 45 PATTERN CATA (DB)                               .
```

```
00100      12*    C .       TAU    TILT ANGLE                                                                   .
00100      13*    C .                                                                                           .
00100      14*    C .............................................................................................
00100      15*    C
00101      16*          SUBROUTINE TILT(P1,P2,P3,P4,TAU)
00103      17*          TEN(X) = 10.0**(X / 10.0)
00104      18*          A = TEN(P3) - TEN(P4)
00105      19*          IF(A) 1, 2, 1
00110      20*        1 TAU = 0.5 * ATAN2(A,TEN(P1) - TEN(P2))
00111      21*          RETURN
00112      22*        2 TAU = 0.0
00113      23*          RETURN
00114      24*          END

     END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

7.9 SUBROUTINE MISALN

```
& FOR MISALN,MISALN                                                          00&34&30
UNIVAC 1108 FORTRAN V LEVEL    2206 0016 F5016M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT 00&34&30

   SUBROUTINE MISALN      ENTRY POINT 000022

   STORAGE USED (BLOCK, NAME, LENGTH)

         0001    *CODE    000040
         0000    *DATA    000012
         0002    *BLANK   000000
         0003    DATA     000031


   EXTERNAL REFERENCES (BLOCK, NAME)

         0004    NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0003 R 000000 ACURCY     0003 R 000001 EPSLNR     0003 R 000002 EPSLNO     0003 R 000003 HALFPI     0003
    0003 R 000005 RO         0003 R 000006 SIGMA      0003 R 000007 TIME       0003 R 000013 TWOPI      0003
    0003 R 000015 UNITY      0003 R 000026 ZERO


00100   C ...........................................................................
00100   C .   SUBROUTINE MISALN                              C. ARGILA, JUNE '68      .
00100   C .                                                                           .
00100   C .   DESCRIPTION-                                                            .
00100   C .      COMPUTES POLARIZATION MIS-ALIGNMENT ANGLE FOR THE VHFASP PROGRAM.     .
00100   C .                                                                           .
00100   C .   ARGUMENTS-                                                              .
00100   C .      TAU1  TILT ANGLE OF FIRST POLARIZATION ELLIPSE                        .
00100   C .      TAU2  TILT ANGLE OF SECOND POLARIZATION ELLIPSE                       .
00100   C .      RHO   COORDINATE SYSTEM MIS-ALIGNMENT ANGLE                           .
```

```
00100   C .       BETA  MIS-ALIGNMENT ANGLE BETWEEN POLARIZATION ELLIPSES                 .
00100   C .                                                                               .
00100   C ................................................................................
00100   C
00101         SUBROUTINE MISALN(TAU1,TAU2,RHO,BETA)
00103         DIMENSION TIME(4), UNITY(3,3), ZERO(3)
00104         COMMON /DATA/ ACURCY, EPSLNR, EPSLNO, HALFPI, PI, RO, SIGMA, TIME,
00104        $TWOPI, TWOPIC, UNITY, ZERO
00105         BETA = TAU2 - TAU1 + RHO
00106         IF(BETA .GT. HALFPI) BETA = BETA - PI
00110         RETURN
00111         END

   END OF UNIVAC 1108 FORTRAN V COMPILATION.      0 *DIAGNOSTIC* MESSAGE(S)
```

7.10 SUBROUTINE GAINRR

```
& FOR GAINRR,GAINRR                                                    00&34&31
UNIVAC 1108 FORTRAN V LEVEL   2206 0016 F5016M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT 00&34&31

  SUBROUTINE GAINRR     ENTRY POINT 000042

  STORAGE USED (BLOCK, NAME, LENGTH)

        0001    *CODE    000056
        0000    *DATA    000015
        0002    *BLANK   000000

  EXTERNAL REFERENCES (BLOCK, NAME)

        0003    NEXP6$
        0004    NERR3$

  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

   0000 R 000000 EL          0000 R 000001 ER

00100   C ...........................................................................
00100   C .   SUBROUTINE GAINRR                                C. ARGILA, JUNE '68     .
00100   C .                                                                            .
00100   C .   DESCRIPTION-                                                             .
00100   C .      COMPUTES THE ANTENNA GAIN AND AXIAL RATIO OF AN ANTENNA FOR THE       .
00100   C .   VHFASP PROGRAM.                                                          .
00100   C .                                                                            .
00100   C .   ARGUMENTS-                                                               .
00100   C .      PLCPDB  LEFT CIRCULARLY POLARIZED PATTERN DATA (DB)                   .
00100   C .      PRCPDB  RIGHT CIRCULARLY POLARIZED PATTERN DATA (DB)                  .
00100   C .      GAIN    ANTENNA GAIN                                                  .
00100   C .      RATIO   AXIAL RATIO                                                   .
```

```
00100      13*    C .                                                                  .
00100      14*    C ....................................................................
00100      15*    C
00101      16*          SUBROUTINE GAINRR(PLCPDB,PRCPDB,GAIN,RATIO)
00103      17*          EL = 10.0**(PLCPDB / 20.0)
00104      18*          ER = 10.0**(PRCPDB / 20.0)
00105      19*          RATIO =(ER + EL) / (ER - EL)
00106      20*          GAIN = EL**2 + ER**2
00107      21*          RETURN
00110      22*          END

     END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

7.11 FUNCTION SUBROUTINE DB

```
& FOR DB,DB                                                          00&34&32
UNIVAC 1108 FORTRAN V LEVEL    2206 0016 F5016M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT 00&34&32

  FUNCTION DB            ENTRY POINT 000036

  STORAGE USED (BLOCK, NAME, LENGTH)

        0001    *CODE    000045
        0000    *DATA    000012
        0002    *BLANK   000000

  EXTERNAL REFERENCES (BLOCK, NAME)

        0003    ALOG10
        0004    NERR3$

  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

   0001   000015 2L        0001   000021 3L       0003 R 000000 ALOG10     0000 R 000000 DB

00100       1*    C ...........................................................................
00100       2*    C .   FUNCTION SUBROUTINE DB                          C. ARGILA, MAY '68     .
00100       3*    C .                                                                          .
00100       4*    C .   DESCRIPTION-                                                           .
00100       5*    C .      CHANGES POWER RATIO TO DECIBELS.                                    .
00100       6*    C .                                                                          .
00100       7*    C .   ARGUMENT-                                                              .
00100       8*    C .      RATIO   POWER RATIO                                                 .
00100       9*    C .                                                                          .
00100      10*    C ...........................................................................
00100      11*    C
00101      12*          FUNCTION DB(RATIO)
```

```
00103     13*          IF(RATIO) 1, 2, 3
00106     14*        1 DB = 10.0 * ALOG10(-RATIO)
00107     15*          RETURN
00110     16*        2 DB = 0.0
00111     17*          RETURN
00112     18*        3 DB = 10.0 * ALOG10(RATIO)
00113     19*          RETURN
00114     20*          END

    END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

7.12 SUBROUTINE POLAR

```
& FOR POLAR,POLAR                                                    00&34&33
UNIVAC 1108 FORTRAN V LEVEL   2206 0016 F5016M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT 00&34&33

  SUBROUTINE POLAR      ENTRY POINT 000C46

  STORAGE USED (BLOCK, NAME, LENGTH)

        0001    *CODE    000062
        0000    *DATA    000022
        0002    *BLANK   000000

  EXTERNAL REFERENCES (BLOCK, NAME)

        0003    COS
        0004    NERR3$

  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

   0000 R 000000 A            0000 R 000001 B

00100   C ......................................................................
00100   C .   SUBROUTINE POLAR                               C. ARGILA, MAY '68   .
00100   C .                                                                     .
00100   C .   DESCRIPTION-                                                      .
00100   C .      COMPUTES THE POLARIZATION LOSSES FOR THE VHFASP PROGRAM.       .
00100   C .                                                                     .
00100   C .   ARGUMENTS-                                                        .
00100   C .      R1     FIRST AXIAL RATIO                                       .
00100   C .      R2     SECOND AXIAL RATIO                                      .
00100   C .      BETA   ANGLE BETWEEN AXES                                      .
00100   C .      P      POLARIZATION LOSS                                       .
00100   C .                                                                     .
```

```
00100      13*   C ....................................................................................
00100      14*   C
00101      15*         SUBROUTINE POLAR(R1,R2,BETA,P)
00103      16*         A = R1**2
00104      17*         B = R2**2
00105      18*         P = 0.5 * (1.0 + (4.0 * R1 * R2 + (1.0 - A) * (1.0 - B) * COS(2.0
00105      19*        $    * BETA)) / ((1.C + A) * (1.0 + B)))
00106      20*         RETURN
00107      21*         END

    END OF UNIVAC 1108 FORTRAN V COMPILATION.      0 *DIAGNOSTIC* MESSAGE(S)
```

7.13 SUBROUTINE POLARS

```
& FOR POLARS,POLARS                                                          12815638
UNIVAC 1108 FORTRAN V LEVEL   2206 0016 E5016M
THIS COMPILATION WAS DONE ON 02 JUL 68 AT 12815638

  SUBROUTINE POLARS     ENTRY POINT 000434

  STORAGE USED (BLOCK, NAME, LENGTH)

        0001    *CODE    000476
        0000    *DATA    000073
        0002    *BLANK   000000
        0003    DATA     000031

  EXTERNAL REFERENCES (BLOCK, NAME)

        0004    VECDOT
        0005    VECMAG
        0006    VECSUB
        0007    VECSCA
        0010    VECEQU
        0011    VECCRS
        0012    GMPRD
        0013    ACOS
        0014    SIN
        0015    ASIN
        0016    COS
        0017    ATAN2
        0020    NERR3$

  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0000 R 000011 A          0000 R 000022 AAC      0000 R 000022 AAL     0000 R 000044 AAS      000
 0003 R 000000 ACURCY     0000 R 000022 A01      0000 R 000033 A12     0000 R 000033 A2S      000
 0000 R 000013 C          0000 R 000021 CP       0000 R 000017 CT      0003 R 000001 EPSLNR   000
 0003 R 000003 HALFPI     0000 R 000015 PHIO     0003 R 000004 PI      0000 R 000044 RCAPC    000
 0003 R 000005 RO         0003 R 000006 SIGMA    0000 R 000020 SP      0000 R 000016 ST       000
```

```
0003 R 000007 TIME        0003 R 000013 TWOPI       0003 R 000014 TWOPIC      0000 R 000052 U           0001
0000 R 000033 V           0003 R 000026 ZERO
```

```
00100   C ..............................................................................
00100   C .   SUBROUTINE POLARS                                 C. ARGILA, JUNE '68    .
00100   C .                                                                            .
00100   C .   DESCRIPTION-                                                             .
00100   C .      COMPUTES THE MULTIPATH POLARIZATICN ANGLES, RHO1S AND RHO2S, FOR THE  .
00100   C .   VHFASP PROGRAM.                                                          .
00100   C .                                                                            .
00100   C .   ARGUMENTS-                                                               .
00100   C .      ROCB    POSITION VECTOR OF CSM                                        .
00100   C .      ROLB    POSITION VECTOR OF LEM                                        .
00100   C .      ROSB    POSITION VECTOR OF BOUNCE POINT                               .
00100   C .      ACB     CSM MATRIX                                                    .
00100   C .      ALB     LEM MATRIX                                                    .
00100   C .      PSI     GRAZING ANGLE                                                 .
00100   C .      THTACS  CSM LOOK ANGLE                                                .
00100   C .      PHICS   CSM LOOK ANGLE                                                .
00100   C .      THTALS  LEM LOOK ANGLE                                                .
00100   C .      PHILS   LEM LOOK ANGLE                                                .
00100   C .      RHO1S   POLARIZATION ANGLE                                            .
00100   C .      RHO2S   POLARIZATION ANGLE                                            .
00100   C .                                                                            .
00100   C ..............................................................................
00100   C
00101         SUBROUTINE POLARS(ROCB,ROLB,ROSB,ACB,ALB,PSI,THTACS,PHICS,THTALS,
00101        $PHILS,RHO1S,RHO2S)
00103         DIMENSION ROCB(3), ROLB(3), ROSB(3), RCAPC(3), RCAPL(3), U(3), V(3
00103        $), ACB(3,3), ALB(3,3), A01(3,3), A12(3,3), AB2(3,3), A2S(3,3), AAL
00103        $(3,3), AAC(3,3), AAS(3,3)
00104         EQUIVALENCE (AAL,AAC), (AAL,A01), (A2S,A12), (AAS(1,1),RCAPC(1)),
00104        $(AAS(1,2),RCAPL(1)), (AAS(1,3),U(1)), (A2S(1,1),V(1))
00105         DIMENSION TIME(4), UNITY(3,3), ZERO(3)
00106         COMMON /DATA/ ACURCY, EPSLNR, EPSLNC, HALFPI, PI, RO, SIGMA, TIME,
00106        $TWOPI, TWOPIC, UNITY, ZERO
00107         CALL VECDOT(ROSB,ROCB,3,A)
00110         CALL VECMAG(ROSB,3,B)
```

```
      CALL VECMAG(ROCB,3,C)
      THETAO = HALFPI - PSI - ACOS(A/(B*C))
      CALL VECSUB(ROCB,ROSB,3,U)
      CALL VECMAG(U,3,A)
      PHIO = ASIN(A*SIN(THETAO)/B)
      A12(1,1) = 0.0
      A12(2,1) = 0.0
      A12(3,1) = 1.0
      A12(1,2) = COS(PHIO)
      A12(2,2) = SIN(PHIO)
      A12(3,2) = 0.0
      A12(1,3) = -A12(2,2)
      A12(2,3) = A12(1,2)
      A12(3,3) = 0.0
      CALL VECMAG(ROLB,3,A)
      CALL VECSCA(ROCB,1.0/C,3,RCAPC)
      CALL VECSCA(ROLB,1.0/A,3,RCAPL)
      CALL VECEQU(A01(1,1),RCAPC,3)
      CALL VECCRS(RCAPC,RCAPL,V)
      CALL VECMAG(V,3,A)
      CALL VECSCA(V,1.0/A,3,V)
      CALL VECEQU(A01(1,3),V,3)
      CALL VECCRS(V,RCAPC,U)
      CALL VECEQU(A01(1,2),U,3)
      CALL GMPRD(A01,A12,AB2,3,3,3)
      ST = SIN(THTALS)
      CT = COS(THTALS)
      SP = SIN(PHILS)
      CP = COS(PHILS)
      AAL(1,1) = -ST
      AAL(2,1) = 0.0
      AAL(3,1) = CT
      AAL(1,2) = -CT*SP
      AAL(2,2) = -CP
      AAL(3,2) = -ST*SP
      AAL(1,3) = CT*CP
      AAL(2,3) = -SP
      AAL(3,3) = ST*CP
      A2S(1,1) = -1.0
      A2S(2,1) = 0.0
      A2S(3,1) = 0.0
```

```
      CALL GMPRD(AAL,ALB,AAS,3,3,3)
      CALL GMPRD(AAS,AB2,AAL,3,3,3)
      CALL GMPRD(AAL,A2S,AAS,3,3,3)
      RHO2S = ATAN2(AAS(2,1),AAS(1,1))
      ST = SIN(THTACS)
      CT = COS(THTACS)
      SP = SIN(PHICS)
      CP = COS(PHICS)
      AAC(1,1) = ST
      AAC(2,1) = 0.0
      AAC(3,1) = -CT
      AAC(1,2) = -CT*SP
      AAC(2,2) = CP
      AAC(3,2) = -ST*SP
      AAC(1,3) = CT*CP
      AAC(2,3) = ST
      AAC(3,3) = ST*CP
      A2S(1,1) = -1.0
      A2S(2,1) = 0.0
      A2S(3,1) = 0.0
      CALL GMPRD(AAC,ACB,AAS,3,3,3)
      CALL GMPRD(AAS,AB2,AAC,3,3,3)
      CALL GMPRD(AAC,A2S,AAS,3,3,3)
      RHO1S = ATAN2(AAS(2,1),AAS(1,1))
      RETURN
      END

END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

7.14 BLOCK DATA SUBROUTINE DATA

```
& FOR DATA,DATA                                                    00834837
UNIVAC 1108 FORTRAN V LEVEL   2206 0016 F5016M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT 00834837

  BLOCK DATA

  STORAGE USED (BLOCK, NAME, LENGTH)

        0003    DATA      000031
        0004    GNREF     000044

  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

   0003 R 000000 ACURCY     0003 R 000001 EPSLNR     0003 R 000002 EPSLNO     0004 R 000000 GNREFC    000
   0003 R 000003 HALFPI     0000 I 000000 I$         0003 R 000004 PI         0003 R 000005 RO        000
   0003 R 000007 TIME       0003 R 000013 TWOPI      0003 R 000014 TWOPIC     0003 R 000015 UNITY     000

00100      1*    C .......................................................................
00100      2*    C .   SUBROUTINE DATA                                C. ARGILA, JUNE '68    .
00100      3*    C .                                                                       .
00100      4*    C .   DESCRIPTION-                                                        .
00100      5*    C .      DEFINES CONSTANTS USED IN THE VHFASP PROGRAM.                    .
00100      6*    C .                                                                       .
00100      7*    C .......................................................................
00100      8*    C
00101      9*          BLOCK DATA
00102     10*          DIMENSION TIME(4), UNITY(3,3), ZERO(3)
00103     11*          DIMENSION GNREFC(6,2), GNREFL(6,4)
00104     12*          COMMON /DATA/ ACURCY, EPSLNR, EPSLNO, HALFPI, PI, RO, SIGMA, TIME,
00104     13*         $TWOPI, TWOPIC, UNITY, ZERO
00105     14*          COMMON /GNREF/ GNREFC, GNREFL
00106     15*          DATA ACURCY /0.01/
00110     16*          DATA EPSLNR /2.8000000/
00112     17*          DATA EPSLNO /8.8419411E-12/
00114     18*          DATA HALFPI /1.5707963/
```

```
00116          DATA PI      /3.1415927/
00120          DATA RO      /938.44492/
00122          DATA SIGMA   /0.0040000/
00124          DATA TIME    /86400.0, 3600.0, 60.0, 1.0/
00126          DATA TWOPI   /6.2831853/
00130          DATA TWOPIC  /33.817181/
00132          DATA UNITY   /1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0/
00134          DATA ZERO    /0.0, 0.0, 0.0/
00136          DATA GNREFC, GNREFL /5.52, 5.48, 3.76, 5.45, 5.41, 3.74, 6.19,
00136         $   6.29, 3.27, 5.10, 5.12, 3.25, 6.79, 3.56, 5.68, 7.64, 1.58,
00136         $   4.77, 4.25, 4.29, 5.60, 3.44, 3.08, 5.30, 2.47, 2.30, 5.96,
00136         $   6.04, 5.10, 6.28, 3.51, 1.69, 4.91, 3.06, 3.24, 4.19/
00141          END
```

END OF UNIVAC 1108 FORTRAN V COMPILATION.   0 *DIAGNOSTIC* MESSAGE(S)

7.15 UTILITY PROGRAM UTILT1

```
& FOR UTILT1,UTILT1                                                          00&34&37
UNIVAC 1108 FORTRAN V LEVEL   2206 0016 F5016M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT 00&34&38

  MAIN PROGRAM

  STORAGE USED (BLOCK, NAME, LENGTH)

       0001   *CODE    000100
       0000   *DATA    000124
       0002   *BLANK   000000


  EXTERNAL REFERENCES (BLOCK, NAME)

       0003   NREW$
       0004   NRNL$
       0005   NWDU$
       0006   NIO1$
       0007   NIO2$
       0010   NRBU$
       0011   NWNL$
       0012   NSTOP$


  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

   0000   000050 1F        0001   000052 136G      0001   000045 2L        0000 R 000074 ACB      000
   0000 R 000065 ARRAY     0000 I 000001 FILM      0000 I 000004 I$        0000 I 000003 NRECDS   000
   0000 R 000066 ROCB      0000 R 000071 RCLB      0000 R 000065 T         0000 I 000000 TAPE


00100     1*   C .............................................................................
00100     2*   C .                                                                           .
00100     3*   C .   UTILT1                                        C. ARGILA, JUNE '68           .
00100     4*   C .                                                                           .
```

```
00100    C .   DESCRIPTION-                                                             .
00100    C .      UTILT1 IS A UTILITY PROGRAM FOR USE WITH THE VHFASP PROGRAM.  UTILT1  .
00100    C .   DUMPS THE VHFASP TRAJECTORY TAPE ONTO MICROFILM OR PRINT.  UTILT1 ALSO   .
00100    C .   COUNTS THE NUMBER OF RECORDS ENCOUNTERED.                                .
00100    C .                                                                            .
00100    C .   INPUT-                                                                   .
00100    C .      INPUT IS VIA THE FORTRAN NAMELIST '$INPUT.'  INPUT PARAMETERS ARE     .
00100    C .       TAPE    TRAJECTORY TAPE IDENTIFICATION NUMBER                        .
00100    C .       FILM    FILM FLAG (0,1)                                              .
00100    C .       PRINT   PRINT FLAG (0,1)                                             .
00100    C .      THE VHFASP TRAJECTORY TAPE IS MOUNTED ON UNIT E (FORTRAN I. D. NUMBER .
00100    C .   7) AND IS ASSUMED TO BE IN THE 800 BPI DENSITY MODE.                     .
00100    C .                                                                            .
00100    C .   OUTPUT-                                                                  .
00100    C .      MICROFILM OUTPUT IS PROCUCED IF FILM = 1.  PRINT OUTPUT IS PRODUCED   .
00100    C .   IF PRINT = 1.                                                            .
00100    C .                                                                            .
00100    C ..............................................................................
00100    C
00101          INTEGER TAPE, FILM, PRINT
00103          DIMENSION ROCB(3), ROLB(3), ACB(3,3), ALB(3,3), ARRAY(25)
00104          EQUIVALENCE (ARRAY(1),T), (ARRAY(2),ROCB(1)), (ARRAY(5),ROLB(1)),
00104         $(ARRAY(8),ACB(1,1)), (ARRAY(17),ALB(1,1))
00105          NAME LIST /I N P U T/ TAPE, FILM, PRINT
00106          NAME LIST /O U T P U T/ T, ROCB, ACB, ROLB, ALB, NRECDS
00106    C
00106    C *** INITIALIZATION
00107          REWIND 7
00110          TAPE = 0
00111          FILM = 0
00112          PRINT = 0
00113          NRECDS = 0
00113    C
00113    C *** READ INPUT
00114          READ(5,INPUT)
00114    C
00114    C *** PRINT HEADING
00117          IF(FILM .NE. 0) WRITE(17,1)TAPE, FILM, PRINT
00125          IF(PRINT .NE. 0) WRITE(6,1)TAPE, FILM, PRINT
00133        1 FORMAT(17H1DUMP OF TAPE NO.I7,14H.  FILM FLAG =I2,15H.  PRINT FLAG
00133         $ =I2,1H./1H0)
```

```
00133     46*    C
00133     47*    C *** PROCESS RECORDS
00134     48*        2 READ(7) ARRAY
00142     49*          NRECDS = NRECDS + 1
00143     50*          IF(FILM .NE. 0) WRITE(17,OUTPUT)
00147     51*          IF(PRINT .NE. 0) WRITE(6,OUTPUT)
00153     52*          GO TO 2
00154     53*          END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
```

7.16 UTILITY PROGRAM UTILT2

```
& FOR UTILT2,UTILT2                                                        00&34&39
UNIVAC 1108 FORTRAN V LEVEL   2206 0016 F5016M
THIS COMPILATION WAS DONE ON 27 JUN 68 AT C0&34&39

  MAIN PROGRAM

  STORAGE USED (BLOCK, NAME, LENGTH)

        0001    *CODE    000076
        0000    *DATA    CC0077
        0002    *BLANK   CC0000

  EXTERNAL REFERENCES (BLOCK, NAME)

        0003    STABL
        0004    STREL
        0005    SSORT
        0006    STRTN
        0007    NREW$
        0010    NRNL$
        0011    NRBU$
        0012    NIO1$
        0013    NIO2$
        0014    NWBU$
        0015    NWEF$
        0016    NSTOP$

  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001    000040 10L        0001    000C21 117G        0001    000026 123G        0001    000051 135G        000(
    0000 I  000062 I          0000 I  000063 IFLAG       0000 I  000061 I$          0000 I  000060 NRECDS      000(

00100      1*    C .......................................................................
```

```
00100   C .                                                                              .
00100   C .   UTILT2                                        C. ARGILA, JUNE '68          .
00100   C .                                                                              .
00100   C .   DESCRIPTION-                                                               .
00100   C .      UTILT2 IS A UTILITY PROGRAM FOR USE WITH THE VHFASP PROGRAM.  UTILT2    .
00100   C .   SORTS THE VHFASP TRAJECTORY TAPE SO THAT TIME-POINTS ARE IN INCREASING     .
00100   C .   ORDER.                                                                     .
00100   C .                                                                              .
00100   C .   INPUT-                                                                     .
00100   C .      INPUT IS VIA THE FORTRAN NAMELIST '$INPUT.'  INPUT PARAMETERS ARE-      .
00100   C .            NRECDS  NUMBER OF RECORDS ON TRAJECTORY TAPE.                     .
00100   C .   THE '$SORT$' SORT TAPE MUST BE ENTERED INTO THE USER PCF AREA.  THE        .
00100   C .   TRAJECTORY TAPE TO BE SORTED IS MOUNTED ON UNIT D (FORTRAN I. D. NUMBER    .
00100   C .   4) AND IS ASSUMED TO BE IN THE 800 BPI DENSITY MODE.                       .
00100   C .                                                                              .
00100   C .   OUTPUT-                                                                    .
00100   C .      THE SORTED TRAJECTORY TAPE IS WRITTEN ON UNIT E (FORTRAN UNIT 7)        .
00100   C .      FASTRAND FILES MUST BE ASSIGNED TO UNITS I, J AND L FOR SCRATCH USE.    .
00100   C .                                                                              .
00100   C ................................................................................
00100   C
00101         INTEGER SORTAB(23)
00103         DIMENSION AREA(25)
00104         NAMELIST /INPUT/ NRECDS
00105         DATA SORTAB /150,6H     I,6H     J,6H     L,13*0,6,1,6H     B,6H
00105        $   A,0,0/
00105   C
00105   C *** INITIALIZATION
00107         REWIND 4
00110         REWIND 7
00111         NRECDS = 0
00111   C
00111   C *** READ INPUT
00112         READ(5,INPUT)
00112   C
00112   C *** SORT
00115         CALL STABL(SORTAB)
00116         DO 1 I = 1, NRECDS
00121         READ(4) AREA
00127       1 CALL STREL(AREA)
00131         CALL SSORT
```

```
00132          10 CALL STRTN(AREA,IFLAG)
00133             WRITE(7) AREA
00141             IF(IFLAG .NE. -1) GO TO 10
00143             END FILE 7
00144             REWIND 4
00145             REWIND 7
00146             STOP
00147             END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.      0 *DIAGNOSTIC* MESSAGE(S)
```

8. APPENDIX A -- Plotting Routine

We include here the most current documentation available on the TRWPLT General Plotting Program by Mr. C. Coflin. This program has been modified by Mr. R. Cunningham to print abscissa labels in days, hours and minutes rather than just a single unit. Modifications for this option have been included in this documentation. Reference should be made to the sample case (Section 4.) which illustrates the usage of this plot program.

CCDD80 PLOT PROGRAM

1. DESCRIPTION

CCDD80 is a compact, general plot program requiring minimum input while combining many of the best features of current plot packages. The basic capabilities are now available even though many additional features are presently in the planning stage.

The capability now available includes the generation of microfilm and/or CALCOMP plots, multiple traces per plot, multiple files or phases per trace, multiple symbol designation, automatic scaling, and axis labeling. In addition, this package can plot variables calculated at different time frequencies, and can plot the sum, difference, product, or quotient of any two of these variables. This, perhaps unique, capability of plotting variables appearing at different time frequencies is possible by the elimination of those data points for which there is not associated a common point in time.

CCDD80 is presently designed to process a variable format data tape consisting of the variable names (symbols) and the corresponding data points for each record type used. A unique record type should be associated with each group of variables which have a common basis. For example, a mod deck is available which will slightly alter the format of the VLIST tape generated by the TRW six degree of freedom program so that it can be processed by CCDD80. These alters assign a record type to each vehicle for every distinct time frequency. The variables appearing in each group are input via a VLISTi card where i = 1, 5.

Record Type	Input VLIST	Vehicle	Print Group
1	VLIST1	1	All 3-D
2	VLIST2	2	All 3-D
3	VLIST3	6-D	6-D Time (T, TT, PHTIME)
4	VLIST4	6-D	Six Degree
5	VLIST5	6-D	SCS Control or SCS Monitor

If N different record types are employed, the first N records of each tape file must contain the variable name or symbols associated with each record type. The symbol records are designated as such by a negative record type (IREC). The format of the N symbol records is as follows:

$-IREC_1$, NWDS, $SYMBOL_1$, $SYMBOL_2$, ..., $SYMBOL_{NWDS}$
$-IREC_2$, NWDS, $SYMBOL_1$, $SYMBOL_2$, ..., $SYMBOL_{NWDS}$
⋮ ⋮ ⋮ ⋮ ⋮
$-IREC_N$, NWDS, $SYMBOL_1$, $SYMBOL_2$, ..., $SYMBOL_{NWDS}$

where IRECi are the N different record types, NWDS is the number of symbols appearing in that record, and SYMBOLi are the BCD variable names associated with that record type (IREC).

The data records, designated by a positive record type, follow the symbol records with each record type appearing at its own natural frequency. The format of a data record is as follows:

$$IREC, NWDS, DATA_1, DATA_2, \ldots, DATA_{NWDS}$$

where IREC is the record type, NWDS is the number of data points appearing in that record, and DATAi are the data points associated with the SYMBOLi appearing in the same type record.

An end of file mark signals an end of the data records (i.e. end of the phase).

1.1. INPUT

The inputs to CCDD80 are divided into two blocks, the plot block and the phase block. A plot block consists of all inputs pertaining to one particular grid or graph. The phase block is comprised of all plot blocks relative to one particular tape file or program phase. The last card of each block must be an end card - ENDPLT for the plot block, ENDPHA for the phase block. An ENDJØB card must terminate the input deck.

1.1.1 DD80 and CALCOMP INPUTS

ICCØMP - integer value of the plot selector (initialized 0)
= 0 DD80
= 1 CALCOMP

IPRINT - integer value of the print selector (initialized 0)
= 0 suppress print
= 1 print titles and data points
= -1 print titles only

NFILES - integer value of the number of files to be plotted on the same graph (initialized 0)

PLØT - symbols or variable names to be plotted and the corresponding record types.
= $XSYM, IREC, YSYM_1, IREC, YSYM_2, IREC, \ldots, YSYM_N, IREC, ENDLST$

where XSYM denotes the abscissa symbol,
YSYMi denotes the N ordinate symbols,
IREC denotes the record type of the preceding symbol,
and ENDLST indicates the end of the list.

NOIREC - disregard above "IREC" parameter if NOIREC = 1 (initialized 0)

STEP - step function flag (initialized 0)
= 0 normal mode
= 1 plot step functions

This specification will result in N traces for this graph (max. N = 10),
XSYM vs. $YSYM_1$, XSYM vs. $YSYM_2$, ..., XSYM vs. $YSYM_N$

A special feature permits the user to plot the sum, difference, product, or quotient of any two variables, specified by the inputs SUM, DIFF, PRØD, or QUØT respectively. When one of these inputs appears in a PLØT list, rather than plotting the next two symbols, the result of the operation requested will be plotted. For example,

PLØT = X, 1, Y, 2, Z, 4, SUM, Y, 2, Z, 4, ENDLST

will result in three traces on this graph.

X vs. Y, X vs. Z, and X vs. Y + Z

where X appears in record type 1,
      Y appears in record type 2,
and   Z appears in record type 4.

CADD - Additive factor
= ADD_1, ADD_2, ..., ADD_{N+1}
where ADDi refers to the respective symbols in the PLØT list.
(i.e. ADD_1 refers to XSYM,
ADD_2 refers to $YSYM_1$,
ADD_3 refers to $YSYM_2$,
⋮ ⋮
and ADD_{N+1} refers to $YSYM_N$)

CMULT - Multiplicative factor
= $MULT_1$, $MULT_2$, ..., $MULT_{N+1}$
where MULTi refers to the respective symbols in the PLØT list.
(i.e. $MULT_1$ refers to XSYM,
$MULT_2$ refers to $YSYM_1$,
$MULT_3$ refers to $YSYM_2$,
⋮ ⋮
$MULT_{N+1}$ refers to $YSYM_N$)

IFREQ - integer value of the plot symbol frequency per trace (max. N = 10)
= K_1, K_2, ..., K_N
The i^{th} trace will display a plot symbol every Ki^{th} point.

XLØ - abscissa lower limit

XHI - abscissa upper limit

YLØ - ordinate lower limit

YHI - ordinate upper limit

TITLE - graph title
= ID = title

XLABEL - X-axis title
= ID = title

YLABEL - Y-axis title
= ID = title

1.1.2 DD80 INPUTS

NCHAR - integer value of the plot symbols for each trace on the graph. (max of ten values) If the value is negative, the points will not be connected. Refer to Table 1-1.

NTYPE - integer value of the type of grid desired (initialized 0)
= 0 Cartesian graph with grid lines
= 1 Cartesian graph without grid lines
= 2 X-axis logarithmic, with grid lines
= 3 Y-axis logarithmic, with grid lines
= 4 X, Y axes logarithmic, with grid lines
= 5 Horizontal bar chart, without grid lines
= 6 Vertical bar chart, without grid lines

NXL - left margin in raster counts (initialized 24)

NXR - right margin in raster counts (initialized 0)

NYL - bottom margin in raster counts (initialized 24)

NYH - top margin in raster counts (initialized 24)

KDAY - plot abscissa label in days, hours, minutes and seconds (initialized 0)
= 0, Normal mode
= 1, Abscissa in seconds
= 2, Abscissa in minutes
= 3, Abscissa in hours
= 4, Abscissa in days

1.1.3 CALCOMP INPUTS

CCHAR - integer value of the plot symbol for each trace in the graph (initialized 0)

= 0 no symbol
= 1 + plus sign
= 2 Δ triangle
= 3 Z letter Z
= 4 octagon
= 5 ↑ arrow
= 6 X letter X
= 7 diamond
(max 10)

PPNM - grid selector

= 1.	10 x 7	Form C
= 2.	10 x 10	Form A or B
= 3.	15 x 10	Form D
= 4.	20 x 10	Form A or B

YLAB1 - Y-axis title for first trace
= ID = title

YLAB2 - Y-axis title for second trace
= ID = title

YLAB3 - Y-axis title for third trace
= ID = title

YLAB4 - Y-axis title for fourth trace
= ID = title

YLAB5 - Y-axis title for fifth trace
= ID = title

1.1.4 CONTROL INPUTS

BACK - integer value of the number of file marks over which the plot tape will be back spaced.

KUNIT - integer number of the unit on which the input data tape will be mounted (initialized 8). Avoid units 1, 2, 4, 12, and 13.

RECØRD - integer values indicating which records of the data tape will be plotted. RECØRD should be used only if it is desired to plot only part of a file (3 values, initialized 0, 0, 0)
= N_1, N_2, N_3

Data will be extracted from the data tape beginning with the N_1^{st} data record. N_2 records of record type N_3 will be processed.

N_1=0 Begin with the first data record.

N_2=0 Process data until an end of file mark is encountered.

N_3=0 Process N_2 number of records of any record type.

RECSKP - integer value of the number of symbol records to be skipped at the beginning of the file

REPEAT - causes the current file to be processed with the same inputs used in the preceding file.

REWIND - causes the input data tape to be rewound.

SKIP - integer value of the number of file marks over which the input data tape will be forward spaced.

ENDPLT - terminates a block of data for one plot.

ENDPHA - terminates a block of data for the file.

ENDJØB - terminates all inputs.

1.2. GENERAL INFORMATION

1.2.1 If XLØ and XHI or YLØ and YHI are not input, they will be computed from the maxima and minima of the data points for the first trace. Care should be taken to input the variables if the bounds of the first trace are exceeded by those of another trace. It should be noted that in the case of CALCOMP plots, the values of XLØ, XHI, YLØ, and YHI may be adjusted to correspond to the optimum scaling for the paper size requested.

1.2.2 All inputs carry over from block to block

1.2.3 If the CALCOMP option is selected, then the inputs applicable to DD80 are ignored and vice versa. As a result, both DD80 and CALCOMP inputs may both appear in an input deck. The set of inputs used is controlled by the setting of ICCØMP.

1.2.4 In the case of CALCOMP log plots, the logarithm of each point is plotted. Since no provision is made for log paper, the scale will be linear.

Table 1-1. dd80 Plot Symbol Designation

NCHAR	Symbol on Plot	NCHAR	Symbol on Plot
0	∂	32	)
1	"	33	-
2	π	34	+
3	δ	35	· (plotting dot)
4	α	36	=
5	blank	37	γ
6	A	38	~
7	B	39	$
8	C	40	*
9	D	41	(
10	E	42	d
11	F	43	o
12	G	44	∫
13	H	45	Σ
14	I	46	,
15	J	47	□
16	K	48	0
17	L	49	1
18	M	50	2
19	N	51	3
20	O	52	4
21	P	53	5
22	Q	54	6
23	R	55	7
24	S	56	8
25	T	57	9
26	U	58	' (apostrophe)
27	V	59	β
28	W	60	/
29	X	61	. (period)
30	Y	62	?
31	Z	63	±

Note: All alphabetic and numeric symbols correspond to 1108 field data code.

Table 1-2. Example of DD80 Inputs

KUNIT = 8	* The input plot tape is mounted on Unit 8.
IPRINT = 1	* Print titles and points to be plotted on the standard output tape.
PLØT = X, 1, X, 2, ENDLST	* First frame plots X (record type 1) vs. X (record type 2)
CMULT = 1.E-6, 1.E-3	* Each point of X(1) is multiplied by $1. \times 10^{-6}$ and each point of X(2) is multiplied by $1. \times 10^{-3}$.
CADD = 5., -5.	* After being multiplied by CMULT, 5. is added to each point of X(1) and -5. is added to each point of X(2).
TITLE = ID = GRAPH TITLE XLABEL = ID = X-AXIS TITLE YLABEL = ID = Y-AXIS TITLE ENDPLT	Terminates inputs for first frame.
PLØT = X, 1, Y, 1, Y, 2, Y, 3, ENDLST	* The second frame consists of three traces, X(1) vs. Y(1), X(1) vs. Y(2), and X(1) vs. Y(3).
CMULT = 0., 0.	* Multipliers are reset (CMULT = 0. will cause the multiplier to be ignored!)
CADD = 0., 0.	* Adders are reset.
NCHAR = 29, 35, 34	* Plot symbol for the traces will be X, ., and +, respectively.
YLØ = -5.	* Lower limit of the Y-axis.
YHI = 10.	* Upper limit of the Y-axis.
TITLE = ID = GRAPH TITLE XLABEL = ID = X-AXIS TITLE YLABEL = ID = Y-AXIS TITLE ENDPHA	Terminates inputs for second frame and for phase.
SKIP = 2	* Skip the second and third phases.
RECORD = 10, 150, 0	* Plot data beginning with the 10th data record and continuing for 150 records.
REPEAT	Plot on the third and fourth frames the same variables for the fourth phase as for the first phase.
BACK = 1	* Backspace to beginning of the fourth phase.

Table 1-2. Example of DD80 Inputs (Continued)

RECORD = 0, 0, 0	* Reset record selector.
NFILES = 3	* Plot data for the fourth, fifth, and sixth phases on the same frame.
REPEAT	Plot on the fifth and sixth frames the same variables for the next three phases as for the first phase.
REWIND	Rewind the input plot tape and position to the first phase.
NFILES = 36	* Plot data for the next thirty-six phases on the same frame.
PLØT = T, 1, PRØD, Y, 2, Z, 5, ENDLST	* The seventh frame plots T(1) vs. Y(2)*Z(5)
YLØ = 0.	* Resets lower limit of Y-axis so the minima of the ordinate points will be used.
YHI = 0.	* Resets upper limit of Y-axis so the maxima of the ordinate points will be used.
TITLE = ID = GRAPH TITLE XLABEL = ID = X-AXIS TITLE YLABEL = ID = Y-AXIS TITLE ENDPHA	Terminates inputs for the seventh frame and for the phase.
ENDJØB	Terminates all inputs.

* Only those inputs requiring values (i.e. inputs other than REPEAT, ENDPLT, etc.) may be terminated by an asterisk and followed by comments.

Table 1-3. Example of CALCOMP Inputs

KUNIT = 8	* The input tape is mounted on Unit 8.
IPRINT = 1	* Print titles and points to be plotted on the standard output tape.
ICCØMP = 1	* Generate CALCØMP plots.
PPNM = 1.0	* Plot grid size set to 10 x 7 (abscissa x ordinate).
CCHAR = 1	* Plot symbol for first trace is +.
IFREQ = 2	* Plot requested symbol at every other plot point.
PLØT = X, 1, X, 2, ENDLST	* First frame plots X (record type 1) vs. X (record type 2).
CMULT = 1.E-6, 1.E-3	* Each point of X(1) is multiplied by 1. x 10^{-6} and each point of X(2) is multiplied by 1. x 10^{-3}.
CADD = 5., -5.	* After being multiplied by CMULT, 5. is added to each point of X(1) and -5. is added to each point of X(2).
TITLE = ID = GRAPH TITLE XLABEL = ID = X-AXIS TITLE YLABEL = ID = Y-AXIS TITLE ENDPLT	Terminates inputs for first frame.
PLØT = X, 1, Y, 1, Y, 2, Y, 3, ENDLST	* The second frame consits of three traces, X(1) vs. Y(1), X(1) vs. Y(2), and X(1) vs. Y(3).
CMULT = 0., 0.	* Multipliers are reset (CMULT = 0. will cause the multiplier to be ignored!)
CADD = 0., 0.	* Adders are reset.
CCHAR = 2, 3, 6	* Plot symbol for the traces is Δ, Z, and X, respectively.
IFREQ = 5, 10, 50	* The plot symbol will be displayed at every 5th, 10th, and 50th point, respectively.
YLØ = -5.	* Lower limit of Y-axis.
YHI = 10.	* Upper limit of Y-axis.

Table 1-3. Example of CALCOMP Inputs (Continued)

TITLE = ID = GRAPH TITLE XLABEL = ID = X-AXIS TITLE YLAB1 = ID = FIRST TRACE Y-AXIS TITLE YLAB2 = ID = SECOND TRACE Y-AXIS TITLE YLAB3 = ID = THIRD TRACE Y-AXIS TITLE	
ENDPHA	Terminates inputs for the second frame and for the phase.
SKIP = 2	* Skip the second and third phases.
RECØRD = 10, 150, 0	* Plot data beginning with the 10th data record and continuing for 150 records.
REPEAT	Plot on the third and fourth frames the same variables for the fourth phase as for the first phase.
BACK = 1	* Backspace to beginning of the fourth phase.
RECØRD = 0, 0, 0	* Reset record selector.
NFILES = 3	* Plot data for the fourth, fifth, and sixth phases on the same frame.
REPEAT	Plot on the fifth and sixth frames the same variables for the next three phases as for the first phase.
REWIND	Rewind the input plot tape and position to the first phase.
NFILES = 36	* Plot data for the next thirty-six phases on the same frame.
PLØT = T, 1, PRØD, Y, 2, Z, 5, ENDLST	* The seventh frame plots T(1) vs. Y(2)*Z(5).
YLØ = 0.	* Resets lower limit of Y-axis so the minima of the ordinate points will be used.
YHI = 0.	* Resets upper limit of Y-axis so the maxima of the ordinate points will be used.
TITLE = ID = GRAPH TITLE XLABEL = ID = X-AXIS TITLE YLABEL = ID = Y-AXIS TITLE	
ENDPHA	Terminates inputs for the seventh frame and for the phase.
ENDJØB	Terminates all inputs.

* Only those inputs requiring values (i.e. inputs other than REPEAT, ENDPLT, etc.) may be terminated by an asterisk and followed by comments.

JOB DECK SETUP

The Univac 1108 control cards necessary to execute the CCDD80 program are listed below.

Control Cards	Comment
∇ JØB	Form specified by local directive.
∇ ASG X = alpha character or label	Tape number of the CCDD80 PCF tape.
∇ ASG F = input tape	Tape number of the user's input plot tape.
∇ ASG P = P, Q = Q	Output tapes generated for CALCOMP - neither P nor Q need be assigned if DD80 plots are requested.
∇ XQT CUR	
TRW X	Rewind PCF tape
IN X	Load PCF
TRI X	
∇ XQT TRWPLT	Execute program (overlay MAP)
DATA DECK (plot input)	
∇ EØF	

NOTE: For a CALCOMP run, the color of ink, the pen thickness, and the paper form must be specified.

Color of ink: BLACK or BK, BLUE or BL, RED or R.
Pen thickness: FINE or FN, MEDIUM or MED, BROAD or BRD.
Paper form: Plain white or A, grid or B, 11 x 8 1/2 or C, 15 x 10 or D.

The local form provided for this purpose is:

```
NAME      DATE       PROJ
PLØTS     SEQ. NO.   TAPE NO.
PEN 02-04-06**    INK BL-BK-R
FORM--1730A--1731B--1732C--1733D
```

9. APPENDIX B -- Tape Formats

The only input tapes required by the VHF Antenna Simulation Program are the trajectory tape and the antenna patterns tape. Normally, the antenna patterns tape is included as a second file on the production PCF tape so that the user is not aware of its existence. A trajectory tape must be mounted each time the program is executed. Both of these tapes are FORTRAN binary tapes operated in the 800BPI density mode.

9.1 FORMAT OF TRAJECTORY TAPE

The VHFASP trajectory tape consists of one file. Each record in this file corresponds to precisely one time-point. Each record consists of twenty-five words as described below. The records are assumed to be ordered by increasing times.

Word(s)	Description
1	Trajectory time (in seconds)
2 - 4	Position vector of CSM in basic coordinates, $\bar{R}_{oc} = \begin{bmatrix} \text{Word } 2 \\ \text{Word } 3 \\ \text{Word } 4 \end{bmatrix}$.
5 - 7	Position vector of LM in basic coordinates, $\bar{R}_{ol} = \begin{bmatrix} \text{Word } 5 \\ \text{Word } 6 \\ \text{Word } 7 \end{bmatrix}$.
8 - 16	Direction cosine matrix of CSM, $A_{cb} = \begin{bmatrix} \text{Word } 8 & \text{Word } 11 & \text{Word } 14 \\ \text{Word } 9 & \text{Word } 12 & \text{Word } 15 \\ \text{Word } 10 & \text{Word } 13 & \text{Word } 16 \end{bmatrix}$.
17 - 25	Direction cosine matrix of LM, $A_{lb} = \begin{bmatrix} \text{Word } 17 & \text{Word } 20 & \text{Word } 23 \\ \text{Word } 18 & \text{Word } 21 & \text{Word } 24 \\ \text{Word } 19 & \text{Word } 20 & \text{Word } 25 \end{bmatrix}$.

9.2 FORMAT OF ANTENNA PATTERNS TAPE

The VHFASP antenna patterns tape is in an extremely compact format, packing as much information in as little tape as possible. In this manner it is possible to reduce tape manipulation to a minimum and hence to increase reliability while decreasing run time. The antenna patterns tape consists of one twelve-record file. Each record corresponds to precisely one CSM or LM antenna configuration. Records appear in the following order:

Record	Description
1	Pattern data for CSM low frequency antenna A
2	Pattern data for CSM low frequency antenna B
3	Pattern data for CSM high frequency antenna A
4	Pattern data for CSM high frequency antenna B
5	Pattern data for LM low frequency, descent, antenna 1
6	Pattern data for LM low frequency, descent, antenna 2
7	Pattern data for LM low frequency, ascent, antenna 1
8	Pattern data for LM low frequency, ascent, antenna 2
9	Pattern data for LM high frequency, descent, antenna 1
10	Pattern data for LM high frequency, descent, antenna 2
11	Pattern data for LM high frequency, ascent, antenna 1
12	Pattern data for LM high frequency, ascent, antenna 2

Each record consists of a single 90-row by 180-column matrix representing, in two degree increments, all of the points located on a sphere about the antenna. Each entry in this matrix is a twelve digit octal integer representing all of the antenna pattern data for this particular antenna configuration at this point. The information is coded into these octal digits as follows:

Digits	Description
1 - 2	P_{rcp} antenna pattern data
3 - 4	P_{lcp} antenna pattern data
5 - 6	$P_{\phi+45^o}$ antenna pattern data
7 - 8	$P_{\theta+45^o}$ antenna pattern data
9 - 10	P_{ϕ} antenna pattern data
11 - 12	P_{θ} antenna pattern data

The antenna pattern data thus obtained must then be converted to a decimal integer and subtracted from the appropriate gain reference value to obtain the gain (in decibels) of the antenna at this point. The gain reference values for the various antenna configurations is given below.

Vehicle	Frequency	A/D Status	Antenna	$P_\theta - P_\phi$	$P_{\theta+45} - P_{\phi+45}$	$P_L - P_R$
CSM	LOW	--	A	5.52	5.48	3.76
CSM	LOW	--	B	5.45	5.41	3.74
CSM	HIGH	--	A	6.19	6.29	3.27
CSM	HIGH	--	B	5.10	5.12	3.25
LM	LOW	D	1	6.79	3.56	5.68
LM	LOW	D	2	7.64	1.58	4.77
LM	LOW	A	1	4.25	4.29	5.60
LM	LOW	A	2	3.44	3.08	5.30
LM	HIGH	D	1	2.47	2.30	5.96
LM	HIGH	D	2	6.04	5.10	6.28
LM	HIGH	A	1	3.51	1.69	4.91
LM	HIGH	A	2	3.06	3.24	4.19

9.2.1 An Example

It is known that at the trajectory time-point 4.02.56.31.7 the look angles of the CSM along the direct path to the LM are $\theta = 3.123$ and $\phi = 3.074$. What are the gains of the six antenna patterns at this point for antenna A operated in the low frequency mode?

Solution: The antenna pattern data for CSM low frequency antenna A is located in the first record of the antenna patterns tape. We must locate the proper entry of this matrix to obtain the pattern data.

Converting the look angles to the nearest two degree increments we have: $\theta = 178^\circ$, $\phi = 176^\circ$. Hence, the entry we are looking for is located at row 89, column 88. Refering to a dump of this tape we find this entry to be 130710261214. With the aid of Table 9.1 we determine the antenna pattern data as 11,7,8,22,10,12. The desired antenna pattern gains are then:

$P_{rcp} = 3.76 - 11 = -7.24$db, $P_{lcp} = 3.76 - 7 = -3.24$db,

$P_{\phi+45^\circ} = 5.48 - 8 = -2.52$db, $P_{\theta+45^\circ} = 5.48 - 22 = -16.52$db,

$P_\phi = 5.52 - 10 = -4.48$db, $P_\theta = 5.52 - 12 = -6.48$db.

TABLE 9.1

OCTAL TO DECIMAL INTEGER CONVERSION TABLE

	0	1	2	3	4	5	6	7
00	00	01	02	03	04	05	06	07
10	08	09	10	11	12	13	14	15
20	16	17	18	19	20	21	22	23
30	24	25	26	27	28	29	30	31
40	32	33	34	35	36	37	38	39
50	40	41	42	43	44	45	46	47
60	48	49	50	51	52	53	54	55
70	56	57	58	59	60	61	62	63

10. APPENDIX C -- Utility Routines

The VHFASP package contains two utility programs which dump and order the trajectory tape as may be required when a new trajectory tape is obtained.

10.1 UTILT1

The program UTILT1 dumps the VHFASP trajectory tape onto either microfilm or print or both. Input is via the FORTRAN namelist "$INPUT." The input parameters are TAPE (trajectory tape identification number), FILM (0 indicates no microfilm output desired, 1 indicates microfilm output desired) and PRINT (0 indicates no print output desired, 1 indicates print output desired). In addition to listing each record on the trajectory tape, UTILT1 provides the number of records, NRECDS, on the tape.

The trajectory tape must be mounted on unit "E" (FORTRAN I. D. Number 7) in the 800BPI density mode. UTILT1 may be executed before or after VHFASP or independently of VHFASP. The required control cards are indicated below. It is assumed that the HV014B PCF tape has already been entered into the user PCF area.

```
c
c
1
∇△ASG△E=(trajectory tape number)
∇△XQT△UTILT1
[input data
∇△EOF
```

10.2 UTILT2

The program UTILT2 reorders a given VHFASP trajectory tape so that records are in order of increasing time. Input is via the FORTRAN namelist "$INPUT." The sole input parameter is NRECDS (number of records on the old trajectory tape).

The old trajectory tape must be mounted on unit "D" (FORTRAN I. D. Number 4) in the 800BPI density mode. The new trajectory tape (to be saved) must be mounted on unit "E" (FORTRAN I. D. Number 7). UTILT2 may be executed before or after VHFASP or independently of VHFASP. The required control cards are indicated below. It is assumed that the HV014B PCF tape has already been entered into the user PCF area.

```
c
c
1
∇△ASG△D=(old trajectory tape number)
∇△ASG△E=SAVE (new trajectory tape to be saved)
∇△ASG△I,J,L (three FASTRANDS required)
∇△XQT△UTILT2
[
[input data
[
∇△EOF
```

D A S (initials)

D. A. S.

TRW
SYSTEMS GROUP

INTEROFFICE CORRESPONDENCE

5512.30-005

TO: File CC: Task E-78 File DATE: 14 January 1969

SUBJECT: Modifications to the VHF Antenna Simulation Program (HV014B)

FROM: C. A. Argila
BLDG. H1 ROOM 2076 EXT. 2503

A number of minor modifications have been made to the VHF Antenna Simulation Program (HV014B) to alter the Trajectory Tape input format and to correct the program for very small look angles. A third Utility Program (UTILT3) has also been prepared to enable the user to input trajectory data from cards rather than tape.

The attached pages have been prepared to update the HV014B User's Manual (HCC Report No. 3145.30-012) to include these modifications. The following table indicates which pages are to be deleted (from the present manual) and which pages are to be inserted (from the attached pages).

Replace Page (s)	By Page (s)
7-2 to 7-15	7-2 (Mod I) to 7-15b (Mod I)
7-25	7-25 (Mod I)
7-27	7-27 (Mod I), 7-27a (Mod I)
7-29	7-29 (Mod I), 7-29a (Mod I)
7-34, 7-35	7-34 (Mod I), 7-35 (Mod I)
9-1	9-1 (Mod I)
10-2	10-2 (Mod I)

C. A. Argila
Electrical and Electronics Group

Approved: W. P. Bennett
W. P. Bennett, Head
Subsystem Analysis Section

CAA/du

7.1 THE MAIN DRIVER, VHFASP

```
&  FOR,* VHFASP,VHFASP                                        17&07&14
UNIVAC 1108 FORTRAN V LEVEL    2206 0018 F5018H
THIS COMPILATION WAS DONE ON 14 JAN 69 AT 17&07&14

- 188,188

 300 READ(7) (ARRAY(I,NPOINT),I=1,7),

    $(ARRAY(I,NPOINT),I=8,14,3),

    $(ARRAY(I,NPOINT),I=9,15,3),

    $(ARRAY(I,NPOINT),I=10,16,3),

    $(ARRAY(I,NPOINT),I=17,23,3),

    $(ARRAY(I,NPOINT),I=18,24,3),

    $(ARRAY(I,NPOINT),I=19,25,3)

- 246

     IF(ARRAY(30,INDEX) .LT. 0.034903 .OR. ARRAY(30,INDEX) .GT. 6.2482)

    $ARRAY(30,INDEX) = 0.034923

     IF(ARRAY(31,INDEX) .LT. 0.034903 .OR. ARRAY(31,INDEX) .GT. 6.2482)

    $ARRAY(31,INDEX) = 0.034923

- 252

     IF(ARRAY(34,INDEX) .LT. 0.034903 .OR. ARRAY(34,INDEX) .GT. 6.2482)

    $ARRAY(34,INDEX) = 0.034923

     IF(ARRAY(35,INDEX) .LT. 0.034903 .OR. ARRAY(35,INDEX) .GT. 6.2482)

    $ARRAY(35,INDEX) = 0.034923

- 298
```

```
 IF(ARRAY(32,INDEX) .LT. 0.034903 .OR. ARRAY(32,INDEX) .GT. 6.2482)
$ARRAY(32,INDEX) = 0.034923
 IF(ARRAY(33,INDEX) .LT. 0.034903 .OR. ARRAY(33,INDEX) .GT. 6.2482)
$ARRAY(33,INDEX) = 0.034923
- 304
 IF(ARRAY(36,INDEX) .LT. 0.034903 .OR. ARRAY(36,INDEX) .GT. 6.2482)
$ARRAY(36,INDEX) = 0.034923
 IF(ARRAY(37,INDEX) .LT. 0.034903 .OR. ARRAY(37,INDEX) .GT. 6.2482)
$ARRAY(37,INDEX) = 0.034923

MAIN PROGRAM

STORAGE USED (BLOCK, NAME, LENGTH)

      0001   *CODE    003760
      0000   *DATA    117155
      0002   *BLANK   000000
      0003   DATA     000031
      0004   GNREF    000044


EXTERNAL REFERENCES (BLOCK, NAME)

      0005   INITAL
      0006   REFLKT
      0007   LOOKCM
      0010   LOOKLM
      0011   VECSUB
      0012   VECMAG
      0013   POLANG
```

```
0014    TILT
0015    MISALN
0016    GAINRR
0017    DB
0020    POLAR
0021    VECDOT
0022    POLARS
0023    NREW$
0024    NRBU$
0025    NIO1$
0026    NIO2$
0027    NWBU$
0030    NRNL$
0031    NWDU$
0032    NRDU$
0033    NERR2$
0034    ACOS
0035    COS
0036    CSQRT
0037    SIN
0040    CDV$
0041    ATAN2
0042    TAN
0043    CEXP
0044    CABS
0045    ATAN
0046    ASIN
0047    SQRT
0050    NWEF$
0051    NBSP$
0052    NSTOP$
```

STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

```
0001   000113 100L     0000   116464 1000F    0000   116200 1001F    0000   116206 1003F    0001
0001   000227 1004L    0000   116211 1005F    0001   003160 1007G    0001   000311 1007L    0000
0001   003103 1009L    0000   116474 1010F    0001   003135 1011L    0000   116176 102F     0001
0001   003232 1034G    0001   003242 1040G    0001   003266 1052G    0001   003300 1056G    0001
0001   003314 1063G    0001   003324 1067G    0001   000415 107L     0001   003334 1073G    0001
```

0001		003354	1103G	0001		003400	1116G	0001		003463	1146G	0001		003473	1152G	0001	
0001		003523	1166G	0001		003535	1172G	0001		003551	1177G	0001		003561	1203G	0001	
0001		003601	1213G	0001		000022	122G	0001		003613	1220G	0001		003671	1237G	0001	
0001		003721	1257G	0001		000033	130G	0001		000041	136G	0001		000046	142G	0001	
0001		000065	156G	0001		000072	162G	0001		000103	170G	0001		000170	217G	0001	
0001		000215	237G	0001		000222	243G	0001		000235	253G	0001		000242	257G	0000	
0000		116143	28F	0001		000164	29L	0001		000277	300G	0001		000423	300L	0001	
0001		000671	303L	0001		000304	304G	0001		000737	304L	0001		000331	321G	0001	
0001		000374	346G	0001		001114	350L	0001		003015	356L	0001		000461	365G	0001	
0001		000475	375G	0001		001526	400L	0001		000503	401G	0001		003017	401L	0001	
0001		000517	411G	0001		000525	415G	0001		000772	450G	0001		003633	450L	0001	
0001		003713	452L	0000		117013	453F	0000		117031	454F	0001		001000	454G	0000	
0001		001007	461G	0001		001015	465G	0001		001050	501G	0001		001056	505G	0001	
0001		001402	544G	0001		001746	607G	0001		002075	624G	0001		002157	636G	0001	
0001		002455	666G	0001		002501	673G	0001		003725	703L	0001		002653	712G	0001	
0001		003111	763G	0000		116505	87F	0000		116362	93F	0001		000370	94L	0001	
0000		116416	998F	0001		000764	999L	0000	R	116067	A	0000	C	037525	AA	0003	R
0000	I	000000	ADFLAG	0000	R	037547	ALPHA	0000	R	037551	ARRAY	0000	R	116102	B	0000	C
0000	R	116054	BETA	0000	R	056121	BETAS	0000	R	116073	C	0000	I	000001	COMENT	0000	I
0000	R	116103	D	0017	R	000000	DB	0000	R	116066	DELTAR	0000	R	116022	DELTAT	0003	R
0003	R	000002	EPSLNO	0000	R	116035	FREQ	0000	I	037512	FREQCY	0000	R	116036	FREQRD	0004	R
0004	R	000014	GNREFL	0000	R	116104	GP	0000	R	116105	GPDB	0000	R	116055	G1	0000	R
0000	R	116074	G1S	0000	R	116100	G1SDB	0000	R	116057	G2	0000	R	116062	G2DB	0000	R
0000	R	116101	G2SDB	0003	R	000003	HALFPI	0000	I	116024	I	0000	I	116051	ICODED	0000	I
0000	I	116041	III	0000	I	116044	INDEX	0000	I	116050	IPHI	0000	I	116047	ITHETA	0000	I
0000	I	116026	J	0000	I	116042	JJJ	0000	R	037515	K	0000	I	056123	LEMPAT	0000	I
0000	I	116043	LINK	0000	R	037516	LP	0000	R	037517	LPDB	0000	R	037520	LPS	0000	R
0000	R	037523	LR	0000	I	116023	MULTIP	0000	I	116027	N	0000	I	116030	NCSM	0000	I
0000	I	116037	NPOINT	0000	R	115633	PATERN	0003	R	000004	PI	0000	I	037513	PLOT	0000	I
0000	R	116070	PSI	0000	C	037531	P1S	0000	C	037535	P2S	0000	C	037541	Q	0000	R
0000	C	037543	RA	0000	C	037545	RB	0000	R	116046	RHO	0000	R	116071	RHO1S	0000	R
0000	R	115663	RLCB	0000	R	115666	RRS	0000	R	116056	RR1	0000	R	116075	RR1S	0000	R
0000	R	116077	RR2S	0000	R	116065	RS	0000	R	115670	RSCB	0000	R	115673	RSLB	0003	R
0000	R	116063	R1S	0000	R	116064	R2S	0003	R	000006	SIGMA	0000	R	115676	TAUS	0000	R
0000	R	115700	TAU1S	0000	R	116053	TAU2	0000	R	115702	TAU2S	0003	R	000007	TIME	0000	R
0000	R	116012	TMAX	0000	R	116016	TMIN	0003	R	000013	TWOPI	0003	R	000014	TWOPIC	0003	R
0000	R	116034	VMAX	0000	R	116033	VMIN	0003	R	000026	ZERO						

```
C .
C .   VHF ANTENNA SIMULATION PROGRAM (VHFASP)        C. ARGILA, 1968
C .
C .   DESCRIPTION-
C .      THE VHF ANTENNA SIMULATION PROGRAM SIMULATES THE CSM-LM VHF
C .   COMMUNICATION LINK FOR ALL APOLLO MISSIONS.  IN THE STANDARD
C .   CONFIGURATION VHFASP PRINTS VALUES OF CSM AND LM ANTENNA GAINS,
C .   POLARIZATION LOSSES, CSM-LM RANGE AND GAIN PRODUCT AS FUNCTIONS OF
C .   TIME.  IN AN OPTIONAL MODE EVERY VARIABLE INVOLVED IN THE SIMULATION
C .   MAY BE PRINTED OUT.  A PLOT TAPE IS GENERATED FOR USE WITH THE TRWPLT
C .   GENERAL PLOTTING PROGRAM.  THIS PROGRAM IS WRITTEN IN FORTRAN V FOR USE
C .   ON THE SRU 1108 SYSTEM.
C .
C .   INPUT-
C .     CARD INPUT- CARD INPUT IS VIA THE FORTRAN NAMELIST '$INPUT.'  INPUT
C .   PARAMETERS ARE
C .      FREQCY   FREQUENCY FLAG.  FREQCY=-1 INDICATES LOW FREQUENCY,
C .               FREQCY=+1 INDICATES HIGH FREQUENCY.  FREQCY INITIALIZED TO -1.
C .      ADFLAG   ASCENT/DESCENT FLAG.  ADFLAG=-1 INDICATES DESCENT CONDITION,
C .               ADFLAG=+1 INDICATES ASCENT CONDITION.  ADFLAG INITIALIZED
C .               TO -1.
C .      MULTIP   MULTI-PATH FLAG.  MULTIP=0 INDICATES OMIT MULTI-PATH
C .               CALCULATIONS, MULTIP=1 INDICATES MULTI-PATH CALCULATIONS
C .               ARE TO BE EXECUTED.  MULTIP INITIALIZED TO 1.
C .      PRINT    PRINT FLAG.  PRINT=0 INDICATES ONLY STANDARD PRINT OUTPUT IS
C .               DESIRED.  PRINT=1 INDICATES OPTIONAL PRINT OUTPUT IS ALSO
C .               DESIRED.  PRINT INITIALIZED TO 0.
C .      PLOT     PLOT FLAG.  PLOT=0 INDICATES NO PLOT TAPE IS TO BE GENERATED,
C .               PLOT=1 INDICATES PLOT TAPE IS TO BE GENERATED.  PLOT INITIAL-
C .               IZED TO 1.
C .      TMIN     FOUR-POINT ARRAY REPRESENTING MINIMUM TIME TO BE CONSIDERED,
C .               IN DAYS, HOURS, MINUTES AND SECONDS.  INITIALIZED TO 0,0,0,0.
C .      TMAX     FOUR-POINT ARRAY REPRESENTING MAXIMUM TIME TO BE CONSIDERED,
C .               IN DAYS, HOURS, MINUTES AND SECONDS.  INITIALIZED TO 0,0,0,0.
C .      DELTAT   MINIMUM TIME (IN SECONDS) BETWEEN TIME-POINTS.  INITIALIZED
C .               TO 0.0.
C .     TAPE INPUT- A TRAJECTORY TAPE (MOUNTED ON UNIT E) AND A 'PACKED'
C .   ANTENNA PATTERNS TAPE ARE THE ONLY TAPE INPUTS REQUIRED.  THEESE TAPES
C .   ARE IN THE FORMAT SPECIFIED IN THE HV014B USER'S MANUAL.
C .
```

```
00100   C .   OUTPUT-                                                              .
00100   C .      IN ADDITION TO THE STANDARD AND OPTIONAL PRINT OUTPUT, A PLOT TAPE  .
00100   C .   IN THE FORMAT REQUIRED BY 'TRWPLT' MAY ALSO BE GENERATED.             .
00100   C .                                                                         .
00100   C .   SUBROUTINE USAGE-                                                     .
00100   C .      INITAL                                                             .
00100   C .      REFLKT                                                             .
00100   C .        VECTOR                                                           .
00100   C .        GMPRD                                                            .
00100   C .        GMTRA                                                            .
00100   C .      LOOK                                                               .
00100   C .        VECTOR                                                           .
00100   C .        GMPRD                                                            .
00100   C .      POLANG                                                             .
00100   C .        GMTRA                                                            .
00100   C .        GMPRD                                                            .
00100   C .      TILT                                                               .
00100   C .      MISALN                                                             .
00100   C .      GAINPR                                                             .
00100   C .      DB                                                                 .
00100   C .      POLAR                                                              .
00100   C .      POLARS                                                             .
00100   C .        VECTOR                                                           .
00100   C .        GMPRD                                                            .
00100   C .      DATA (BLOCK DATA SUBROUTINE)                                       .
00100   C .                                                                         .
00100   C .   NAMED COMMON USAGE-                                                   .
00100   C .      DATA                                                               .
00100   C .      GNREF                                                              .
00100   C .                                                                         .
00100   C ...........................................................................
00100   C
00101         INTEGER ADFLAG, COMENT, CSMPAT(90,180), FREQCY, PLOT, PRINT
00103         REAL K, LP, LPDB, LPS(2), LPSDB, LR(2)
00104         COMPLEX AA, BB, P1S(2), P2S(2), Q, RA, RB
00105         DIMENSION ALPHA(2), ARRAY(37,200), BETAS(2), GNREFC(6,2), GNREFL(6
00105        $,4), LEMPAT(90,180), PATERN(6,4), RLCB(3), RRS(2), RSCB(3), RSLB(3
00105        $), TAUS(2), TAU1S(2), TAU2S(2), TIME(4), TITLE(14,5), TMAX(4), TMI
00105        $N(4), UNITY(3,3), ZERO(3)
00106         COMMON /DATA/ ACURCY, EPSLNR, EPSLNO, HALFPI, PI, RO, SIGMA, TIME,
00106        $TWOPI, TWOPIC, UNITY, ZERO
```

7-7 (Mod 1)

```
00107         COMMON /GNREF/ GNREFC, GNREFL
00110         NAME LIST /INPUT/ FREQCY, ADFLAG, COMENT, PRINT, PLOT, TMIN, TMAX,
00110        $DELTAT, MULTIP
00110   C
00110   C *** INITIALIZE TAPES
00111         REWIND 1
00112         REWIND 2
00113         REWIND 3
00114         REWIND 8
00114   C
00114   C *** DUMP ANTENNA PATTERNS ONTO HIGH SPEED DRUMS
00115         DO 99 I = 1, 4
00120         READ(27) CSMPAT
00126      99 WRITE(1) CSMPAT
00135         DO 98 I = 1, 6
00140         READ(27) LEMPAT
00146      98 WRITE(2) LEMPAT
00155         DO 86 I = 1, 2
00160         READ(27) LEMPAT
00166      86 WRITE (3) LEMPAT
00175         REWIND 27
00175   C
00175   C *** REWIND TAPE DRIVES
00176     100 REWIND 1
00177         REWIND 2
00200         REWIND 3
00201         REWIND 7
00201   C
00201   C *** INITIALIZE INPUT PARAMETERS
00202         CALL INITAL(FREQCY, ADFLAG, COMENT, PRINT, PLOT, TMIN, TMAX,
00202        $DELTAT, MULTIP)
00202   C
00202   C *** READ INPUT PARAMETERS
00203         READ(5,INPUT)
00203   C
00203   C *** READ COMMENT/DESCRIPTION CARDS
00206         IF(COMENT .EQ. 0) GO TO 103
00210         IF(COMENT .LE. 5) GO TO 29
00212         WRITE(6,28)
00214      28 FORMAT(1044H1INPUT CARD DECK ERROR.  NUMBER OF COMMENT/DESCRIPTION
00214        $CARDS EXCEEDS FIVE.  EXTRA CARDS WILL BE IGNORED./27HOBREVITY IS T
```

```
00214            $HE SOUL OF WIT/15X,12H-SHAKESPEARE)
00215             COMENT = 5
00216          20 DO 101 J = 1, COMENT
00221         101 READ(5,102) (TITLE(I,J),I=1,14)
00230         102 FORMAT(13A6,1A2)
00230       C
00230       C *** PRINT PARAMETER LISTING AND REPORT HEADING
00231         103 WRITE(6,1001)
00233        1001 FORMAT(1H1,43X,22HVHF ANTENNA SIMULATION)
00234             IF(COMENT .EQ. 0) GO TO 1004
00236             DO 1002 J = 1, COMENT
00241        1002 WRITE(6,1003) (TITLE(I,J),I=1,14)
00250        1003 FORMAT(15X,13A6,1A2)
00251        1004 WRITE(6,1005) TMIN, TMAX, DELTAT, FREQCY, ADFLAG, MULTIP, PRINT,
00251            $PLOT, COMENT
00272        1005 FORMAT(1H0,45X,17HPARAMETER LISTING                         /33X
00272            $,31HMINIMUM TIME (D.H.M.S)          = 3F3.0,F4.1            /33X
00272            $,31HMAXIMUM TIME (D.H.M.S)          = 3F3.0,F4.1            /33X
00272            $,31HTIME INCREMENT (SECONDS)        = F13.1                 /33X
00272            $,31HFREQUENCY FLAG (-1,1)           = I13                   /33X
00272            $,31HASCENT/DESCENT FLAG (-1,1)      = I13                   /33X
00272            $,31HMULTI-PATH FLAG (0,1)           = I13                   /33X
00272            $,31HPRINT FLAG (0,1)                = I13                   /33X
00272            $,31HPLOT FLAG (0,1)                 = I13                   /33X
00272            $,31HCOMMENT/DESCRIPTION CARDS     . = I13)
00273             WRITE(6,1001)
00275             IF(COMENT .EQ. 0) GO TO 1007
00277             DO 1006 J = 1, COMENT
00302        1006 WRITE(6,1003) (TITLE(I,J),I=1,14)
00311        1007 WRITE(6,1008)
00313        1008 FORMAT(1H0,109(1H-),/7X,4HTIME,10X,5HRANGE,6X,9HG-PRODUCT,5X,14HCS
00313            $M GAINS (DB),8X,14HLEM GAINS (DB),5X,23HPOLARIZATION GAINS (DB)/4X
00313            $,9H(D.H.M.S),9X,4H(NM),8X,4H(DB),7X,3(6HDIRECT,4X,8HSPECULAR,4X)/1
00313            $X,109(1H-))
00313       C
00313       C *** POSITION DRUMS FOR THIS DATA CASE
00314             N = FREQCY + 1
00315             NCSM = N / 2 + 1
00316             IF(N .EQ. 0) GO TO 96
00320             DO 97 I = 1, N
00323          97 READ(1)
```

```
00326        96 N = 2 * FREQCY + ADFLAG + 3
00327           NLEM = N / 2 + 1
00330           LEMTAP = 2 + N/6
00331           N = MOD(N,6)
00332           IF(N .EQ. 0) GO TO 94
00334           DO 95 I = 1, N
00337        95 READ(2)
00342        94 CONTINUE
00342     C
00342     C *** COMPUTE TIME LIMITS (IN SECONDS) AND SET FREQUENCY
00343           VMIN = 0.0
00344           VMAX = 0.0
00345           DO 104 I = 1, 4
00350           VMIN = VMIN + TIME(I)*TMIN(I)
00351       104 VMAX = VMAX + TIME(I)*TMAX(I)
00353           IF(FREQCY) 105,105,106
00356       105 FREQ = 259.7
00357           GO TO 107
00360       106 FREQ = 296.8
00361       107 FREQRD = FREQ * 1.0E+06 * TWOPI
00361     C
00361     C *** SCAN TRAJECTORY TAPE AND COMPUTE LOOK ANGLES
00362           NPOINT = 1
00363       300 READ(7) (ARRAY(I,NPOINT),I=1,7),                                *NEW
00363          $(ARRAY(I,NPOINT),I=8,14,3),                                      *NEW
00363          $(ARRAY(I,NPOINT),I=9,15,3),                                      *NEW
00363          $(ARRAY(I,NPOINT),I=10,16,3),                                     *NEW
00363          $(ARRAY(I,NPOINT),I=17,23,3),                                     *NEW
00363          $(ARRAY(I,NPOINT),I=18,24,3),                                     *NEW
00363          $(ARRAY(I,NPOINT),I=19,25,3)                                      *NEW
00421           IF(ARRAY(1,NPOINT) .LT. VMIN) GO TO 300                          **-1
00423           IF(ARRAY(1,NPOINT) .GT. VMAX) GO TO 999
00425           CALL REFLKT(ARRAY(8,NPOINT),ARRAY(2,NPOINT),ARRAY(17,NPOINT),
00425          $ARRAY(5,NPOINT),UNITY,ZERO,RO,ACURCY,ARRAY(27,NPOINT),IFLAG)
00426           IFLAG = IFLAG + 5 * MULTIP
00427           ARRAY(26,NPOINT) = IFLAG
00430           GO TO (303,304,304,303,303,302,304,304,303,304), IFLAG
00431       302 CALL LOOKCM(ARRAY(2,NPOINT),ARRAY(27,NPOINT),ARRAY(8,NPOINT),
00431          $ARRAY(32,NPOINT),ARRAY(33,NPOINT))
00432           CALL LOOKLM(ARRAY(5,NPOINT),ARRAY(27,NPOINT),ARRAY(17,NPOINT),
00432          $ARRAY(36,NPOINT),ARRAY(37,NPOINT))
```

```
00433         303 CALL LOOKCM(ARRAY(2,NPOINT),ARRAY(5,NPOINT),ARRAY(8,NPOINT),
00433            $ARRAY(30,NPOINT),ARRAY(31,NPOINT))
00434             CALL LOOKLM(ARRAY(5,NPOINT),ARRAY(2,NPOINT),ARRAY(17,NPOINT),
00434            $ARRAY(34,NPOINT),ARRAY(35,NPOINT))
00435         304 VMIN = ARRAY(1,NPOINT) + DELTAT
00436             NPOINT = NPOINT + 1
00437             IF(VMIN .GT. VMAX) GO TO 999
00441             IF(NPOINT .LE. 200) GO TO 300
00443             WRITE(6,93)
00445          93 FORMAT(1H0,86HNUMBER OF TRAJECTORY POINTS EXCEEDS 200.  ONLY THE F
00445            $IRST 200 POINTS WILL BE PROCESSED.)
00446         999 NPOINT = NPOINT - 1
00446       C
00446       C *** SCAN ANTENNA PATTERNS AND PROCESS EACH DATA POINT
00447             DO 501 III = 1, 2
00452             READ(1) CSMPAT
00460             DO 500 JJJ = 1, 2
00463             READ(LEMTAP) LEMPAT
00463       C
00463       C *** PRINT LINK BEING SIMULATED
00471             LINK = 2*(III-1)+JJJ
00472             WRITE(6,26) LINK, III, JJJ
00477          26 FORMAT(1H0,33X,9HLINK NO. ,I1,15H.  CSM ANTENNA ,I1,14H, LEM ANTEN
00477            $NA ,I1,1H.)
00500             DO 703 INDEX = 1, NPOINT
00500       C
00500       C
00500       C *** CONVERT TIME TO (D.H.M.S)
00503             TMIN(4) = ARRAY(1,INDEX)
00504             DO 27 I = 1, 3
00507             TMIN(I) = AINT(TMIN(4)/TIME(I))
00510          27 TMIN(4) = TMIN(4) - TMIN(I)*TIME(I)
00510       C
00510       C *** TRANSFER IF NO CALCULATIONS ARE TO BE EXECUTED THIS TIME-POINT
00512             IFLAG = ARRAY(26,INDEX)
00513             GO TO (350,450,450,350,350,350,450,450,350,450), IFLAG
00513       C
00513       C *** PERFORM DIRECT PATH CALCULATIONS
00513       C *** COMPUTE THE RANGE BETWEEN THE CSM AND LEM, R
00514         350 CALL VECSUB(ARRAY(2,INDEX),ARRAY(5,INDEX),3,RLCB)
00515             CALL VECMAG(RLCB,3,R)
```

```
00515    C
00515    C *** COMPUTE POLARIZATION ANGLE ALONG DIRECT PATH, RHO
00516          CALL POLANG(ARRAY(30,INDEX),ARRAY(31,INDEX),ARRAY(34,INDEX),
00516         $ARRAY(35,INDEX),ARRAY(8,INDEX),ARRAY(17,INDEX),RHO)
00516    C
00516    C *** FIND DIRECT PATH ANTENNA PATTERN DATA
00517          IF(ARRAY(30,INDEX) .LT. 0.034903 .OR. ARRAY(30,INDEX) .GT. 6.2482)        *NEW
00517         $ARRAY(30,INDEX) = 0.034923                                                 *NEW
00521          IF(ARRAY(31,INDEX) .LT. 0.034903 .OR. ARRAY(31,INDEX) .GT. 6.2482)        *NEW
00521         $ARRAY(31,INDEX) = 0.034923                                                 *NEW
00523          ITHETA = (IFIX(ARRAY(30,INDEX) * 57.296) + 1) / 2
00524          IPHI = (IFIX(ARRAY(31,INDEX) * 57.296) + 1) / 2
00525          ICODED = CSMPAT(ITHETA,IPHI)
00526          DO 91 I = 1, 6
00531          J = 3*(III-1) + (I+1)/2
00532       91 PATERN(I,1) = GNREFC(J,NCSM) - FLD(36-6*I,6,ICODED)
00534          IF(ARRAY(34,INDEX) .LT. 0.034903 .OR. ARRAY(34,INDEX) .GT. 6.2482)        *NEW
00534         $ARRAY(34,INDEX) = 0.034923                                                 *NEW
00536          IF(ARRAY(35,INDEX) .LT. 0.034903 .OR. ARRAY(35,INDEX) .GT. 6.2482)        *NEW
00536         $ARRAY(35,INDEX) = 0.034923                                                 *NEW
00540          ITHETA = (IFIX(ARRAY(34,INDEX) * 57.296) + 1) / 2
00541          IPHI = (IFIX(ARRAY(35,INDEX) * 57.296) + 1) / 2
00542          ICODED = LEMPAT(ITHETA,IPHI)
00543          DO 90 I = 1, 6
00546          J = 3*(JJJ-1) + (I+1)/2
00547       90 PATERN(I,3) = GNREFL(J,NLEM) - FLD(36-6*I,6,ICODED)
00547    C
00547    C *** COMPUTE POLARIZATION ELLIPSE TILT ANGLES, TAU1 AND TAU2, ALONG DIRECT PATH
00551          CALL TILT(PATERN(1,1),PATERN(2,1),PATERN(3,1),PATERN(4,1),TAU1)
00552          CALL TILT(PATERN(1,3),PATERN(2,3),PATERN(3,3),PATERN(4,3),TAU2)
00552    C
00552    C *** COMPUTE COORDINATE SYSTEM MIS-ALIGNMENT ALONG DIRECT PATH, BETA
00553          CALL MISALN(TAU1,TAU2,RHO,BETA)
00553    C
00553    C *** COMPUTE ANTENNA GAINS, G1, G2, AND AXIAL RATIOS, RR1, RR2, ALONG DIRECT P
00554          CALL GAINRR(PATERN(5,1),PATERN(6,1),G1,RR1)
00555          CALL GAINRR(PATERN(5,3),PATERN(6,3),G2,RR2)
00556          G1DB = DB(G1)
00557          G2DB = DB(G2)
00557    C
00557    C *** COMPUTE POLARIZATION LOSS ALONG DIRECT PATH, LP
```

```
00560            CALL POLAR(RR1,RR2,BETA,LP)
00561            LPDB = DB(LP)
00561      C
00561      C *** TRANSFER IF MULTI-PATH CALCULATIONS ARE NOT REQUIRED
00562            IF(IFLAG .EQ. 6) GO TO 400
00564            K = 0.0
00565            GO TO 401
00565      C
00565      C *** PERFORM MULTIPATH CALCULATIONS
00565      C *** COMPUTE SPECULAR PATH LENGTHS R1S, R2S, RS, DELTAR AND THE ANGLE PSI
00566        400 CALL VECSUB(ARRAY(5,INDEX),ARRAY(27,INDEX),3,RSLB)
00567            CALL VECSUB(ARRAY(2,INDEX),ARRAY(27,INDEX),3,RSCB)
00570            CALL VECMAG(RSCB,3,R1S)
00571            CALL VECMAG(RSLB,3,R2S)
00572            RS = R1S + R2S
00573            DELTAR = RS - R
00574            CALL VECDOT(ARRAY(27,INDEX),RSLB,3,A)
00575            PSI = HALFPI - ACOS(A / (RO * R2S))
00575      C
00575      C *** COMPUTE MULTIPATH POLARIZATION ANGLES, RHO1S AND RHO2S
00576            CALL POLARS(ARRAY(2,INDEX),ARRAY(5,INDEX),ARRAY(27,INDEX),ARRAY(8,
00576           $INDEX),ARRAY(17,INDEX),PSI,ARRAY(32,INDEX),ARRAY(33,INDEX),ARRAY(3
00576           $6,INDEX),ARRAY(37,INDEX),RHO1S,RHO2S)
00576      C
00576      C *** FIND SPECULAR PATH ANTENNA PATTERN DATA
00577            IF(ARRAY(32,INDEX) .LT. 0.034903 .OR. ARRAY(32,INDEX) .GT. 6.2482)        *NEW
00577           $ARRAY(32,INDEX) = 0.034923                                                *NEW
00601            IF(ARRAY(33,INDEX) .LT. 0.034903 .OR. ARRAY(33,INDEX) .GT. 6.2482)        *NEW
00601           $ARRAY(33,INDEX) = 0.034923                                                *NEW
00603            ITHETA = (IFIX(ARRAY(32,INDEX) * 57.296) + 1) / 2
00604            IPHI = (IFIX(ARRAY(33,INDEX) * 57.296) + 1) / 2
00605            ICODED = CSMPAT(ITHETA,IPHI)
00606            DO 89 I = 1, 6
00611            J = 3*(III-1) + (I+1)/2
00612         89 PATERN(I,2) = GNREFC(J,NCSM) - FLD(36-6*I,6,ICODED)
00614            IF(ARRAY(36,INDEX) .LT. 0.034903 .OR. ARRAY(36,INDEX) .GT. 6.2482)        *NEW
00614           $ARRAY(36,INDEX) = 0.034923                                                *NEW
00616            IF(ARRAY(37,INDEX) .LT. 0.034903 .OR. ARRAY(37,INDEX) .GT. 6.2482)        *NEW
00616           $ARRAY(37,INDEX) = 0.034923                                                *NEW
00620            ITHETA = (IFIX(ARRAY(36,INDEX) * 57.296) + 1) / 2
00621            IPHI = (IFIX(ARRAY(37,INDEX) * 57.296) + 1) / 2
```

```
00622          ICODED = LEMPAT(ITHETA,IPHI)
00623          DO 88 I = 1, 6
00626          J = 3*(JJJ-1) + (I+1)/2
00627       88 PATERN(I,4) = GNREFL(J,NLEM) - FLD(36-6*I,6,ICODED)
00627    C
00627    C *** COMPUTE POLARIZATION ELLIPSE TILT ANGLE OF RECEIVING ANTENNA, TAU2S
00631          CALL TILT(PATERN(1,2),PATERN(2,2),PATERN(3,2),PATERN(4,2),TAU2S(1)
00631         $)
00632          CALL TILT(PATERN(1,4),PATERN(2,4),PATERN(3,4),PATERN(4,4),TAU2S(2)
00632         $)
00632    C
00632    C *** COMPUTE POLARIZATION ELLIPSE TILT ANGLE OF TRANSMITTING ANTENNA, TAU1S
00633          TAU1S(1) = TAU2S(1) - RHO1S
00634          TAU1S(2) = TAU2S(2) - RHO2S
00635          DO 351 I = 1, 2
00640      351 IF(TAU1S(I) .LT. 0.0) TAU1S(I) = TAU1S(I) + PI
00640    C
00640    C *** COMPUTE MULTI-PATH REFLECTION COEFFICIENTS, RA AND RB
00643          AA = CMPLX(EPSLNR,-SIGMA/(FREFQRD * EPSLNO))
00644          BB = CSQRT(AA - COS(PSI)**2)
00645          C = SIN(PSI)
00646          A = AA * C
00647          RA = (C - BB) / (C + BB)
00650          RB = (A - BB) / (A + BB)
00650    C
00650    C *** COMPUTE CSM AND LEM ANTENNA GAINS AND AXIAL RATIOS ALONG SPECULAR PATH
00651          CALL GAINRR(PATERN(5,2),PATERN(6,2),G1S,RR1S)
00652          CALL GAINRR(PATERN(5,4),PATERN(6,4),G2S,RR2S)
00653          G1SDB = DB(G1S)
00654          G2SDB = DB(G2S)
00654    C
00654    C *** COMPUTE POLARIZATION RATIO OF INCIDENT WAVE, P1S
00655          ALPHA(1) = ATAN2(1.0,RR1S) * 2.0
00656          ALPHA(2) = ATAN2(1.0,RR2S) * 2.0
00657          DO 352 I = 1, 2
00662      352 P1S(I) = TAN(0.5 * ACOS(COS(ALPHA(I)) * COS(2.0 * TAU1S(I)))) *
00662         $CEXP(CMPLX(0.0,ATAN2(TAN(ALPHA(I)),SIN(2.0 * TAU1S(I)))))
00662    C
00662    C *** COMPUTE DEPOLARIZATION FACTOR, Q, AND POLARIZATION RATIO OF REFLECTED WAVE
00664          Q = RA / RB
00665          DO 353 I = 1, 2
```

```
  353 P2S(I) = Q * P1S(I)
C
C *** COMPUTE TILT ANGLE, TAUS, AND AXIAL RATIO, RRS, OF REFLECTED WAVE
      DO 354 I = 1, 2
      A = 2.0 * ATAN(CABS(P2S(I)))
      B = ATAN2(AIMAG(P2S(I)),REAL(P2S(I)))
      RRS(I) = 1.0 / TAN(0.5 * ASIN(SIN(A) * SIN(B)))
  354 TAUS(I) = 0.5 * ATAN(TAN(A) * COS(B))
C
C *** COMPUTE THE COORDINATE SYSTEM MIS-ALIGNMENT ALONG THE SPECULAR PATH, BETAS
      CALL MISALN(TAU2S(1),TAUS(1),RHO2S,BETAS(1))
      CALL MISALN(TAU2S(2),TAUS(2),RHO1S,BETAS(2))
C
C *** COMPUTE SPECULAR POLARIZATION LOSS, LPS
      CALL POLAR(RRS(1),RR2S,BETAS(1),LPS(1))
      CALL POLAR(RRS(2),RR1S,BETAS(2),LPS(2))
      LPSDB = DB(AMIN1(LPS(1),LPS(2)))
C
C *** COMPUTE THE LUNAR REFLECTION LOSS, LR
      A = CABS(RA) **2
      B = CABS(RB) **2
      DO 355 I = 1, 2
      C = CABS(P1S(I)) **2
  355 LR(I) = (B + A * C) / (1.0 + C)
C
C *** COMPUTE MULTI-PATH DIVERGENCE FACTOR, D
      A = 2.0 * R1S * R2S
      B = SIN(PSI)
      D = 1.0 / SQRT(1.0 + A * (1.0 + B**2) / (RS * RO * B) + (A / RO)**
     $   2 / RS)
C
C *** COMPUTE MULTIPATH REFLECTION FACTOR, K
      A = R * G1S * G2S * D / (RS * G1 * G2)
      ALPHA(1) = A * LR(1) * SQRT(LPS(1) / LP)
      ALPHA(2) = A * LR(2) * SQRT(LPS(2) / LP)
      C = COS(AMOD(TWOPIC*FREQ*DELTAR,TWOPI))
      IF(ABS(ALPHA(1) + C) .LT. ABS(ALPHA(2) + C)) GO TO 356
      K = ALPHA(2)
      GO TO 401
  356 K = ALPHA(1)
C
```

```
00732   C *** COMPUTE GAIN PRODUCT GP
00733     401 GP = LP * G1 * G2 * (1.0 + K * (K + 2.0 * C))
00734         GPDB = DB(GP)
00734   C
00734   C *** PRINT STANDARD OUTPUT
00735         GPDB = G1DB + G2DB + LPDB
00736         IF(IFLAG .EQ. 6) GO TO 1009
00740         IF(IFLAG .EQ. 9) WRITE(6,998)
00743     998 FORMAT(1X,107HTHE FOLLOWING TIME-POINT HAS THE LM AND CSM LYING AL
00743        $ONG A LUNAR RADIUS.  THE EXISTENCE OF A 'BOUNCE-POINT' /103H IN SU
00743        $CH A SITUATION IS MORE ACADEMIC THAN ACTUAL.  HENCE, ONLY DIRECT P
00743        $ATH CALCULATIONS WERE EXECUTED.)
00744         WRITE(6,1000) TMIN, R, GPDB, G1DB, G2DB, LPDB
00757    1000 FORMAT(1X,3F3.0,F4.1,6X,F6.2,7X,F7.2,4X,3(F7.2,15X))
00760         GO TO 1011
00761    1009 WRITE(6,1010) TMIN, R, GPDB, G1DB, G1SDB, G2DB, G2SDB, LPDB, LPSDB
00777    1010 FORMAT(1X,3F3.0,F4.1,6X,F6.2,7X,F7.2,4X,3(F7.2,3X,F7.2,5X))
00777   C
00777   C *** PRINT OPTIONAL OUTPUT
01000    1011 IF(PRINT .NE. 0) WRITE(6,87) (ARRAY(I,INDEX),I=1,37), PATERN, R,
01000        $RHO, TAU1, TAU2, BETA, G1, G2, RR1, RR2, LP, R1S, R2S, RS, DELTAR,
01000        $PSI, RHO1S, RHO2S, TAU1S, TAU2S, RA, RB, G1S, G2S, RR1S, RR2S,
01000        $P1S, P2S, Q, TAUS, RRS, BETAS, LPS, LR, D, K, GP
01112      87 FORMAT(9H0TIME   =1PE10.3/9H ROCB   =E10.3,2(1H,E10.3)/9H ROLB   =
01112        $E10.3,2(1H,E10.3)/9H ACB    =E10.3,8(1H,E10.3)/9H ALB    =E10.3,8(
01112        $1H,E10.3)/9H IFLAG  =E10.3/9H ROSB   =E10.3,2(1H,E10.3)/9H THTAC
01112        $=E10.3/9H PHIC    =E10.3/9H THTACS =E10.3/9H PHICS  =E10.3/9H THTAL
01112        $  =E10.3/9H PHIL    =E10.3/9H THTALS =E10.3/9H PHILS  =E10.3/9H PAT
01112        $ERN =3(6(E10.3,1H,)/9X)5(E10.3,1H,)E10.3/9H R       =E10.3/9H RHO
01112        $  =E10.3/9H TAU1    =E10.3/9H TAU2    =E10.3/9H BETA    =E10.3/9H G1
01112        $     =E10.3/9H G2      =E10.3/9H RR1      =E10.3/9H RR2     = E10.3/9H
01112        $LP      =E10.3/9H R1S     =E10.3/9H R2S     =E10.3/9H RS      =E10.3/9
01112        $H DELTAR =E10.3/9H PSI     =E10.3/9H RHO1S  =E10.3/9H RHO2S  =E10.3
01112        $/9H TAU1S  =E10.3,1H,E10.3/9H TAU2S  =E10.3,1H,E10.3/9H RA      =1H
01112        $(E9.3, 6H) + J(E9.3, 1H)/9H RB      =1H(E9.3, 6H) + J(E9.3, 1H)/9H
01112        $G1S     =E10.3/9H G2S     =E10.3/9H RR1S    =E10.3/9H RR2S    =E10.3/9
01112        $H P1S     =1H(E9.3, 6H) + J(E9.3, 2H),     2H (E9.3, 6H) + J(E9.3,1H
01112        $)/9H P2S     =1H(E9.3, 6H) + J(E9.3, 2H),   2H (E9.3, 6H) + J(E9.3
01112        $,1H)/9H Q       =1H(E9.3, 6H) + J(E9.3, 1H)/9H TAUS    =E10.3,1H,E10
01112        $.3/9H RRS      =E10.3,1H,E10.3/9H BETAS  =E10.3,1H,E10.3/9H LPS     =
01112        $E10.3,1H,E10.3/9H LR      =E10.3,1H,E10.3/9H D       =E10.3/9H K
```

```
01112          $  =E10.3/9H GP      =E10.3/)
01112      C
01112      C *** WRITE PLOT TAPE
01113            IF(PLOT .NE. 0) WRITE(8) (ARRAY(I,INDEX),I=1,37), R, RHO, TAU1,
01113           $TAU2, BETA, G1, G1DB, G2, G2DB, RR1, RR2, LP, LPDB, R1S, R2S, RS,
01113           $DELTAR, PSI, RHO1S, RHO2S, TAU1S, TAU2S, RA, RB, G1S, G1SDB, G2S,
01113           $G2SDB, RR1S, RR2S, P1S, P2S, Q, TAUS, RRS, BETAS, LPS, LPSDB, LR,
01113           $D, K, GP, GPDB
01230            GO TO 703
01230      C
01230      C *** NO CALCULATIONS HAVE BEEN EXECUTED - PRINT APPROPRIATE MESSAGE
01231        450 IF((IFLAG .EQ. 2) .OR. (IFLAG .EQ. 7)) GO TO 451
01233            IF((IFLAG .EQ. 3) .OR. (IFLAG .EQ. 8)) GO TO 452
01235            WRITE(6,453) TMIN
01243        453 FORMAT(1X,3F3.0,F4.1,5X,62HTHE 'BOUNCE-POINT' WAS NOT DETERMINED T
01243           $O THE DESIRED ACCURACY.)
01244            GO TO 703
01245        451 WRITE(6,454) TMIN
01253        454 FORMAT(1X,3F3.0,F4.1,5X74HLM AND CSM ARE ON OPPOSITE SIDES OF THE
01253           $LUNAR HORIZON.  TRY SMOKE SIGNALS.)
01254            GO TO 703
01255        452 WRITE(6,455) TMIN
01263        455 FORMAT(1X,3F3.0,F4.1,5X,78HLM AND CSM ARE ON DIAMETRICALLY OPPOSIT
01263           $E SIDES OF THE MOON-- SORRY 'BOUT THAT.)
01264        703 CONTINUE
01264      C
01264      C *** END-FILE PLOT TAPE
01266            IF(PLOT .NE. 0) END FILE 8
01270        500 CONTINUE
01270      C
01270      C *** RE-POSITION LEM TAPE
01272            BACK SPACE LEMTAP
01273            BACK SPACE LEMTAP
01274        501 CONTINUE
01276            GO TO 100
01277            END

    END OF UNIVAC 1108 FORTRAN V COMPILATION.      0 *DIAGNOSTIC* MESSAGE(S)
  VHFASP           SYMBOLIC                        02 JUL 68  12&15&01   0  01336740   14
  VHFASP   CODE    RELOCATABLE                     02 JUL 68  12&15&01   1  01353516   60
                                                                         0  01353612   14
```

(this page intentionally left blank)

```
& FOR,* DXO,DXO                                                                                      17&07&21
UNIVAC 1108 FORTRAN V LEVEL   2206 0018 F5018H.
THIS COMPILATION WAS DONE ON 14 JAN 69 AT 17&07&21

- 2

    DR1(THETA) = (RO + H1)**2 * SIN(2.0*THETA) / (2.0*SQRT((( RO +
   1            H1)*COS(THETA) )**2 - H1 * (2.0*RO+H1) )) - (RO +
   2            H1)*SIN(THETA)

  FUNCTION DXO          ENTRY POINT 000065

  STORAGE USED (BLOCK, NAME, LENGTH)

       0001    *CODE    000074
       0000    *DATA    000015
       0002    *BLANK   000000
       0003    TEMPRY   000004


  EXTERNAL REFERENCES (BLOCK, NAME)

       0004    R1
       0005    SIN
       0006    COS
       0007    SQRT
       0010    NERR3$


  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

  0000 R 000000 DXO        0003 R 000000 H1        0003 R 000001 RO        0004 R 000000 R1        0003
  0003 R 000003 Y2
```

```
00101            FUNCTION DXO    (THETA)
00103            COMMON /TEMPRY/ H1, RO, X2, Y2
00104            DR1(THETA) = (RO + H1)**2 * SIN(2.0*THETA) / (2.0*SQRT((( RO +      *NEW
00104           1                H1)*COS(THETA) )**2 - H1 * (2.0*RO+H1) )) - (RO +   *NEW
00104           2                H1)*SIN(THETA)                                       *NEW
00105            DXO         = R1(THETA) * SIN(THETA) - DR1(THETA) * COS(THETA)
00106            RETURN
00107            END

     END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
   DXO               SYMBOLIC                          02 JUL 68   12&15809   0  01372552   14
   DXO       CODE    RELOCATABLE                       02 JUL 68   12&15809   1  01372660   24
                                                                              0  01372710   14
```

```
& FOR,* DYO,DYO                                                                    17&07&22
UNIVAC 1108 FORTRAN V LEVEL    2206 0018 E5018H
THIS COMPILATION WAS DONE ON 14 JAN 69 AT 17&07&22

- 2

     DR1(THETA) = (RO + H1)**2 * SIN(2.0*THETA) / (2.0*SQRT((( RO +
   1             H1)*COS(THETA) )**2 - H1 * (2.0*RO+H1) )) - (RO +
   2             H1)*SIN(THETA)

  FUNCTION DYO          ENTRY POINT 000066

  STORAGE USED (BLOCK, NAME, LENGTH)

       0001    *CODE    000075
       0000    *DATA    000015
       0002    *BLANK   000000
       0003    TEMPRY   000004

  EXTERNAL REFERENCES (BLOCK, NAME)

       0004    R1
       0005    COS
       0006    SIN
       0007    SQRT
       0010    NERR3$

  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

  0000 R 000000 DYO        0003 R 000000 H1        0003 R 000001 RO        0004 R 000000 R1        0003
  0003 R 000003 Y2
```

```
00101             FUNCTION DYO    (THETA)
00103             COMMON /TEMPRY/ H1, RO, X2, Y2
00104             DRI(THETA) = (RO + H1)**2 * SIN(2.0*THETA) / (2.0*SQRT((( RO +        *NEW
00104            1                  H1)*COS(THETA) )**2 - H1 * (2.0*RO+H1) )) - (RO +   *NEW
00104            2                  H1)*SIN(THETA)                                       *NEW
00105             DYO          = R1(THETA) * COS(THETA) + DR1(THETA) * SIN(THETA)
00106             RETURN
00107             END

     END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
   DYO                SYMBOLIC                                02 JUL 68  12&15&11   0  01373226   14
   DYO       CODE     RELOCATABLE                             02 JUL 68  12&15&11   1  01373334   24
                                                                                    0  01373364   14
```

```
& FOR,* DPHI,DPHI                                                        17&07&23
UNIVAC 1108 FORTRAN V LEVEL    2206 0018 F5018H .
THIS COMPILATION WAS DONE ON 14 JAN 69 AT 17&07&23

- 2

      DR1(THETA) = (RO + H1)**2 * SIN(2.0*THETA) / (2.0*SQRT((( RO +
    1              H1)*COS(THETA) )**2 - H1 * (2.0*RO+H1) )) - (RO +
    2              H1)*SIN(THETA)

   FUNCTION DPHI          ENTRY POINT 000105

   STORAGE USED (BLOCK, NAME, LENGTH)

         0001    *CODE    000114
         0000    *DATA    000017
         0002    *BLANK   000000
         0003    TEMPRY   000004

   EXTERNAL REFERENCES (BLOCK, NAME)

         0004    R1
         0005    COS
         0006    SIN
         0007    SQRT
         0010    NERR3$

   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

   0000 R 000000 DPHI       0003 R 000000 H1       0003 R 000001 RO       0004 R 000000 R1       0003
   0003 R 000003 Y2
```

```
00101          FUNCTION DPHI   (THETA)
00103          COMMON /TEMPRY/ H1, RO, X2, Y2
00104          DR1(THETA) = (RO + H1)**2 * SIN(2.0*THETA) / (2.0*SQRT((( RO +        *NEW
00104         1              H1)*COS(THETA) )**2 - H1 * (2.0*RO+H1) )) - (RO +        *NEW
00104         2              H1)*SIN(THETA)                                            *NEW
00105          DPHI        = (R1(THETA) * COS(THETA) + DR1(THETA) * SIN(THETA)) /
00105         $              SQRT((RO + R1(THETA) * SIN(THETA)) * (RO - R1(THETA)
00105         $              * SIN(THETA)))
00106          RETURN
00107          END

     END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
  DPHI              SYMBOLIC                       02 JUL 68   12&15&15   0  01374446   14
  DPHI     CODE     RELOCATABLE                    02 JUL 68   12&15&15   1  01374610   24
                                                                          0  01374640   14
```

9. APPENDIX B -- Tape Formats

The only input tapes required by the VHF Antenna Simulation Program are the trajectory tape and the antenna patterns tape. Normally, the antenna patterns tape is included as a second file on the production PCF tape so that the user is not aware of its existence. A trajectory tape must be mounted each time the program is executed. Both of these tapes are FORTRAN binary tapes operated in the 800BPI density mode.

9.1 FORMAT OF TRAJECTORY TAPE

The VHFASP trajectory tape consists of one file. Each record in this file corresponds to precisely one time-point. Each record consists of twenty-five words as described below. The records are assumed to be ordered by increasing times.

Word(s)	Description
1	Trajectory time (in seconds)
2 - 4	Position vector of CSM in basic coordinates, $\bar{R}_{oc} = \begin{bmatrix} \text{Word 2} \\ \text{Word 3} \\ \text{Word 4} \end{bmatrix}$.
5 - 7	Position vector of LM in basic coordinates, $\bar{R}_{ol} = \begin{bmatrix} \text{Word 5} \\ \text{Word 6} \\ \text{Word 7} \end{bmatrix}$.
8 - 16	Direction cosine matrix of CSM, $A_{cb} = \begin{bmatrix} \text{Word 8} & \text{Word 9} & \text{Word 10} \\ \text{Word 11} & \text{Word 12} & \text{Word 13} \\ \text{Word 14} & \text{Word 15} & \text{Word 16} \end{bmatrix}$.
17 - 25	Direction cosine matrix of LM, $A_{lb} = \begin{bmatrix} \text{Word 17} & \text{Word 18} & \text{Word 19} \\ \text{Word 20} & \text{Word 21} & \text{Word 22} \\ \text{Word 23} & \text{Word 24} & \text{Word 25} \end{bmatrix}$.

```
c
c
1
∇△ASG△D=(old trajectory tape number)
∇△ASG△E=SAVE (new trajectory tape to be saved)
∇△ASG△I,J,L (three FASTRANDS required)
∇△XQT△UTILT2
[input data
∇△EOF
```

10.3 UTILT3

The Utility Program UTILT3 allows the user to input trajectory data for the VHFASP program on cards rather than from a trajectory tape. Input is via the FORTRAN namelist "$INPUT." The input parameters are T (time in seconds), RØCB (position vector of CSM), RØLB (position vector of LM), ACB (direction cosine matrix of CSM) and ALB (direction cosine matrix of LM). Input is terminated by setting all parameters to zero.

Unit "E" (FORTRAN I. D. Number 7) must be assigned to a FASTRAND file. UTILT3 must be executed immediately before VHFASP. The required control cards are indicated below.

```
c
c
1
∇△ASG△E
∇△XQT△UTILT3
[input data
∇△XQT△HV014B
[input data
∇△EOF
```

D. A. S.

TRW
SYSTEMS GROUP

INTEROFFICE CORRESPONDENCE

5512.30-024

TO: File

CC: Task E-78 File

DATE: 14 February 1969

SUBJECT: Modifications to the VHF Antenna Simulation Program (VHFASP), HV014B

FROM: C. A. Argila

BLDG.	ROOM	EXT.
H1	2061	2503

A minor modification has been made to the VHF Antenna Simulation Program (VHFASP). The attached pages have been prepared to update the HV014B User's Manual (HCC Report No. 3145.30-012) to include these modifications. Pages 7-2 (Mod I) to 7-15b (Mod I) should be replaced by pages 7-2 (Mod II) to 7-15b (Mod II).

C. A. Argila
Electrical and Electronics Group

Approved: W. P. Bennett, Head
Subsystems Analysis Section

CAA/idf

Attachment

7.1 The Main Driver, VHFASP

```
& FOR,* VHFASP,VHFASP                                          15&25&46
UNIVAC 1108 FORTRAN V LEVEL   2206 0018 F5018H
THIS COMPILATION WAS DONE ON 13 FEB 69 AT 15&25&46

-416,416

   MAIN PROGRAM

   STORAGE USED (BLOCK, NAME, LENGTH)

        0001    *CODE    003754
        0000    *DATA    117155
        0002    *BLANK   000000
        0003    DATA     000031
        0004    GNREF    000044

   EXTERNAL REFERENCES (BLOCK, NAME)

        0005    INITAL
        0006    REFLKT
        0007    LOOKCM
        0010    LOOKLM
        0011    VECSUB
        0012    VECMAG
        0013    POLANG
        0014    TILT
        0015    MISALN
        0016    GAINRR
        0017    DB
        0020    POLAR
        0021    VECDOT
        0022    POLARS
        0023    NREW$
        0024    NRBU$
        0025    NIO1$
        0026    NIO2$
        0027    NWBU$
        0030    NRNL$
```

0031	NWDU$
0032	NRDU$
0033	NERR2$
0034	ACOS
0035	COS
0036	CSQRT
0037	SIN
0040	CDV$
0041	ATAN2
0042	TAN
0043	CEXP
0044	CABS
0045	ATAN
0046	ASIN
0047	SQRT
0050	NWEF$
0051	NBSP$
0052	NSTOP$

STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001	000113	100L	0000	116464	1000F	0000	116200	1001F	0001	003141	1002G	0000
0001	000227	1004L	0000	116211	1005F	0001	003154	1006G	0001	000311	1007L	0000
0001	003077	1009L	0000	116474	1010F	0001	003131	1011L	0000	116176	102F	0001
0001	003226	1033G	0001	003236	1037G	0001	003262	1051G	0001	003274	1055G	0001
0001	003310	1062G	0001	003320	1066G	0001	000415	107L	0001	003330	1072G	0001
0001	003350	1102G	0001	003374	1115G	0001	003457	1145G	0001	003467	1151G	0001
0001	003517	1165G	0001	003531	1171G	0001	003545	1176G	0001	003555	1202G	0001
0001	003575	1212G	0001	003607	1217G	0001	000022	122G	0001	003665	1236G	0001
0001	003715	1256G	0001	000033	130G	0001	000041	136G	0001	000046	142G	0001
0001	000065	156G	0001	000072	162G	0001	000103	170G	0001	000170	217G	0001
0001	000215	237G	0001	000222	243G	0001	000235	253G	0001	000242	257G	0000
0000	116143	28F	0001	000164	29L	0001	000277	300G	0001	000423	300L	0001
0001	000671	303L	0001	000304	304G	0001	000737	304L	0001	000331	321G	0001
0001	000374	346G	0001	001114	350L	0001	003015	356L	0001	000461	365G	0001
0001	000475	375G	0001	001526	400L	0001	000503	401G	0001	003017	401L	0001
0001	000517	411G	0001	000525	415G	0001	000772	450G	0001	003627	450L	0001
0001	003707	452L	0000	117013	453F	0000	117031	454F	0001	001000	454G	0000
0001	001067	461G	0001	001015	465G	0001	001050	501G	0001	001056	505G	0001

```
0001     001402 544G      0001     001746 607G      0001     002075 624G      0001     002157 636G      0001
0001     002455 666G      0001     002501 673G      0001     003721 703L      0001     002653 712G      0001
0001     003105 762G      0000     116505 87F       0000     116362 93F       0001     003370 94L       0001
0000     116416 998F      0001     000764 999L      0000 R 116067 A         0000 C 037525 AA        0003
0000 I 000000 ADFLAG      0000 R 037547 ALPHA     0000 R 037551 ARRAY     0000 R 116102 B         0000
0000 R 116054 BETA        0000 R 056121 BETAS     0000 R 116073 C         0000 I 000001 COMENT    0000
0000 R 116103 D           0017 R 000000 DB        0000 R 116066 DELTAR    0000 R 116022 DELTAT    0003
0003 R 000002 EPSLNO      0000 R 116035 FREQ      0000 I 037512 FREQCY    0000 R 116036 FREQRD    0004
0004 R 000014 GNREFL      0000 R 116104 GP        0000 R 116105 GPDB      0000 R 116055 G1        0000
0000 R 116074 G1S         0000 R 116100 G1SDB     0000 R 116057 G2        0000 R 116062 G2DB      0000
0000 R 116101 G2SDB       0003 R 000003 HALFPI    0000 I 116024 I         0000 I 116051 ICODED    0000
0000 I 116041 III         0000 I 116044 INDEX     0000 I 116050 IPHI      0000 I 116047 ITHETA    0000
0000 I 116026 J           0000 I 116042 JJJ       0000 R 037515 K         0000 I 056123 LEMPAT    0000
0000 I 116043 LINK        0000 R 037516 LP        0000 R 037517 LPDB      0000 R 037520 LPS       0000
0000 R 037523 LR          0000 I 116023 MULTIP    0000 I 116027 N         0000 I 116030 NCSM      0000
0000 I 116037 NPOINT      0000 R 115633 PATERN    0003 R 000004 PI        0000 I 037513 PLOT      0000
0000 R 116070 PSI         0000 C 037531 P1S       0000 C 037535 P2S       0000 C 037541 Q         0000
0000 C 037543 RA          0000 C 037545 RB        0000 R 116046 RHO       0000 R 116071 RHO1S     0000
0000 R 115663 RLCB        0000 R 115666 RRS       0000 R 116056 RR1       0000 R 116075 RR1S      0000
0000 R 116077 RR2S        0000 R 116065 RS        0000 R 115670 RSCB      0000 R 115673 RSLB      0003
0000 R 116063 R1S         0000 R 116064 R2S       0003 R 000006 SIGMA     0000 R 115676 TAUS      0000
0000 R 115700 TAU1S       0000 R 116053 TAU2      0000 R 115702 TAU2S     0003 R 000007 TIME      0000
0000 R 116012 TMAX        0000 R 116016 TMIN      0003 R 000013 TWOPI     0003 R 000014 TWOPIC    0003
0000 R 116034 VMAX        0000 R 116033 VMIN      0003 R 000026 ZERO
```

```
00100   C ..........................................................................
00100   C .                                                                          .
00100   C .   VHF ANTENNA SIMULATION PROGRAM (VHFASP)        C. ARGILA, 1968          .
00100   C .                                                                          .
00100   C .   DESCRIPTION-                                                           .
00100   C .      THE VHF ANTENNA SIMULATION PROGRAM SIMULATES THE CSM-LM VHF         .
00100   C .   COMMUNICATION LINK FOR ALL APOLLO MISSIONS.  IN THE STANDARD           .
00100   C .   CONFIGURATION VHFASP PRINTS VALUES OF CSM AND LM ANTENNA GAINS,        .
00100   C .   POLARIZATION LOSSES, CSM-LM RANGE AND GAIN PRODUCT AS FUNCTIONS OF     .
00100   C .   TIME.  IN AN OPTIONAL MODE EVERY VARIABLE INVOLVED IN THE SIMULATION   .
00100   C .   MAY BE PRINTED OUT.  A PLOT TAPE IS GENERATED FOR USE WITH THE TRWPLT  .
00100   C .   GENERAL PLOTTING PROGRAM.  THIS PROGRAM IS WRITTEN IN FORTRAN V FOR USE .
00100   C .   ON THE SRU 1108 SYSTEM.                                                .
```

```
C .                                                                         .
C .   INPUT-                                                                .
C .     CARD INPUT- CARD INPUT IS VIA THE FORTRAN NAMELIST '$INPUT.'  INPUT  .
C .   PARAMETERS ARE                                                        .
C .      FREQCY    FREQUENCY FLAG.  FREQCY=-1 INDICATES LOW FREQUENCY,       .
C .               FREQCY=+1 INDICATES HIGH FREQUENCY.  FREQCY INITIALIZED TO -1..
C .      ADFLAG    ASCENT/DESCENT FLAG.  ADFLAG=-1 INDICATES DESCENT CONDITION, .
C .               ADFLAG=+1 INDICATES ASCENT CONDITION.  ADFLAG INITIALIZED  .
C .               TO -1.                                                    .
C .      MULTIP    MULTI-PATH FLAG.  MULTIP=0 INDICATES OMIT MULTI-PATH      .
C .               CALCULATIONS, MULTIP=1 INDICATES MULTI-PATH CALCULATIONS  .
C .               ARE TO BE EXECUTED.  MULTIP INITIALIZED TO 1.             .
C .      PRINT     PRINT FLAG.  PRINT=0 INDICATES ONLY STANDARD PRINT OUTPUT IS .
C .               DESIRED.  PRINT=1 INDICATES OPTIONAL PRINT OUTPUT IS ALSO .
C .               DESIRED.  PRINT INITIALIZED TO 0.                         .
C .      PLOT      PLOT FLAG.  PLOT=0 INDICATES NO PLOT TAPE IS TO BE GENERATED,.
C .               PLOT=1 INDICATES PLOT TAPE IS TO BE GENERATED.  PLOT INITIAL- .
C .               IZED TO 1.                                                .
C .      TMIN      FOUR-POINT ARRAY REPRESENTING MINIMUM TIME TO BE CONSIDERED, .
C .               IN DAYS, HOURS, MINUTES AND SECONDS.  INITIALIZED TO 0,0,0,0. .
C .      TMAX      FOUR-POINT ARRAY REPRESENTING MAXIMUM TIME TO BE CONSIDERED, .
C .               IN DAYS, HOURS, MINUTES AND SECONDS.  INITIALIZED TO 0,0,0,0. .
C .      DELTAT    MINIMUM TIME (IN SECONDS) BETWEEN TIME-POINTS.  INITIALIZED .
C .               TO 0.0.                                                   .
C .     TAPE INPUT- A TRAJECTORY TAPE (MOUNTED ON UNIT E) AND A 'PACKED'    .
C .   ANTENNA PATTERNS TAPE ARE THE ONLY TAPE INPUTS REQUIRED.  THEESE TAPES .
C .   ARE IN THE FORMAT SPECIFIED IN THE HV014B USER'S MANUAL.              .
C .                                                                         .
C .   OUTPUT-                                                               .
C .      IN ADDITION TO THE STANDARD AND OPTIONAL PRINT OUTPUT, A PLOT TAPE .
C .   IN THE FORMAT REQUIRED BY 'TRWPLT' MAY ALSO BE GENERATED.             .
C .                                                                         .
C .   SUBROUTINE USAGE-                                                     .
C .      INITAL                                                             .
C .      REFLKT                                                             .
C .        VECTOR                                                           .
C .        GMPRD                                                            .
C .        GMTRA                                                            .
C .      LOOK                                                               .
C .        VECTOR                                                           .
C .        GMPRD                                                            .
```

```
00100     C .       POLANG                                                                  .
00100     C .         GMTRA                                                                 .
00100     C .         GMPRD                                                                 .
00100     C .       TILT                                                                    .
00100     C .       MISALN                                                                  .
00100     C .       GAINRR                                                                  .
00100     C .       DB                                                                      .
00100     C .       POLAR                                                                   .
00100     C .       POLARS                                                                  .
00100     C .         VECTOR                                                                .
00100     C .         GMPRD                                                                 .
00100     C .       DATA (BLOCK DATA SUBROUTINE)                                            .
00100     C .                                                                               .
00100     C .    NAMED COMMON USAGE-                                                        .
00100     C .       DATA                                                                    .
00100     C .       GNREF                                                                   .
00100     C .                                                                               .
00100     C .................................................................................
00100     C
00101            INTEGER ADFLAG, COMENT, CSMPAT(90,180), FREQCY, PLOT, PRINT
00103            REAL K, LP, LPDB, LPS(2), LPSDB, LR(2)
00104            COMPLEX AA, BB, P1S(2), P2S(2), Q, RA, RB
00105            DIMENSION ALPHA(2), ARRAY(37,200), BETAS(2), GNREFC(6,2), GNREFL(6
00105           $,4), LEMPAT(90,180), PATERN(6,4), RLCB(3), RRS(2), RSCB(3), RSLB(3
00105           $), TAUS(2), TAU1S(2), TAU2S(2), TIME(4), TITLE(14,5), TMAX(4), TMI
00105           $N(4), UNITY(3,3), ZERO(3)
00106            COMMON /DATA/ ACURCY, EPSLNR, EPSLNO, HALFPI, PI, RO, SIGMA, TIME,
00106           $TWOPI, TWOPIC, UNITY, ZERO
00107            COMMON /GNREF/ GNREFC, GNREFL
00110            NAME LIST /INPUT/ FREQCY, ADFLAG, COMENT, PRINT, PLOT, TMIN, TMAX,
00110           $DELTAT, MULTIP
00110     C
00110     C *** INITIALIZE TAPES
00111            REWIND 1
00112            REWIND 2
00113            REWIND 3
00114            REWIND 8
00114     C
00114     C *** DUMP ANTENNA PATTERNS ONTO HIGH SPEED DRUMS
00115            DO 99 I = 1, 4
00120            READ(27) CSMPAT
```

```
00126      99 WRITE(1) CSMPAT
00135         DO 98 I = 1, 6
00140         READ(27) LEMPAT
00146      98 WRITE(2) LEMPAT
00155         DO 86 I = 1, 2
00160         READ(27) LEMPAT
00166      86 WRITE (3) LEMPAT
00175         REWIND 27
00175   C
00175   C *** REWIND TAPE DRIVES
00176     100 REWIND 1
00177         REWIND 2
00200         REWIND 3
00201         REWIND 7
00201   C
00201   C *** INITIALIZE INPUT PARAMETERS
00202         CALL INITAL(FREQCY, ADFLAG, COMENT, PRINT, PLOT, TMIN, TMAX,
00202        $DELTAT, MULTIP)
00202   C
00202   C *** READ INPUT PARAMETERS
00203         READ(5,INPUT)
00203   C
00203   C *** READ COMMENT/DESCRIPTION CARDS
00206         IF(COMENT .EQ. 0) GO TO 103
00210         IF(COMENT .LE. 5) GO TO 29
00212         WRITE(6,28)
00214      28 FORMAT(104H1INPUT CARD DECK ERROR.  NUMBER OF COMMENT/DESCRIPTION
00214        $CARDS EXCEEDS FIVE.  EXTRA CARDS WILL BE IGNORED./27H0BREVITY IS T
00214        $HE SOUL OF WIT/15X,12H-SHAKESPEARE)
00215         COMENT = 5
00216      29 DO 101 J = 1, COMENT
00221     101 READ(5,102) (TITLE(I,J),I=1,14)
00230     102 FORMAT(13A6,1A2)
00230   C
00230   C *** PRINT PARAMETER LISTING AND REPORT HEADING
00231     103 WRITE(6,1001)
00233    1001 FORMAT(1H1,43X,22HVHF ANTENNA SIMULATION)
00234         IF(COMENT .EQ. 0) GO TO 1004
00236         DO 1002 J = 1, COMENT
00241    1002 WRITE(6,1003) (TITLE(I,J),I=1,14)
00250    1003 FORMAT(15X,13A6,1A2)
```

```
00251      1004 WRITE(6,1005) TMIN, TMAX, DELTAT, FREQCY, ADFLAG, MULTIP, PRINT,
00251          $PLOT, COMENT
00272      1005 FORMAT(1H0,45X,17HPARAMETER LISTING                        /33X
00272          $,31HMINIMUM TIME (D.H.M.S)        = 3F3.0,F4.1           /33X
00272          $,31HMAXIMUM TIME (D.H.M.S)        = 3F3.0,F4.1           /33X
00272          $,31HTIME INCREMENT (SECONDS)      = F13.1                /33X
00272          $,31HFREQUENCY FLAG (-1,1)         = I13                  /33X
00272          $,31HASCENT/DESCENT FLAG (-1,1)    = I13                  /33X
00272          $,31HMULTI-PATH FLAG (0,1)         = I13                  /33X
00272          $,31HPRINT FLAG (0,1)              = I13                  /33X
00272          $,31HPLOT FLAG (0,1)               = I13                  /33X
00272          $,31HCOMMENT/DESCRIPTION CARDS     = I13)
00273           WRITE(6,1001)
00275           IF(COMENT .EQ. 0) GO TO 1007
00277           DO 1006 J = 1, COMENT
00302      1006 WRITE(6,1003) (TITLE(I,J),I=1,14)
00311      1007 WRITE(6,1008)
00313      1008 FORMAT(1H0,109(1H-),/7X,4HTIME,10X,5HRANGE,6X,9HG-PRODUCT,5X,14HCS
00313          $M GAINS (DB),8X,14HLEM GAINS (DB),5X,23HPOLARIZATION GAINS (DB)/4X
00313          $,9H(D.H.M.S),9X,4H(NM),8X,4H(DB),7X,3(6HDIRECT,4X,8HSPECULAR,4X)/1
00313          $X,109(1H-))
00313      C
00313      C *** POSITION DRUMS FOR THIS DATA CASE
00314           N = FREQCY + 1
00315           NCSM = N / 2 + 1
00316           IF(N .EQ. 0) GO TO 96
00320           DO 97 I = 1, N
00323        97 READ(1)
00326        96 N = 2 * FREQCY + ADFLAG + 3
00327           NLEM = N / 2 + 1
00330           LEMTAP = 2 + N/6
00331           N = MOD(N,6)
00332           IF(N .EQ. 0) GO TO 94
00334           DO 95 I = 1, N
00337        95 READ(2)
00342        94 CONTINUE
00342      C
00342      C *** COMPUTE TIME LIMITS (IN SECONDS) AND SET FREQUENCY
00343           VMIN = 0.0
00344           VMAX = 0.0
00345           DO 104 I = 1, 4
```

```
00350          VMIN = VMIN + TIME(I)*TMIN(I)
00351      104 VMAX = VMAX + TIME(I)*TMAX(I)
00353          IF(FREQCY) 105,105,106
00356      105 FREQ = 259.7
00357          GO TO 107
00360      106 FREQ = 296.8
00361      107 FREQRD = FREQ * 1.0E+06 * TWOPI
00361    C
00361    C *** SCAN TRAJECTCRY TAPE AND COMPUTE LOOK ANGLES
00362          NPOINT = 1
00363      300 READ(7) (ARRAY(I,NPOINT),I=1,7),
00363         $(ARRAY(I,NPOINT),I=8,14,3),
00363         $(ARRAY(I,NPCINT),I=9,15,3),
00363         $(ARRAY(I,NPOINT),I=10,16,3),
00363         $(ARRAY(I,NPCINT),I=17,23,3),
00363         $(ARRAY(I,NPCINT),I=18,24,3),
00363         $(ARRAY(I,NPCINT),I=19,25,3)
00421          IF(ARRAY(1,NPOINT) .LT. VMIN) GO TO 300
00423          IF(ARRAY(1,NPOINT) .GT. VMAX) GO TO 999
00425          CALL REFLKT(ARRAY(8,NPOINT),ARRAY(2,NPOINT),ARRAY(17,NPOINT),
00425         $ARRAY(5,NPOINT),UNITY,ZERO,RC,ACURCY,ARRAY(27,NPOINT),IFLAG)
00426          IFLAG = IFLAG + 5 * MULTIP
00427          ARRAY(26,NPCINT) = IFLAG
00430          GO TO (303,304,304,303,303,302,304,304,303,304), IFLAG
00431      302 CALL LCOKCM(ARRAY(2,NPOINT),ARRAY(27,NPOINT),ARRAY(8,NPOINT),
00431         $ARRAY(32,NPCINT),ARRAY(33,NPOINT))
00432          CALL LCOKLM(ARRAY(5,NPCINT),ARRAY(27,NPOINT),ARRAY(17,NPOINT),
00432         $ARRAY(36,NPCINT),ARRAY(37,NPOINT))
00433      303 CALL LCOKCM(ARRAY(2,NPOINT),ARRAY(5,NPOINT),ARRAY(8,NPOINT),
00433         $ARRAY(30,NPCINT),ARRAY(31,NPOINT))
00434          CALL LCOKLM(ARRAY(5,NPOINT),ARRAY(2,NPOINT),ARRAY(17,NPOINT),
00434         $ARRAY(34,NPCINT),ARRAY(35,NPOINT))
00435      304 VMIN = ARRAY(1,NPOINT) + DELTAT
00436          NPOINT = NPOINT + 1
00437          IF(VMIN .GT. VMAX) GO TO 999
00441          IF(NPOINT .LE. 200) GO TO 300
00443          WRITE(6,93)
00445       93 FORMAT(1H0,86HNUMBER CF TRAJECTORY POINTS EXCEEDS 200.  ONLY THE F
00445         $IRST 200 POINTS WILL BE PROCESSED.)
00446      999 NPOINT = NPOINT - 1
00446    C
```

```
00446   C *** SCAN ANTENNA PATTERNS AND PROCESS EACH DATA POINT
00447         DO 501 III = 1, 2
00452         READ(1) CSMPAT
00460         DO 500 JJJ = 1, 2
00463         READ(LEMTAP) LEMPAT
00463   C
00463   C *** PRINT LINK BEING SIMULATED
00471         LINK = 2*(III-1)+JJJ
00472         WRITE(6,26) LINK, III, JJJ
00477      26 FORMAT(1H0,33X,9HLINK NO. ,I1,15H.  CSM ANTENNA ,I1,14H, LEM ANTEN
00477        $NA ,I1,1H.)
00500         DO 703 INDEX = 1, NPCINT
00500   C
00500   C
00500   C *** CONVERT TIME TO (D.H.M.S)
00503         TMIN(4) = ARRAY(1,INDEX)
00504         DO 27 I = 1, 3
00507         TMIN(I) = AINT(TMIN(4)/TIME(I))
00510      27 TMIN(4) = TMIN(4) - TMIN(I)*TIME(I)
00510   C
00510   C *** TRANSFER IF NO CALCULATIONS ARE TO BE EXECUTED THIS TIME-POINT
00512         IFLAG = ARRAY(26,INDEX)
00513         GO TO (350,450,450,350,350,350,450,450,350,450), IFLAG
00513   C
00513   C *** PERFORM DIRECT PATH CALCULATIONS
00513   C *** COMPUTE THE RANGE BETWEEN THE CSM AND LEM, R
00514     350 CALL VECSUB(ARRAY(2,INDEX),ARRAY(5,INDEX),3,RLCB)
00515         CALL VECMAG(RLCB,3,R)
00515   C
00515   C *** COMPUTE POLARIZATION ANGLE ALONG DIRECT PATH, RHO
00516         CALL PCLANG(ARRAY(30,INDEX),ARRAY(31,INDEX),ARRAY(34,INDEX),
00516        $ARRAY(35,INDEX),ARRAY(8,INDEX),ARRAY(17,INDEX),RHO)
00516   C
00516   C *** FIND DIRECT PATH ANTENNA PATTERN DATA
00517         IF(ARRAY(30,INDEX) .LT. 0.034903 .OR. ARRAY(30,INDEX) .GT. 6.2482)
00517        $ARRAY(30,INDEX) = 0.034923
00521         IF(ARRAY(31,INDEX) .LT. 0.034903 .OR. ARRAY(31,INDEX) .GT. 6.2482)
00521        $ARRAY(31,INDEX) = 0.034923
00523         ITHETA = (IFIX(ARRAY(30,INDEX) * 57.296) + 1) / 2
00524         IPHI = (IFIX(ARRAY(31,INDEX) * 57.296) + 1) / 2
00525         ICODED = CSMPAT(ITHETA,IPHI)
```

```
00526        DO 91 I = 1, 6
00531        J = 3*(III-1) + (I+1)/2
00532     91 PATERN(I,1) = GNREFC(J,NCSM) - FLD(36-6*I,6,ICODED)
00534        IF(ARRAY(34,INDEX) .LT. 0.034903 .OR. ARRAY(34,INDEX) .GT. 6.2482)
00534       $ARRAY(34,INDEX) = 0.034923
00536        IF(ARRAY(35,INDEX) .LT. 0.034903 .OR. ARRAY(35,INDEX) .GT. 6.2482)
00536       $ARRAY(35,INDEX) = 0.034923
00540        ITHETA = (IFIX(ARRAY(34,INDEX) * 57.296) + 1) / 2
00541        IPHI = (IFIX(ARRAY(35,INDEX) * 57.296) + 1) / 2
00542        ICODED = LEMPAT(ITHETA,IPHI)
00543        DO 90 I = 1, 6
00546        J = 3*(JJJ-1) + (I+1)/2
00547     90 PATERN(I,3) = GNREFL(J,NLEM) - FLD(36-6*I,6,ICODED)
00547  C
00547  C *** COMPUTE POLARIZATION ELLIPSE TILT ANGLES, TAU1 AND TAU2, ALONG DIRECT PATH
00551        CALL TILT(PATERN(1,1),PATERN(2,1),PATERN(3,1),PATERN(4,1),TAU1)
00552        CALL TILT(PATERN(1,3),PATERN(2,3),PATERN(3,3),PATERN(4,3),TAU2)
00552  C
00552  C *** COMPUTE COORDINATE SYSTEM MIS-ALIGNMENT ALONG DIRECT PATH, BETA
00553        CALL MISALN(TAU1,TAU2,RHO,BETA)
00553  C
00553  C *** COMPUTE ANTENNA GAINS, G1, G2, AND AXIAL RATIOS, RR1, RR2, ALONG DIRECT P
00554        CALL GAINRR(PATERN(5,1),PATERN(6,1),G1,RR1)
00555        CALL GAINRR(PATERN(5,3),PATERN(6,3),G2,RR2)
00556        G1DB = DB(G1)
00557        G2DB = DB(G2)
00557  C
00557  C *** COMPUTE POLARIZATION LOSS ALONG DIRECT PATH, LP
00560        CALL POLAR(RR1,RR2,BETA,LP)
00561        LPDB = DB(LP)
00561  C
00561  C *** TRANSFER IF MULTI-PATH CALCULATIONS ARE NOT REQUIRED
00562        IF(IFLAG .EQ. 6) GO TO 400
00564        K = 0.0
00565        GO TO 401
00565  C
00565  C *** PERFORM MULTIPATH CALCULATIONS
00565  C *** COMPUTE SPECULAR PATH LENGTHS R1S, R2S, RS, DELTAR AND THE ANGLE PSI
00566    400 CALL VECSUB(ARRAY(5,INDEX),ARRAY(27,INDEX),3,RSLB)
00567        CALL VECSUB(ARRAY(2,INDEX),ARRAY(27,INDEX),3,RSCB)
00570        CALL VECMAG(RSCB,3,R1S)
```

```
      CALL VECMAG(RSLB,3,R2S)
      RS = R1S + R2S
      DELTAR = RS - R
      CALL VECDOT(ARRAY(27,INDEX),RSLB,3,A)
      PSI = HALFPI - ACOS(A / (RO * R2S))
C
C *** COMPUTE MULTIPATH POLARIZATION ANGLES, RHO1S AND RHO2S
      CALL POLARS(ARRAY(2,INDEX),ARRAY(5,INDEX),ARRAY(27,INDEX),ARRAY(8,
     $INDEX),ARRAY(17,INDEX),PSI,ARRAY(32,INDEX),ARRAY(33,INDEX),ARRAY(3
     $6,INDEX),ARRAY(37,INDEX),RHO1S,RHO2S)
C
C *** FIND SPECULAR PATH ANTENNA PATTERN DATA
      IF(ARRAY(32,INDEX) .LT. 0.034903 .OR. ARRAY(32,INDEX) .GT. 6.2482)
     $ARRAY(32,INDEX) = 0.034923
      IF(ARRAY(33,INDEX) .LT. 0.034903 .OR. ARRAY(33,INDEX) .GT. 6.2482)
     $ARRAY(33,INDEX) = 0.034923
      ITHETA = (IFIX(ARRAY(32,INDEX) * 57.296) + 1) / 2
      IPHI = (IFIX(ARRAY(33,INDEX) * 57.296) + 1) / 2
      ICODED = CSMPAT(ITHETA,IPHI)
      DO 89 I = 1, 6
      J = 3*(III-1) + (I+1)/2
   89 PATERN(I,2) = GNREFC(J,NCSM) - FLD(36-6*I,6,ICODED)
      IF(ARRAY(36,INDEX) .LT. 0.034903 .OR. ARRAY(36,INDEX) .GT. 6.2482)
     $ARRAY(36,INDEX) = 0.034923
      IF(ARRAY(37,INDEX) .LT. 0.034903 .OR. ARRAY(37,INDEX) .GT. 6.2482)
     $ARRAY(37,INDEX) = 0.034923
      ITHETA = (IFIX(ARRAY(36,INDEX) * 57.296) + 1) / 2
      IPHI = (IFIX(ARRAY(37,INDEX) * 57.296) + 1) / 2
      ICODED = LEMPAT(ITHETA,IPHI)
      DO 88 I = 1, 6
      J = 3*(JJJ-1) + (I+1)/2
   88 PATERN(I,4) = GNREFL(J,NLEM) - FLD(36-6*I,6,ICODED)
C
C *** COMPUTE POLARIZATION ELLIPSE TILT ANGLE OF RECEIVING ANTENNA, TAU2S
      CALL TILT(PATERN(1,2),PATERN(2,2),PATERN(3,2),PATERN(4,2),TAU2S(1)
     $)
      CALL TILT(PATERN(1,4),PATERN(2,4),PATERN(3,4),PATERN(4,4),TAU2S(2)
     $)
C
C *** COMPUTE POLARIZATION ELLIPSE TILT ANGLE OF TRANSMITTING ANTENNA, TAU1S
      TAU1S(1) = TAU2S(1) - RHO1S
```

```
      TAU1S(2) = TAU2S(2) - RHO2S
      DO 351 I = 1, 2
  351 IF(TAU1S(I) .LT. 0.0) TAU1S(I) = TAU1S(I) + PI
C
C *** COMPUTE MULTI-PATH REFLECTION COEFFICIENTS, RA AND RB
      AA = CMPLX(EPSLNR,-SIGMA/(FREQRD * EPSLNO))
      BB = CSQRT(AA - COS(PSI)**2)
      C = SIN(PSI)
      A = AA * C
      RA = (C - BB) / (C + BB)
      RB = (A - BB) / (A + BB)
C
C *** COMPUTE CSM AND LEM ANTENNA GAINS AND AXIAL RATIOS ALONG SPECULAR PATH
      CALL GAINRR(PATERN(5,2),PATERN(6,2),G1S,RR1S)
      CALL GAINRR(PATERN(5,4),PATERN(6,4),G2S,RR2S)
      G1SDB = DB(G1S)
      G2SDB = DB(G2S)
C
C *** COMPUTE POLARIZATION RATIO OF INCIDENT WAVE, P1S
      ALPHA(1) = ATAN2(1.0,RR1S) * 2.0
      ALPHA(2) = ATAN2(1.0,RR2S) * 2.0
      DO 352 I = 1, 2
  352 P1S(I) = TAN(0.5 * ACOS(COS(ALPHA(I)) * COS(2.0 * TAU1S(I)))) *
     $CEXP(CMPLX(0.0,ATAN2(TAN(ALPHA(I)),SIN(2.0 * TAU1S(I)))))
C
C *** COMPUTE DEPOLARIZATION FACTOR, Q, AND POLARIZATION RATIO OF REFLECTED WAVE
      Q = RA / RB
      DO 353 I = 1, 2
  353 P2S(I) = Q * P1S(I)
C
C *** COMPUTE TILT ANGLE, TAUS, AND AXIAL RATIO, RRS, OF REFLECTED WAVE
      DO 354 I = 1, 2
      A = 2.0 * ATAN(CABS(P2S(I)))
      B = ATAN2(AIMAG(P2S(I)),REAL(P2S(I)))
      RRS(I) = 1.0 / TAN(0.5 * ASIN(SIN(A) * SIN(B)))
  354 TAUS(I) = 0.5 * ATAN(TAN(A) * COS(B))
C
C *** COMPUTE THE COORDINATE SYSTEM MIS-ALIGNMENT ALONG THE SPECULAR PATH, BETAS
      CALL MISALN(TAU2S(1),TAUS(1),RHO2S,BETAS(1))
      CALL MISALN(TAU2S(2),TAUS(2),RHO1S,BETAS(2))
C
```

```
00703    C *** COMPUTE SPECULAR POLARIZATION LOSS, LPS
00704          CALL POLAR(RRS(1),RR2S,BETAS(1),LPS(1))
00705          CALL POLAR(RRS(2),RR1S,BETAS(2),LPS(2))
00706          LPSDB = DB(AMIN1(LPS(1),LPS(2)))
00706    C
00706    C *** COMPUTE THE LUNAR REFLECTION LOSS, LR
00707          A = CABS(RA) **2
00710          B = CABS(RB) **2
00711          DO 355 I = 1, 2
00714          C = CABS(P1S(I)) **2
00715      355 LR(I) = (B + A * C) / (1.0 + C)
00715    C
00715    C *** COMPUTE MULTI-PATH DIVERGENCE FACTOR, D
00717          A = 2.0 * R1S * R2S
00720          B = SIN(PSI)
00721          D = 1.0 / SQRT(1.0 + A * (1.0 + B**2) / (RS * RO * B) + (A / RO)**
00721         $   2 / RS)
00721    C
00721    C *** COMPUTE MULTIPATH REFLECTION FACTOR, K
00722          A = R * G1S * G2S * D / (RS * G1 * G2)
00723          ALPHA(1) = A * LR(1) * SQRT(LPS(1) / LP)
00724          ALPHA(2) = A * LR(2) * SQRT(LPS(2) / LP)
00725          C = COS(AMOD(TWOPIC*FREQ*DELTAR,TWOPI))
00726          IF(ABS(ALPHA(1) + C) .LT. ABS(ALPHA(2) + C)) GO TO 356
00730          K = ALPHA(2)
00731          GO TO 401
00732      356 K = ALPHA(1)
00732    C
00732    C *** COMPUTE GAIN PRODUCT GP
00733      401 GP = LP * G1 * G2 * (1.0 + K * (K + 2.0 * C))
00734          GPDB = DB(GP)
00734    C
00734    C *** PRINT STANDARD OUTPUT
00735          IF(IFLAG .EQ. 6) GO TO 1009                                        **-1
00737          IF(IFLAG .EQ. 9) WRITE(6,998)
00742      998 FORMAT(1X,107HTHE FOLLOWING TIME-POINT HAS THE LM AND CSM LYING AL
00742         $ONG A LUNAR RADIUS.  THE EXISTENCE OF A 'BOUNCE-POINT' /103H IN SU
00742         $CH A SITUATION IS MORE ACADEMIC THAN ACTUAL.  HENCE, ONLY DIRECT P
00742         $ATH CALCULATIONS WERE EXECUTED.)
00743          WRITE(6,1000) TMIN, R, GPDB, G1DB, G2DB, LPDB
00756     1000 FORMAT(1X,3F3.0,F4.1,6X,F6.2,7X,F7.2,4X,3(F7.2,15X))
```

```
00757          GO TO 1011
00760     1009 WRITE(6,1010) TMIN, R, GPDB, G1DB, G1SDB, G2DB, G2SDB, LPDB, LPSDB
00776     1010 FORMAT(1X,3F3.0,F4.1,6X,F6.2,7X,F7.2,4X,3(F7.2,3X,F7.2,5X))
00776     C
00776     C *** PRINT OPTIONAL OUTPUT
00777     1011 IF(PRINT .NE. 0) WRITE(6,87) (ARRAY(I,INDEX),I=1,37), PATERN, R,
00777         $RHO, TAU1, TAU2, BETA, G1, G2, RR1, RR2, LP, R1S, R2S, RS, DELTAR,
00777         $PSI, RHO1S, RHO2S, TAU1S, TAU2S, RA, RB, G1S, G2S, RR1S, RR2S,
00777         $P1S, P2S, Q, TAUS, RRS, BETAS, LPS, LR, D, K, GP
01111       87 FORMAT(9H0TIME    =1PE10.3/9H ROCB    =E10.3,2(1H,E10.3)/9H ROLB    =
01111         $E10.3,2(1H,E10.3)/9H ACB     =E10.3,8(1H,E10.3)/9H ALB     =E10.3,8(
01111         $1H,E10.3)/9H IFLAG   =E10.3/9H ROSB    =E10.3,2(1H,E10.3)/9H THTAC
01111         $=E10.3/9H PHIC    =E10.3/9H THTACS =E10.3/9H PHICS   =E10.3/9H THTAL
01111         $  =E10.3/9H PHIL    =E10.3/9H THTALS =E10.3/9H PHILS   =E10.3/9H PAT
01111         $ERN =3(6(E10.3,1H,)/9X)5(E10.3,1H,)E10.3/9H R       =E10.3/9H RHO
01111         $  =E10.3/9H TAU1    =E10.3/9H TAU2    =E10.3/9H BETA    =E10.3/9H G1
01111         $    =E10.3/9H G2      =E10.3/9H RR1     =E10.3/9H RR2     = E10.3/9H
01111         $LP      =E10.3/9H R1S     =E10.3/9H R2S     =E10.3/9H RS      =E10.3/9
01111         $H DELTAR =E10.3/9H PSI     =E10.3/9H RHO1S   =E10.3/9H RHO2S   =E10.3
01111         $/9H TAU1S  =E10.3,1H,E10.3/9H TAU2S  =E10.3,1H,E10.3/9H RA      =1H
01111         $(E9.3, 6H) + J(E9.3, 1H)/9H RB      =1H(E9.3, 6H) + J(E9.3, 1H)/9H
01111         $G1S     =E10.3/9H G2S     =E10.3/9H RR1S    =E10.3/9H RR2S    =E10.3/9
01111         $H P1S    =1H(E9.3, 6H) + J(E9.3, 2H),    2H (E9.3, 6H) + J(E9.3,1H
01111         $)/9H P2S     =1H(E9.3, 6H) + J(E9.3, 2H),    2H (E9.3, 6H) + J(E9.3
01111         $,1H)/9H Q       =1H(E9.3, 6H) + J(E9.3, 1H)/9H TAUS    =E10.3,1H,E10
01111         $.3/9H RRS     =E10.3,1H,E10.3/9H BETAS  =E10.3,1H,E10.3/9H LPS     =
01111         $E10.3,1H,E10.3/9H LR      =E10.3,1H,E10.3/9H D       =E10.3/9H K
01111         $  =E10.3/9H GP      =E10.3/)
01111     C
01111     C *** WRITE PLOT TAPE
01112          IF(PLOT .NE. 0) WRITE(8) (ARRAY(I,INDEX),I=1,37), R, RHO, TAU1,
01112         $TAU2, BETA, G1, G1DB, G2, G2DB, RR1, RR2, LP, LPDB, R1S, R2S, RS,
01112         $DELTAR, PSI, RHO1S, RHO2S, TAU1S, TAU2S, RA, RB, G1S, G1SDB, G2S,
01112         $G2SDB, RR1S, RR2S, P1S, P2S, Q, TAUS, RRS, BETAS, LPS, LPSDB, LR,
01112         $D, K, GP, GPDB
01227          GO TO 703
01227     C
01227     C *** NO CALCULATIONS HAVE BEEN EXECUTED - PRINT APPROPRIATE MESSAGE
01230      450 IF((IFLAG .EQ. 2) .OR. (IFLAG .EQ. 7)) GO TO 451
01232          IF((IFLAG .EQ. 3) .OR. (IFLAG .EQ. 8)) GO TO 452
01234          WRITE(6,453) TMIN
```

```
01242    453 FORMAT(1X,3F3.0,F4.1,5X,62HTHE 'BOUNCE-POINT' WAS NOT DETERMINED T
01242       $O THE DESIRED ACCURACY.)
01243        GO TO 703
01244    451 WRITE(6,454) TMIN
01252    454 FORMAT(1X,3F3.0,F4.1,5X74HLM AND CSM ARE ON OPPOSITE SIDES OF THE
01252       $LUNAR HORIZON.  TRY SMOKE SIGNALS.)
01253        GO TO 703
01254    452 WRITE(6,455) TMIN
01262    455 FORMAT(1X,3F3.0,F4.1,5X,78HLM AND CSM ARE ON DIAMETRICALLY OPPOSIT
01262       $E SIDES OF THE MOON-- SORRY 'BOUT THAT.)
01263    703 CONTINUE
01263  C
01263  C *** END-FILE PLOT TAPE
01265        IF(PLOT .NE. 0) END FILE 8
01267    500 CONTINUE
01267  C
01267  C *** RE-POSITION LEM TAPE
01271        BACK SPACE LEMTAP
01272        BACK SPACE LEMTAP
01273    501 CONTINUE
01275        GO TO 100
01276        END

   END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
 VHFASP        SYMBOLIC                          14 JAN 69  17&07&21   0  01401564    14
 VHFASP  CODE  RELOCATABLE                       14 JAN 69  17&07&21   1  01417026    60
                                                                       0  01417122    14
```