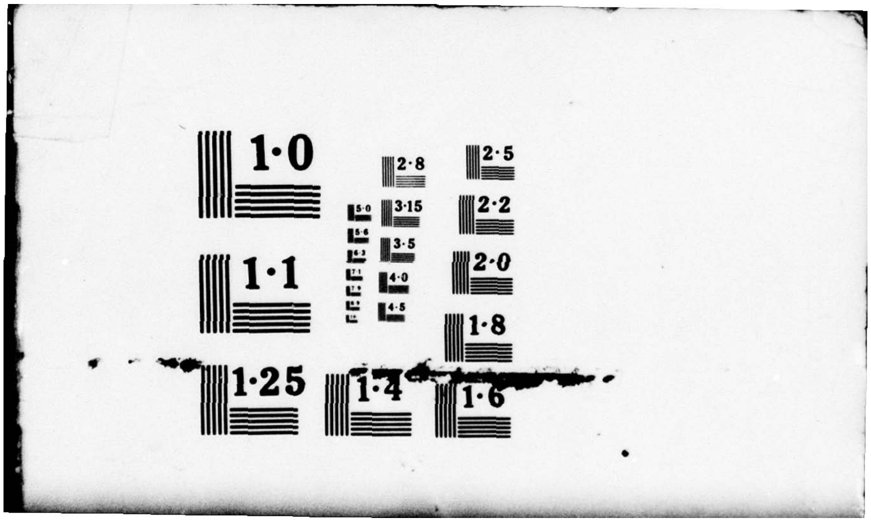
AD-A065 178   SYRACUSE UNIV N Y   F/G 20/14
COMPUTER PROGRAM MAINTENANCE MANUAL FOR THE ANTENNA PATTERN DIS--ETC(U)
JAN 79 J PERINI, S WANG, K HIRASAWA   F30602-75-C-0121
UNCLASSIFIED   RADC-TR-78-264   NL

1 OF 2
ADA
065178


1·0
2·8
2·5
5·0
3·15
2·2
5·6
3·5
6·3
2·0
7·1
4·0
8·0
4·5
1·1
1·8
1·25
1·4
1·6

RADC-TR-78-264
Phase Report
January 1979

COMPUTER PROGRAM MAINTENANCE MANUAL FOR THE ANTENNA PATTERN DISTORTION COMPUTER PROGRAM - VERSION IV

Syracuse University

Dr. J. Perini
Dr. S. Wang
Dr. K. Hirasawa

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

This report contains a large percentage of machine-produced copy which is not of the highest printing quality but because of economical consideration, it was determined in the best interest of the government that they be used in this publication.

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-78-264 has been reviewed and is approved for publication.

APPROVED: Jacob Scherer

JACOB SCHERER
Project Engineer

APPROVED: Joseph J Naresky

JOSEPH J. NARESKY
Chief, Reliability & Compatibility Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (RBC) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19 REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
18 1. REPORT NUMBER RADC-TR-78-264	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER 9
6 4. ~~TITLE (and Subtitle)~~ COMPUTER PROGRAM MAINTENANCE MANUAL FOR THE ANTENNA PATTERN DISTORTION COMPUTER PROGRAM - VERSION IV.		5. ~~TYPE OF REPORT & P~~ERIOD COVERED Phase ~~Report~~,
		6. ~~PERFORMING ORG.~~ REPORT NUMBER N/A
10 7. ~~AUTHOR(s)~~ ~~Dr.~~ J. Perini, ~~Dr.~~ S. Wang ~~Dr.~~ K. Hirasawa		8. CONTRACT OR GRANT NUMBER(s) 15 F30602-75-C-0121
9. ~~PERFORMING ORGANIZATI~~ON NAME AND ADDRESS Syaracuse University Syracuse NY 13210		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 16 95670016 17 00
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (RBC) Griffiss AFB NY 13441		11 12. ~~REPORT DATE~~ Janu~~ary~~ 1979
		13. ~~NUMBER OF PAGES~~ 140
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Same 12 158 p.		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

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

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Jacob Scherer (RBC)

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

Antennas
Patterns
Coupling

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

This report describes in reasonable detail the Antenna Pattern Distortion Computer (DISTORT) so that any software maintenance or modification can be carried out without much difficulty. A description of every subsystem, program and subroutine as well as all the file requirements is given and is accompanied by detailed flow charts.

339600

PREFACE

This effort was conducted by Syracuse University under the sponsorship of the Rome Air Development Center Post-Doctoral Program for Rome Air Development Center. Mr. Marvin Hirshman (EEIT) and Capt. Jerry Rintella (EEIMC), 1842 EEG, were the task project engineers and provided overall technical direction and guidance.

The RADC Post-Doctoral Program is a cooperative venture between RADC and some sixty-five universities eligible to participate in the program. Syracuse University (Department of Electrical Engineering), Purdue University (School of Electrical Engineering), Georgia Institute of Technology (School of Electrical Engineering), and State University of New York at Buffalo (Department of Electrical Engineering) act as prime contractor schools with other schools participating via sub-contracts with prime schools. The U.S. Air Force Academy (Department of Electrical Engineering), Air Force Institute of Technology (Department of Electrical Engineering), and the Naval Post Graduate School (Department of Electrical Engineering) also participate in the program.

The Post-Doctoral Program provides an opportunity for faculty at participating universities to spend up to one year full time on exploratory development and problem-solving efforts with the post-doctorals splitting their time between the customer location and their educational institutions. The program is totally customer-funded with current projects being undertaken for Rome Air Development Center (RADC), Space and Missile Systems Organization (SAMSO), Aeronautical System Division (ASD), Electronics Systems Division (ESD), Air Force Avionics Laboratory (AFAL), Foreign Technology Division (FTD), Air Force Weapons Laboratory (AFWL), Armament Development and Test Center (ADTC), Air Force Communications Service (AFCS), Aerospace Defense Command (ADC), HQ USAF, Defense Communications Agency (DCA), Navy, Army, Aerospace Medical Division (AMD), and Federal Aviation Administration (FAA).

Further information about the RADC Post-Doctoral Program can be obtained from Mr. Jacob Scherer, RADC/RBC, Griffiss AFB, NY, 13441, telephone Autovon 587-2543, Commercial (315) 330-2543.

TABLE OF CONTENTS

1. GENERAL DESCRIPTION

1.1 Purpose of the Program Maintenance Manual

The objective of this Computer Program Maintenance Manual for the Antenna Pattern Distortion Computer Program is to provide the maintenance programmer personnel with the information necessary to effectively maintain the system.

1.2 System Application

The Antenna Pattern Distortion Computer Program calculates antenna coupling coefficients, vertical and horizontal radiation patterns, communication range contour and pattern distribution of antenna farms such as those found in many AF air-to-ground communication stations.

1.3 Equipment Environment

1.3.1 Computer. Honeywell 6180, GCOS SRH.2 and HP 7202A Graphic Plotter.

1.3.2 Input/Output Devices. (a) on-line disks

(b) terminal

Keyboard
Printer
Plotter

1.4 Program Environment

1.4.1 Programming Language. With the exception of three GMAP (Honeywell DD08A) subroutines, YSFUTL, PACK, DIMXXX, all programs and subroutines are written in series 60 (level 66)/6000 FORTRAN (Honeywell DD-02)

1.4.2 Mode of Processing

The program is in the time sharing mode for the inputing of data. Although the main calculations have to be done in the remote batch mode (CARDIN), this is completely transparent to the user since the CARDIN JOB is spawned by the time-sharing program with the use of specially written utility subroutines.

The subsystems /RADIO/DISTORT and /RADIO/DISTORTP have been compiled and are stored in files with these names in the user library of AFCS.LIB so that they can be accessed as follows:

a. In response to "SYSTEM?" type FORT NEW

b. The computer will print

```
READY
*
```

c. Type RUN AFCS.LIB/RADIO/DISTORT (or RUN AFCS.LIB/RADIO/DISTORTP)

d. The program will then start asking questions.

2. SYSTEM DESCRIPTION

The system consists of three subsystems

a) Subsystem /RADIO/DISTORT

b) Subsystem /RADIO/USA

c) Subsystem /RADIO/DISTORTP

d) Utility packages.

2.1.1 Subsystem /RADIO/DISTORT

This system consists of a series of time-sharing programs that accept the necessary input data from the terminal, process it and generate a file that will be used by the /RADIO/USA subsystem.

2.1.2 Subsystem /RADIO/USA

This subsystem consists of a series of batch programs that accept the output generated by /RADIO/DISTORT and carries out all the computations to provide the requested outputs. This subsystem is spawned by the timesharing /RADIO/DISTORT through callable routines of the utility packages.

2.1.3 Subsystem /RADIO/DISTORTP

This subsystem consists of a series of time-sharing programs that accept the output generated by /RADIO/USA and produces a plot on the HP 7202A graphic plotter

2.2 Detailed Description

Each of the subsystems contain the following functions/ subroutines.

Subsystem	Function/Subroutine
/RADIO/DISTORT	LINKMN
	LINKM1
	LINKM2
	LINKM3
/RADIO/USA	DIMXXX
	MAIN
	PATT
	FUN
	ELE
	ELK
	PXYZ
	QZP
	NPAT
	PATD
/RADIO/DISTORTP	DSTPLT
	PPLOT
	LABEL

Subsystem	Function/Subroutine
Utility Packages	YSFUTL
	PACK

2.2.1 Subroutine LINKMN

[1] Description: This is the resident time-sharing program that controls the link/overlays LINKM1, LINKM2, LINKM3 in and out of core. At the beginning it creates two temporary files 01 and 02. File 01 will hold the data that will be used by the subsystem /RADIO/USA. File 02 will contain the $LIMITS and $PRMFL statements required to run /RADIO/USA. It also attaches the file BLA00001/RADIO /NORUN that keeps track of the numbers of the executed runs (NORUN).

[2] Language: TSS FORTRAN

[3] Input: Terminal

[4] Output: Terminal

2.2.2 Subroutine LINKM1

[1] Description: This time sharing program asks all the pertinent questions with regard to the calculation of antenna patterns, coupling coefficients, pattern distributions, or communication range contour. If the user responses are outside certain specified ranges, the program will print an error message and asks the question again until an acceptable response is entered.

[2] Language: TSS FORTRAN

[3] Input: Terminal

[4] Output: terminal

[5] Called by: resident subroutine LINKMN

2.2.3 Subroutine LINKM2

[1] Description: This time-sharing program asks questions about the site parameters and topographical data for the calculation of the communication range contours in a similar fashion as LINKM1. When all the pertinent data is entered, it prints out a summary of all the data entered in LINKM1 and LINKM2 for verification by the user.

[2] Language: TSS FORTRAN

[3] Input: Terminal

[4] Output: Terminal

[5] Called by: resident subroutine LINKMN.

2.2.4 Subroutine LINKM3

[1] Description: This time-sharing program, using the information of LINKM1 and LINKM2, generates the necessary data to be used by the batch programs and writes it in the temporary file 01. It also calculates the necessary size to be used in the variable dimension statements, the amount of core required, as well as an estimate of the running time. This information is printed out and is also written in the proper $LIMITS and $PRMFL cards of file 02. Next the file BLA00001/DISTRT/NORUN is created. This file will hold the input data for the plotter subsystem /RADIO/DISTORTP. LINKM3 then calls the CARDIN system and initiates the batch portion of the /RADIO/DISTORT subsystem through the JOB control file /RADIO/USA.RUN1

[2] Language: TSS FORTRAN

[3] Input: data generated by LINKM1 and LINKM2 through COMMON

[4] Output: Terminal Printer

Temporary Files 01 and 02

[5] Called by: resident subroutine LINKMN

2.2.5 Subroutine DIMXXX

[1] Description: This batch program defines the size of the variable dimension statements of the MAIN subroutine and allocates the necessary core.

[2] Language: GMAP

[3] Input: data generated by the TSS programs

[4] Output: DIMENSION OF ARRAYS C1 and C2 of MAIN

[5] Called by: LINKM3.

2.2.6 Subroutine MAIN

[1] Description: This batch program calculates all that is requested by the user. To do this, it uses a series of subroutines/ functions which are described below.

[2] Language: FORTRAN

[3] Input: Temporary file 01 and permanent file /RADIO/(filename) created by another AFCS program called SCREEN.* (filename) was assigned to this file at the time SCREEN was run.

[4] Output: File 02 (/DISTRT/XXXX) where XXXX is a number internally generated by LINKM3 to identify the run.

[5] Called by Printer DIMXXX

*User manual for SCREEN system (DSD MO57) March 1976

2.2.7 Subroutine PATT

[1] Description: This batch program calculates the radiation patterns and the communication range contours

[2] Language: FORTRAN

[3] Input: /RADIO/(filename)
COMMON and arguments from MAIN

[4] Output: File 02 (/DISTORT/XXXX) and the printer
Argument-AMAX
COMMON - AZ

[5] Called by MAIN

2.2.8 Subroutine FUN

[1] Description: This batch program performs numerical integration by the trapezoidal rule.

[2] Language: FORTRAN

[3] Input Arguments: DD, DE, Y9, NDIM

[4] Output Argument: SUM

[5] Called by: MAIN

2.2.9 Function ELE

[1] Description: This batch program performs the numerical integration of elliptic integrals of the second kind.

[2] Language: FORTRAN

[3] Inpur Argument: P

[4] Output: Return value to MAIN

[5] Called by: MAIN

2.2.10 Function ELK

[1] Description: This batch program performs the numerical integration of elliptic integrals of the first kind.

[2] Language: FORTRAN

[3] Input Argument: P

[4] Output: return value to MAIN

[5] Called by: MAIN

2.2.11 Subroutine PXYZ

[1] Description: In order to perform all desired calculations every antenna has to be subdivided into small subsections. This program computes the coordinates of the end points of each of test subsections.

[2] Language: FORTRAN

[3] Input: Temporary file 01
COMMON

[4] Output: COMMON (IT, X, Y, Z ,X1, Y1, Z1)

[5] Called by MAIN

2.2.12 Subroutine QZP

[1] Description: This batch program calculates the redundant elements of the impedance matrix.

[2] Language: FORTRAN

[3] Input: Arguments
COMMON

[4] Output: Argument MZ

[5] Called by: MAIN

2.2.13 Subroutine NPAT

[1] Description: This batch program normalizes the radiation pattern to its maximum value.

[2] Language: FORTRAN

[3] Input: Arguments AINT, AZ, AMAX, AIT
COMMON

[4] Output: File 02 (/DISTRT/XXXX)
Arguments: ATH, FM, FDB, I

[5] Called by: MAIN

2.2.14 Subroutine PATD

[1] Description: This batch program calculates the pattern distribution of normalized horizontal patterns.

[2] Language: FORTRAN

[3] Input Arguments: DPER, IDB, NFD, AZ

[4] Output: File 02 (/DISTRT/XXXX)
Arguments: P,G, or P, G, H or P, H

[5] Called by MAIN

2.2.15 Subroutine DSTPLT

[1] Description: This time-sharing program was developed by AFCS personnel. It controls the HP7202A graphic plotter and produces a plot of the horizontal and vertical patterns calculated by the subsystem /RADIO/DISTORT

[2] Language: FORTRAN

[3] Input: File (DISTRT/XXXX)

[4] Output: Plotter (HP 7202A graphic plotter)
Terminal

[5] Called by: /RADIO/DISTORTP subsystem

2.2.16 Subroutine PPLOT

[1] Description: This time-sharing program was developed by AFCS personnel. It calculates the (X,Y) coordinates of the points to be plotted.

[2] Language: FORTRAN

[3] Input: Arguments

[4] Output: Plotter (HP 7202A graphic plotter)
Terminal

[5] Called by: DSTPLT

2.2.17 Subroutine LABEL

[1] Description: This time-sharing program was developed by AFCS personnel. Its function is to generate alphanumerics on HP 7202A graphic plotter.

[2] Language: FORTRAN

[3] Input: Argument INPTFILE
COMMON

[4] Output: Plotter (HP 7202A graphic plotter)

[5] Called by: DSTPLT

2.2.18 Utility Package YSFUTL

[1] Description: This utility package was developed by AFCE personnel and is designed to be used as a subroutine for TSS FORTRAN programs. It performs miscellaneous worthwhile functions which are difficult or impossible to pull off without the aid of such a subroutine. There are eight entry points to this routine, the characteristics of which are described below.

a) Call FLGBRK (FLAG)

Where FLAG is a logical variable, will allow recovery from breaks during execution. When FLGBRK is called, FLAG will be set to the logical value FALSE. If a break occurs subsequently, the logical variable FLAG will be set to TRUE to indicate recognition of a break. Fhis FLAG may be tested by the calling routine at any time, and any appropriate action taken. To portect the user against infinite loops, however, breaks are counted as they are received; if twenty or more breaks are recognized, execution is temporarily suspended and the following message appears --

```
*BREAK STOP OR CONTINUE
```

At this point the user may enter any of three responses:

S will immediately terminate the program
A will terminate the program with a DRL ABORT
C will reset counters and continue execution

Note: The DRL report allows dumping the subsystem to a file for debugging purposes.

b) Call BRKOFF

Calling this entry point nullifies the effects of a previous call to FLGBRK.

c) Call KEYOT

Calling this entry point will force any output accumulated by the subsystem to be output. This is of significance whenever small amounts of output are being generated, with extended processing intervening. If this function is not used, in such a case the output will not be printed until the executive's buffer fills, or input is requested, either of which might not occur for a long time.

d) Call KEYIN (BUFFIN)

This entry point allows the user to retrieve the last line of input so long as it remains available. Thus it effectively functions as a backspace command for the teletype. The input is placed in the Array, BUFF, in 21A4 format. N is an integer variable into which will be placed the number of characters transmitted. Note that N may be zero, indicating that the data is no longer available due to intervening output. The data placed in BUFF is not edited or blank filled in any way, and will include a carriage return character at the end of the line. N includes the carriage return.

e) Call CALLTS (SSNAME, LINE, NCHAR)

This entry point allows the user to place data in the KIN buffer or call another TSS subsystem or both. If a subsystem is to be called, the first four characters of the subsystem name should be placed in the ASCII variable SSNAME. If SSNAME contains all blanks or is an integer zero, the subsystem call will be bypassed. If data is to be placed in the KIN buffer, the count of characters to be moved should be placed in the integer variable NCHAR and the ASCII array, LINE should contain the data to be placed in the buffer in 21A4 format. Up to 81 characters may be placed in the buffer in this manner. If NCHAR is zero, this function will be bypassed. As an example of usage of this function, assume the user wishes to obtain the status of a batch job with the SNUMB 1234T. The command at system level would be: JSTS 1234T<CR>. This may be accomplished by the TSS FORTRAN program by placing the subsystem name (JSTS) in SSNAME. Placing the command(JSTS 1234T<CR>) in line and the character count (11) in NCHAR. The JSTS subsystem is then called to produce the job status message.

f) Call ULASCI (ARRAY,CPOS,NCHAR,L)

This entry point is used to force ASCII data in an array to upper case or lower case ASCII code. This is occasionally useful, inasmuch as most teletypes transmit only upper case, and certain pieces of software (TSS FORTRAN, for example) recognize only upper case ASCII, whereas lower case is somewhat easier to work with and hence is required by other timesharing software. ULASCI will transliterate either way. The data in the ASCII array, ARRAY, is acted upon, starting at character position CPOS (INTEGER) with NCHAR characters. If L is even, the data will be forced upper case; if L is odd, it will be forced lower case. Note: The character position, CPOS is handled in the same way as in the standard TSS subroutines, Get #C, Put #C and Move #S.

g) Call BCDASC (FROM, IPOS, TO, JPOS,NCHAR,FILL)

h) Call ASCBCD (FROM, IPOS, TO, JPOS,NCHAR,FILL)

These two routines function as a move #S with transliteration as indicated. IPOS and JPOS are starting character positions in FROM and TO, respectively. Bear in mind that there are 4 ASCII characters per word, and 6 BCD characters per word. FILL is a blank fill control. If the integer variable fill is zero, the array, TO, is not blank-filled. If FILL is nonzero, the last receiving word of the array will be blank-filled.

[2] Language: GMAP

[3] Input: Arguments

[4] Output: Return Arguments

[5] Called by: LINKM3, DSTPLT

2.2.19 Utility Package PACK

[1] Description: This utility package was developed by AFCS personnel to manipulate character strings. It contains three entry points UNPACK, PACK, and MOVE which are described below.

UNPACK is used to convert a packed character string, in either BCD or ASCII, into an unpacked character string (one character per word, left justified, with trailing blanks in each word) of the same mode (BCD or ASCII). The calling sequence is:

```
CALL UNPACK(LINE,I,ARRAY,J,N)
```

where LINE is the packed character string, and ARRAY is the unpacked character string. N characters are moved, starting with the Ith character of LINE and the Jth character of ARRAY. The mode of the character strings is assumed to be ASCII; this may be altered as follows:

```
CALL UNPACK( ,0)      changes the mode to BCD
CALL UNPACK( ,1)      changes the mode back to ASCII.
```

PACK has the opposite effect of UNPACK; it moves characters from the first to third arguments, packing them as it goes. MOVE moves a packed string from the first to third arguments. PACK and MOVE each have a sixth argument which, if nonzero, causes the last word into which data was moved to be filled with blanks to the end of the word. Calling PACK or MOVE with a null first argument will set the mode (BCD or ASCII) as above; the mode currently in effect applies to all three entry points.

[2] Language: GMAP
[3] Input: Arguments
[4] Output: As described above
[5] Called by: DSTPLT

3. INPUT/OUTPUT FILES DESCRIPTION

3.1 General Description

In order to communicate among the various systems used here, a series of temporary and permanent files have to be created. A list of these files with a description of their purpose follows:

3.1.1 List of Files

The list of the files is presented in the order they are created or used.

03 / RADIO/NORUN — Permanent file where the numbers of the different runs are stored for reference with the output data and plots attached by LINKMN

01 — Temporary file created by LINKMN to save all the data computed by /RADIO/DISTORT time-sharing subsystem to be used by the batch /RADIO/USA subsystem.

02 — Temporary file created by LINKMN where the $LIMITS and $PRMFL cards required by /RADIO/USA are stored. These statements are written by LINKM3.

DISTRT/XXXX — Permanent file created by LINKM3 that will be used by /RADIO/USA as its output file. XXXX is the number of the run stored in /DISTRT/NORUN.

/RADIO/USA.RUN1 — Permanent file containing all the necessary job stream cards and files to run /RADIO/USA

/RADIO/IDENT — Permanent file that has the $IDENT card. The reason for this is so that it can be altered by different users.

RADIO/(filename) — Permanent file created by the AFCS system SCREEN where all the pertinent topographical data of a site designated by (filename) is stored. This site is used by RADIO/USA to calculate communication range contours.

3.1.2 Detailed File Description

3.1.2.1 File 03 /RADIO/NORUN

[1] File Content: This file contains the numbers assigned to the different runs for identification purposes.

[2] Written by: LINKM3

[3] Read by: LINKM3

[4] File Type: Sequential

[5] File Mode: Binary

$:PRMFL:03,W/R,R, BLA00001/RADIO/(filename)

[2] Written by: LINKM3

[3] Read by: /RADIO/USA.RUN1

[4] File Type: Sequential

[5] File Mode: ASCAII

3.1.2.2 File 01

[1] File Content: This file contains the output of the /RADIO/DISTORT time-sharing system.

[2] Written by: LINKM3

[3] Read by: /RADIO/USA

[4] File Type: Sequential

[5] File Mode: ASCII

3.1.2.3 File 02

[1] File Content: This file contains the $LIMITS and $PRMFL cards necessary to run /RADIO/USA. It can have at most the following

```
$:LIMITS:XX,XXXK, 10K
$:PRMFL:02,W,S,BLA00001/DISTRT/XXXX
```

3.1.2.4 File / DISTRT/XXXX

[1] File Content: This file contains the output of the batch /RADIO/USA subsystem to be used by the /RADIO/DISTORTP subsystem.

[2] Written by: MAIN, PATT, NPAT, PATD

[3] Read by: DSTPLT

[4] File Type: Sequential

[5] File Mode: Binary

[6] Record Length: Variable

[7] Record Format:

Record Type	DATA
1	ANT#(FED),FREQ(MHZ)
2	EMAX,GAIN,GAIN(DB)
3	THETA or PHI, NMAG,NMAG(dB)
4	PHI (vertical pat.)
5	THETA (horizontal pat.)

3.1.2.5 File / RADIO/USA.RUN1

[1] File Contents: This file contains the necessary JCL to run /RADIO/USA. A listing of this file looks like

```
SYSTEM ?LIST /RADIO/USA.RUN1
##ASIS N
$       SELECTA BLA00001/RADIO/IDENT
$       OPTION  FORTRAN

$       LIBRARY LB
$       USE     .......
$       ENTRY   ......
$       EXECUTE DUMP
$       SELECTA 02

$       DATA    01
$       SELECTA 01
$       PRMFL   LB,R,R,BLA00001/RADIO/USA
$       ENDJOB
```

[2] Written by: Created by the user

[3] Read by: LINKM3

[4] File Type: Sequential

[5] File Mode: ASCII

3.1.2.6 File /RADIO/IDENT

[1] File Content: This file contains the $IDENT card to allow different users to modify it easily:

```
$    IDENT    BLA00001,PERINI-SW,956700160121,DISTORT
```

[2] Written by: Created by the user

[3] Read by: /RADIO/USA.RUN1

[4] File Type: Sequential

[5] File Mode: ASCII

3.1.2.7 File RADIO/(filename)

[1] File Content: This file contains the topographical data of

the site identified by (filename). This file is created by the AFCS system SCREEN.

[2] Written by: SCREEN System (AFCS)

[3] Read by: PATT

[4] File Type: Random

[5] File Mode: Binary

[6] Record Length: 9 words

[7] Record Format:

Field Name	Content Description
ZZ	Azimuth in radians
ELANG	Elevation Angle in radians
DIST	Distance to screen in feet

RNG(1)
⋮
RNG(6)
	Ranges in nautical miles

4. PROGRAM ASSEMBLING, LOADING, AND MAINTENANCE PROCEDURES

4.1 Source Program Storage

The source programs are stored in the file RADIO/RADAR and also on magnetic tape.

4.2 System and Subsystem Creation

This is done under the FILE EDIT system. All the necessary cards are stored in the file /RADIO/JOB. A list with explanation follows.

4.2.1 List of /RADIO/JOB

```
10$      IDENT   BLA00001,PERINI-SW,956700160121,RADIO-JOB
20$      FILEDIT SOURCE,NOBJECT,UPDATE
30$      LIMITS  1,,,9K
40$      PRMFL   H*,R,S,BLA00001/RADIO/RADAR
50$      FILE    K*,D1SR,6L
60$      DATA    *C,,COPY
70$      MODIFY  SOURCE,,DSTRTP
80$      FORTRAN ASCII,NFORM,NLNO,OPTZ                                    DSTRTP
90$      UPDATE  LIST
100$      ALTER   1
110       SUBROUTINE DSTPLT
120$      ENDEDIT
130$      ENDCOPY
140$      FILEDIT SOURCE,OBJECT,INITIALIZE,NONE
150$      LIMITS  5,,,9K
160$      FILE    *C,D1R
170$      FILE    K*,NULL
180$      FILE    R*,D2SR,5L
190$      FILE    P*,NULL
200$      SYSOUT  *5
210$      PROGRAM RANLIB
220$      LIMITS  1,,,9K
230$      FILE    R*,D2SR
240$      FILE    A4,D3SR,10R
250$      RELCOM  40
260$      OPTTON  FORTRAN,NOFCB,NOGO,SAVE/......
270$      LIBRARY LB
280$      USE     LINKMN
290$      USE     .GTLIT,.TSGF.,.FTSU.,.FXEMA
300$      ENTRY   LINKMN
310$      EXECUTE DUMP
320$      LIMITS  1,20K,,9K
330$      FILE    LB,D3SR
340$      PRMFL   H*,R/W,R,BLA00001/RADIO/DISTORT
350$      OPTION  FORTRAN,NOFCB,NOGO,SAVE/......
360$      LIBRARY LB
370$      USE     DSTPLT
380$      USE     .GTLIT,.TSGF.,.FTSU.,.FXEMA
390$      ENTRY   DSTPLT
400$      EXECUTE DUMP
410$      LIMITS  1,15K,,9K
420$      FILE    LB,D3R
430$      PRMFL   H*,R/W,R,BLA00001/RADIO/DISTORTP
440$      BREAK
450$      FILEDIT NOSOURCE,OBJECT,NONE
460$      LIMITS  1,,,9K
470$      FILE    *R,D2R
480$      FILE    R*,D4SR,5L
490$      DATA    *C,,COPY
500$      DELETE  ,,PACK
510$      ENDEDIT
520$      ENDCOPY
530$      PROGRAM RANLIB
540$      LIMITS  1,,,9K
550$      FILE    R*,D4R
560$      PRMFL   A4,R/W,R,BLA00001/RADIO/USA
570$      ENDJOB
```

4.2.2 Explanation of /RADIO/JOB

The file BLA00001/RADIO/JOB contains all of the programs that were on tape we sent you, and they are in the same format as on the tape. The first activity, a FILEDIT, merely modifies the main plotting routine, DSTRTP. A SUBROUTINE DSTPLT statement is added to change the SYMDEF of the routine from .... to DSTPLT so that all of the routines will have unique SYMDEFs. Without the change, there would be two routines (DSTRTP and DIMXXX) with the same SYMDEF (......), and since programs which have been placed on a random library by the program RANLIB are accessed by SYMDEF, the second of the two would then be inaccessible.

The second activity, another FILEDIT, compiles or assembles all of the programs and places the object decks on a single output file, R*. This file is used as input to the third activity, RANLIB, where the object decks are reformatted and placed on a random library file.

The fourth activity loads the timesharing input programs and saves the core image of these programs onto an H* file. The $ RELCOM card is required since the programs are highloaded, causing the COMMON to be allocated in low core. Without the $RELCOM, COMMON allocation would begin at 100 octal, just above the slave prefix. This is correct for a batch program; timesharing, however, requires a larger slave previx (144 octal instead of 100 octal words), so COMMON must be relocated above timesharing's slave prefix. This is not required in the fifth activity, since there is no blank common. The name following the SAVE option of the $OPTION card is arbitrary, since there is only a single core image saved on the file, and it is not used as an overlay nor accessed by a MME GECALL/GERSTR or

DRL RESTOR. The $ USE and $ ENTRY cards are used to inform the loader that the program whose SYMDEF is LINKMN is to be loaded (in this case from the random library created in the previous activity), and that execution is to begin with this program (later on, in timesharing, when the program is actually executed). The other $ USE card forces loading of special versions of several FORTRAN library routines which replace the BCD- and batch-oriented routines and allow executing in an ASCII time-sharing environment. The fifth activity is similar to the fourth, except that the timesharing output/plotting routines are being loaded and saved. The sixth and seventh activities delete the timesharing programs from the collection of object decks produced in activity 2, leaving only the batch object decks. These are then placed on a random library file for later execution.

The procedure described above, using FILEDIT and RANLIB, was used in our current implementation of the DISTORT system; however, you should be aware that many variations of this procedure are possible.

4.2.3 Program operation

The file BLA00001/RADIO/NORUN is a binary file containing a single work of binary information, along with the three words of standard GFRC control information. This single word is a binary integer; it is read, incremented, and written back in lines 7790 thru 7840 of LNKM3A. It is used thereafter in LNKM3A to specify a unique file name. This file is created and erased, deaccessed, specified for file code 02 in the spawned batch job, used by a batch job to contain output to be passed to the plot routines, read and released by the plot routines, and lastly, the file name is printed out as a "program execution number" to allow the user to keep track of the input and output for each execution of the program.

File 01 in the timesharing input program contains the data specifying the problem; file 02 contains the $ LIMITS card specifying the appropriate output files for the batch job. The amount of core required for a given execution of the batch program is calculated on line 7700 of LNKM3A; this statement allows approximately 27K words for the program and file buffers, and an additional 1K for each 1K or fraction thereof required for the two variable arrays, C1 and C2. The amount of time required is calculated on line 7710; it is arbitrarily set at 50/100 hour, since we are not aware of an algorithm by which a better estimate of the time required can be calculated.

REWIND and FCLOSE are used on files 01 and 02 in line 8040 of LNKM3A, since the buffers for these two files must be completed and written out to the files prior to calling the CARDIN subsystem. The subroutine CALLTS (part of YSFUTL) is called to invoke the ACCESS, REMOVE, and CARDIN subsystems of timesharing to create, erase, and deaccess a permanent file, and to spawn a batch job.

CALLTS is one of several FORTRAN-callable entry points within the YSFUTL utility package. Nearly all of the code pertinent to its operation is on the listing of YSFUTL. Its operation involves two steps. The first step, lines 15020 thru 15230 is to perform a DRL PSEUDO, which simulates input from the keyboard. Keyboard input is simulated because the subsystem which is subsequently called will frequently examine the last line of input typed in at the keyboard to determine the exact nature of the function to be performed. The second step of the procedure, lines 15240 thru 15380, is to invoke the desired subsystem via DRL CALLSS. When this DRL is executed, the program that executed the instruction is interrupted, its state is automatically saved by the TSS Executive, and the specified subsystem is loaded and executed. After the subsystem has terminated, the program which executed the DRL CALLSS is reinstated and its execution resumed at the point where it was interrupted.

UNPACK is one of the FORTRAN-callable entry points within the PACK utility package. UNPACK is used to convert a packed character string, in either BCD or ASCII, into an unpacked character string (one character per word, left justified, with trailing blanks in each word) of the same mode (BCD or ASCII). The calling sequence is:

```
CALL UNPACK (LINE,I,ARRAY,J,N)
```

where LINE is the packed character string, and ARRAY is the unpacked character string. N characters are moved, starting with the Ith character of LINE and the Jth character of ARRAY. The mode of the character strings is assumed to be ASCII; this may be altered as follows:

```
CALL UNPACK (,0)     changes the mode to BCD
CALL UNPACK (,1)     changes the mode back to ASCII.
```

PACK has the opposite effect of UNPACK; it moves characters from the first to third arguments, packing them as it goes. MOVE moves a packed string from the first to third arguments. PACK and MOVE each have a sixth argument which, if nonzero, causes the last word into which data was moved to be filled with blanks to the end of the word. Calling PACK or MOVE with a null first argument will set the mode (BCD or ASCII) as above; the mode currently in effect applies to all three entry points.

DIMXXX automatically allocates storage for the two arrays, C1 and C2, depending upon the amount of storage available to be allocated to these two arrays, which is in turn dependent upon the $ LIMITS card generated by LNKM3A when the batch job was spawned. When the batch program is loaded for execution, the instructions are loaded at the high end of the momory segment allocated by the $ LIMITS card. As a result, "unused" space may exist near the low end of memory. DIMXXX calls .FOPEN to allocate I/O buffers for files 01, 02, and 06, since these buffers are taken from this "unused" space. The amount of "unused" space remaining is then calculated, and the maximum dimension for C1 and C2 which will fit into the remaining space is computed. A MME GESNAP instruction is executed to document the locations and dimensions of C1 and C2, and this information is then passed to the main batch routines via a CALL statement and adjustable dimension procedures.

The batch FORTRAN routines were modified wherever a WRITE to file code 02 appeared. In some cases, such as the coupling coefficient printouts (which are not plotted) and where the printout is purely textual with no variable information, the WRITE statements were deleted altogether (converted to COMMENTS). In the other cases the original WRITE statement was converted to a COMMENT and a new BINARY (unformatted) WRITE statement was inserted in its place. These new statements convey the same variable information as the original (unformatted) WRITE statements, with the addition of RECORD TYPE and RECORD LENGTH variables at the beginning of each record, and without the textual information. These changes have resulted in a somewhat reduced filespace requirement for the output file (150 words versus 198 words, or a 25% reduction, for the sample run included in this package; in general, the reduction is on the order of 13%). More importantly, though, the file is more readily machine-readable than it was before. The record can be read at once with a single binary read statement; no format conversion is required; the exact nature and amount of data read is immediately available to the plotting program, without having to go through the trial-and-error series of IF and DECODE statements required to interpret the formatted output.

The plotting routines (DSTRTP, DSTRP1, and Y.LABL, with utility routines YSFUTL and PACK) allow the user to plot the antenna patterns generated by a given execution of the batch portion of the DISTORT system. DSTRTP first requests the "program execution number" which is the name of the file to be plotted; the file is accessed and an initial scan of the data on the file is made, tabulating gain figures and printing out a summary of the data on the file. The user is then asked a series of questions to determine which portion of the data on the file is to be plotted and the way in which the user desires to have the data presented. After this information has been obtained, the plotter is activated and the plot produced. Upon completion of the plot, the user is given several options to either rerun a portion of the plotting program or release the data file if it is no longer needed.

The user is given the option of plotting the vertical or horizontal patterns for a given run, or both may be plotted on the same graph. The user may also plot an isotropic circle if he so desires. The plot may be presented in any one of three different ways: scaled, normalized or family. If "normalized" is selected, the horizontal and/or vertical plots will (both) be normalized, i.e. will extend to the usable limits of the graph paper at the points of maximum gain. The size of the isotropic circle will then be scaled to indicate the relative magnitudes of the isotropic and the pattern plotted. Since only one isotropic circle is plotted, the user will be asked which pattern the circle is to be related to in the event that both the horizontal and vertical patterns are being plotted in normalized form. If "scaled" is selected, the horizontal, vertical, and isotropic patterns for the run to be plotted will all be scaled relative to each other so that their relative magnitudes can be readily seen. Finally, if "family" is selected all runs for this file will be scaled relative to each other. This allows the user to see how the gain, as well as the shape of the pattern, varies from run to run as the frequency is changed.

After the appropriate scaling factors have been computed, the user is given an opportunity to ready the plotter for operation; plotting then begins. The data points to be plotted are read by DSTRTP; the scaling and output of the data to the plotter for both patterns and the isotropic circle is done by DSTRP1, which is called once for each data point to be plotted. Only a few points on the isotropic circle are plotted; these appear as dots at 45 degree intervals. If the user requested that the program stop between plots, the message CHANGE PENS will appear between each pattern, affording the user an opportunity to change the color of the pen before continuing. Lastly, Y.LABL is called to place the program execution number and run number in the upper lefthand corner of the plot. The user may then plot another run from the same file, or he may release the file if no longer needed and then begin plotting from another file or terminate the plotting program.

Y.LABL produces alphanumeric characters on the plotter. It is capable of producing any of the 95 printable ASCII characters, although in this application it produces only numerics and the hyphen. It can get its data from labeled common, as it does in this case, or from the keyboard or from a disk file containing the answers that would be typed at the keyboard. It is capable of many variations in scaling. A file called BLA00001/PLOTTER/ALPHANUM, a random data file, contains descriptions of the shapes of the characters to be plotted. You do not have a copy of the data in this file, and it would be somewhat difficult for you to duplicate its contents; this is of little consequence, however, since you also do not have a plotter to use the data and since the program, when it is unable to access the file, will simply print an error message to that effect and return.

4.2.4 HP 7202A graphic plotter

The graphic plotter produces graphs by drawing straight line segments between successive coordinates or by plotting points at the coordinates. Every point is uniquely defined by a 4 digit X and a 4 digit Y coordinate pair. Each coordinate may range from 0000 to 9999. Line segments may be up to 3 inches in length; coordinates are accepted at a maximum rate of 1.1 coordinate pairs per second. Since data is transmitted from the computer at 30 characters per second, this last requirement necessitates adding "fill" characters to take up additional time. If each plotter command or coordinate pair is preceded by about 10 blanks, the combined delay of the blanks, the data itself, and the several nonprinting character inserted in each line by timesharing should be sufficient.

The plotter is manually adjusted so that position 0.0 corresponds to the lower left corner of the usable space on the graph paper; it is then manually adjusted so that 9999,9999 corresponds to the upper right corner. Thus, any coordinate pair sent to the plotter will cause the pen to move to a corresponding point on the grid area of the graph paper.

When the plotter is turned on, receipt of the characters PLTL enables the plotter; line segments will be drawn between the coordinates transmitted on lines following this command. PLTP causes the points to be plotted; the pen will move to the specified coordinates, make a dot on the paper, and then lift from the paper before moving to the next coordinate. PLTT terminates plotting and disables the plotter. The carriage return character causes the pen to move to a new position, and the line feed character prepares the plotter's logic to accept the next pair of coordinates. The up-arrow or circumflex character immediately following the Y coordinate in line mode causes the pen to lift prior to moving to that position. This allows pen lift while plotting lines with PLTL. A typical line containing coordinate information will begin with 10 spaces, followed by a 4 digit X coordinate, followed by one space, followed by a 4 digit Y coordinate, followed by a carriage return and line feed.

4.2.5 Sample Timesharing Runs

The $ IDENT card image is contained on a file separate from the rest of the JCL for the batch portion of DISTORT, so that the $ IDENT card can readily be altered as needed by using personnel, without disturbing the rest of the JCL. The $ IDENT card image is inserted into the JCL stream by CARDIN in response to the $ SELECTA card image. The $ LIBRARY, $ USE, $ ENTRY, and $ PRMFL cards allow the loader to load and execute the batch

DISTORT program from the random library file on which the program resides. Note that the batch program must be reloaded each time it is to be used, since varying memory size allocations dictate loading the program at different locations in memory and making corresponding changes to constants used by FORTRAN's library routines, a task accomplished by the loader. This is in contrast with the timesharing segments of the system, whose memory allocation sizes do not vary from one execution to the next and which, therefore, can be loaded once by the loader; thereafter they are simply recalled from the H* file and executed, without going through the usual loading procedure. The two $ SELECTA cards following the $ EXECUTE cause CARDIN to include the $ LIMITS and $ PRMFL 02 card images and the data card images in the JCL stream.

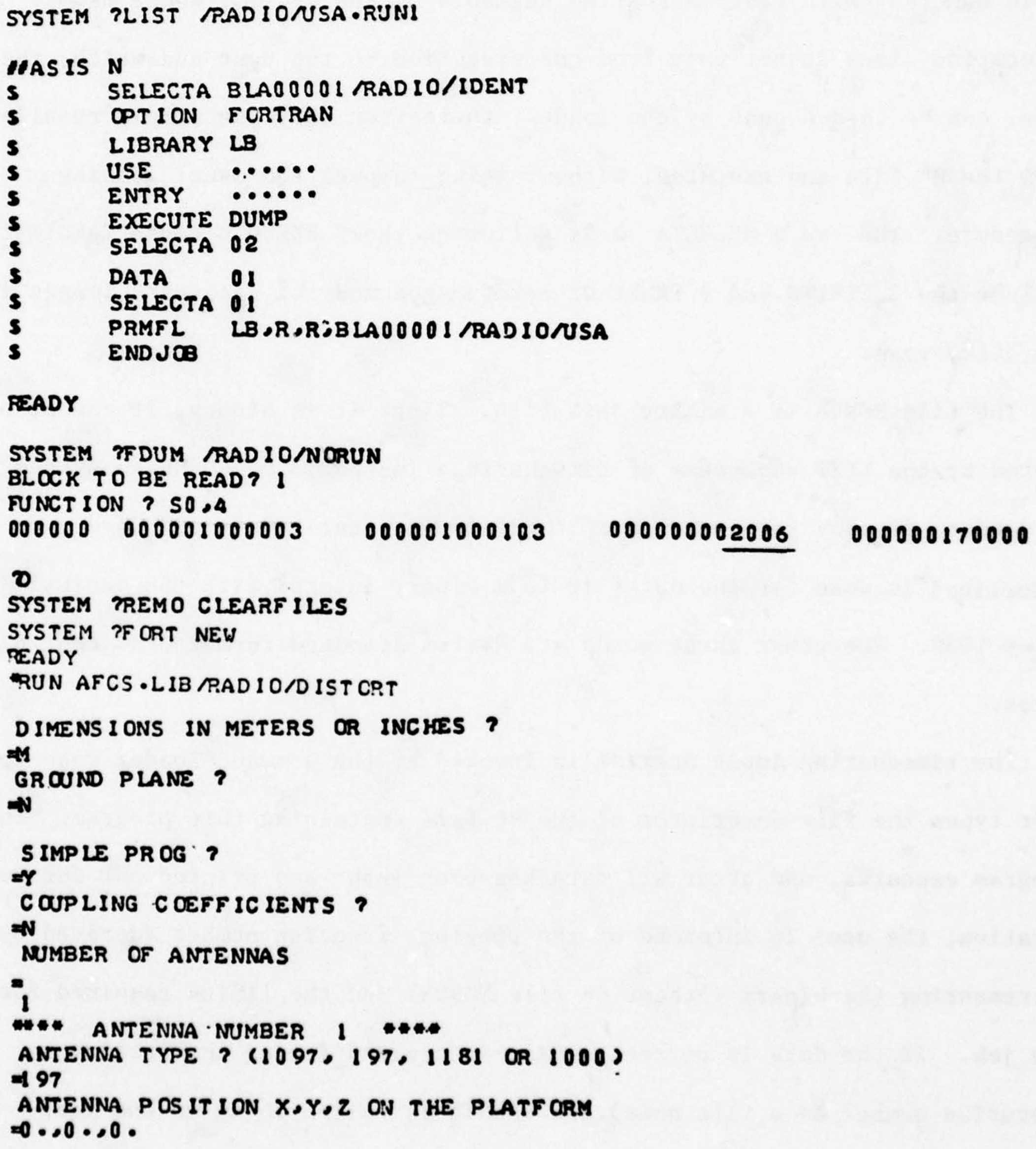
The file NORUN is a binary data file. Since it is binary, it cannot be listed by the LIST subsystem of timesharing; instead, the FDUMP subsystem is used to display the contents of the file in octal. Only one word underlined is used for the data; it is a binary integer with the decimal value 1030. The other three words are system standard format GFRC control words.

The timesharing input program is invoked by the command loader when the user types the file descriptor of the H* file containing this program. The program executes, and after all data has been input and printed out for verification, the user is informed of the program execution number (derived by incrementing the binary integer on file NORUN) and the limits required for the job. If the data is correct a file is created (using the program execution number as a file name) and CARDIN is called to spawn the job.

Upon completion of the batch job the plotting routines are similarly invoked using the command loader.

A list of a sample run follows.

4.2.6 List of Sample Timesharing Runs



```
SYSTEM ?LIST /RADIO/USA.RUN1

//ASIS N
$       SELECTA BLA00001/RADIO/IDENT
$       OPTION  FORTRAN
$       LIBRARY LB
$       USE     .......
$       ENTRY   .......
$       EXECUTE DUMP
$       SELECTA 02
$       DATA    01
$       SELECTA 01
$       PRMFL   LB,R,R;BLA00001/RADIO/USA
$       ENDJOB

READY

SYSTEM ?FDUM /RADIO/NORUN
BLOCK TO BE READ? 1
FUNCTION ? S0,4
000000  000001000003    000001000103    000000002006    000000170000

?D
SYSTEM ?REMO CLEARFILES
SYSTEM ?FORT NEW
READY
*RUN AFCS.LIB/RADIO/DISTORT

 DIMENSIONS IN METERS OR INCHES ?
=M
 GROUND PLANE ?
=N

  SIMPLE PROG ?
=Y
 COUPLING COEFFICIENTS ?
=N
 NUMBER OF ANTENNAS
=
 1
****  ANTENNA NUMBER  1  ****
 ANTENNA TYPE ? (1097, 197, 1181 OR 1000)
=197
 ANTENNA POSITION X,Y,Z ON THE PLATFORM
=0.,0.,0.
```

```
+++ RADIATION PATTERN +++
 VERTICAL PATTERN ?
=Y
 PHI (DEGREES)
=0.
 PLOTTING INCREMENT (NON-ZERO NUMBER) (DEGREES)
=10
 HORIZONTAL PATTERN ?
=
 N
 COMMUNICATION RANGE CONTOUR ?
=N

  NUMBER OF RUNS
=1
**** RUN # 1 ****
  FREQUENCY (MHZ)
=320.
  FED ANTENNA (#)
=1

 DIM= M  GP= N  SIMP= Y  COUPL= N  NR=  1
 ANT# =     1
 TYPE =     197
 X    =   0.
 Y    =   0.
 Z    =   0.
 VER PAT=  Y
 PHI  =   0.    PLOT INC=  10.00
 HOR PAT=  N
 COM RNG=  N


 RUN#      FREQ(MHZ)  ANT FED(#)
  1        320.00         1
 DATA CORRECT ?
 =Y
 PROGRAM EXECUTION NUMBER-1031
 ARRAY DIMENSIONS =  7
 JOB REQUIRES 50/100 HR TIME AND 028K WORDS CORE.
   SNUMB # 3168T
 RERUN?
 =N



SYSTEM ?STAT


CHANNEL 4200
USER STATUS ON MAR 16,1978 AT 15:50:35   LOG-ON AT 15:30:40
PROC TIME USED  2.54 SEC., 60 FILE I/O 6400 CHAR KEY I/O
LIST OF OPEN FILES: 01         02          USA.RUN1


SYSTEM ?LIST 01;02
```

```
  1  26    1 0 0. 0.
 0.480   0.030   0.450   0.450   0.450   0.450   0.450   0.450   0.450
 0.450   0.450   0.450   0.450   0.450   0.170   0.170   0.170   0.170
 0.170   0.170   0.170   0.170   0.170   0.170   0.170   0.170
0.01750 0.00200 0.00450 0.00450 0.00450 0.00450 0.00450 0.00450
0.00450 0.00450 0.00450 0.00450 0.00450 0.00450 0.00350 0.00350

0.00350 0.00350 0.00350 0.00350 0.00350 0.00350 0.00350 0.00350

0.00350 0.00350
 0.      0.      0.      0.      0.      0.      0.      0.      0.
 0.      0.      0.      0.      0.      0.      0.      0.      0.
 0.      0.      0.      0.      0.      0.      0.      0.
 0.      0.      0.      0.      0.      0.      0.      0.      0.
 0.      0.      0.      0.      0.      0.      0.      0.      0.
 0.      0.      0.      0.      0.      0.      0.      0.
 0.      0.480   0.480   0.480   0.480   0.480   0.480   0.480   0.480
 0.480   0.480   0.480   0.480   0.480   0.510   0.510   0.510   0.510
 0.510   0.510   0.510   0.510   0.510   0.510   0.510   0.510
 1   0   0   0    0.      0.     10.00    0.      0.
 197
  7    0    1    0    0 320.00
0.      1.000   0.      0.       0.      0.      0.      0.      0.
0.      0.      0.      0.       0.      0.      0.      0.      0.
0.      0.      0.      0.       0.      0.      0.      0.
   2    1    3    3    3    3    3    3    3    3    3    3    3    3    1    1
   1    1    1    1    1    1    1    1    1    1
   1    3    4    4    4    4    4    4    4    4    4    4    4    4    7    7

   7    7    7    7    7    7    7    7    7    7
   1    3    4    4    4    4    4    4    4    4    4    4    4    4    7    7
   7    7    7    7    7    7    7    7    7    7


$:LIMITS:50,028K,,20K

$:PRMFL:02,W,S,AFCS.LIB/DISTRT/1031


READY

SYSTEM ?CATA DISTRT


    LIST OF CATALOG-DISTRT
    CATALOGS
      FILES



      1031

SYSTEM ?FDUM /RADIO/NORUN
 BLOCK TO BE READ? 1
 FUNCTION ? S0,4
 000000  000001000003     000001000103     000000002007     000000170000

 D
```

4.3 Program Changes Procedures

As stated in 4.1 all programs are stored in source format in the RADAR file. Therefore, they can be changed by the normal procedures available to any TSS user. If new programs/subroutines/functions are written, they should be merged in the file RADAR with the proper $ (System) card for compilation. The File EDIT system as lised in /RADIO/JOB can then be rerun, updating all subsystems.

4.4 Program Verification

Even if the program has been modified, it should still produce the same results as previous runs (see User's Manual) that are not affected by the changes. Programmers should provide test runs to verify that modifications are working properly.

4.5 Error Conditions

There are two types of error conditions: system errors and user's errors.

4.5.1 System Errors

These are due to improper use of the computer system and are listed in the appropriate manuals.

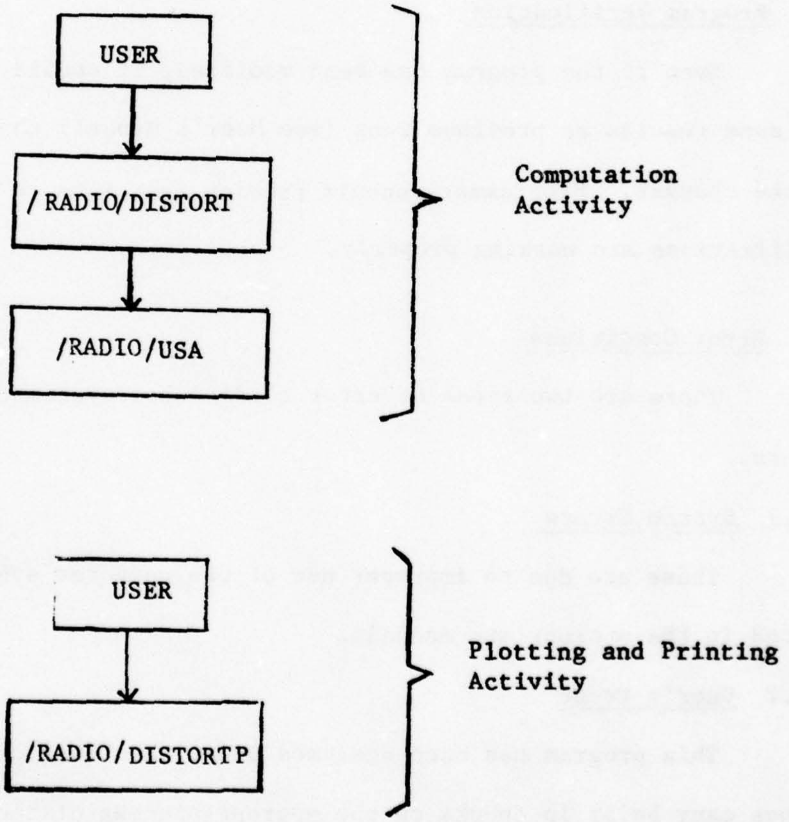
4.5.2 User's Error

This program has been designed under the "user oriented" philosophy. It has many built in checks on the appropriateness of the answers. For example, if the user tries to feed a VHF antenna with a UHF frequency, the program will detect the error and request the frequency again (see User's Manual). Of course, if the wrong VHF frequency is entered, the program will accept it, since it is a valid frequency for the VHF antenna.

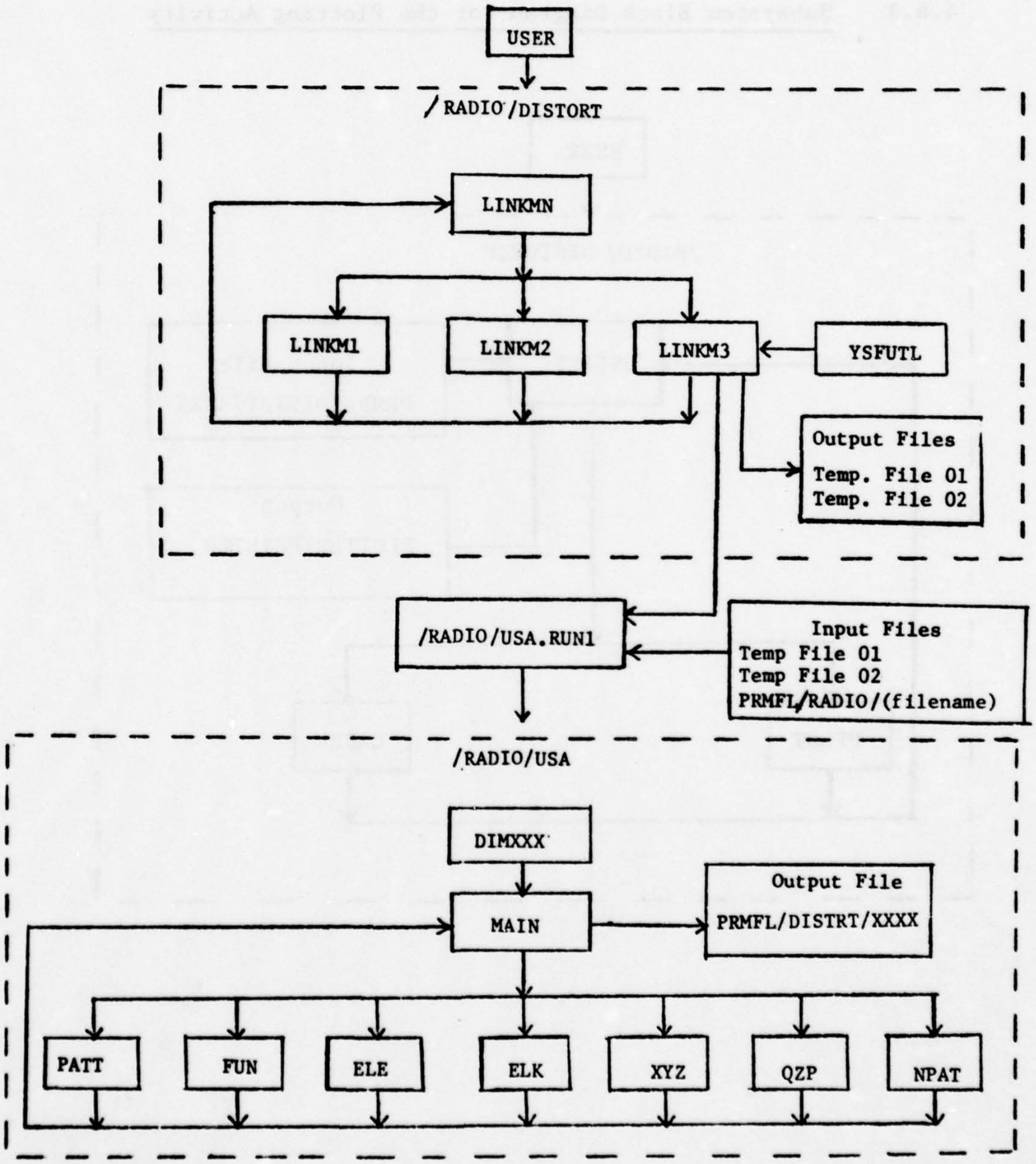
4.6 Flow Charts/Block Diagrams

In the next sections flow charts/block diagrams for the overall system and for each program are presented.

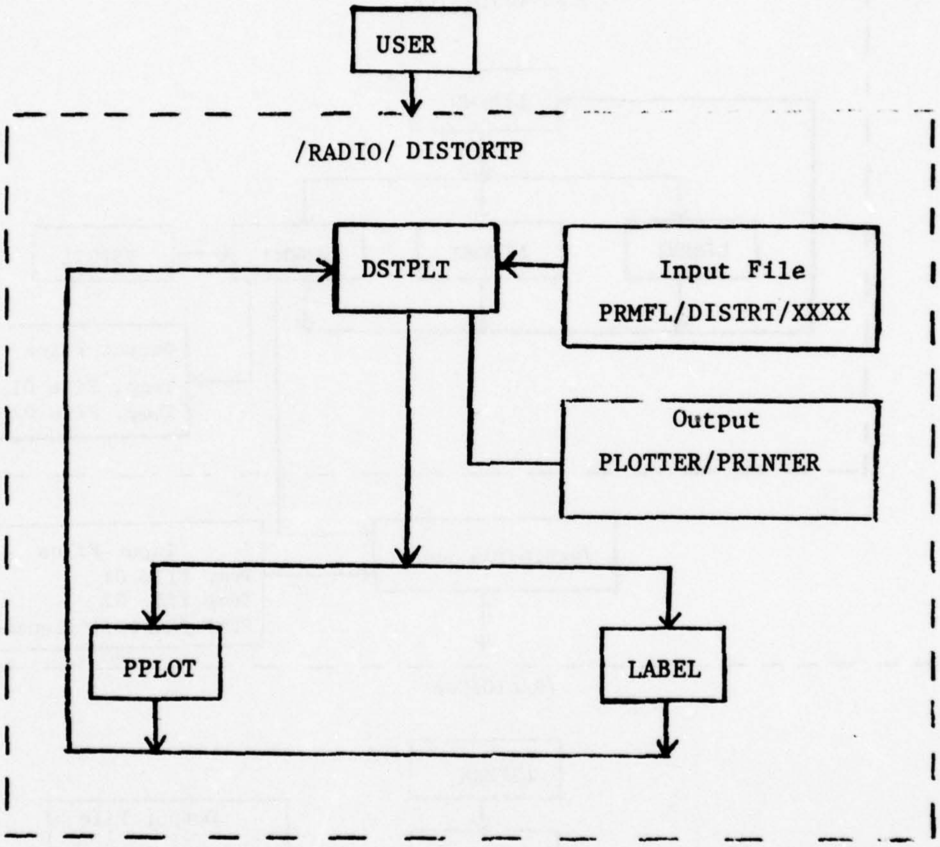
4.6.1 Overall System Block Diagram



4.6.2 Subsystem Interconnection Block Diagram for the Computing Activity

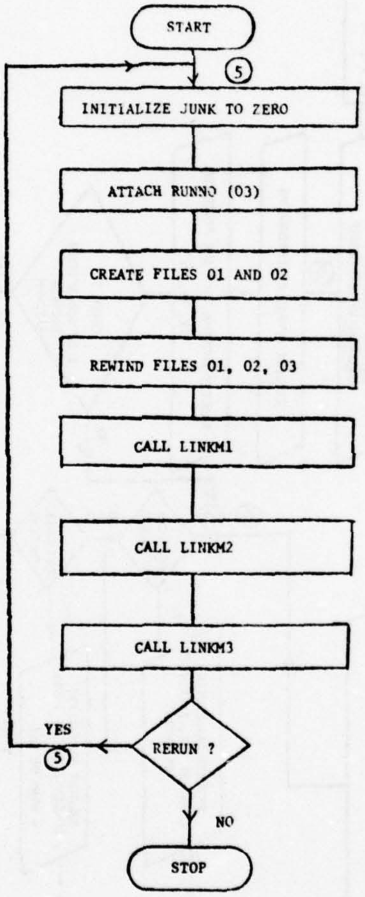


4.6.3 Subsystem Block Diagram for the Plotting Activity

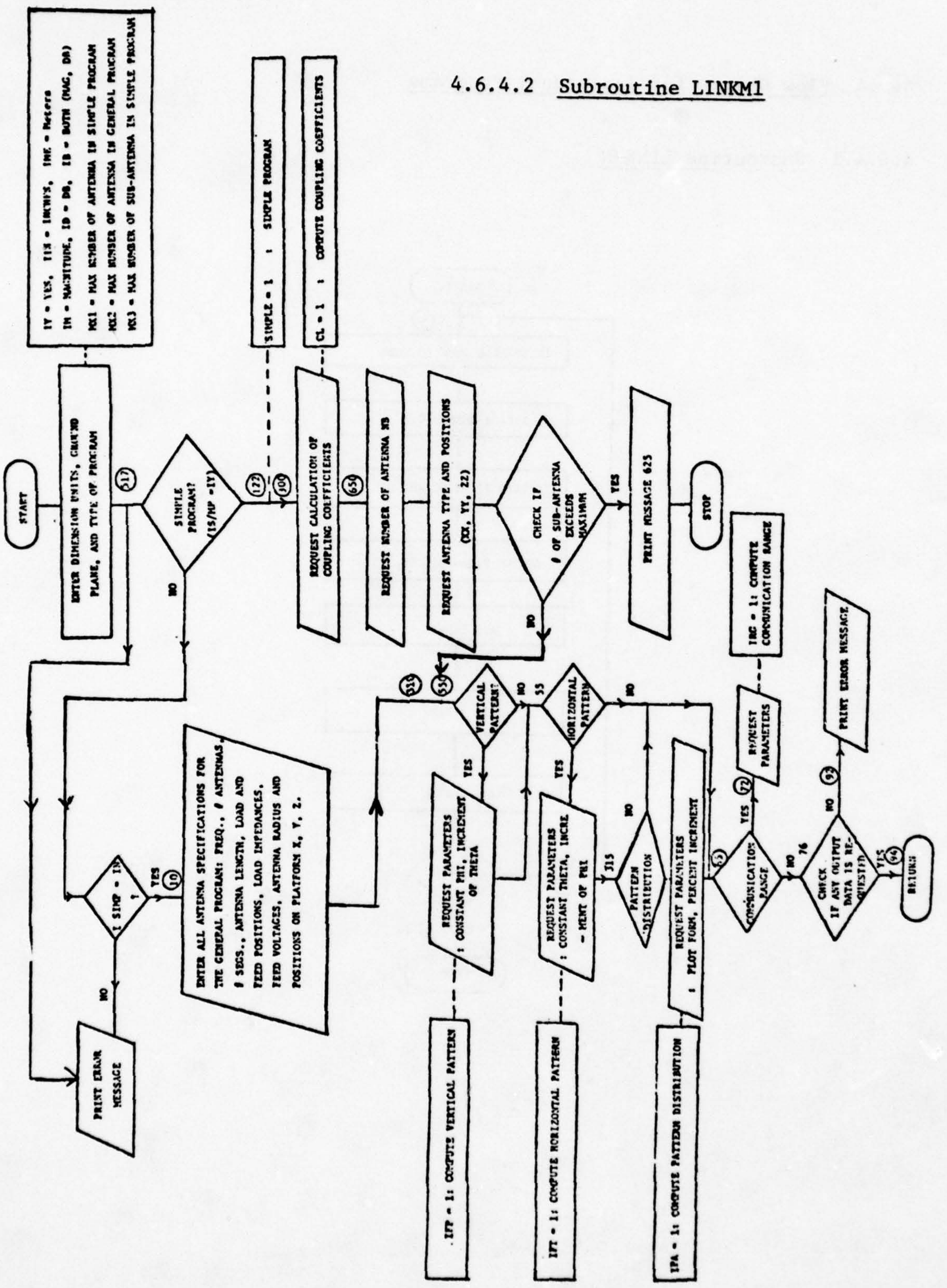


4.6.4 Flow Charts for Individual Programs

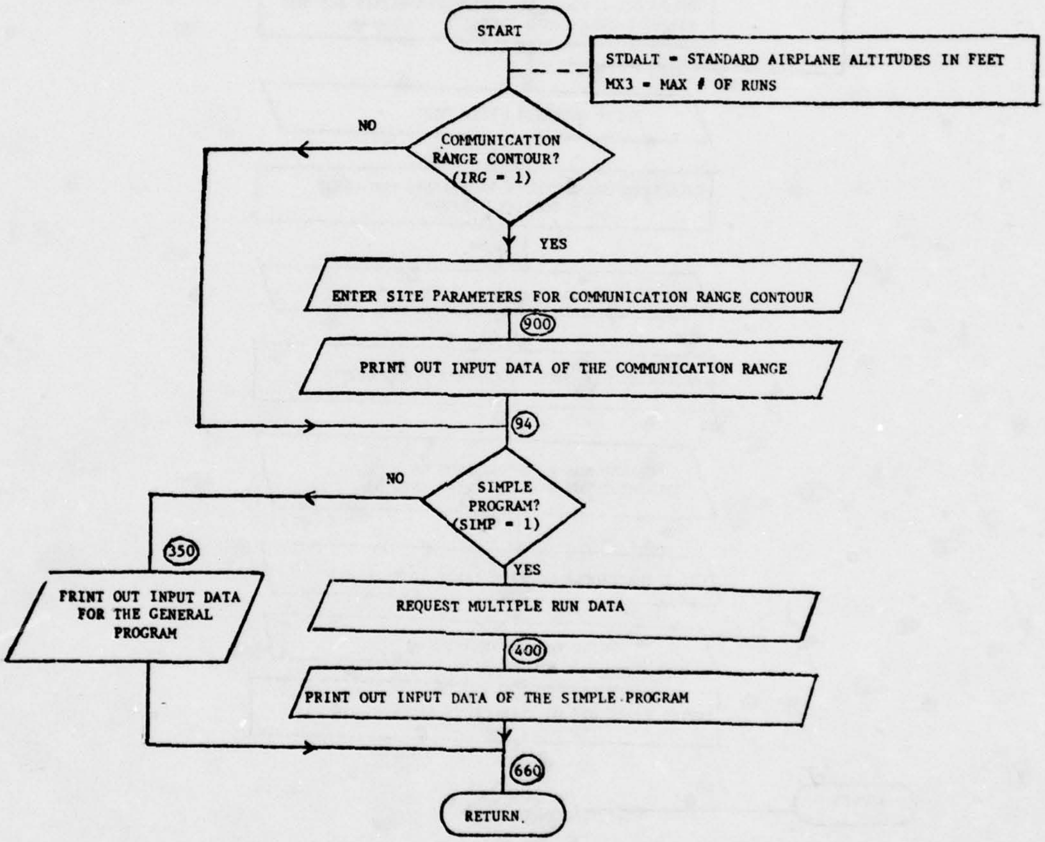
4.6.4.1 Subroutine LINKMN



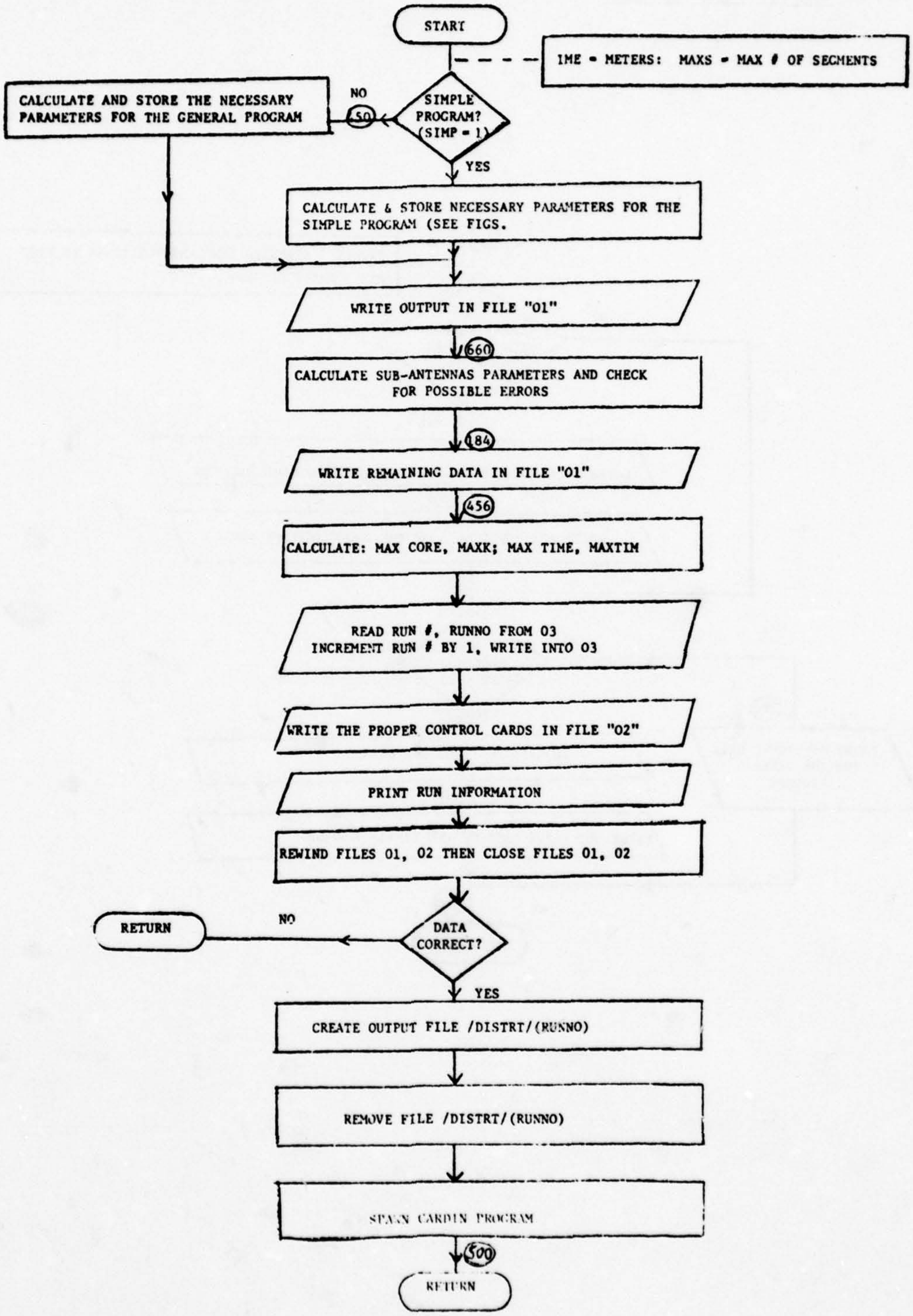
4.6.4.2 Subroutine LINKM1


START
ENTER DIMENSION UNITS, GROUND PLANE, AND TYPE OF PROGRAM
IY = YES, IIN = INCHES, INE = Meters
IM = MAGNITUDE, ID = DB, IB = BOTH (MAG, DB)
MX1 = MAX NUMBER OF ANTENNA IN SIMPLE PROGRAM
MX2 = MAX NUMBER OF ANTENNA IN GENERAL PROGRAM
MX3 = MAX NUMBER OF SUB-ANTENNA IN SIMPLE PROGRAM
517
SIMPLE PROGRAM? (IS/NP = IY)
122
300
SIMPLE = 1 : SIMPLE PROGRAM
CL = 1 : COMPUTE COUPLING COEFFICIENTS
REQUEST CALCULATION OF COUPLING COEFFICIENTS
650
REQUEST NUMBER OF ANTENNA NS
REQUEST ANTENNA TYPE AND POSITIONS (XX, YY, ZZ)
CHECK IF # OF SUB-ANTENNA EXCEEDS MAXIMUM
YES
PRINT MESSAGE 625
STOP
NO
NO
I SIMP = IN ?
YES
10
NO
PRINT ERROR MESSAGE
ENTER ALL ANTENNA SPECIFICATIONS FOR THE GENERAL PROGRAM: FREQ., # ANTENNAS, # SEGS., ANTENNA LENGTH, LOAD AND FEED POSITIONS, LOAD IMPEDANCES, FEED VOLTAGES, ANTENNA RADIUS AND POSITIONS ON PLATFORM X, Y, Z.
535
550
VERTICAL PATTERN?
YES
REQUEST PARAMETERS : CONSTANT PHI, INCREMENT OF THETA
IVP = 1: COMPUTE VERTICAL PATTERN
NO
55
HORIZONTAL PATTERN?
YES
REQUEST PARAMETERS : CONSTANT THETA, INCRE - MENT OF PHI
IPT = 1: COMPUTE HORIZONTAL PATTERN
NO
315
PATTERN DISTRIBUTION
NO
REQUEST PARAMETERS : PLOT FORM, PERCENT INCREMENT
IPA = 1: COMPUTE PATTERN DISTRIBUTION
65
COMMUNICATION RANGE
YES
72
REQUEST PARAMETERS
IRC = 1: COMPUTE COMMUNICATION RANGE
NO
76
CHECK IF ANY OUTPUT DATA IS RE- QUESTED
NO
92
PRINT ERROR MESSAGE
YES
64
RETURN

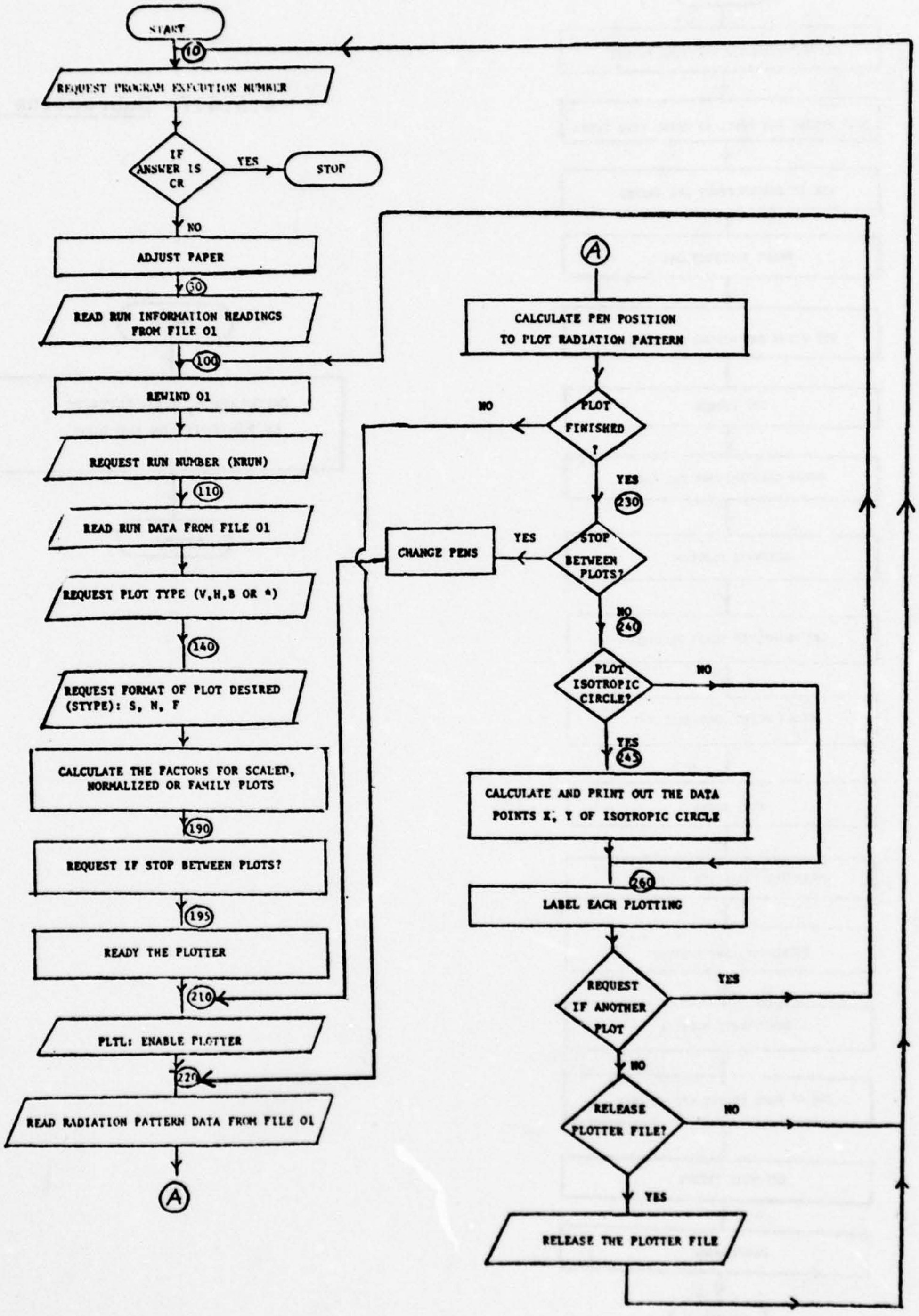
4.6.4.3 Subroutine LINKM2


START
STDALT = STANDARD AIRPLANE ALTITUDES IN FEET
MX3 = MAX # OF RUNS
NO
COMMUNICATION RANGE CONTOUR? (IRC = 1)
YES
ENTER SITE PARAMETERS FOR COMMUNICATION RANGE CONTOUR
900
PRINT OUT INPUT DATA OF THE COMMUNICATION RANGE
94
NO
SIMPLE PROGRAM? (SIMP = 1)
YES
350
PRINT OUT INPUT DATA FOR THE GENERAL PROGRAM
REQUEST MULTIPLE RUN DATA
400
PRINT OUT INPUT DATA OF THE SIMPLE PROGRAM
660
RETURN.

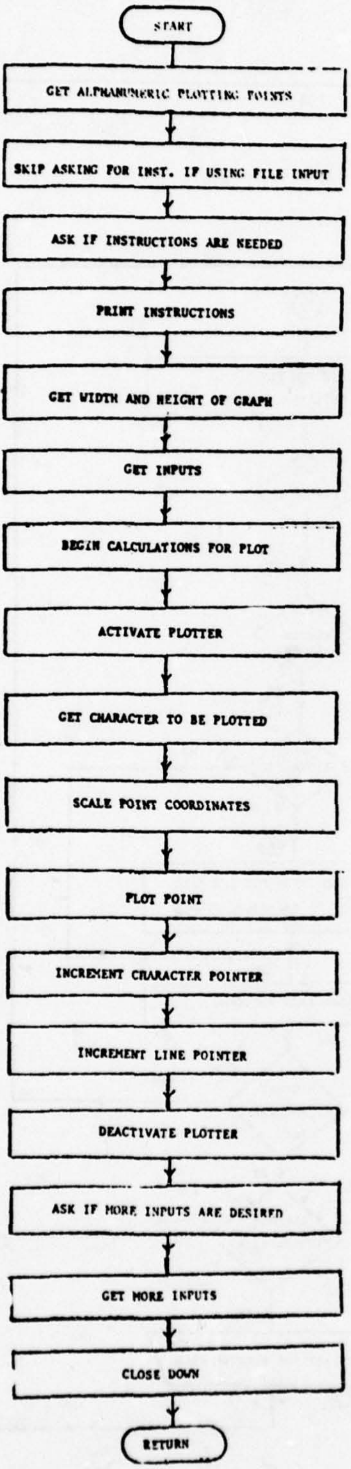
4.6.4.4 Subroutine LINKM3


START
IME = METERS: MAXS = MAX # OF SEGMENTS
SIMPLE PROGRAM? (SIMP = 1)
NO
450
CALCULATE AND STORE THE NECESSARY PARAMETERS FOR THE GENERAL PROGRAM
YES
CALCULATE & STORE NECESSARY PARAMETERS FOR THE SIMPLE PROGRAM (SEE FIGS.
WRITE OUTPUT IN FILE "01"
660
CALCULATE SUB-ANTENNAS PARAMETERS AND CHECK FOR POSSIBLE ERRORS
184
WRITE REMAINING DATA IN FILE "01"
456
CALCULATE: MAX CORE, MAXK; MAX TIME, MAXTIM
READ RUN #, RUNNO FROM 03 INCREMENT RUN # BY 1, WRITE INTO 03
WRITE THE PROPER CONTROL CARDS IN FILE "02"
PRINT RUN INFORMATION
REWIND FILES 01, 02 THEN CLOSE FILES 01, 02
DATA CORRECT?
NO
RETURN
YES
CREATE OUTPUT FILE /DISTRT/(RUNNO)
REMOVE FILE /DISTRT/(RUNNO)
SPAWN CARDIN PROGRAM
500
RETURN

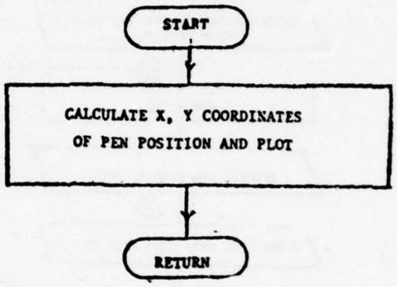
4.6.4.5 Subroutine DSPLT


START
10
REQUEST PROGRAM EXECUTION NUMBER
IF ANSWER IS CR
YES
STOP
NO
ADJUST PAPER
30
READ RUN INFORMATION HEADINGS FROM FILE 01
100
REWIND 01
REQUEST RUN NUMBER (NRUN)
110
READ RUN DATA FROM FILE 01
REQUEST PLOT TYPE (V,H,B OR *)
140
REQUEST FORMAT OF PLOT DESIRED (STYPE): S, N, F
CALCULATE THE FACTORS FOR SCALED, NORMALIZED OR FAMILY PLOTS
190
REQUEST IF STOP BETWEEN PLOTS?
195
READY THE PLOTTER
210
PLTL: ENABLE PLOTTER
220
READ RADIATION PATTERN DATA FROM FILE 01
A
A
CALCULATE PEN POSITION TO PLOT RADIATION PATTERN
PLOT FINISHED ?
NO
YES
230
STOP BETWEEN PLOTS?
YES
CHANGE PENS
NO
240
PLOT ISOTROPIC CIRCLE?
NO
YES
245
CALCULATE AND PRINT OUT THE DATA POINTS X, Y OF ISOTROPIC CIRCLE
260
LABEL EACH PLOTTING
REQUEST IF ANOTHER PLOT
YES
NO
RELEASE PLOTTER FILE?
NO
YES
RELEASE THE PLOTTER FILE

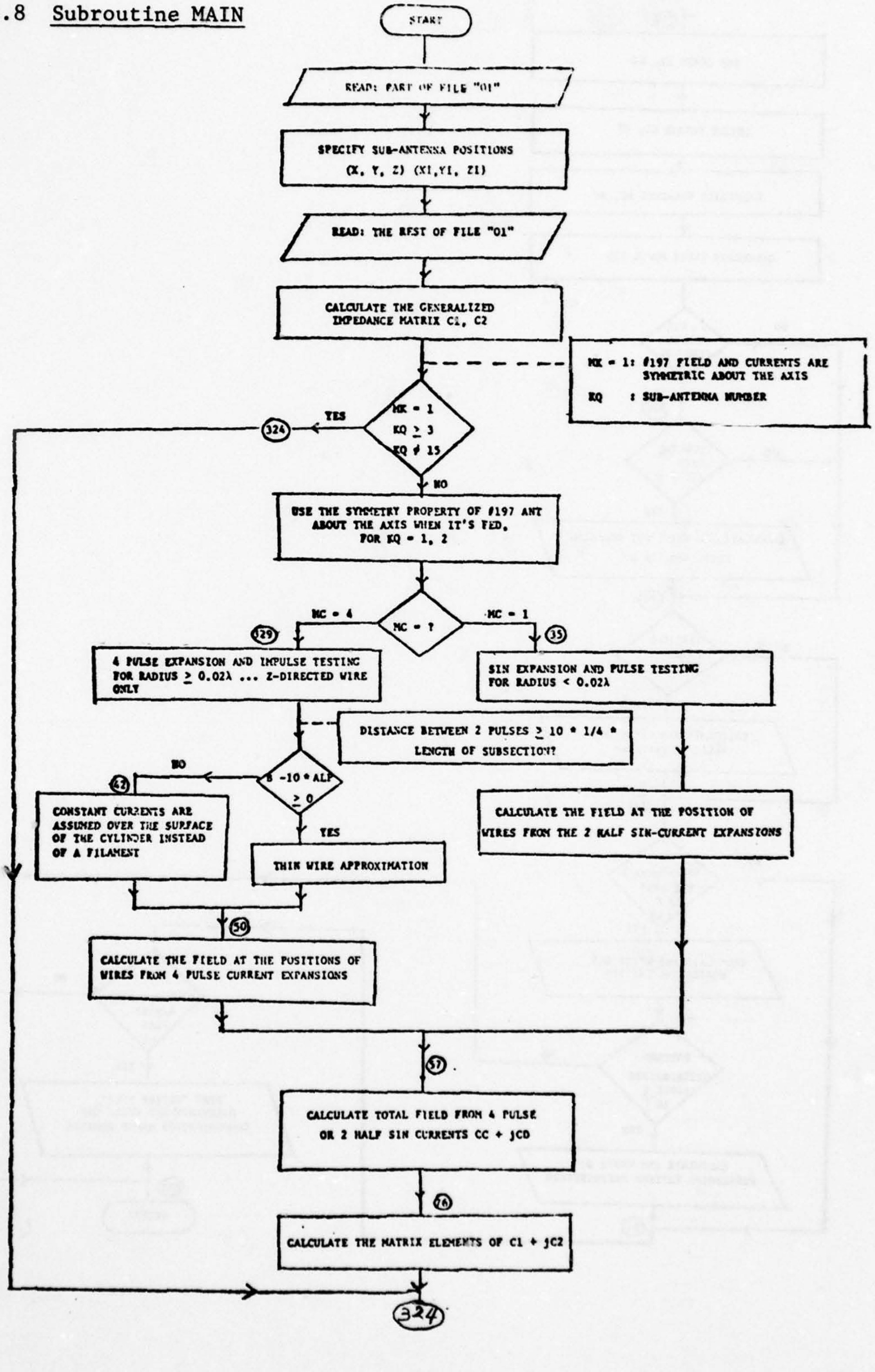
4.6.4.6 Subroutine LABEL

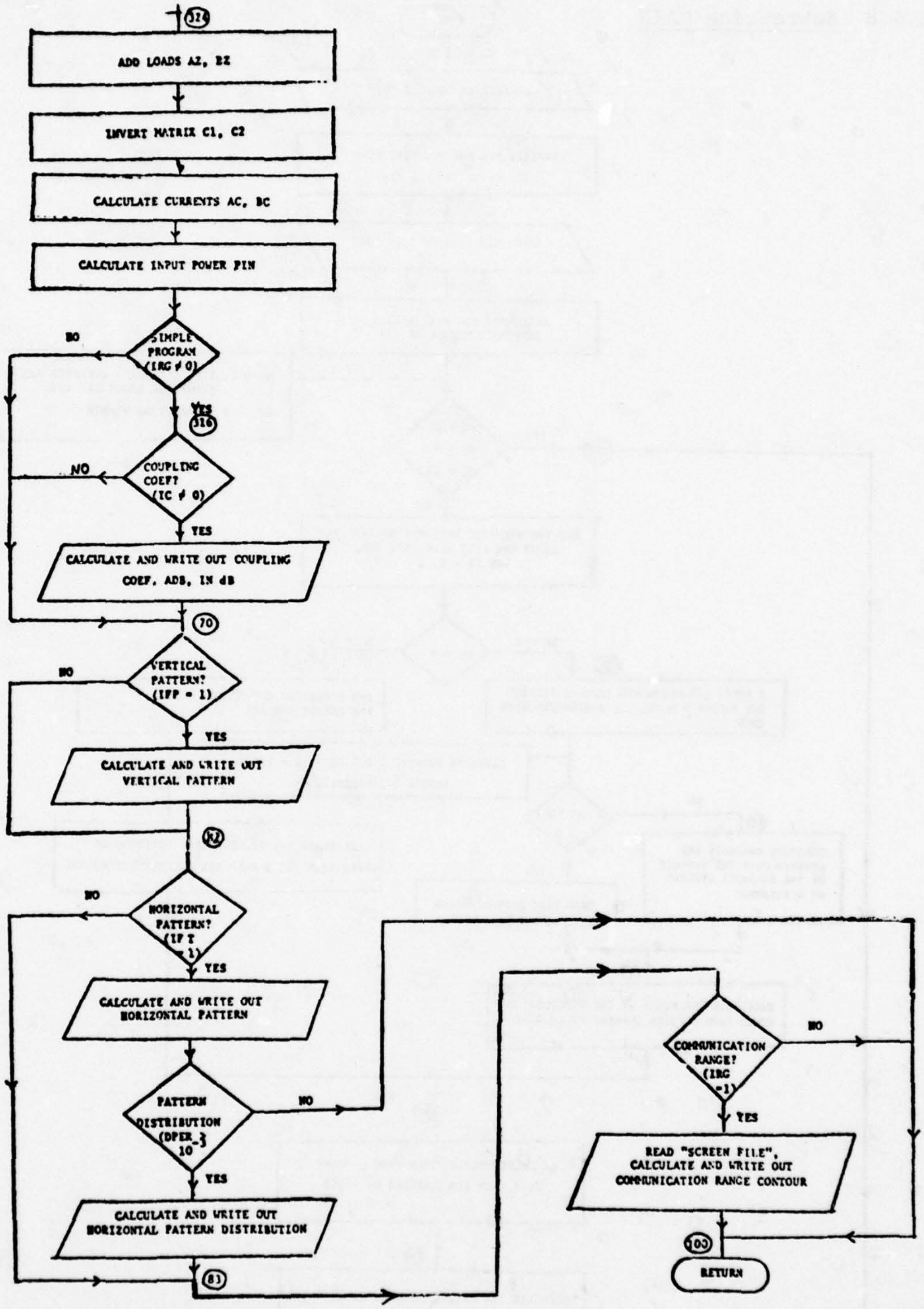

START
GET ALPHANUMERIC PLOTTING POINTS
SKIP ASKING FOR INST. IF USING FILE INPUT
ASK IF INSTRUCTIONS ARE NEEDED
PRINT INSTRUCTIONS
GET WIDTH AND HEIGHT OF GRAPH
GET INPUTS
BEGIN CALCULATIONS FOR PLOT
ACTIVATE PLOTTER
GET CHARACTER TO BE PLOTTED
SCALE POINT COORDINATES
PLOT POINT
INCREMENT CHARACTER POINTER
INCREMENT LINE POINTER
DEACTIVATE PLOTTER
ASK IF MORE INPUTS ARE DESIRED
GET MORE INPUTS
CLOSE DOWN
RETURN

4.6.4.7 Subroutine PPLOT



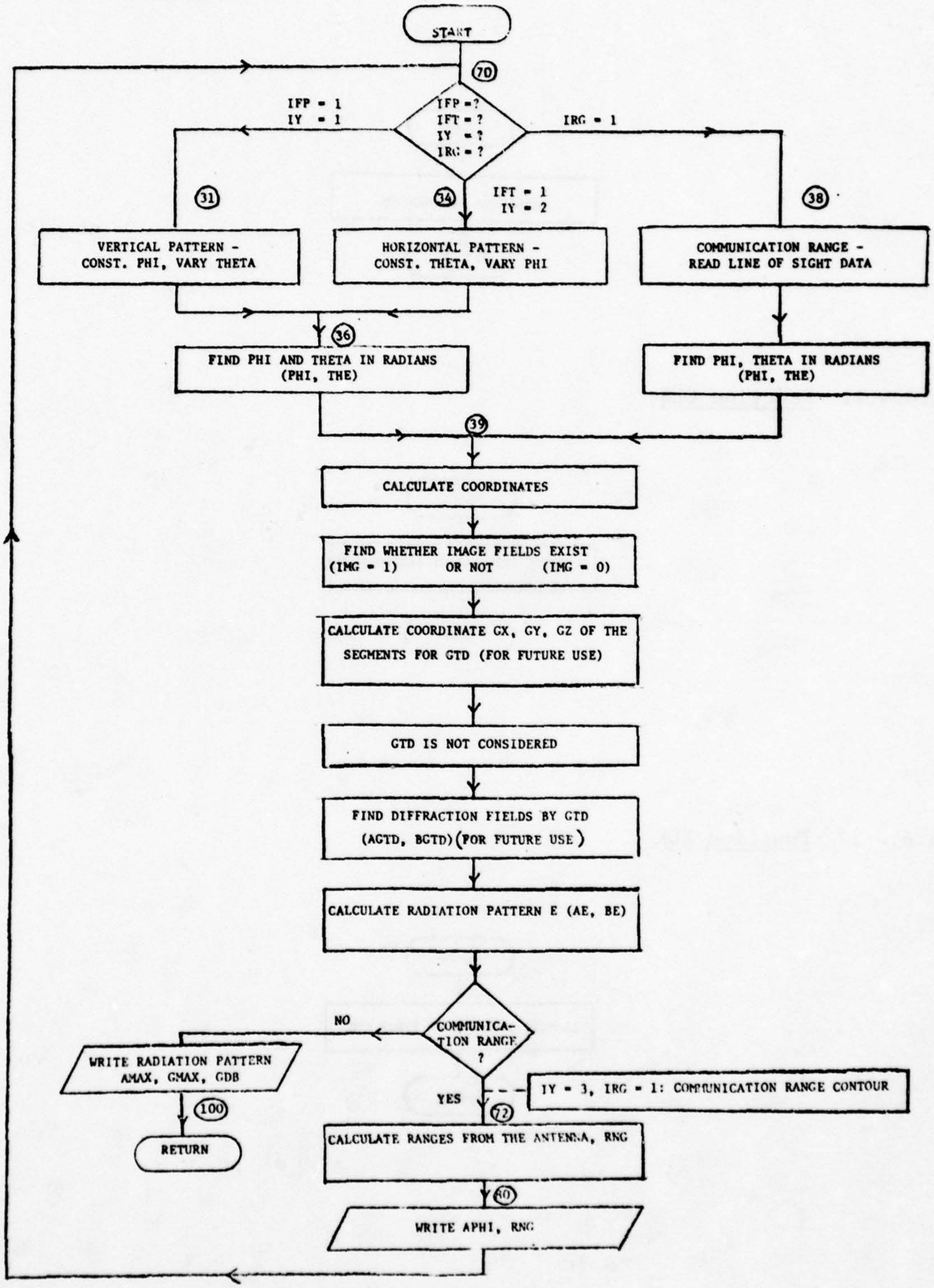
4.6.4.8 Subroutine MAIN


START
READ: PART OF FILE "01"
SPECIFY SUB-ANTENNA POSITIONS
(X, Y, Z) (X1,Y1, Z1)
READ: THE REST OF FILE "01"
CALCULATE THE GENERALIZED
IMPEDANCE MATRIX C1, C2
MK = 1: #197 FIELD AND CURRENTS ARE SYMMETRIC ABOUT THE AXIS
KQ : SUB-ANTENNA NUMBER
MK = 1
KQ ≥ 3
KQ ≠ 15
YES
324
NO
USE THE SYMMETRY PROPERTY OF #197 ANT
ABOUT THE AXIS WHEN IT'S FED,
FOR KQ = 1, 2
MC = ?
MC = 4
MC = 1
329
35
4 PULSE EXPANSION AND IMPULSE TESTING
FOR RADIUS ≥ 0.02λ ... Z-DIRECTED WIRE
ONLY
SIN EXPANSION AND PULSE TESTING
FOR RADIUS < 0.02λ
DISTANCE BETWEEN 2 PULSES ≥ 10 * 1/4 *
LENGTH OF SUBSECTION?
B -10 * ALP
≥ 0
NO
YES
42
CONSTANT CURRENTS ARE
ASSUMED OVER THE SURFACE
OF THE CYLINDER INSTEAD
OF A FILAMENT
THIN WIRE APPROXIMATION
CALCULATE THE FIELD AT THE POSITION OF
WIRES FROM THE 2 HALF SIN-CURRENT EXPANSIONS
50
CALCULATE THE FIELD AT THE POSITIONS OF
WIRES FROM 4 PULSE CURRENT EXPANSIONS
57
CALCULATE TOTAL FIELD FROM 4 PULSE
OR 2 HALF SIN CURRENTS CC + JCD
76
CALCULATE THE MATRIX ELEMENTS OF C1 + JC2
324

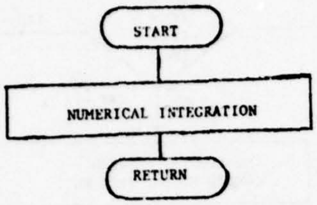


324
ADD LOADS AZ, BZ
INVERT MATRIX C1, C2
CALCULATE CURRENTS AC, BC
CALCULATE INPUT POWER PIN
SIMPLE PROGRAM (IRG ≠ 0)
NO
YES
316
COUPLING COEF? (IC ≠ 0)
NO
YES
CALCULATE AND WRITE OUT COUPLING COEF. ADB, IN dB
70
VERTICAL PATTERN? (IFP = 1)
NO
YES
CALCULATE AND WRITE OUT VERTICAL PATTERN
82
HORIZONTAL PATTERN? (IF T = 1)
NO
YES
CALCULATE AND WRITE OUT HORIZONTAL PATTERN
PATTERN DISTRIBUTION (DPER-3 10)
NO
YES
CALCULATE AND WRITE OUT HORIZONTAL PATTERN DISTRIBUTION
83
COMMUNICATION RANGE? (IRG =1)
NO
YES
READ "SCREEN FILE", CALCULATE AND WRITE OUT COMMUNICATION RANGE CONTOUR
100
RETURN

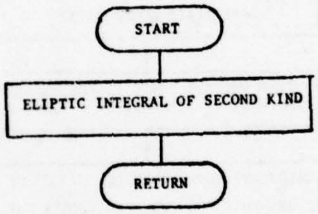
4.6.4.9 Subroutine PATT


START
70
IFP = ?
IFT = ?
IY = ?
IRG = ?
IFP = 1
IY = 1
IRG = 1
IFT = 1
IY = 2
31
34
38
VERTICAL PATTERN - CONST. PHI, VARY THETA
HORIZONTAL PATTERN - CONST. THETA, VARY PHI
COMMUNICATION RANGE - READ LINE OF SIGHT DATA
36
FIND PHI AND THETA IN RADIANS (PHI, THE)
FIND PHI, THETA IN RADIANS (PHI, THE)
39
CALCULATE COORDINATES
FIND WHETHER IMAGE FIELDS EXIST (IMG = 1) OR NOT (IMG = 0)
CALCULATE COORDINATE GX, GY, GZ OF THE SEGMENTS FOR GTD (FOR FUTURE USE)
GTD IS NOT CONSIDERED
FIND DIFFRACTION FIELDS BY GTD (AGTD, BGTD)(FOR FUTURE USE)
CALCULATE RADIATION PATTERN E (AE, BE)
COMMUNICA-TION RANGE ?
NO
YES
IY = 3, IRG = 1: COMMUNICATION RANGE CONTOUR
WRITE RADIATION PATTERN AMAX, GMAX, GDB
100
RETURN
72
CALCULATE RANGES FROM THE ANTENNA, RNG
80
WRITE APHI, RNG

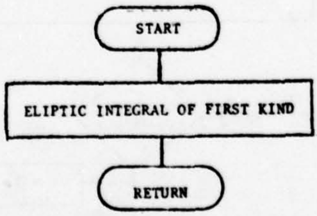
4.6.4.10 Subroutine FUN



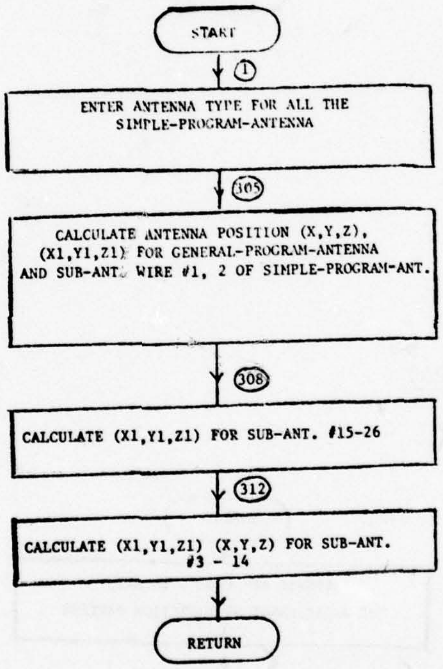
4.6.4.11 Function ELE



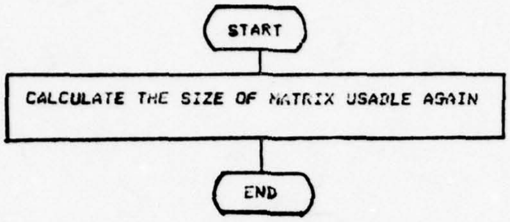
4.6.4.12 Function ELK



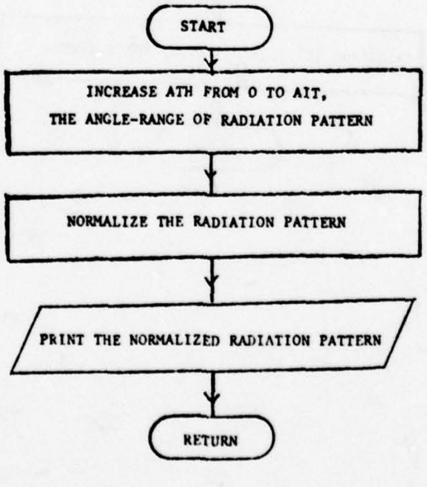
4.6.4.13 Subroutine PXYZ


START
1
ENTER ANTENNA TYPE FOR ALL THE SIMPLE-PROGRAM-ANTENNA
305
CALCULATE ANTENNA POSITION (X,Y,Z), (X1,Y1,Z1) FOR GENERAL-PROGRAM-ANTENNA AND SUB-ANT. WIRE #1, 2 OF SIMPLE-PROGRAM-ANT.
308
CALCULATE (X1,Y1,Z1) FOR SUB-ANT. #15-26
312
CALCULATE (X1,Y1,Z1) (X,Y,Z) FOR SUB-ANT. #3 - 14
RETURN

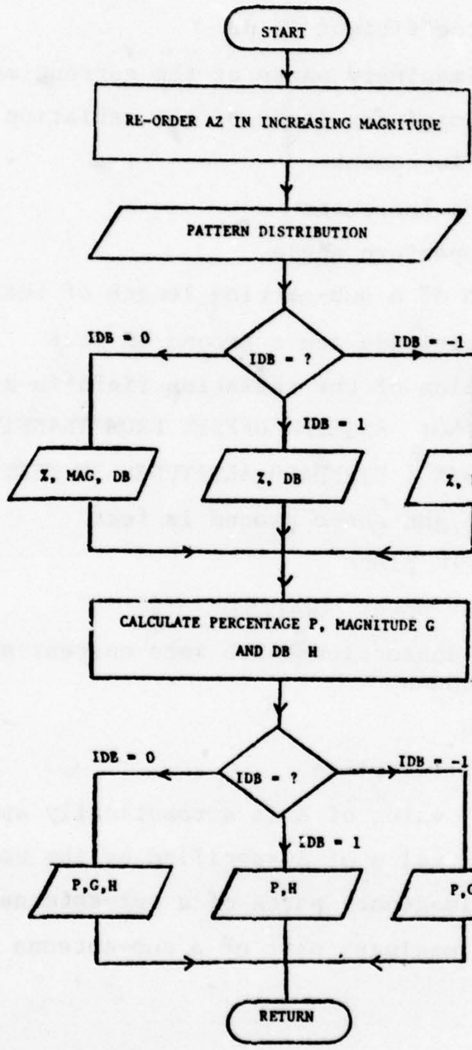
4.6.4.14 Subroutine QZP



4.6.4.15 Subroutine NPAT


START
INCREASE ATH FROM 0 TO AIT,
THE ANGLE-RANGE OF RADIATION PATTERN
NORMALIZE THE RADIATION PATTERN
PRINT THE NORMALIZED RADIATION PATTERN
RETURN

4.6.4.16 <u>Subroutine PATD</u>



4.7 List of Principal Symbols

A

A	Number of subsections per wavelength
ADB	Coupling coefficient in dB
AC, BC	Real and imaginary parts of the current matrix
AE, BE	Real and imaginary parts of the radiation field
AINP	Angle Phi increment
AINT	Angle Theta increment
AIT	Radiation pattern angle
ALP	One-fourth of a sub-section length of the test wire.
ALT	Airplane altitude above ground in feet
AMAX	Maximum value of the radiation field in a specific pattern
ANSV	ANSV= YES/NO: ANTENNA OFFSET FROM TRANSIT?
ANSW	ANSW = YES/NO: STANDARD ALTITUDES IN FEET
ANTEL	Antenna height above ground in feet
APHI	Constant Phi plane
APIN	Transmitter power (WATTS)
AS	Number of subsections with zero current at each end of the antenna
AT	Same as AS
ATHI	Constant Theta plane
AUTO	AUTO = 1: value of A is automatically specified
	AUTO = 0: value of A specified by the user
AV, BV	Real and imaginary parts of a sub-antenna feed voltage
AZ, BZ	Real and imaginary part of a sub-antenna load impedance

B

B	Distance between two pulses of expansion or testing functions.
BEAR	Antenna bearing in Degrees and Minutes (DD, MM).
BTA	Antenna bearing from the North

C

CL	CL = 1: Compute coupling coefficients CL = 0: Do not compute coupling coefficients
C1, C2	Real and imaginary parts of the generalized impedance matrix

D

DBM	Receiver sensitivity in dBm
DIST	Distance between screen and transit in feet
PPER	Percent increment of pattern distribution
DTA	Distance between antenna and transit in feet

E

ELANG	Elevation angle from transit in radians

F

FA	Height of the bottom of the fed antenna in SIMPLE PROGRAM
FILENAME	Name of SCREEN file
FP	Operating frequency in Hz
FR	Operating frequency in MHz

G

GMAX	Maximum gain

GP	GP = 1; with ground plane
	GP = 0; without ground plane

GREL	Ground elevation above mean sea level (MSL) in feet

H

H1	Load position measured from sub-antenna bottom
H3	Feed position measured from sub-antenna bottom
H4	Lighting rod length
H5	Lighting rod diameter
HO	Sub-antenna length
HSCRN	Screen height in meters

I

IALP	Vertical pattern
IALT	Horizontal pattern
IAUTO	IAUTO = YES/NO: answer to AUTO SPEC # SEGS
IB	Both (MAG, DB)
IC	IC = 1: compute coupling coefficients IC = 0: do not compute coupling coefficients.
ICL	ICL = YES/NO answer to COUPLING COEFFICIENTS?
ID	in dB

IDB	IDB = -1, pattern in magnitude
	IDB = 0, pattern in dB
	IDB = 1, pattern in magnitude and dB

IDD	IDD = M, plot in magnitude
	IDD = D, plot in dB
	IDD = B, plot in magnitude and dB

IDIM	Dimensions of generalized impedance matrix (Cl, C2)
IF	Feed position of sub-antenna

IFP	IFP = 1, vertical pattern
	IFP = 0, otherwise

IFT	IFT = 1, horizontal pattern
	IFT = 0, otherwise

IG	IG = 1, with ground plane
	IG = 0, without ground plane

IGP	IGP = YES/NO: answer to GROUND PLANE?

II2	II2 = 1, $(BZ)^2 \neq 0$
	II2 = 0, otherwise

II4	II4 = 1, $(BV)^2 \neq 0$
	II4 = 0, otherwise

II6	II6 = 1, AZ > 0
	II6 = 0, otherwise

IIN	Dimensions in inches
IM	In magnitude
IME	Dimension in meters
IN	No
IPA	IPA = 1, Compute pattern distribution IPA = 0, Do not compute pattern distribution
IPD	IPD = YES/NO: answer to PATTERN DISTRIBUTION
IRG	IRG = 1, Compute communication range IRG = 0, Do not compute communication range
IRGE	IRGE = YES/NO: answer to COMMUNICATION RANGE CONTOUR?
ISIMP	ISIMP = YES/NO: answer to SIMPLE PROGRAM?
IT	Type of antenna in SIMPLE PROGRAM
IV	Fed antenna number
IZ	Feed and load position

<u>J</u>

JQ	Number of test field calculations for numerical integration of the pulse testing

<u>K</u>

K	Total number of sub-antennas (wires), column index of the Z-matrix

K2	K2 = 1, without ground plane
	K2 = 2, with ground plane

KQ	Sub-antenna (wire) number. column index of the Z-matrix
KS	Antenna number, column index of the Z-matrix
KW	Number of runs

L

L	Total number of sub-antennas, row index of the Z-matrix
L1	Load positions of sub-antenna
LQ	Sub-antenna number, row index of the Z-matrix
LS	Antenna number, row index of the Z-matrix

M

MAXK	Maximum core
MAXS	Maximum number of sub-sections
MAXTIM	Maximum time required by job
MC	MC = 1, sine expansion and pulse testing for radius less than 0.02 of wavelength MC = 4, Four pulse expansion (approximation of a triangle) and impulse testing for radius greater than or equal to 0.02 of wavelength ... z-directed wire only,
MK	MK = 1, AT 197 fed, currents assumed symmetrical about the axis MK = 0, Otherwise
ML	Same as MK
MM	Maximum number of segments

MV	MV = 0, vertical wires
	MV = 1, #13-#14 wires of AT 197 antenna
	AV = 2, #15-#16 wires of AT 197 antenna

MW	Same as MV

N

NA	Number of airplane altitudes above ground, in feet, to be calculated (1 to 6 altitudes)
NB	Number of antennas in the SIMPLE program, NB = 0, GENERAL program
NFP	Number of field points in the horizontal pattern
NK	Number of sub-antennas (wires): . NK = 1 for AT 1181; NK = 5 for AT 1097; NK = 26 for AT 197

NL	Same as NK
NM	Maximum number of segments
NN	Number of antennas in the GENERAL PROGRAM; number of sub-antennas in the SIMPLE PROGRAM
NP	Number of non-zero currents on an antenna
NS	Number of sub-sections of sub-antenna
NZ	Total number of feed and load with coupling coefficients computed

R

R2	Earth diameter in feet
R3	Equivalent earth radius in meters for ray tracing (4/3 radius)
RA	Radius of sub-antenna
RNG	Ranges in nautical miles for antenna

S

SIMP	SIMP = 0, GENERAL program
	SIMP = 1, SIMPLE program

STDALT	Standard airplane altitudes in feet

T

TRANEL	Transit elevation
TTH	Elevation angle from the antenna bottom to screen in radians

X

X,Y,Z	Bottom position of antenna in GENERAL program
XX,YY,ZZ	Bottom position of antenna in SIMPLE program
X1,Y1,Z1	Top end position of sub-antenna

APPENDIX I

Program Listing

```
10$       FORTRAN ASCII,NFORM,NLNO                                    LNKMNA
20 SUBROUTINE LINKMN
30 COMMON JUNK(3423)
40 IY=11953782816
50 5 DO 10 I=1,3423
60 10 JUNK(I)=0
70 CALL ATTACH(03,"BLA00001/RADIO/NORUN;",3,0,ISTAT,)
80 CALL CREATE(01,4000,0,ISTAT)
90 CALL CREATE(02,4000,0,ISTAT)
100 REWIND 01; REWIND 02; REWIND 03
110 CALL LINKM1
120 CALL LINKM2
130 CALL LINKM3
140 PRINT:"RERUN?"; READ 1000,JUNK(1)
150 1000 FORMAT(A1)
160 IF(JUNK(1).NE.IY)STOP
170 GO TO 5
180 END
190$      FORTRAN ASCII,NFORM,NLNO                                    LNKM1A
200 SUBROUTINE LINKM1
210 COMMON A,H0,H1,H3,RA,X,Y,Z,AV,BV,AZ,BZ,NS,L1,IF,IT,XX,YY,ZZ,IZ,
220&FR,IV,IIM,GP,SIMP,NN,AUTO,KW,CL,NB,AINT,AINP,APHI,ATHE,IPA,IPD
230&,H4,H5,IFP,IFT,IRG,DBM,ISIMP,ICL,IALP,IALT,IRGE,IAUTO,IGP,IDD
240&,ALT(6),GREL,TRANEL,ANTEL,NA,DTA,BTA,APIN,FA,IDB,DPER,FILENAME
250 DIMENSION A(200),H0(200),H1(200),H3(200),RA(200),X(200),Y(200),
260&Z(200),AV(200),BV(200),AZ(361),BZ(200),NS(200),L1(200),IF(200),
270&IT(20),XX(20),YY(20),ZZ(20),IZ(40),FR(10),IV(10),H4(20),H5(20),
280&FA(40),STDALT(6) ;CHARACTER FILENAME*8
290 IY=11953782816;IN=10477387808
300 IB=8865775072;ID=9135210528;IM=10343170080
310*
320*    ***IY=YES, IN=NO, IIN=INCHES, IME=METERS
330*    ***IM=MAGNITUDE, ID=DB, IB=BOTH(MAG,DB)
340*
350 IIN=9806299168;IME=10343170080
360 ABW=0.
370 MX1=20;MX2=200
380*
390*    ***MX1=MAXIMUM NUMBER OF ANTENNAS IN THE SIMPLE PROGRAM
400*    ***MX2=MAXIMUM NUMBER OF ANTENNAS IN THE GENERAL PROGRAM
410*    ***MX2=MAXIMUM NUMBER OF SUB-ANTENNAS IN THE SIMPLE PROGRAM
420*    ***ENTER DIMENSION UNITS, GROUND PLANE AND TYPE OF PROGRAM
430*
440 510 PRINT:" DIMENSIONS IN METERS OR INCHES ?"
450 READ 515,IIM
460 IF(IIM-IIN) 102,106,102
470 102 IF(IIM-IME) 104,106,104
480 104 PRINT 502;GO TO 510
490 502 FORMAT(" ..... INPUT ERROR, TRY AGAIN .....")
500 106 PRINT:" GROUND PLANE ?"
510 READ 515,IGP
520 IF(IGP-IY) 112,116,112
```

```
530 112 IF(IGP-IN) 114,517,114
540 114 PRINT 502;GO TO 106
550 116 GP=1.
560*
570*   ***GP=1. : WITH THE GROUND PLANE
580*
590 515 FORMAT(A1)
600 517 PRINT:" SIMPLE PROG ?"
610 READ 515,ISIMP
620 IF(ISIMP-IY) 124,122,124
630 122 SIMP=1.;GO TO 300
640*
650*   ***SIMP=1.  :SIMPLE PROGRAM
660*
670 124 IF(ISIMP-IN) 126,10,126
680 126 PRINT 502;GO TO 517
690*
700*  ***ENTER ALL ANTENNA SPECIFICATIONS FOR THE GENERAL PROGRAM :
710*  ***FREQ. ,# ANTENNAS, #SEGS., ANTENNA LENGTH,LOAD  & FEED POSITIONS
720*  ***LOAD IMPEDANCE, FEED VOLTAGE, RADIUS OF ANTENNA AND POSITION OF
730*  ***ANTENNA ON PLATFORM X,Y,Z
740*
750 10 PRINT:" FREQUENCY (MHZ)";READ:FR(1)
760 IF(FR(1).LE.0.) PRINT 502
770 IF(FR(1).LE.0.) GO TO 10
780 605 PRINT:" NUMBER OF ANTENNAS ";READ:NN
790 IF(NN.GT.0.AND.NN.LE.MX2) GO TO 520
800 IF(NN.LE.0) PRINT 502;IF(NN.LE.0) GO TO 605
810 PRINT 625;GO TO 605
820 520 PRINT:" AUTO SPEC #SEGS ?"
830 READ 515,IAUTO
840 IF(IAUTO-IY) 132,136,132
850 132 IF(IAUTO-IN) 134,138,134
860 134 PRINT 502;GO TO 520
870 136 AUTO=1.
880*
890* ***AUTO=1. : WITH AUTO SPEC #SEGS
900*
910 138 DO 285 I=1,NN
920 PRINT 274,I
930 274 FORMAT(" ****  ANTENNA NUMBER",I3,"  ****")
940 IF(IAUTO.EQ.IY) GO TO 2
950 3 PRINT:" NUMBER OF SEGMENTS PER WAVE-LENGTH "
960 READ:A(I)
970 IF(A(I).LE.0.) PRINT 502;IF(A(I).LE.0.) GO TO 3
980 2 PRINT:" ANTENNA LENGTH"
990 READ:HO(I)
1000 IF(HO(I).LE.0.) PRINT 502;IF(HO(I);LE.0.) GO TO 2
1010 610 PRINT:" LOAD POSITION"
1020 READ:H1(I)
1030 IF(H1(I).GE.0..AND.H1(I).LT.HO(I)) GO TO 620
1040 PRINT 502;GO TO 610
```

```
1050 620 PRINT:" FEED POSITION"
1060 READ:H3(I)
1070 IF(H3(I).GE.0..AND.H3(I).LT.HO(I)) GO TO 622
1080 PRINT 502;GO TO 620
1090 622 PRINT:" ANTENNA RADIUS"
1100 READ:RA(I)
1110 IF(RA(I).LE.0.) PRINT 502;IF(RA(I).LE.0.) GO TO 622
1120 IF(IIM-IME) 162,166,166
1130 162 IF(RA(I)-1240./FR(1)) 168,168,164
1140 164 PRINT 502;GO TO 622
1150 166 IF(RA(I)-31./FR(1)) 168,168,164
1160 168 PRINT:" ANTENNA POSITION X,Y,Z ON THE PLATFORM"
1170 READ:X(I),Y(I),Z(I)
1180 IF(IIM.EQ.IIN) FA(I)=Z(I)*.0254
1190 IF(IIM.EQ.IME) FA(I)=Z(I)
1200 IF(GP.LT..5.OR.Z(I).GE.0.) GO TO 67
1210 PRINT 502;GO TO 168
1220 67 PRINT:" FEED VOLTAGE (REAL,IMAG)"
1230 READ:AV(I),BV(I)
1240 PRINT:" LOAD IMPEDANCE (REAL,IMAG) "
1250 READ:AZ(I),BZ(I)
1260 ABW=ABW+AV(I)**2+BV(I)**2
1270 285 CONTINUE
1280 IF(ABW-1.E-8) 56,56,310
1290 56 PRINT 573
1300 573 FORMAT(" **<F>**  << NO ANTENNAS ARE FED >>")
1310 DO 57 I=1,NN
1320 PRINT 274,I
1330 PRINT:" FEED VOLTAGE (REAL,IMAG) "
1340 57 READ:AV(I),BV(I)
1350 GO TO 310
1360*
1370*  ***REQUEST CALCULATION OF COUPLING COEFFICIENTS
1380*
1390 300 PRINT:" COUPLING COEFFICIENTS ?"
1400 CL=0.
1410 READ 515,ICL
1420 IF(ICL-IY) 172,176,172
1430 172 IF(ICL-IN) 174,650,174
1440 174 PRINT 502;GO TO 300
1450 176 CL=1.
1460*
1470*   *** CL=1.  :COMPUTE COUPLING COEFFICIENTS
1480*
1490*   ***REQUEST NUMBER OF ANTENNAS , NB
1500*
1510 650 PRINT:" NUMBER OF ANTENNAS"
1520 READ:NB
1530 IF(NB.LE.0) PRINT 502;IF(NB.LE.0) GO TO 650
1540 IF(CL.GT..5.AND.NB.EQ.1) PRINT 7
1550 IF(CL.GT..5.AND.NB.EQ.1) GO TO 300
1560 7 FORMAT(" **<<F>>**  << # OF ANT. GREATER THAN ONE (COUPLING
15704 COEF.) >>")
1580 IF(NB.GT.MX1) PRINT 625
1590 625 FORMAT(" **<F>**  << TOO MANY ANTENNAS >>")
1600 IF(NB.GT.MX1) GO TO 650
1610*
1620*    ***REQUEST ANTENNA TYPES AND POSITIONS XX, YY, ZZ
1630*
1640 DO 295 I=1,NB
1650 PRINT 274,I
1660 530 PRINT:" ANTENNA TYPE ? (1097, 197, 1181 OR 1000)"
1670 READ:IT(I)
1680 IF(NB.EQ.1.AND.IT(I).EQ.1000) PRINT 502
1690 IF(NB.EQ.1.AND.IT(I).EQ.1000) GO TO 650
1700 IF(IT(I)-1097) 182,586,182
1710 182 IF(IT(I)-1181) 184,585,184
1720 184 IF(IT(I)-197) 200,586,200
1730 200 IF(IT(I)-1000) 202,186,202
1740 186 PRINT:" LENGTH";READ:H4(I)
1750 IF(H4(I).LE.0.) PRINT 502;IF(H4(I).LE.0.) GO TO 186
1760 4 PRINT:" DIAMETER";READ:H5(I)
1770 IF(H5(I).LE.0.) PRINT 502;IF(H5(I),LE.0.) GO TO 4
1780 GO TO 586
1790 202 PRINT 502;GO TO 530
1800 586 PRINT:" ANTENNA POSITION X,Y,Z ON THE PLATFORM"
1810 READ:XX(I),YY(I),ZZ(I)
1820 41 IF(GP.LT..5.OR.ZZ(I).GE.0.) GO TO 69
1830 PRINT 502;GO TO 586
1840 69 IF(IIM.EQ.IIN) FA(I)=ZZ(I)*.0254
1850.IF(IIM.EQ.IME) FA(I)=ZZ(I)
1860 N9=1
1870 IF(IT(I).EQ.1097) N9=5
1880 IF(IT(I).EQ.197) N9=26
1890 NN=NN+N9
1900 295 CONTINUE
1910*
1920*    ***CHECK IF THE # OF SUB-ANTENNAS NN EXCEEDS MAXIMUM
1930*
1940 IF(NN.GT.MX2) PRINT 625
1950 IF(NN.GT.MX2) STOP
1960 310 PRINT:"+++ RADIATION PATTERN +++"
1970*
1980*    *** VERTICAL PATTERN ?
1990*
2000 550 PRINT:" VERTICAL PATTERN ?"
2010 READ 515,IALP
2020 IF(IALP-IY) 232,236,232
2030 232 IF(IALP-IN) 234,55,234
2040 234 PRINT 502;GO TO 550
2050*
2060*    ***IFP=1  : COMPUTE VERTICAL PATTERN
2070*
2080*    ***REQUEST PARAMETERS :CONSTANT PHI, INCREMENT OF THETA
```

```
2090*
2100 236 IFP=1
2110 79 PRINT:" PHI (DEGREES) ";READ:APHI
2120 580 PRINT 380;READ:AINT
2130 IF(ABS(AINT).GE..01) GO TO 55
2140 PRINT 502;GO TO 580
2150*
2160*    ***HORIZONTAL PATTERN ?
2170*
2180 55 PRINT:" HORIZONTAL PATTERN ?"
2190 READ 515,IALT
2200 IF(IALT-IY) 242,246,242
2210 242 IF(IALT-IN) 244,65,244
2220 244 PRINT 502;GO TO 55
2230 246 IFT=1
2240*
2250*    ***IFT=1  : COMPUTE HORIZONTAL PATTERN
2260*
2270*    ***REQUEST PARAMETERS :CONSTANT THETA, INCREMENT OF PHI
2280*
2290 89 PRINT:" THETA (DEGREES) ";READ:ATHE
2300 590 PRINT 380;READ:AINP
2310 IF(ABS(AINP).GE..01) GO TO 315
2320 PRINT 502;GO TO 590
2330 380 FORMAT("  PLOTTING INCREMENT (NON-ZERO NUMBER) (DEGREES)")
2340*
2350*    ***PATTERN DISTRIBUTION ?
2360*
2370 315 PRINT:" PATTERN DISTRIBUTION ?"
2380 READ 515,IPD
2390 IF(IPD-IY) 320,330,320
2400 320 IF(IPD-IN) 325,65,325
2410 325 PRINT 502;GO TO 315
2420 330 IPA=1
2430*
2440*    ***IPA=1  :COMPUTE PATTERN DISTRIBUTION
2450*
2460*    ***REQUEST PARAMETERS :PLOT FORM , PERCENT INCREMENT
2470*
2480 335 PRINT:" PLOT IN MAG, DB, OR BOTH ?"
2490 READ 515,IDD
2500 IF(IDD.EQ.IM) IDB=-1
2510 IF(IDD.EQ.IB) IDB=0
2520 IF(IDD.EQ.ID) IDB=1
2530 IF(IDD.NE.IM.AND.IDD.NE.ID.AND.IDD;NE.IB) PRINT 502
2540 IF(IDD.NE.IM.AND.IDD.NE.ID.AND.IDD;NE.IB) GO TO 335
2550 340 PRINT:" PERCENT INCREMENT";READ:DPER
2560 IF(DPER.LE..01) PRINT 502
2570 IF(DPEF.LE..01) GO TO 340
2580 65 IF(SIMP.LT..5) GO TO 76
2590*
2600*    ***COMMUNICATION RANGE ?
```

```
2610*
2620 500 FORMAT(A8)
2630 64 PRINT:" COMMUNICATION RANGE CONTOUR ?"
2640 IRG=0
2650 READ 515,IRGE
2660 IF(IRGE-IY) 70,72,70
2670 70 IF(IRGE-IN) 74,76,74
2680 74 PRINT 502;GO TO 64
2690 72 IRG=1
2700*
2710*    ***IRG=1  : COMPUTE COMMUNICATION RANGE
2720*
2730*    ***REQUEST PARAMETERS
2740 PRINT:"NAME OF SCREEN FILE"
2750 READ 500,FILENAME
2760 40 PRINT:" RECEIVER SENSITIVITY (DBM)"
2770 READ:DBM
2780 IF(DBM.GE.0.) PRINT 502;IF(DBM.GE.0.) GO TO 40
2790 15 PRINT:" GROUND TRANSMITTER POWER (WATT)"
2800 READ:APIN
2810 IF(APIN.LE.0.) PRINT 502;IF(APIN.LE.0.) GO TO 15
2820 76 IF(CL.GT..5) GO TO 94
2830 IAA=IFP+IFT+IRG
2840*
2850*    ***CHECK IF ANY OUTPUT DATA IS REQUESTED
2860*
2870 IF(IAA) 92,92,94
2880*
2890*    ***PRINT ERROR MASSAGE
2900*
2910 92 PRINT 502;GO TO 550
2920 94 RETURN;END
2930$       FORTRAN ASCII,NFORM,NLNO                                        LNKM2A
2940 SUBROUTINE LINKM2
2950 ASCII ANSW,ANSV
2960 COMMON A,H0,H1,H3,RA,X,Y,Z,AV,BV,AZ,BZ,NS,L1,IF,IT,XX,YY,ZZ,IZ,
2970&FR,IV,IIM,GP,SIMP,NN,AUTO,KW,CL,NB,AINT,AINP,APHI,ATHE,IPA,IPD
2980&,H4,H5,IFP,IFT,IRG,DBM,ISIMP,ICL,IALP,IALT,IRGE,IAUTO,IGP,IDD
2990&,ALT(6),GREL,TRANEL,ANTEL,NA,DTA,BTA,APIN,FA,IDB,DPER,
3000&FILENAME
3010 DIMENSION A(200),H0(200),H1(200),H3(200),RA(200),X(200),Y(200),
3020&Z(200),AV(200),BV(200),AZ(361),BZ(200),NS(200),L1(200),IF(200),
3030&IT(20),XX(20),YY(20),ZZ(20),IZ(40),FR(10),IV(10),H4(20),H5(20),
3040&FA(40),STDALT(6);CHARACTER FILENAME *8
3050 IY=11953782816;IN=10477387808
3060 STDALT(1)=1000.
3070 STDALT(2)=5000.
3080 STDALT(3)=10000.
3090 STDALT(4)=15000.
3100 STDALT(5)=20000.
3110 STDALT(6)=35000.
3120 IIN=9806299168
```

```
3130 MMMMM=1
3140*
3150*    ***STDALT=STANDARD AIRPLANE ALTITUDES IN FEET
3160*
3170 CDTR=3.1415927/180.
3180 MX3=10
3190*
3200*    ***MX3=MAXIMUM NUMBER OF RUNS
3210*
3220 IF(IRG.EQ.0) GO TO 94
3230*
3240*    ***ENTER SITE PARAMETER FOR COMMUNICATION RANGE CONTOUR
3250*
3260 PRINT:" "
3270 PRINT:"+++++ ENTER SITE PARAMETERS +++++"
3280 PRINT : " "
3290 PRINT:" GROUND ELEVATION IN FEET"
3300 READ:GREL
3310 PRINT:" TRANSIT ELEVATION IN FEET"
3320 READ:TRANEL
3330 PRINT:" ANTENNA ELEVATION IN FEET"
3340 READ : ANTEL
3350 45 PRINT:" ANTENNA OFFSET FROM TRANSIT ?"
3360 READ:ANSV
3370 IF(ANSV.EQ.IN) GO TO 60
3380 IF(ANSV.NE.IY.AND.ANSV.NE.IN) PRINT 502
3390 IF(ANSV.NE.IY.AND.ANSV.NE.IN) GO TO 45
3400 70 PRINT:" ENTER DISTANCE TO ANTENNA IN FEET"
3410 READ:DTA
3420 IF(DTA.LT.0.) PRINT 502;IF(DTA.LT.0.) GO TO 70
3430 PRINT:" BEARING TO ANTENNA IN DEG AND MIN (DD.MM)"
3440 READ:BEAR
3450 DEG=INT(BEAR)
3460 BTA=(DEG+(BEAR-DEG)/6.)*CDTR
3470 60 PRINT: " STANDARD ALTITUDES ?"
3480 READ : ANSW
3490 IF(ANSW .EQ. IY)GO TO 20
3500 IF(ANSW.NE.IY.AND.ANSW.NE.IN) PRINT 502
3510 IF(ANSW.NE.IY.AND.ANSW.NP.IN) GO TO 60
3520 24 PRINT : " NUMBER OF ALTITUDES TO BE CALCULATED (1-6)"
3530 READ : NA
3540 IF(NA.LT.1.OR.NA.GT.6) PRINT 502
3550 IF(NA .LT. 1 .OR. NA .GT. 6)GO TO 24
3560 PRINT:" DESIRED ALTITUDES IN FEET"
3570 READ : (ALT(I), I = 1,NA)
3580 GO TO 900
3590 20 NA=6
3600 DO 26 I = 1,NA
3610 ALT(I) = STDALT(I)
3620 26 CONTINUE
3630*
3640*    ***PRINT OUT INPUT DATA OF THE COMMUNICATION RANGE
```

```
3650*
3660 900 PRINT 915,GREL,TRANEL,ANTEL
3670 IF(ANSV.EQ.IY) PRINT 916,DTA,BEAR
3680 916 FORMAT(" DIST TO ANT (FT)  =",F8.1/,
3690&" BEAR TO ANT(DD.MM)=",F9.2,/)
3700 PRINT:"AIRCRAFT ALT'S ARE IN FT AGL."
3710 PRINT 917,(ALT(I),I=1,NA)
3720 245 CONTINUE
3730 915 FORMAT(//,"OGROUND  ELEV =",F8.1," FT";/,
3740&" TRANSIT ELEV =",F8.1," FT",/,
3750&" ANTENNA ELEV =",F8.1," FT",/)
3760 917 FORMAT(" ALT'S=",6F8.1,//)
3770        DTA=DTA*.3048
3780 ANTEL=ANTEL*.3048
3790 TRANEL=TRANEL*.3048
3800 GREL=GREL*.3048
3810 502 FORMAT(" ..... INPUT ERROR, TRY AGAIN .....")
3820 94 IF(SIMP.LT..5) GO TO 350
3830*
3840*   ***REQUEST MULTIPLE RUN DATA
3850*
3860 PRINT:"  "
3870 192 PRINT:" NUMBER OF RUNS"
3880 READ:KW
3890 IF(KW.LE.0) GO TO 194
3900 IF(KW-MX3) 196,196,194
3910 194 PRINT 502;GO TO 192
3920 196 DO 228 J=1,KW
3930 IV(J)=0
3940 PRINT 140,J
3950 140 FORMAT(" **** RUN #",I3," ****")
3960 223 PRINT:" FREQUENCY (MHZ)";READ:FR(J)
3970 IF(FR(J).LE.0.) PRINT 502;IF(FR(J).LE.0.) GO TO 223
3980 FRE=FR(J)
3990 628 PRINT:" FED ANTENNA (#)";READ:IV(J)
4000 IW=IV(J)
4010 IF(IW) 224,224,222
4020 222 IF(IW-NB) 225,225,224
4030 224 PRINT 502;GO TO 628
4040 225 IF(IT(IW).EQ.1000) GO TO 224
4050 IF(FRE-200.) 226,226,227
4060 226 IF(IT(IW)-1097) 210,229,210
4070 210 IF(IT(IW)-197) 228,229,228
4080 227 IF(IT(IW)-1181) 228,229,228
4090 229 PRINT 502;GO TO 223
4100 228 CONTINUE
4110 GO TO 400
4120*
4130*   ***PRINT OUT INPUT DATA OF THE GENERAL PROGRAM
4140*
4150 350 PRINT 42,IIM,IGP,IAUTO,FR(1),NN
4160 NQ=NN
```

```
4170 IF(NN.GT.8) NQ=8
4180 PRINT 555,(I,I=1,NQ)
4190 555 FORMAT(" ANT# =",I6,7I8)
4200 IF(AUTO.GT..5) GO TO 4
4210 PRINT 14,(A(I),I=1,NN)
4220 4 PRINT 16,(PO(I),I=1,NN)
4230 PRINT 18,(H1(I),I=1,NN)
4240 PRINT 22,(H3(I),I=1,NN)
4250 PRINT 62,(RA(I),I=1,NN)
4260 PRINT 64,(X(I),I=1,NN)
4270 PRINT 28,(Y(I),I=1,NN)
4280 PRINT 30,(Z(I),I=1,NN)
4290 PRINT 32,(AV(I),I=1,NN)
4300 PRINT 34,(BV(I),I=1,NN)
4310 PRINT 36,(AZ(I),I=1,NN)
4320 PRINT 38,(BZ(I),I=1,NN)
4330 KW=1
4340 GO TO 450
4350*
4360*    ***PRINT OUT INPUT DATA OF THE SIMPLE PROGRAM
4370*
4380 400 PRINT 12,IIM,IGP,ISIMP,ICL,NB
4390 NQ=NB
4400 IF(NB.GT.8) NQ=8
4410 PRINT 555,(I,I=1,NQ)
4420 PRINT 44,(IT(I),I=1,NB)
4430 PRINT 64,(XX(I),I=1,NB)
4440 PRINT 28,(YY(I),I=1,NB)
4450 PRINT 30,(ZZ(I),I=1,NB)
4460 HH1=0.05
4470 IF(IIM.EQ.IIN)HH1=HH1*0.0254
4480 DO 99 I=1,NB
4490 IF(IT(I).NE.1181) GO TO 99
4500 ZZ(I)=ZZ(I)+HH1
4510 99 CONTINUE
4520 DO 156 I=1,NB
4530 IF(H4(I).GT.1.E-10) II8=1
4540 156 CONTINUE
4550 IF(II8.NE.0) PRINT 152,(H4(I),I=1,NB)
4560 IF(II8.NE.0) PRINT 154,(H5(I),I=1,NB)
4570 152 FORMAT(" L    =",8F8.3/(7X,8F8.3))
4580 154 FORMAT(" D    =",8F8.3/(7X,8F8;3))
4590 39 FORMAT(" VER PAT=",2X,A1)
4600 450 PRINT 39,IALP
4610 IF(IFP.EQ.0) GO TO 453
4620 PRINT 66,APHI,AINT
4630 453 PRINT 558,IALT
4640 IF(IFT.EQ.0) GO TO 78
4650 558 FORMAT(" HOR PAT=",2X,A1)
4660 PRINT 41,ATHE,AINP
4670 PRINT 300,IPD
4680 IF(IPA.EQ.0) GO TO 78
```

```
4690 300 FOFMAT(" PAT DIST=",1X,A1)
4700 PRINT 310,IDD,DPER
4710 310 FORMAT(" PLOT=",1X,A1,1X," INC=",F6.1," %")
4720 78 IF(SIMP.GT..5) PRINT 80,IRGE
4730 80 FORMAT(" COM RNG=",2X,A1)
4740 IF(IRG.EQ.0) GO TO 660
4750 PRINT 100,DBM,APIN
4760 100 FORMAT(" REC SEN (DBM)=",F7.1," GROUND TPW (WATT)=",F6.1)
4770 41 FORMAT(" THETA=",F6.1,1X," PLOT INC=",F7.2)
4780 12 FORMAT(///" DIM=",1X,A1,1X," GP=",1X,A1;1X," SIMP=",1X,A1,1X,
4790&" COUPL=",1X,A1,1X," NR=",I3)
4800 14 FORMAT(" #SEGS=",8F8.2/(7X,8F8.2))
4810 16 FORMAT(" L    =",8F8.3/(7X,8F8.3))
4820 18 FORMAT(" LP   =",8F8.3/(7X,8F8.3))
4830 22 FORMAT(" FP   =",8F8.3/(7X,8F8.3))
4840 62 FORMAT(" R    =",8F8.5/(7X,8F8.5))
4850 64 FORMAT(" X    =",8F8.3/(7X,8F8.3))
4860 28 FORMAT(" Y    =",8F8.3/(7X,8F8.3))
4870 30 FORMAT(" Z    =",8F8.3/(7X,8F8.3))
4880 32 FORMAT(" RE V =",8F8.3/(7X,8F8.3))
4890 34 FORMAT(" IM V =",8F8.3/(7X,8F8.3))
4900 36 FORMAT(" RE LD=",8F8.1/(7X,8F8.1))
4910 38 FORMAT(" IM LD=",8F8.1/(7X,8F8.1))
4920 42 FORMAT(///" DIM=",1X,A1,1X," GP=",1X,A1;1X," AUTO=",1X,A1,1X,
4930&" FREQ=",F8.2,1X," NR=",I3)
4940 44 FORMAT(" TYPE =",8I8/(7X,8I8))
4950 66 FORMAT(" PHI  =",F6.1,1X," PLOT INC=",F7.2)
4960 660 RETURN;END
4970$        FORTRAN ASCII,NFORM,NLNO                                  LNKM3A
4980 SUBROUTINE LINKM3
4990 COMMON A,H0,H1,H3,RA,X,Y,Z,AV,BV,AZ,BZ,NS,L1,IF,IT,XX,YY,ZZ,IZ,
5000&FR,IV,IIM,GP,SIMP,NN,AUTO,KW,CL,NB,AINT,AINP,APHI,ATHE,IPA,IPD
5010&,H4,H5,IFP,IFT,IRG,DBM,ISIMP,ICL,IALP,IALT,IRGE,IAUTO,IGP,IDD
5020&,ALT(6),GREL,TRANEL,ANTEL,NA,DTA,BTA,APIN,FA,IDB,DPER,FILENAME
5030 DIMENSION A(200),H0(200),H1(200),H3(200),RA(200),X(200),Y(200),
3040 &Z(200),AV(200),BV(200),AZ(361),BZ(200),NS(200),L1(200),IF(200),
5050 &IT(20),XX(20),YY(20),ZZ(20),IZ(40),FR(10),IV(10),H4(20),H5(20),
5060 &FA(40),STDALT(6);CHARACTER FILENAME*8
5070 INTEGER RUNNO
5080 MAXS=500;IME=10343170080
5090 Iy=11953782816
5100*
5110*    ***IME=METERS
5120*    ***MAXS=MAXIMUM NUMBER OF SEGMENTS
5130*
5140 250 FORMAT(3I4,I2,2F3.0)
5150 252 FORMAT(9F7.3)
5160 254 FORMAT(8F8,5)
5170 256 FORMAT(6E11.3)
5180 258 FORMAT(4I3,4F7.2,F6.1)
5190 260 FORMAT(I5,15I4)
5200 AIN=1.
```

```
5210 IF(IIM.NE.IME) AIN=.0254
5220 MX=1
5230 IF(SIMP.GT..5) PRINT 10
5240 10 FORMAT(//" RUN#     FREQ(MHZ)  ANT FED(#)")
5250 DO 456 L=1,KW
5260 LXY=0
5270 IJ=0
5280 FRE=FR(L)
5290 IF(SIMP.LT..5) GO TO 450
5300*
5310*   ***CALCULATE & STORE NECESSARY PARAMETERS FOR THE SIMPLE PROGRAM
5320*
5330 PRINT 30,L,FR(L),IV(L)
5340 30 FORMAT(I4,5X,F8.2,3X,I6)
5350 DO 350 I=1,NB
5360 NM=1
5370 IF(IT(I).EQ.1097) NM=5
5380 IF(IT(I).EQ.197) NM=26
5390 DO 340 J=1,NM
5400 AJ=J;IJ=IJ+1
5410 AV(IJ)=0.;BV(IJ)=0.
5420 AZ(IJ)=50.;BZ(IJ)=0.
5430 X(IJ)=XX(I)*AIN;Y(IJ)=YY(I)*AIN
5440 IF(IT(I).EQ.1000) GO TO 132
5450 IF(CL-1.) 142,145,145
5460 142 IF(FRE-200.) 143,143,144
5470 143 A(IJ)=8.5*250./FRE;GO TO 148
5480 144 A(IJ)=13.3*250./FRE;GO TO 148
5490 145 IF(FRE-200.) 144,144,146
5500 146 A(IJ)=14.8*300./FRE
5510 148 IF(IT(I).NE.197) GO TO 158
5520 AZ(IJ)=0.
5530 IF(J.EQ.1) GO TO 82
5540 IF(J.EQ.2) GO TO 84
5550 IF(J.LE.14) GO TO 86
5560 Z(IJ)=.51+ZZ(I)*AIN
5570 A(IJ)=11.8*300./FRE
5580 HO(IJ)=.17;RA(IJ)=.0035
5590 GO TO 340
5600 82 A(IJ)=4.2*300./FRE
5610 IF(CL.GT..5.OR.FRE.GT.200.) A(IJ)=6.25*300./FRE
5620 HO(IJ)=.48;RA(IJ)=.0175;Z(IJ)=ZZ(I)*AIN
5630 GO TO 340
5640 84 AZ(IJ)=50.;HO(IJ)=.03;RA(IJ)=.002
5650 Z(IJ)=.48+ZZ(I)*AIN
5660 IF(FRE.LE.200.) GO TO 340
5670 IF(IV(L).NE.I) GO TO 340
5680 AV(IJ)=1.;AZ(IJ)=0.
5690 LXY=1
5700 GO TO 340
5710 86 Z(IJ)=.48+ZZ(I)*AIN
5720 HO(IJ)=.45;RA(IJ)=.0045
```

```
5730 A(IJ)=6.66*300./FRE
5740 IF(FRE.LE.200.) GO TO 46
5750 IF(IV(L).EQ.I.OR.CL.GT..5) A(IJ)=8.89*300./FRE
5760 GO TO 340
5770 46 IF(CL.LT..5) A(IJ)=4.44*300./FRE
5780 GO TO 340
5790 158 IF(NM.EQ.1) GO TO 320
5800 RA(IJ)=.0754;H0(IJ)=.54;H1(IJ)=.27;H3(IJ)=.27
5810 IF(J.EQ.5) GO TO 370
5820 GO TO 330
5830 370 RA(IJ)=.0191;H0(IJ)=.279;AZ(IJ)=0.;H1(IJ)=0.;H3(IJ)=0.
5840 A(IJ)=7.4*300/FRE
5850 GO TO 330
5860 320 RA(IJ)=.0628;H0(IJ)=1.1684;H1(IJ)=.8001;H3(IJ)=.8001
5870 330 Z(IJ)=ZZ(I)*AIN+(AJ-1.)*(H0(IJ)+.1461)
5880 IF(J.EQ.5) Z(IJ)=ZZ(I)*AIN+4.*.54+3.*.1461
5890 IF(IV(L).NE.I.OR.J.EQ.5) GO TO 340
5900*
5910*    ***CALCULATE ANTENNA AT1097 FEED VOLTAGES AV,BV
5920*
5930 AAA=J-1
5940 BETA=2.*3.141593*.04166*FRE/(.6951*300.)
5950 ABC=BETA*AAA
5960 IF(J.EQ.4) ABC=.00378094*FRE
5970 AV(IJ)=COS(ABC);BV(IJ)=-SIN(ABC);AZ(IJ)=0.
5980 A(IJ)=13.3*250./FRE
5990 IF(FRE.GT.200.) A(IJ)=14.8*300./FRE
6000 GO TO 340
6010 132 AZ(IJ)=0.;H1(IJ)=0.;Z(IJ)=ZZ(I)*AIN
6020 H0(IJ)=H4(I)*AIN;RA(IJ)=H5(I)*AIN/2.
6030 A(IJ)=4.*300./FRE
6040 IF(FRE.GT.200.) A(IJ)=6.*300./FRE
6050 340 CONTINUE
6060 350 CONTINUE
6070*
6080*    ***CALCULATE & STORE NECESSARY PARAMETERS FOR THE GENERAL PROGRAM
6090*
6100 450 KK=0.
6110 IF(L.GT.1) GO TO 660
6120 IF(SIMP.GT..5)GO TO 180
6130 DO 50 I=1,NN
6140 IF(IIM.EQ.IME) GO TO 560
6150 H0(I)=.0254*H0(I)
6160 H1(I)=.0254*H1(I)
6170 H3(I)=.0254*H3(I)
6180 RA(I)=.0254*RA(I)
6190 X(I)=.0254*X(I)
6200 Y(I)=.0254*Y(I)
6210 Z(I)=.0254*Z(I)
6220 560 ABV=AV(I)**2+BV(I)**2
6230 IF(ABV.LT.1.E-20) GO TO 50
6240 KK=KK+1
```

```
6250 KF=I
6260 50 CONTINUE
6270 180 II2=0
6280 DO 182 I=1,NN
6290 IF(BZ(I)*BZ(I).GT.1.E-10) II2=1
6300 182 CONTINUE
6310*
6320*    ***WRITE OUTPUT IN FILE 01
6330*
6340 WRITE(01,250) NB,NN,KW,II2,GP,CL
6350 WRITE(01,252) (HO(I),I=1,NN)
6360 WRITE(01,254) (RA(I),I=1,NN)
6370 WRITE(01,252) (X(I),I=1,NN)
6380 WRITE(01,252) (Y(I),I=1,NN)
6390 WRITE(01,252) (Z(I),I=1,NN)
6400 IF(II2.NE.0) WRITE(01,256) (BZ(I),I=1,NN)
6410 42 FORMAT(4F8.1,F9.6,2F6.1)
6420 WRITE(01,258) IFP,IFT,IDB,IRG,ATHE,APHI,AINT,AINP,DPER
6430 IF(IRG.EQ.1) WRITE(01,42) GREL,TRANEL,ANTEL,DTA,BTA,DBM,APIN
6440 IF(IRG.EQ.1) WRITE(01,252) (FA(I),I=1,NB)
6450 IF(IRG.EQ.1) WRITE(01,35) NA,(ALT(I),I=1,NA)
6460 35 FORMAT(I3,(6F8.1))
6470 IF(NB.NE.0) WRITE(01,260) (IT(I),I=1,NB)
6480*
6490*    ***CALCULATE SUB-ANTENNAS PARAMETERS AND CHECK FOR POSSIBLE ERRORS
6500 660 WAVE=300./FRE
6510 MM=0;II=0
6520 DO 210 I=1,NN
6530 CC=1
6540 IF(GP.GT..5.AND.Z(I).LT.1.E-10) CC=0.
6550 IF(AUTO.LT..5) GO TO 100
6560 IF(KK.EQ.1) GO TO 60
6570 GO TO 70
6580 60 DD=(X(KF)-X(I))**2+(Y(KF)-Y(I))**2
6590 DS=SQRT(DD)
6600 BB=10
6610 A(I)=BB*.8
6620 IF(DS/WAVE.GT.2.5) A(I)=BB*.6
6630 IF(DS/WAVE.LT.1.) A(I)=BB
6640 IF(I.EQ.KF) A(I)=BB*1.5
6650 GO TO 100
6660 70 A(I)=8
6670 IF(AV(I)*AV(I).GT.1.E-10.OR.BV(I)*BV(I).GT.1.E-10) A(I)=15
6680 100 DEL=WAVE/A(I)
6690 IF(SIMP.GT..5) GO TO 120
6700 IF(DEL.LT.2.*RA(I).AND.RA(I).LE..02) GO TO 110
6710 IF(DEL.LT..7*RA(I)) GO TO 110
6720 GO TO 120
6730 110 AA=WAVE/RA(I)/2.
6740 IF(RA(I).GT..02) AA=3.*AA
6750 GO TO 140
6760 120 AO=HO(I)/DEL-CC
```

```
6770 A1=H1(I)/DEL+1.-CC
6780 A3=H3(I)/DEL+1.-CC
6790 I0=A0;I1=A1;I3=A3;B0=I0;B1=I1;B3=I3
6800 NS(I)=I0+1;J1=I1+1;J3=I3+1
6810 IF(A0.LT.B0+.5) NS(I)=I0
6820 IF(I0.EQ.0) NS(I)=1
6830 IF(A1.LT.B1+.5.OR.J1.GT.NS(I)) J1=I1
6840 IF(I1.EQ.0) J1=1
6850 IF(A3.LT.B3+.5.OR.J3.GT.NS(I)) J3=I3
6860 IF(I3.EQ.0) J3=1
6870 IF(J1.GT.NS(I)) J1=NS(I)
6880 IF(J3.GT.NS(I)) J3=NS(I)
6890 IF(I.EQ.1) GO TO 130
6900 MM=MM+NS(I-1)
6910 130 L1(I)=MM+J1
6920 IF(I)=MM+J3
6930 IF(IF(I).EQ.L1(I)) GO TO 160
6940 GO TO 210
6950 140 PRINT 150,I,AA
6960 150 FORMAT(" **<F>**  << THE",I3," TH A(I) MUST BE LESS THAN",
6970&F5.1," >>")
6980 GO TO 200
6990 160 ABV=AV(I)*AV(I)+BV(I)*BV(I)
7000 ABZ1=AZ(I)*AZ(I)+BZ(I)*BZ(I)
7010 IF(ABV.LT.1.E-20) GO TO 210
7020 IF(ABZ1.LT.1.E-20) GO TO 210
7030 PRINT 190,I
7040 190 FORMAT(" **<F>**  << THE",I3," TH LOAD FEED ARE TOO CLOSE
7050& >>")
7060 200 II=II+1
7070 210 CONTINUE
7080 IF(II.NE.0) GO TO 500
7090 MM=MM+NS(NN)
7100 IF(NB.EQ.0) GO TO 192
7110 IF(LXY.EQ.0) GO TO 192
7120 MM=0
7130 NR=0
7140 K=0
7150 DO 195 I=1,NB
7160 NX=1
7170 KX=0
7180 IF(IT(I).EQ.1097) NX=5
7190 IF(IT(I).EQ.197) NX=26
7200 IF(NX.EQ.26.AND.IV(L).EQ.I) KX=1
7210 DO 195 J=1,NX
7220 K=K+1
7230 IF(KX.EQ.1.AND.J.GT.3.AND.J.NE.15) GO TO 193
7240 MM=MM+NS(K)
7250 GO TO 194
7260 193 NR=NR+NS(K)
7270 194 L1(K)=L1(K)-NR
7280 IF(K)=IF(K)-NR
```

```
7290 195 CONTINUE
7300 192 IF(MM.LE.MAXS) GO TO 220
7310 PRINT 215,MM
7320 215 FORMAT(" **<F>**  << NO. OF SEGMENTS", I5," ARE TOO BIG >>")
7330 GO TO 500
7340 220 K=0
7350 IF(CL.LT..5) GO TO 600
7360 NZ=0
7370 DO 410 I=1,NB
7380 NM=1
7390 IF(IT(I).EQ.1097) NM=5
7400 IF(IT(I).EQ.197) NM=26
7410 DO 408 J=1,NM
7420 K=K+1
7430 IF(J.EQ.5) GO TO 408
7440 IF(IT(I).EQ.1000) GO TO 408
7450 IF(NM.EQ.26.AND.J.NE.2) GO TO 408
7460 NZ=NZ+1
7470 IZ(NZ)=IF(K)
7480 408 CONTINUE
7490 410 CONTINUE
7500 600 IF(MM.GT.MX) MX=MM
7510 II4=0;II6=0
7520 DO 184 I=1,NN
7530 IF(BV(I)*BV(I).GT.1.E-10) II4=1
7540 IF(AZ(I).GT.1.E-10) II6=1
7550 184 CONTINUE
7560*
7570*    ***WRITE REMAINING DATA IN FILE 01
7580 WRITE(01,40) MM,NZ,IV(L),II4,II6,PR(L)
7590 WRITE(01,252) (AV(I),I=1,NN)
7600 IF(II4.NE.0) WRITE(01,252) (BV(I),I=1,NN)
7610 IF(II6.NE.0) WRITE(01,256) (AZ(I),I=1,NN)
7620 WRITE(01,260) (NS(I),I=1,NN)
7630 WRITE(01,260) (L1(I),I=1,NN)
7640 WRITE(01,260) (IF(I),I=1,NN)
7650 IF(CL.GT..5) WRITE(01,260) (IZ(I),I=1,NZ)
7660 40 FORMAT(3I4,2I3,F7.2)
7670 456 CONTINUE
7680*    ***CALCULATE MAX CORE MAXK,MAX TIME MAXTIM
7690*
7700 MAXK=1018+MX**2*2/1024 +10
7710 MAXTIM=150
7720*
7730*    ***READ RUN #, NORUN FROM 03
7740*    ***INCREMENT RUN # BY 1, WRITE INTO 03
7750*
7760 PRINT:"DATA CORRECT ?"
7770 READ 1303,I;1303 FORMAT(A1)
7780 IF(I.NE.IY) RETURN
7790 READ(03) NORUN
7800 NORUN=NORUN+1
```

```
7810 REWIND 03
7820 WRITE(03)NORUN
7830 REWIND 03
7840 CALL DETACH(03,ISTAT,)
7850*
7860*    ***WRITE THE PROPER CONTROL CARDS IN FILE 02
7870*
7880 WRITE(02,1301) MAXTIM,MAXK,NORUN
7890 1301 FORMAT("$:LIMITS:",I2,",",I3,"K,,20K"/"$:PRMFL:02,W,S,BLA00001/",
7900&"DISTRT/",I4)
7910 IF (IRG.LE.0)GO TO 1440
7920 WRITE(02,1308)FILENAME
7930 1308 FORMAT("$:PRMFL:03,R/W,R,BLA00001/RADIO/",A8)
7940 1440 REWIND 02
7950*
7960*    ***PRINT RUN INFORMATIONS
7970*
7980 PRINT 1302,NORUN,MX,MAXTIM,MAXK
7990 1302 FORMAT(1X,"PROGRAM EXECUTION NUMBER-",I4/1X,"ARRAY DIMENSIONS ="
8000& I3/1X,"JOB REQUIRES ",I2,"/100 HR TIME AND ",I3,"K WORDS CORE.")
8010*
8020*    ***REWIND FILES 01,02, THEN CLOSE FILES
8030*
8040 REWIND 01;REWIND 02;CALL FCLOSE(01);CALL FCLOSE(02)
8050*
8060*    ***CREATE OUTPUT FILE /DISTRT/(NORUN )
8070*
8080 ENCODE(A,1304)NORUN,13,NORUN,13
8090 1304 FORMAT("ACCE CF,BLA00001/DISTRT/",I4,",B/1,100/,W,CLEAR",
8100&A1,2X,"REMO ",I4,A1)
8110*
8120*    ***REMOVE FILE /DISTRT/(NORUN )
8130*    ***SPAWN CARDIN PROGRAM
8140*
8150 CALL CALLTS("ACCE",A,46)
8160 CALL CALLTS("REMO",A(13),10)
8170 CALL CALLTS("CDIN","RUN BLA00001/RADIO/[illegible]SA,RUN1,R",30)
8180 500 RETURN;END
8190$       FORTRAN ASCII,NFORM,NLNO                                         DSTRTP
8200 COMMON /LBLCOM/ GH,GV,SIZE,SPACE,LOC,ANGLE,LENG,TEXT
8210 REAL GH,GV,SIZE,SPACE,LOC(2),ANGLE
8220 INTEGER LENG(10)
8230 CHARACTER TEXT*1(72,10)
8240 DATA GH,GV/3.5,5./,LENG(1),LENG(2)/7,0/
8250 CHARACTER NORUN*4,LINE*26,PTYPE*1,ISO*1,STYPE*1,STOP*1
8260 INTEGER ISTAT,IOK/0400000000000/,IRUN,RTYPE,RSIZE,NRUN,RTYPE1,I,J
8270 REAL AAMAX(3,10),AMAX(3,10),RCRD(4),A
8280 LOC(1)=0.25; LOC(2)=4.75
8290*
8300*    ***REQUEST PROGRAM EXECUTION NUMBER
8310*
8320 10 PRINT 2000
```

```
8330 READ 1000,NORUN
8340 IF(NORUN.EQ." ") STOP
8350 ENCODE(LINF,6000) NORUN
8360 CALL ATTACH(01,LINE,1,0,ISTAT,)
8370 IF(ISTAT.NE.IOK) GO TO 300
8380 DO 20 I=1,10
8390 AAMAX(1,I)=0.
8400 20 AAMAX(2,I)=0.
8410 PTYPE=" "
8420*
8430*    ***ADJUST PAPER
8440*
8450 PRINT 2010
8460 READ 1020,LINE
8470 REWIND 01
8480 IRUN=0
8490 PRINT 2020,NORUN
8500*
8510*    ***READ  RUN INFORMATION HEADINGS FROM FILE 01
8520*
8530 30 READ(01,END=80) RTYPE,RSIZE,(RCRD(I),I=1,RSIZE)
8540 IF(RTYPE.LE.0.OR.RTYPE.GT.5) GO TO 301
8550 GO TO(40,50,30,60,70),RTYPE
8560 40 IF(RSIZE.NE.2.AND.RSIZE.NE.3) GO TO 302
8570 IRUN=IRUN+1
8580 PRINT 2030,IRUN,(RCRD(I),I=1,RSIZE)
8590 GO TO 30
8600 50 IF(RSIZE.NE.3) GO TO 302
8610 IF(IRUN.EQ.0) IRUN=1
8620 PRINT 2040,(RCRD(I),I=1,3)
8630 AAMAX(J,IRUN)=RCRD(2)
8640 GO TO 30
8650 60 IF(RSIZE.NE.1) GO TO 302
8660 PRINT 2050,RCRD(1)
8670 J=1
8680 GO TO 30
8690 70 IF(RSIZE.NE.1) GO TO 302
8700 PRINT 2060,RCRD(1)
8710 J=2
8720 GO TO 30
8730 80 PRINT 2070
8740 100 REWIND 01
8750 IRUN=1
8760*
8770*    ***REQUEST RUN NUMBER ,NRUN
8780*
8790 PRINT 2080
8800 READ 1010,NRUN
8810*
8820*    ***READ RUN DATA FROM FILE 01
8830*
8840 110 READ(01,END=303) RTYPE
```

```
8850 IF(RTYPE.EQ.1.AND.IRUN.EQ.NRUN) GO TO 130
8860 IF(IRUN.EQ.1.AND.RTYPE.NE.1) GO TO 120
8870 IF(RTYPE.NE.1) GO TO 110
8880 IRUN=IRUN+1
8890 GO TO 110
8900 120 BACKSPACE 01
8910 130 PRINT 2090
8920*
8930*    ***REQUEST PLOT TYPE(V,H, B, OR *)
8940*
8950 READ 1020,LINE
8960 IF(LINE.NE."*") PTYPE=LINE
8970 IF(PTYPE.NE."V".AND.PTYPE.NE."H".AND.PTYPE.NE."B") GO TO 130
8980 IF (PTYPE.NE."V")GO TO 66;IF(AAMAX(1,NRUN).EQ.0.)GO TO 130;GO TO 67
8990 66 IF(PTYPE.NE."H")GO TO 68;IF(AAMAX(2,NRUN).EQ.0.)GO TO 130;GO TO 67
9000 68 IF((AAMAX(1,NRUN).EQ.0.).OR.(AAMAX(2,NRUN).EQ.0.)) GO TO 130
9010 67 CONTINJE
9020*
9030*    ***REQUEST IF ISOTROPIC CIRCLE(ISO) DESIRED
9040*
9050 PRINT 2100 ;READ 1020,ISO
9060 IF(LINE.EQ."*") GO TO 195
9070*
9080*    ***REQUEST FORMAT OF PLOT DESIRED( S,N,F),STYPE
9090*
9100 140' PRINT 2110
9110 READ 1020,STYPE
9120 IF(STYPE.NE."S".AND.STYPE.NE."N".AND.STYPE.NE."F") GO TO 140
9130 IF(STYPE.NE."F") GO TO 160
9140 A=1.
9150*    ***CALCULATE THE FACTORS FOR SCALED ,NORMALIZED
9160*    ***  OR FAMILY PLOTS
9170*
9180 DO 150 I=1,10
9190 IF(PTYPE.NE."H") A=AMAX1(A,AAMAX(1,I))
9200 IF(PTYPE.NE."V") A=AMAX1(A,AAMAX(2,I))
9210 150 CONTINUE
9220 A=SQRT(A)
9230 151 DO 155 I=1,10
9240 IF(STYPE.EQ."S") A=SQRT(AMAX1(AAMAX(1,I),AAMAX(2,I),1.))
9250 AMAX(1,I)=SQRT(AAMAX(1,I))/A
9260 AMAX(2,I)=SQRT(AAMAX(2,I))/A
9270 155 AMAX(3,I)=1./A
9280 GO TO 190
9290 160 IF(STYPE.EQ."S") GO TO 151
9300 LINE=PTYPE
9310 IF(PTYPE.NE."B") GO TO 170
9320 IF(ISO.EQ."Y") GO TO 165
9330 J=2
9340 GO TO 175
9350 165 PRINT 2120
9360 READ 1020,LINE
```

```
9370 170 J=1
9380 IF(LINE.EQ."H") J=2
9390 IF(J.EQ.1.AND.LINE.NE."V") GO TO 165
9400 175 DO 180 I=1,10
9410 AMAX(1,I)=1.
9420 AMAX(2,I)=1.
9430 A=SQRT(AMAX1(AAMAX(J,I),1.))
9440 180 AMAX(3,I)=1./A
9450*
9460*    ***REQUEST IF STOP BETWEEN PLOTS
9470*
9480 190 PRINT 2130
9490 READ 1020,STOP
9500*
9510*    ***READY THE PLOTTER
9520*
9530 195 PRINT 2140
9540 200 READ 1020,LINE
9550 205 READ(01,END=304) RTYPE1
9560 IF(PTYPE.EQ."H".AND.RTYPE1.NE.5) GO TO 205
9570 210 READ(01,END=304) A
9580*
9590*    ***PLTL:ENABLE PLOTTER
9600*
9610 PRINT 2150
9620 J=1
9630 IF(RTYPE1.EQ.5) J=2
9640 A=AMAX(J,NRUN)
9650*
9660*    ***READ : RADIATION PATTERN DATA FROM FILE 01
9670*
9680 220 READ(01,END=305) RTYPE,RSIZE,(RCRD(I),I=1,RSIZE)
9690 IF(RTYPE.NE.3) GO TO 230
9700 IF(RSIZE.LT.2.OR.RSIZE.GT.4) GO TO 306
9710*
9720*    ***CALCULATE PEN POSITIONS TO PLOT RADIATION PATTERN
9730*
9740 CALL PPLOT(J,RCRD,A)
9750 GO TO 220
9760 230 IF(RTYPE.NE.5.OR.PTYPE.EQ."V") GO TO 240
9770 RTYPE1=RTYPE
9780 IF(STOP.NE."Y") GO TO 210
9790*
9800*    ***PLTT: TERMINATE PLOT
9810*    ***CHANGE PEN ?
9820*
9830 PRINT 2170
9840 READ 1020,LINE
9850 GO TO 210
9860 240 IF(ISO.NE."Y") GO TO 260
9870 IF(STOP.NE."Y") GO TO 245
9880 PRINT 2170
```

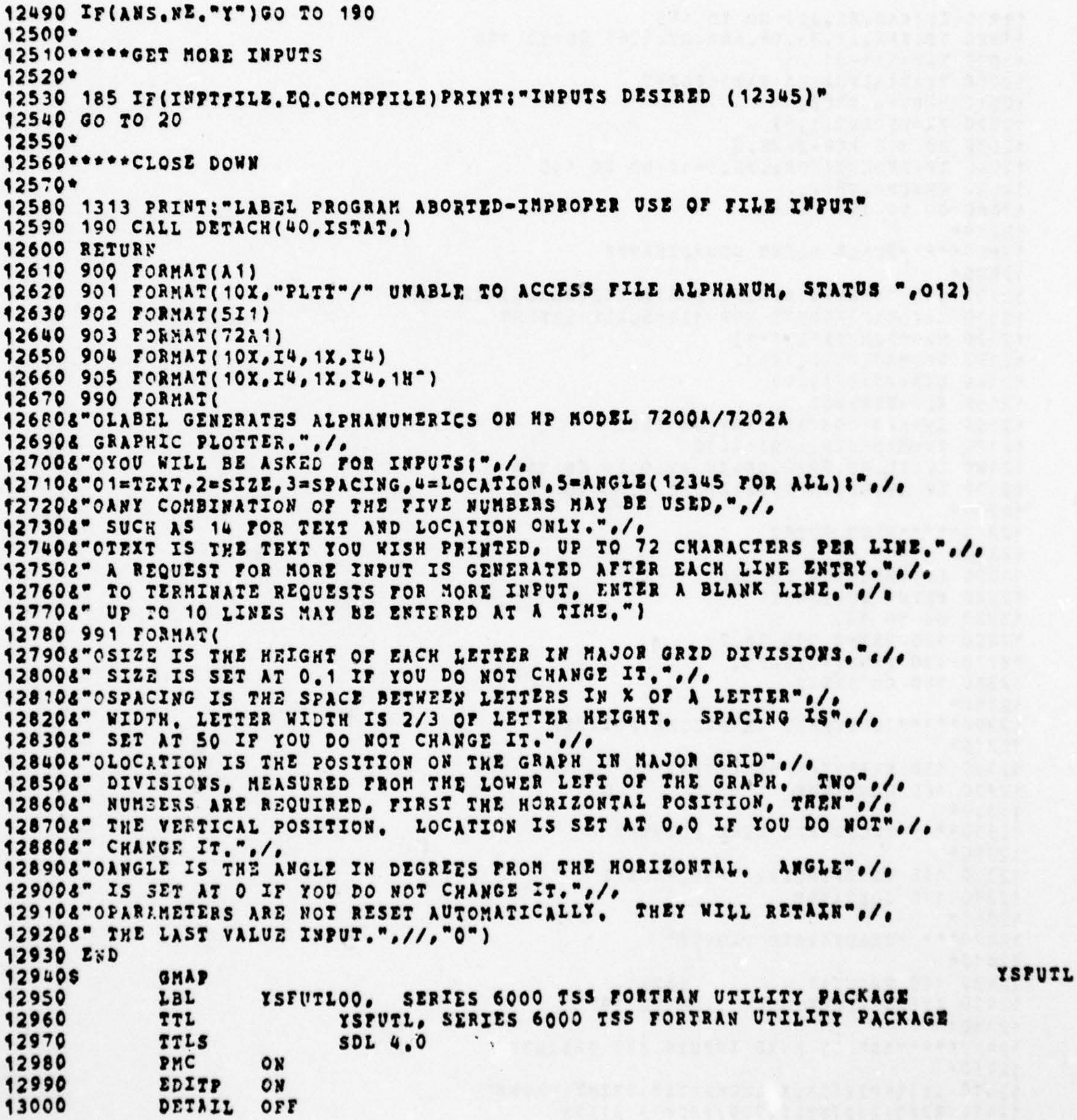
```
9890 READ 1020,LINE
9900*
9910*    ***CALCULATE & PRINT THE DATA POINTS X,Y OF ISOTROPIC CIRCLE
9920*
9930 245 PRINT 2180
9940 RCRD(2)=1.
9950 A=AMAX(3,NRUN)
9960 DO 250 I=1,8
9970 DO 250 J=1,3
9980 RCRD(1)=45*I+2*J-4
9990 250 IF(J.EQ.2.OR.MOD(I,2).EQ.0) CALL PPLOT(2,RCRD,A)
10000*
10010*    ***LEBEL EACH PLOTTING
10020 260 ENCODE(LINE,6020) NORUN,NRUN+100
10030 CALL UNPACK(LINE,1,TEXT,1,7)
10040 CALL LABEL(-1)
10050*
10060*    ***ANOTHER PLOT FROM THIS FILE ?
10070*
10080 PRINT 2200
10090 READ 1020,LINE
10100 IF(LINE.EQ."Y") GO TO 100
10110 REWIND 01
10120 CALL DETACH(01,ISTAT,)
10130*
10140*    ***RELEASE PLOTTER FILE ?
10150*
10160 PRINT 2210
10170 READ 1020,LINE
10180 IF(LINE.NE."Y") GO TO 10
10190 ENCODE(LINE,6010) NORUN,13
10200*
10210*    ***RELEASE THE PLOTTER FILE
10220*
10230 CALL CALLTS("RELE",LINE,26)
10240 GO TO 10
10250 300 PRINT 3000,ISTAT; STOP
10260 301 PRINT 3001,RTYPE; STOP
10270 302 PRINT 3002,RSIZE; STOP
10280 303 PRINT 3003,NRUN; GO TO 100
10290 304 PRINT 3004; STOP
10300 305 RTYPE=0; GO TO 230
10310 306 PRINT 3006; GO TO 302
10320 1000 FORMAT(A4)
10330 1010 FORMAT(V)
10340 1020 FORMAT(A1)
10350 2000 FORMAT(" PROGRAM TO PLOT OUTPUT FROM DISTORT"/
10360&             " ENTER PROGRAM EXECUTION NUMBER")
10370 2010 FORMAT(" ADJUST PAPER")
10380 2020 FORMAT(" PROGRAM EXECUTION NUMBER-",A4)
10390 2030 FORMAT("ORUN NUMBER",I3," ANT# (FED)",I3," FREQ (MHZ)",F8.3,
10400&" TYPE",I5)
```

```
2040 FORMAT(" EMAX=",F8.4," GAIN=",F7.3," GAIN(DB)=",F7.3)
2050 FORMAT(" VERTICAL   PATTERN, PHI  =",F6.1)
2060 FORMAT(" HORIZONTAL PATTERN, THETA=",F6.1)
2070 FORMAT(//////)
2080 FORMAT(" ENTER RUN NUMBER")
2090 FORMAT(" ENTER PLOT TYPE (VER, HOR, BOTH, OR *)")
2100 FORMAT(" PLOT ISOTROPIC CIRCLE?")
2110 FORMAT(" ENTER TYPE OF PLOT DESIRED: SCALED, NORMALIZED, OR ",
&"FAMILY")
2120 FORMAT(" ISOTROPIC CIRCLE RELATIVE TO VER OR HOR PLOT?")
2130 FORMAT(" STOP BETWEEN PLOTS?")
2140 FORMAT(" READY THE PLOTTER")
2150 FORMAT(10X,"PLTL")
2170 FORMAT(10X,"PLTT"/" CHANGE PENS")
2180 FORMAT(10X,"PLTP")
2200 FORMAT(10X,"PLTT"/" ANOTHER PLOT FROM THIS FILE?")
2210 FORMAT(" RELEASE PLOTTER FILE?")
3000 FORMAT(" FILE ACCESS ERROR, STATUS =",O13)
3001 FORMAT(" ILLEGAL RECORD TYPE =",I12)
3002 FORMAT(" ILLEGAL RECORD SIZE =",I12)
3003 FORMAT(" RUN NUMBER",I12," NOT FOUND")
3004 FORMAT(10X,"PLTT"/" UNEXPECTED END OF FILE")
3006 FORMAT(10X,"PLTT")
6000 FORMAT("BLA00001/DISTRT/",A4,";")
6010 FORMAT("RELE BLA00001/DISTRT/",A4,A1)
6020 FORMAT(A4,"-",I2)
END
$       FORTRAN ASCII,NFORM,NLNO                                    DSTRP
*
*    ***SUBROUTINE PPLOT-CALCULATE X,Y COORDINATES OF PEN POSITIONS
*    ***                     AND PLOT
*
SUBROUTINE PPLOT(J,RCRD,A)
REAL RCRD(2)
ANGRAD=RCRD(1)*3.14159/180.
R=RCRD(2)*5000.*A
IX=R*SIN(ANGRAD)
IY=R*COS(ANGRAD)*.7
GO TO (10,20),J
10 IX=IX+5000
IY=IY+5000
GO TO 30
20 IX=-IX+5000
IY=IY+5000
30 PRINT 1000,IX,IY
RETURN
1000 FORMAT(10X,I4,1X,I4)
END
$       FORTRAN ASCII,NFORM,NLNO                                   Y.LABL
SUBROUTINE LABEL(INPTFILE)
*****PROGRAM: LABEL - GENERATES ALPHANUMERICS ON
*     HP MODEL 7200A/7202A GRAPHIC PLOTTER
```

```
10930*
10940 COMMON /LBLCOM/ GH,GV,SIZE,SPACE,LOC,ANGLE,LENG,TEXT
10950 INTEGER INPTFILE,COMPFILE/41/,ALPHANUM/40/,LENG(10),IOK/0400000000000
10960 REAL LOC(2),LINPNT,KAPPNT,MAG
10970 CHARACTER*1 TEXT(72,10)
10980 LOGICAL PENUP
10990 CHARACTER ANS*1,BLANK*1/1H /
11000 DIMENSION INPUT(5),KPOINT(41)
11010 DATA SIZE,SPACE,LOC,ANGLE/0.1,50.,0.,0.,0./
11020*
11030*****GET ALPHANUMERIC PLOTTING POINTS
11040*
11050 CALL ATTACH(40,"BLA00001/PLOTTER/ALPHANUM;",1,1,ISTAT,)
11060 IF(ISTAT.EQ.IOK) GO TO 5; PRINT 901, ISTAT; GO TO 190
11070 5 CALL RANSIZ(40,41,1)
11080*
11090*** SKIP ASKING FOR INST IF USING FILE INPUT
11100*
11110 IF(INPTFILE.EQ.-1) GO TO 105
11120 IF(INPTFILE.LT.1.OR.INPTFILE.GT.39) INPTFILE=41
11130 IF(INPTFILE.NE.COMPFILE) GO TO 10
11140*
11150*****ASK IF INSTRUCTIONS ARE NEEDED
11160*
11170 PRINT:"INSTRUCTIONS ?"
11180 READ 900,ANS
11190 IF(ANS.NE."Y")GO TO 11
11200*
11210*****PRINT INSTRUCTIONS
11220*
11230 PRINT 990
11240 PRINT 991
11250*
11260*****GET WIDTH AND HEIGHT OF GRAPH
11270*
11280 11 PRINT:"INPUT FROM FILE ?"
11290 READ 900,ANS
11300 IF(ANS.EQ."N")GOTO 10
11310 IF(ANS.NE."Y")GOTO 11
11320 PRINT:"FILE NUMBER ?"
11330 READ:INPTFILE
11340 10 IF(INPTFILE.EQ.COMPFILE)PRINT;"GRAPH SIZE IN MAJOR GRID DIVISIONS;
11350 IF(INPTFILE.EQ.COMPFILE)PRINT;"WIDTH"
11360 READ(INPTFILE,119,END=1313)GH
11370 119 FORMAT(V)
11380 IF(INPTFILE.EQ.COMPFILE)PRINT;"HEIGHT"
11390 READ(INPTFILE,119,END=1313)GV
11400*
11410*****GET INPUTS
11420*
11430 IF(INPTFILE.EQ.COMPFILE)PRINT;"INPUTS DESIRED:"
11440 IF(INPTFILE.EQ.COMPFILE)PRINT;
```

```
11450& "1=TEXT,2=SIZE,3=SPACING,4=LOCATION,5=ANGLE(12345 FOR ALL)"
11460 20 READ(INPTFILE,902,END=1313)INPUT
11470 DO 40 I=1,5
11480 INPUTI=INPUT(I)+1
11490 IF(INPUTI.LT.1.OR.INPUTI.GT.6) GO TO 1313
11500 GO TO (100,50,60,70,80,90),INPUTI
11510 50 IF(INPTFILE.EQ.COMPFILE)PRINT:"TEXT:"
11520 DO 55 J=1,10
11530 READ(INPTFILE,903,END=1313)(TEXT(K,J),K=1,72)
11540 LENG(J)=0
11550 DO 53 K=1,72
11560 53 IF(TEXT(K,J).NE.BLANK) LENG(J)=K
11570 IF(LENG(J).EQ.0) GO TO 40
11580 55 CONTINUE
11590 GO TO 40
11600 60 IF(INPTFILE.EQ.COMPFILE)PRINT:"SIZE"
11610 READ(INPTFILE,119,END=1313)SIZE
11620 GO TO 40
11630 70 IF(INPTFILE.EQ.COMPFILE)PRINT:"SPACING (X)"
11640 READ(INPTFILE,119,END=1313)SPACE
11650 GO TO 40
11660 80 IF(INPTFILE.EQ.COMPFILE)PRINT:"LOCATION (X,Y)"
11670 READ(INPTFILE,119,END=1313)LOC(1),LOC(2)
11680 GO TO 40
11690 90 IF(INPTFILE.EQ.COMPFILE)PRINT:"ANGLE"
11700 READ(INPTFILE,119,END=1313)ANGLE
11710 40 CONTINUE
11720 100 IF(INPTFILE.EQ.COMPFILE)PRINT:"DO YOU WANT TO RE-ENTER ANY INPUTS
11730 READ(INPTFILE,900,END=1313)ANS
11740 IF(ANS.EQ."Y")GO TO 185
11750*
11760*****BEGIN CALCULATIONS FOR PLOT
11770*
11780 105 SCALE=SIZE*9999./(60.*GV)
11790 SPAYS=40.*ABS(SCALE)*(1.+SPACE/100.)
11800 XLOC=LOC(1)*9999./GH
11810 YLOC=LOC(2)*9999./GV
11820 ROT=ANGLE*3.1415927/180.
11830 LINPNT=0.
11840*
11850*****ACTIVATE PLOTTER
11860*
11870 PRINT:"          PLTL"
11880 DO 170 LINE=1,10
11890 LENGTH=LENG(LINE)
11900 IF(LENGTH.EQ.0)GO TO 180
11910 KARPNT=0.
11920 DO 160 KN=1,LENGTH
11930*
11940*****GET CHARACTER TO BE PLOTTED
11950*
11960 KAR=FLD(0,9,TEXT(KN,LINE))
```

```
11970 IF(KAR.EQ.32) GO TO 150
11980 IF(KAR.LT.33.OR.KAR.GT.126) GO TO 160
11990 KAR=KAR-31
12000 READ(ALPHANUM'KAR)KPOINT
12010 PENUP=.TRUE.
12020 NK=KPOINT(1)*2
12030 DO 140 KOR=2,NK,2
12040 IF(KPOINT(KOR).NE.(-1))GO TO 110
12050 PENUP=.TRUE.
12060 GO TO 140
12070*
12080*****SCALE POINT COORDINATES
12090*
12100 110 X=FLOAT(KPOINT(KOR))*ABS(SCALE)+KARPNT
12110 Y=FLOAT(KPOINT(KOR+1))*SCALE+LINPNT
12120 MAG=SQRT(X*X+Y*Y)
12130 IF(MAG.EQ.0.)X=1.
12140 DIR=ATAN2(Y,X)
12150 ANG=DIR+ROT
12160 IH=MAG*COS(ANG)*GV/GH+XLOC
12170 IV=MAG*SIN(ANG)+YLOC
12180 IF(IH.GT.9999.OR.IH.LT.0)GO TO 165
12190 IF(IV.GT.9999.OR.IV.LT.0)GO TO 165
12200*
12210*****PLOT POINT
12220*
12230 IF(PENUP)GO TO 120
12240 PRINT 904,IH,IV
12250 GO TO 130
12260 120 PRINT 905,IH,IV
12270 130 PENUP=.FALSE.
12280 140 CONTINUE
12290*
12300*****INCREMENT CHARACTER POINTER
12310*
12320 150 KARPNT=KARPNT+SPAYS
12330 160 CONTINUE
12340*
12350*****INCREMENT LINE POINTER
12360*
12370 165 LINPNT=LINPNT-90.*SCALE
12380 170 CONTINUE
12390*
12400*****DEACTIVATE PLOTTER
12410*
12420 180 PRINT;"           PLTT"
12430 IF(INPTFILE.EQ.-1) GO TO 190
12440*
12450*****ASK IF MORE INPUTS ARE DESIRED
12460*
12470 IF(INPTFILE.EQ.COMPFILE)PRINT;"MORE"
12480 READ(INPTFILE,900,END=1313)ANS
```



```
12490 IF(ANS.NE."Y")GO TO 190
12500*
12510*****GET MORE INPUTS
12520*
12530 185 IF(INPTFILE.EQ.COMPFILE)PRINT:"INPUTS DESIRED (12345)"
12540 GO TO 20
12550*
12560*****CLOSE DOWN
12570*
12580 1313 PRINT:"LABEL PROGRAM ABORTED-IMPROPER USE OF FILE INPUT"
12590 190 CALL DETACH(40,ISTAT,)
12600 RETURN
12610 900 FORMAT(A1)
12620 901 FORMAT(10X,"PLTT"/" UNABLE TO ACCESS FILE ALPHANUM, STATUS ",O12)
12630 902 FORMAT(5I1)
12640 903 FORMAT(72A1)
12650 904 FORMAT(10X,I4,1X,I4)
12660 905 FORMAT(10X,I4,1X,I4,1H")
12670 990 FORMAT(
12680&"OLABEL GENERATES ALPHANUMERICS ON HP MODEL 7200A/7202A
12690& GRAPHIC PLOTTER.",/,
12700&"OYOU WILL BE ASKED FOR INPUTS:",/,
12710&"01=TEXT,2=SIZE,3=SPACING,4=LOCATION,5=ANGLE(12345 FOR ALL):",/,
12720&"OANY COMBINATION OF THE FIVE NUMBERS MAY BE USED,",/,
12730&" SUCH AS 14 FOR TEXT AND LOCATION ONLY.",/,
12740&"OTEXT IS THE TEXT YOU WISH PRINTED, UP TO 72 CHARACTERS PER LINE.",/,
12750&" A REQUEST FOR MORE INPUT IS GENERATED AFTER EACH LINE ENTRY.",/,
12760&" TO TERMINATE REQUESTS FOR MORE INPUT, ENTER A BLANK LINE.",/,
12770&" UP TO 10 LINES MAY BE ENTERED AT A TIME.")
12780 991 FORMAT(
12790&"OSIZE IS THE HEIGHT OF EACH LETTER IN MAJOR GRID DIVISIONS.",/,
12800&" SIZE IS SET AT 0.1 IF YOU DO NOT CHANGE IT.",/,
12810&"OSPACING IS THE SPACE BETWEEN LETTERS IN % OF A LETTER",/,
12820&" WIDTH. LETTER WIDTH IS 2/3 OF LETTER HEIGHT.  SPACING IS",/,
12830&" SET AT 50 IF YOU DO NOT CHANGE IT.",/,
12840&"OLOCATION IS THE POSITION ON THE GRAPH IN MAJOR GRID",/,
12850&" DIVISIONS, MEASURED FROM THE LOWER LEFT OF THE GRAPH.  TWO",/,
12860&" NUMBERS ARE REQUIRED, FIRST THE HORIZONTAL POSITION, THEN",/,
12870&" THE VERTICAL POSITION.  LOCATION IS SET AT 0,0 IF YOU DO NOT",/,
12880&" CHANGE IT.",/,
12890&"OANGLE IS THE ANGLE IN DEGREES FROM THE HORIZONTAL.  ANGLE",/,
12900&" IS SET AT 0 IF YOU DO NOT CHANGE IT.",/,
12910&"OPARAMETERS ARE NOT RESET AUTOMATICALLY.  THEY WILL RETAIN",/,
12920&" THE LAST VALUE INPUT.",//,"0")
12930 END
12940$        GMAP                                                      YSFUTL
12950         LBL     YSFUTL00,  SERIES 6000 TSS FORTRAN UTILITY PACKAGE
12960         TTL            YSFUTL, SERIES 6000 TSS FORTRAN UTILITY PACKAGE
12970         TTLS           SDL 4.0
12980         PMC     ON
12990         EDITP   ON
13000         DETAIL  OFF
```

```
13010       REF     LNRSM,ON
13020       REFMA   ON
13030       LODM    .G3TSM                    DEFINE TSS SYSTEM MACROS
13040       PMC     OFF
13050TRA    BOOL    710200                    TRA OP CODE
13060STUFF  MACRO   REGISTER,LOCATIONS
13070       IDRP    #2
13080       ST#1    #2,$                      STUFF IT!
13090       IDRP
13100       ENDM    STUFF                     THAT WAS SIMPLE, WASN'T IT?
13110.PZER. NULL                              THIS IS THE ORIGIN FOR THIS PROGRAM
13120       BCI     1,740621                  VERSION DATE
13130       DATE                              ASSEMBLY DATE
13140       .SSDRL                            DEFINE DRL EQUIVALENCES
13150       REM
13160       REM
13170       REM     THIS UTILITY PACKAGE IS DESIGNED TO BE USED AS
13180       REM     A SUBROUTINE BY TSS FORTRAN PROGRAMS, IT PERFORMS
13190       REM     MISCELLANEOUS WORTHWHILE(?) FUNCTIONS WHICH ARE
13200       REM     DIFFICULT OR IMPOSSIBLE TO PULL OFF WITHOUT THE
13210       REM     AID OF SUCH A SUBROUTINE.  THERE ARE EIGHT ENTRY
13220       REM     POINTS TO THIS ROUTINE, THE CHARACTERISTICS OF
13230       REM     WHICH ARE DESCRIBED BELOW --
13240       REM
13250       REM       CALL FLGBRK(FLAG)
13260       REM
13270       REM     WHERE FLAG IS A LOGICAL VARIABLE, WILL ALLOW
13280       REM     RECOVERY FROM BREAKS DURING EXECUTION.  WHEN
13290       REM     FLGBRK IS CALLED, FLAG WILL BE SET TO THE LOGICAL
13300       REM     VALUE .FALSE.; IF A BREAK OCCURS SUBSEQUENTLY,
13310       REM     IT WILL BE IGNORED AND EXECUTION CONTINUED, BUT
13320       REM     THE LOGICAL VARIABLE FLAG WILL BE SET TO .TRUE.
13330       REM     TO INDICATE RECOGNITION OF A BREAK.  THIS FLAG MAY
13340       REM     BE TESTED BY THE CALLING ROUTINE AT ANY TIME,
13350       REM     AND ANY APPROPRIATE ACTION TAKEN.  TO PROTECT
13360       REM     THE USER AGAINST INFINITE LOOPS, HOWEVER, BREAKS
13370       REM     ARE COUNTED AS THEY ARE RECEIVED; IF TWENTY OR MORE
13380       REM     BREAKS ARE RECOGNIZED, EXECUTION IS TEMPORARILY
13390       REM     SUSPENDED AND THE FOLLOWING MESSAGE APPEARS --
13400       REM
13410       REM       *BREAK: STOP OR CONTINUE?
13420       REM
13430       REM     AT THIS POINT THE USER MAY ENTER ANY OF THREE
13440       REM     RESPONSES:
13450       REM
13460       REM       S     WILL IMMEDIATELY TERMINATE THE PROGRAM
13470       REM       A     WILL TERMINATE THE PROGRAM WITH A DRL ABORT
13480       REM       C     WILL RESET COUNTERS AND CONTINUE EXECUTION
13490       REM
13500       REM     NOTE:  THE DRL ABORT ALLOWS DUMPING THE SUBSYSTEM
13510       REM     TO A FILE FOR DEBUGGING PURPOSES.
13520       REM
```

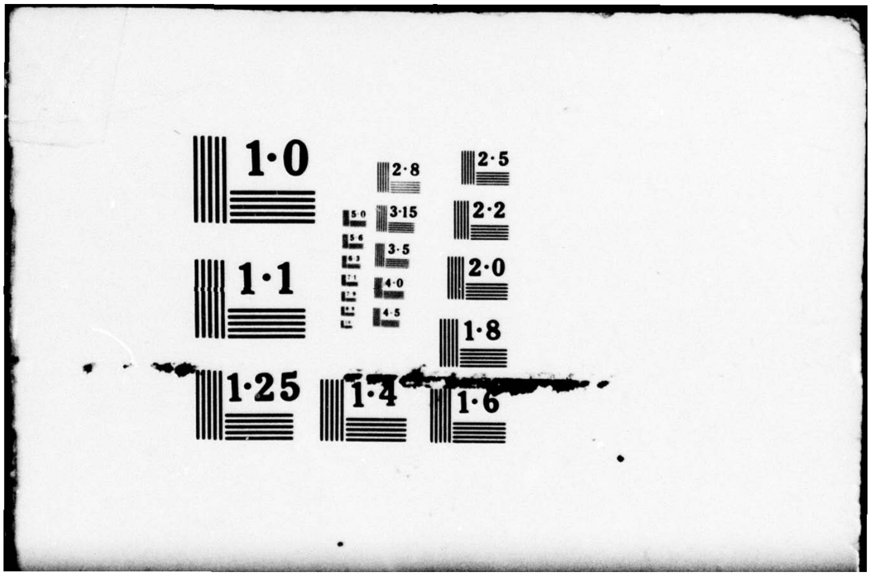
```
13530      REM        CALL BRKOFF
13540      REM
13550      REM     CALLING THIS ENTRY POINT NULLIFIES THE EFFECTS OF A
13560      REM     PREVIOUS CALL TO FLGBRK,
13570      REM
13580      REM        CALL KEYOT
13590      REM
13600      REM     CALLING THIS ENTRY POINT WILL FORCE ANY OUTPUT
13610      REM     ACCUMULATED BY THE SUBSYSTEM TO BE OUTPUT,  THIS
13620      REM     IS OF SIGNIFICANCE WHENEVER SMALL AMOUNTS OF OUTPUT
13630      REM     ARE BEING GENERATED, WITH EXTENDED PROCESSING
13640      REM     INTERVENING; IF THIS FUNCTION IS NOT USED IN
13650      REM     SUCH A CASE, THE OUTPUT WILL NOT BE PRINTED UNTIL
13660      REM     THE EXECUTIVE'S BUFFER FILLS, OR INPUT IS REQUESTED,
13670      REM     EITHER OF WHICH MIGHT NOT OCCUR FOR A LONG TIME,
13680      REM
13690      REM        CALL KEYIN(BUFF,N)
13700      REM
13710      REM     THIS ENTRY POINT ALLOWS THE USER TO RETRIEVE THE LAST
13720      REM     LINE OF INPUT SO LONG AS IT REMAINS AVAILABLE,
13730      REM     THUS IT EFFECTIVELY FUNCTIONS AS A BACKSPACE COMMAND
13740      REM     FOR THE TELETYPE,  THE INPUT IS PLACED IN THE
13750      REM     ARRAY, BUFF, IN 21A4 FORMAT.  N IS AN INTEGER VARIABLE
13760      REM     INTO WHICH WILL BE PLACED THE NUMBER OF CHARACTERS
13770      REM     TRANSMITTED,  NOTE THAT N MAY BE ZERO, INDICATING THAT
13780      REM     THE DATA IS NO LONGER AVAILABLE DUE TO INTERVENING
13790      REM     OUTPUT.  THE DATA PLACED IN BUFF IS NOT EDITED OR BLANK
13800      REM     FILLED IN ANY WAY, AND WILL INCLUDE A CARRIAGE RETURN
13810      REM     CHARACTER AT THE END OF THE LINE,  N INCLUDES THE
13820      REM     CARRIAGE RETURN,
13830      REM
13840      REM        CALL CALLTS(SSNAME,LINE,NCHAR)
13850      REM
13860      REM     THIS ENTRY POINT ALLOWS THE USER TO PLACE DATA IN THE
13870      REM     KIN BUFFER, OR CALL ANOTHER TSS SUBSYSTEM, OR
13880      REM     BOTH.  IF A SUBSYSTEM IS TO BE CALLED, THE FIRST
13890      REM     FOUR CHARACTERS OF THE SUBSYSTEM NAME SHOULD
13900      REM     BE PLACED IN THE ASCII VARIABLE, SSNAME,  IF
13910      REM     SSNAME CONTAINS ALL BLANKS OR IS AN INTEGER ZERO, THE
13920      REM     SUBSYSTEM CALL WILL BE BYPASSED,  IF DATA IS TO BE
13930      REM     PLACED IN THE KIN BUFFER, THE COUNT OF CHARACTERS TO
13940      REM     BE MOVED SHOULD BE PLACED IN THE INTEGER VARIABLE
13950      REM     NCHAR, AND THE ASCII ARRAY, LINE, SHOULD CONTAIN
13960      REM     THE DATA TO BE PLACED IN THE BUFFER, IN 21A4 FORMAT,
13970      REM     UP TO 81 CHARACTERS MAY BE PLACED IN THE BUFFER IN THIS
13980      REM     MANNER.  IF NCHAR IS ZERO, THIS FUNCTION WILL BE
13990      REM     BYPASSED.  AS AN EXAMPLE OF USAGE OF THIS FUNCTION,
14000      REM     ASSUME THE USER WISHES TO OBTAIN THE STATUS OF A
14010      REM     BATCH JOB WITH THE SNUMB, 1234T.  THE COMMAND AT
14020      REM     SYSTEM LEVEL WOULD BE:  JSTS 1234T<CR>,
14030      REM     THIS MAY BE ACCOMPLISHED BY THE TSS FORTRAN PROGRAM
14040      REM     BY PLACING THE SUBSYSTEM NAME(JSTS) IN SSNAME,
14050       REM     PLACING THE COMMAND(JSTS 1234T<CR>) IN LINE,
14060       REM     AND THE CHARACTER COUNT(11) IN NCHAR,  THE
14070       REM     JSTS SUBSYSTEM IS THEN CALLED TO PRODUCE THE JOB
14080       REM     STATUS MESSAGE.
14090       REM
14100       REM       CALL ULASCI(ARRAY,CPOS,NCHAR,L)
14110       REM
14120       REM     THIS ENTRY POINT IS USED TO FORCE ASCII DATA IN AN
14130       REM     ARRAY TO UPPER CASE OR LOWER CASE ASCII CODE.  THIS
14140       REM     IS OCCASIONALLY USEFUL, INASMUCH AS MOST TELETYPES
14150       REM     TRANSMIT ONLY UPPER CASE, AND CERTAIN PIECES
14160       REM     OF SOFTWARE (TSS FORTRAN, FOR EXAMPLE) RECOGNIZE
14170       REM     ONLY UPPER CASE ASCII, WHEREAS LOWER CASE IS
14180       REM     SOMEWHAT EASIER TO WORK WITH AND HENCE IS REQUIRED
14190       REM     BY OTHER TIMESHARING SOFTWARE.  ULASCI WILL
14200       REM     TRANSLITERATE EITHER WAY.  THE DATA IN THE ASCII ARRAY,
14210       REM     ARRAY, IS ACTED UPON, STARTING AT CHARACTER POSITION
14220       REM     CPOS (INTEGER), FOR NCHAR CHARACTERS.  IF L IS EVEN,
14230       REM     THE DATA WILL BE FORCED UPPER CASE; IF L IS ODD,
14240       REM     IT WILL BE FORCED LOWER CASE.
14250       REM     NOTE:  THE CHARACTER POSITION, CPOS, IS
14260       REM     HANDLED IN THE SAME WAY AS IN THE STANDARD TSS
14270       REM     SUBROUTINES, GET#C, PUT#C, AND MOVE#S.
14280       REM
14290       REM       CALL BCDASC(FROM,IPOS,TO,JPOS,NCHAR,FILL)
14300       REM       CALL ASCBCD(FROM,IPOS,TO,JPOS,NCHAR,FILL)
14310.      REM
14320       REM     THESE TWO ROUTINES FUNCTION AS A MOVE#S WITH
14330       REM     TRANSLITERATION AS INDICATED.  IPOS AND JPOS
14340       REM     ARE STARTING CHARACTER POSITIONS IN FROM AND TO,
14350       REM     RESPECTIVELY.  BEAR IN MIND THAT THERE ARE 4 ASCII
14360       REM     CHARACTERS PER WORD, AND 6 BCD CHARACTERS PER WORD.
14370       REM     FILL IS A BLANK FILL CONTROL.  IF THE INTEGER VARIABLE
14380       REM     FILL IS ZERO, THE ARRAY, TO; IS NOT BLANK-FILLED;
14390       AEM     IF FILL IS NONZERO, THE LAST RECEIVING WORD OF THE
14400       REM     ARRAY WILL BE BLANK-FILLED.
14410       TTLS            FLGBRK
14420       REM
14430       REM       FLGBRK ROUTINE--SET BREAK RECOVERY
14440       REM
14450FLGBRK SAVE                           PRIMARY ENTRY POINT
14460       INHIB   ON                     INHIBIT BREAK PROCESSING
14470       EAA     BRKP,$                 GET ADDRESS OF BREAK RTN.
14480       ORA     TRA,DL                 MAKE A TRA OUT OF IT
14490       STA     13                     STORE IN BREAK VECTOR
14500       LDQ     =02000,DU              LOAD BIT 7
14510       DRL     SETSWH                 AND SET SWITCH WORD
14520       EAA     2,1*                   GET LOC OF LOG. VAR.
14530       STA     FLAG,$                 STORE IN POINTER WORD
14540       STZ     FLAG,$*                ZERO THE CALLING VARIABLE
14550       RETURN  FLGBRK                 AND RETURN TO CALLER
14560       TTLS            BRKOFF
```

```
14570       REM
14580       REM      BRKOFF ROUTINE--RESET BREAK RECOVERY
14590       REM
14600BRKOFF SAVE                  ENTRY TO TURN OFF BREAK RECOVERY
14610       LDQ    =02000,DU      GET A BIT
14620       DRL    RSTSWH         RESET SWITCH WORD
14630       STZ    13             ZERO VECTOR
14640       STZ    FLAG,$         AND FLAG WORD
14650       INHIB  OFF            ALL THRU MESSING WITH BREAK PROC.
14660       RETURN BRKOFF         AND RETURN
14670       TTLS         KEYOT
14680       REM
14690       REM      KEYOT ROUTINE--FLUSH TTY OUTPUT BUFFER
14700       REM
14710KEYOT  SAVE                  KOTNOW ENTRY
14720       EAA    MSG,$          FIND A RUBOUT
14730       ORA    64+32+3,DL     ONE CHAR IN POS 03
14740       STA    BUF,$          STORE IN SCRATCH
14750       EAA    BUF,$          FORM DRIVER TALLY
14760       ORA    64,DL          ONE TALLY ONLY
14770       STA    TALLY,$        SAVE
14780       EAA    TALLY,$        LOC OF DRIVER TALLY
14790       STA    2,IC           TO DRL SEQUENCE
14800       DRL    KOTNOW         FLUSH TSS BUFFER
14810       ZERO
14820       RETURN KEYOT          AND RETURN
14830       TTLS         KEYIN
14840       REM
14850       REM      KEYIN ROUTINE--RETRIEVE LAST TTY INPUT
14860       REM
14870KEYIN  SAVE                  ENTRY TO RETRIEVE LAST LINE
14880       EAA    2,1*           GET LOC OF BUFFER
14890       EAQ    3,1*           AND COUNT WORD
14900       ARL    18             SHIFT THEM
14910       LLS    18             INTO A SINGLE WORD
14920       EAQ    STAT,$         SCRATCH
14930      ESTAQ   2,IC           AND STORE IN SEQUENCE
14940       DRL    KIN            GET THE LINE
14950       ZERO
14960       ZERO
14970       RETURN KEYIN          AND RETURN TO CALLER
14980       TTLS         CALLTS
14990       REM
15000       REM      CALLTS ROUTINE--CALL TSS SUBSYSTEM
15010       REM
15020CALLTS SAVE                  ENTRY TO PSEUDO & CALLSS
15030       LDA    2,1*           GET SUBSYSTEM NAME
15040       ORA    ABLNK,$        FORCE LOWER CASE
15050       STA    SSNAME,$       STORE IN CALLSS ARG WORD
15060       LDA    4,1*           GET # OF CHARACTERS TO PSEUDO
15070       TZE    CALTS,$        ZERO, DON'T BOTHER
15080       ALS    6              SHIFT TO TALLY POSITION
```

AD-A065 178   SYRACUSE UNIV N Y   F/G 20/14
COMPUTER PROGRAM MAINTENANCE MANUAL FOR THE ANTENNA PATTERN DIS--ETC(U)
JAN 79   J PERINI, S WANG, K HIRASAWA   F30602-75-C-0121
UNCLASSIFIED   RADC-TR-78-264   NL

2 OF 2
ADA
065178

END
DATE
FILMED
4-79
DDC


1·0
2·8
2·5
5·0
5·6
3·15
2·2
3·5
1·1
4·0
2·0
4·5
1·8
1·25
1·4
1·6

```
15090       ANA     -1,DL           DROP EXTRA CRUD
15100       CMPA    82*64,DL        IS IT GREATER THAN 81?
15110       TMI     2,IC            NO, O. K.
15120       LDA     81*64,DL        FORCE COUNT TO 81 CHARS
15130       ORA     32,DL           MAKE IT A TALLYB
15140       STA     TALLY,$         STORE IN TALLY SCRATCH
15150       EAXO    3,1*            FIND DATA ADDRESS
15160       STXO    TALLY,$         AND PLUG INTO TALLY
15170       EAA     TALLY,$         THE TALLY IS HERE, . .
15180       EAQ     STAT,$          DUMMY STATUS WORD
15190       ARL     18              PUT BOTH
15200       LLS     18              IN ONE WORD
15210       STA     2,IC            AND STORE IN SEQ.
15220       DRL     PSEUDO          PERFORM PSEUDO
15230       ZERO
15240CALTS  LDA     SSNAME,$        GRAB SS NAME AGAIN
15250       CMPA    ABLNK,$         WAS IT BLANK AND/OR ZERO?
15260       TZE     CLTX,$          YES, DON'T DO CALLSS
15270       LDQ     0,DU            GET A ZERO
15280       DRL     SETSWH          SEE WHAT THE SWITCH WORD SAYS
15290       ANQ     =02000,DU       ONLY INTERESTED IN BREAK BIT
15300       STQ     BUF,$           SAVE FOR LATER RESTORE
15310       TZE     2,IC            IF ZERO, DON'T BOTHER RESETTING IT
15320       DRL     RSTSWH          BUT IF NONZERO. . .
15330       DRL     CALLSS          ALL READY,  PUSH TO SS
15340SSNAME ZERO
15350       LDQ     BUF,$           WE'RE BACK,  RETRIEVE BIT
15360       TZE     2,IC            IF ZERO, DON'T SET
15370.      DRL     SETSWH          NONZERO, MUST RESTORE BIT
15380CLTX   RETURN  CALLTS          EXIT TO CALLER
15390       TTLS          ULASCI
15400       REM
15410       REM       ULASCI ROUTINE--FORCE UPPER/LOWER CASE ASCII
15420       REM
15430ULASCI SAVE                    UPPER/LOWER CASE ASCII ROUTINE
15440       LDQ     3,1*            GET STARTING CHARACTER NUMBER
15450       SBQ     1,DL            HIS 1 IS OUR 0
15460       DIV     4,DL            4 CHARS PER WORD
15470       ALS     18              PLACE CHAR POS IN LOWER
15480       LLR     18              AND WORD OFFSET IN UPPER
15490       ADQ     32,DL           MAKE TALLYB
15500       STQ     TALLY,$         AND STORE IN SCRATCH
15510.      LDA     4,1*            GET CHARACTER COUNT
15520       ALS     6               SHIFT
15530       ANA     -1,DL           REMOVE CRUD AGAIN
15540       ASA     TALLY,$         ADD TO THE CONFUSION
15550       EAA     2,1*            GET DATA LOC
15560       ASA     TALLY,$         COMPOUND INTEREST
15570       EAXO    TALLY,$         GET ADDRESS OF OUR TALLY
15580       STUFF   X0,(T1,T2,T3,T4)
15590       LXLO    5,1*            GRAB U/L FLAG
15600       LDQ     LIMITS+2,$      ASSUME UPPER
```

```
15610       ANX0    1,DU             UNLESS WE FIND OUT OTHERWISE
15620       TZE     2,IC             HOW ABOUT THAT?
15630       LDQ     LIMITS+3,$       OH, WELL. . .
15640       EAX1    LIMITS,$         ADDRESS OF OTHER LIMITS
15650       STX1    1,IC             STORE
15660       LDA     **,0             LOAD LOW LIMIT TO BE ALTERED
15670T1     CWL     **,CI            GET A CHARACTER.  IS IT?
15680       TNZ     6,IC             NOPE.  LEAVE IT ALONE
15690T2     LDA     **,CI            AHA!  GRAB HIM
15700       ERA     =040,DL          ZAP!  IF YOU WERE, YOU AREN'T NOW.
15710T3     STA     **,SC            STORE BACK WHERE IT CAME FROM
15720       TTF     -6,IC            MORE TO GO?
15730       TRA     3,IC             ALL DONE, EXIT
15740T4     NOP     **,SC            ADVANCE TALLY
15750       TTF     -8,IC            MORE?
15760       RETURN  ULASCI           THIS WAY OUT
15770       TTLS            BCDASC--ASCBCD
15780       REM
15790       REM        BCDASC/ASCBCD ROUTINES
15800       REM
15810BCDASC SAVE    2,7              BCD TO ASCII TRANSLITERATOR
15820       EAX2    XPARAM,$         GET POINTERS TO PARAMETERS
15830       TSX7    XINIT,$          AND GO INITIALIZE
15840       RETURN  BCDASC           DONE, EXIT
15850ASCBCD SAVE    2,7              ASCII TO BCD ENTRY
15860       EAX2    XPARAM+8,$       NEEDED DATA
15870       TSX7    XINIT,$          INITIALIZE
15880       RETURN  ASCBCD           ALL DONE
15890       EJECT
15900XINIT  EAX0    BUF,$            LOAD TALLY ADDRESS
15910       STX0    S1,$             STORE
15920       EAX0    1,0              STORE TALLY ADDRESS
15930       STUFF   X0,(S5,S7)       STUFF IT TOO
15940       LXL0    4,2              GET # CH / WD IN CODE WE'RE GOING TO
15950       STX0    S2,$             STORE IN DIVIDE
15960       LXL0    5,2              THIS IS WHAT A BLANK LOOKS LIKE
15970       STX0    S6,$             STORE IT TOO
15980       EAX0    6,2              POINTER TO XLIT TABLE POINTER
15990       STX0    S3,$             STORE IT
16000       EAX0    7,2              AND THE PTR TO SHIFT TABLE PTR
16010       STX0    S4,$             DITTO
16020       EAA     BCDTAB,$         ADDRESS OF XLIT TABLES, BCDASC
16030       EAQ     SHFTB,$          AND BCDASC SHIFT TABLE
16040       STAQ    XPARAM+6,$       STORE INDIRECT WORD
16050       EAA     ASCTAB,$         ADDRESS OF ASCBCD XLIT TABLE
16060       EAQ     SHFTA,$          AND SHIFT TABLE
16070       STAQ    XPARAM+14,$      TO ITS INDIRECT WORD
16080       LDQ     3,1*             GET CHARACTER POSITION
16090       SRQ     1,DL             CORRECT
16100       DIV     0,2              DIVIDE BY # CH / WD
16110       ALS     18               SHIFT
16120       LLR     18               TO TALLY FORMAT
```

```
16130       ADQ     1,2               ADD CONDITIONAL TALLYB
16140       STQ     BUF,$             STORE
16150       EAA     2,1*              LOAD DATA ADDRESS
16160       ASA     BUF,$             PLACE IN TALLY
16170       LDQ     5,1*              BUILD OTHER TALLY BY SIMILAR PROCESS
16180       SBQ     1,DL
16190       DIV     2,2
16200       ALS     18
16210       LLR     18
16220       ADQ     3,2
16230       STQ     BUF+1,$           STORE
16240       EAA     4,1*
16250       ASA     BUF+1,$           STORE ADDRESS
16260       LDA     6,1*              GET CHARACTER COUNT
16270       ALS     6                 SHIFT
16280       ANA     -1,DL             DROP GARBAGE
16290       ORSA    BUF+1,$           PLACE IN TALLY
16300S1     LDQ     **,SC             GET A CHARACTER.
16310S2     DIV     **,DL             WHERE IS IT?
16320S3     LDQ     **,*QL            LOAD CORR WORD
16330S4     XEC     **,*AL            MOVE TO LOWER CHAR POS
16340S5     STQ     **,SC             STORE
16350       TTF     -5,IC             CONTINUE
16360       SZN     7,1*              DOES HE WANT BLANK FILL?
16370       TZE     0,7               NO, RETURN
16380S6     LDA     **,DL             YES,  GET A BLANK
16390       LDQ     7,DL              AND A MASK.
16400       CANQ    BUF+1,$           IS THE WORD FILLED?
16410       TZE     0,7               YES, CUIT
16420S7     STA     **,SC             NO, STORE BLANK
16430       TRA     -3,IC             CONTINUE
16440       TTLS              BREAK PROCESSING
16450       REM
16460       REM          BREAK PROCESSOR--FLGBRK
16470       REM
16480       INHIB   ON                BREAK PROCESSING MUST BE INHIBITED
16490BRKP   STA     SVA,$             SAVE AR
16500       LDA     1,DU              GET A .TRUE.
16510       STA     FLAG,$*           STORE IN FLAG VAR
16520       AOS     CNT,$             COUNT THIS BREAK
16530       TPL     OVER,$            OOPS!!!  MAYBE WE'D BETTER ASK.
16540       EAA     BRKP,$            RESET BREAK VECTOR
16550       ORA     TRA,DL            WITH A TRA
16560       STA     13
16570       LDA     SVA,$             RETRIEVE AR
16580       RET     12                AND CONTINUE
16590OVER   LDA     SVA,$             RETRIEVE AR
16600       SREG    REGS,$            MOVE REGISTERS TO A SAFE PLACE
16610       LDA     12                ALSO SAVE IC&I FOR A POSSIBLE RET
16620       STA     REGS+7,$          (DON'T REALLY NEED THE TIMER REGISTER)
16630AGAIN  EAA     0,IC              RESET BREAK VECTOR HERE
16640       ORA     TRA,DL            TO PREVENT DISASTERS
```

```
16650        STA     13               IN THE EVENT OF RECURSIVE BREAKS
16660        EAA     MSG,$            LOC OF A MESSAGE
16670        ORA     30*64+32,DL      MANUFACTURE A TALLY
16680        STA     TALLY+2,$        STORE IN SCRATCH
16690        EAA     TALLY+2,$        AND A TALLY
16700        ORA     64,DL            THAT POINTS TO THE TALLY
16710        STA     TALLY+1,$        IN THE PREVIOUS WORD
16720        EAA     TALLY+1,$        AND YET ANOTHER POINTER
16730        STA     2,IC             STORE IN KOUTN
16740        DRL     KOUTN            WRITE MESSAGE
16750        ZERO
16760        EAA     BUF+2,$          ADDRESS OF INPUT BUFFER
16770        EAQ     TALLY+1,$        SCRATCH
16780        ARL     18               COMBINE
16790        LLS     18
16800        EAQ     STAT,$           MORE SCRATCH
16810       ESTAQ    2,IC             STORE IN CALL
16820        DRL     KIN              RETRIEVE INPUT
16830        ZERO
16840        ZERO
16850        SZN     TALLY+1,$        WAS THERE ANY?(THERE SHOULD BE)
16860        TZE     AGAIN,$          HUH?
16870        LDA     BUF+2,$          GET FIRST WORD OF INPUT
16880        ARL     27               ISOLATE FIRST CHARACTER
16890        ORA     =040,DL          FORCE LOWER CASE
16900        CMPA    =0163,DL         IS IT AN S?
16910        TNZ     2,IC             NO
16920        DRL     RETURN           SCREEEEEECH!!!!!
16930        CMPA    =0141,DL         HOWABOUTANA?
16940        TNZ     2,IC             STILL NO
16950        DRL     ABORT            BOOOOOOOM!!!!!
16960        CMPA    =0143,DL         LAST CHANCE--IS IT A C?
16970        TNZ     AGAIN,$          IT'S YOUR MONEY, NOT MINE.
16980        LCA     20,DL            GET A -20
16990        STA     CNT,$            RESTORE COUNT
17000        EAA     BRKP,$           BUILD A TRA
17010        ORA     TRA,DL           TO THE ORIGINAL BREAK ROUTINE
17020        STA     13               AND STORE IN VECTOR
17030        LREG    REGS,$           GRAB REGISTERS
17040        RET     REGS+7,$         AND RETURN
17050        INHIB   OFF              ALL THROUGH WITH THE HEAVY STUFF
17060        TTLS            STORAGE & CONSTANTS
17070        REM
17080        REM        STORAGE & CONSTANTS SECTION
17090        REM
17100XPARAMEDEC      6,0,4,32,4,32,0,0
17110        DEC     4,32,6,0,6,16,0,0
17120SHFTA  QRL     5*6              ASCBCD SHIFT TABLE
17130        QRL     4*6
17140        QRL     3*6
17150        QRL     2*6
17160        QRL     1*6
```

```
17170         NOP    0,DU
17180SHFTB    QRL    3*9                 BCDASC SHIFT TABLE
17190         QRL    2*9
17200         QRL    1*9
17210         NOP    0,DU
17220ASCTAB   BSS    5
17230         BCI    9,00 !"#$%&'()*+,-,/0123456789:;<=>?@ABCDEFGHIJKLMNOPQRS
17240         BCI    8,TUVWXYZ['']^\0ABCDEFGHIJKLMNOPQRSTUVWXYZ00000
17250BCDTAB   ASCII  13,0123456789[#@:>? ABCDEFGHI&.](<'^JKLMNOPQR-$*);'+/ST
17260         ASCII  3,UVWXYZ\,X="!
17270FLAG     ZERO                       POINTER TO FLGBRK FLAG
17280STAT     BSS    1                   DUMMY STATUS WORD
17290ABLNK    ASCII  1,                  ONE WORD OF ASCII BLANKS
17300TALLY    BSS    3                   SCRATCH FOR TALLIES
17310BUF      BSS    23                  INPUT LINE BUFFER & SCRATCH
17320LIMITS   OCT    141,101,172,132
17330CNT      DEC    -20                 MAX BREAK COUNT
17340MSG      OCT    015012012177        MSG--COUNT EXCEEDED
17350         UASCI  7,*BREAK: STOP OR CONTINUE?
17360REGS    8BSS    8                   BREAK REGISTER SAFE-STORE
17370SVA      EQU    REGS
17380         ORG    REGS                BACK UP TO REG STORE AREA
17390         SYMDEF SAVE                ADD SAVE ENTRY POINT
17400SAVE     LDQ    =3HOH*,DL           LOAD SAVE FILE CODE
17410         MME    GESAVE              AND SAVE US
17420         BCI    1,YSFUTL            GESAVE NAME
17430         ZERO   .PZER.,.PLST.-.PZER; SIZE & LENGTH
17440         ZERO   FLGBRK              AN ENTRY POINT, FER GRINS
17450         MME    GEFINI              WE'SE SAVED; SO STOP ALREADY
17460         EIGHT                      GET US BACK TO THE END OF REGS
17470         LIT
17480         ERLK
17490.PLST.   NULL
17500         END
17510$        GMAP                                                           PACK
17520         LBL    PACK0000,  PACK, UNPACK, MOVE 6000 FORT BCD/ASCII
17530         TTL      PACK, UNPACK, MOVE BCD/ASCII STRINGS  6000 FORTRAN
17540         TTLS
17550         EDITP  ON
17560         SYMDEF PACK
17570         SYMDEF UNPACK
17580         SYMDEF MOVE
17590         BCI    1,740926            VERSION DATE
17600         DATE                       ASSEMBLY DATE
17610MOVE     STX1   .E.L..              SAVE X1
17620         STI    .E.L..              AND INDICATORS FOR RETURN
17630         STX7   RET+1               SAVE X7 FOR CALLER
17640         EAX7   MV                  GET LOC OF MOVE TABLE
17650         TRA    *+10                ENTER COMMON PROCEDURE
17660PACK     STX1   .E.L..              SAVE X1
17670         STI    .E.L..              AND INDICATORS
17680         STX7   RET+1               AND X7
```

```
17690       EAX7   PK                      GET LOC OF PACK TABLE
17700       TRA    *+5                     GO TO COMMON PROCEDURE
17710UNPACK STX1   .E.L..                  SAVE REGISTERS
17720       STI    .E.L..
17730       STX7   RET+1
17740       EAX7   UPK                     GET LOC OF UNPACK TABLE
17750       STX6   RET                     SAVE ANOTHER REGISTER
17760       EAA    2,1*                    CHECK OUT FIRST ARGUMENT
17770       TNZ    *+9                     IS IT NULL?
17780       SZN    3,1*                    IT IS.  CHECK SECOND ARGUMENT
17790       TZE    *+4                     IS IT NONZERO?
17800       EAX6   ASC                     IT IS.  SET MODE TO ASCII
17810       STX6   X6
17820       TRA    RET                     AND RETURN
17830       EAX6   BCD                     SECOND ARG IS ZERO, SET MODE BCD
17840       STX6   X6
17850       TRA    RET                     RETURN
17860       LDX6   X6                      FIRST ARG IS NOT NULL - GET MODE
17870       EAA    2,1*                    GET LOC OF "FROM" AREA
17880       STA    TAL1                    PLACE ADDRESS IN TALLY WORD
17890       EAA    4,1*                    GET LOC OF "TO" AREA
17900       STA    TAL2                    PLACE IN OTHER TALLY WORD
17910       LDA    6,1*                    GET CHARACTER COUNT
17920       ALS    6                       SHIFT FOR TALLY COUNT
17930       ANA    -1,DL                   DROP REFUSE
17940       ORSA   TAL2                    AND "OR" INTO TALLY
17950       LDQ    3,1*                    GET "FROM" CHARACTER POSITION
17960       SBQ    1,DL                    ADJUST FOR FORTRAN'S INDEXING
17970       XED    6,7                     PRODUCE OFFSET & CPOS
17980       LLR    18
17990       XEC    5,7                     ADD CONDITIONAL TALLYB BIT
18000       ASQ    TAL1                    MODIFY THE TALLY AS REQUIRED
18010       LDQ    5,1*                    GET "TO" CHARACTER POSITION
18020       SBQ    1,DL                    ADJUST INDEX
18030       XED    8,7                     PRODUCE OFFSETS
18040       LLR    18
18050       XEC    10,7                    AND CONDITIONAL TALLYB
18060       ASQ    TAL2                    ADD TO TALLY WORD
18070       XED    0,7                     PROCESS THE CHARACTER STRING
18080       XED    2,7                     ACCORDING TO ENTRY POINT AND MODE
18090       TTF    *-2                     USING TALLIES WE JUST BUILT
18100       XEC    4,7                     CHECK FOR BLANK FILL REQUIRED
18110       TZE    RET                     RETURN IF NONE
18120       LDA    1,6                     ELSE GET A BLANK
18130       LDQ    7,DL                    AND A MASK
18140       CANQ   TAL2                    IS THE LAST WORD FULL?
18150       TZE    RET                     IT IS, SO RETURN
18160       STA    TAL2,SC                 IT ISN'T, SO ADD A BLANK
18170       TRA    *-3                     GO CHECK AGAIN
18180RET    LDX6   **,DU                   RESTORE REGISTERS
18190       LDX7   **,DU
18200       RET    .E.L..                  AND RETURN TO CALLER
```

```
18210PK     ELDA     TAL1,ID             GET A WORD
18220        ARL     0,6*                RIGHT JUSTIFY THE CHARACTER
18230        STA     TAL2,SC             STORE IT
18240        TTF     -1,IC               AND CONTINUE
18250        SZN     7,1*                CHECK FOR BLANK FILL
18260        NOP     0,DU
18270        NOP     0,DU
18280        LDA     0,DU                ZERO THE AR
18290        DIV     3,6                 DIVIDE FOR PROPER TALLY CPOS
18300        ALS     18                  PUT REMAINDER IN AU
18310        ORQ     2,6                 POSSIBLY SET TALLYB BIT
18320UPK    ELDA     TAL1,SC             GET A CHARACTER
18330        ALS     0,6*                PUT IT IN AU
18340        ORA     1,6                 BLANK FILL IT
18350        STA     TAL2,ID             AND STORE THE WORD
18360        TRA     RET                 NO BLANK FILL ON UNPACK
18370        ORQ     2,6                 ADD POSSIBLE TALLYB
18380        DIV     3,6                 DIVIDE FOR OFFSET & CPOS
18390        ALS     18                  PLACE REMAINDER (CPOS) IN AU
18400        NOP     0,DU
18410        LDA     0,DU                ZERO AR
18420        NOP     0,DU
18430MV     ELDA     TAL1,SC             GET A CHARACTER
18440        STA     TAL2,SC             STORE IT AGAIN
18450        TTF     -1,IC               LOOP
18460       .TRA     2,IC                BREAK OUT OF LOOP
18470        SZN     7,1*                CHECK FOR BLANK FILL
18480        ORQ     2,6                 SET TALLYB BIT
18490        DIV     3,6                 COMPUTE OFFSET & CPOS
18500        ALS     18                  PUT CPOS IN AU
18510        DIV     3,6                 COMPUT OFFSET & CPOS (THE OTHER ONE)
18520        ALS     18                  POSITION
18530        ORQ     2,6                 SET BIT
18540BCD     ARG     30                  # OF BITS TO SHIFT
18550        BCI     1,0                 BLANK FILL
18560        OCT     0                   NO TALLYB BIT
18570        DEC     6                   6 CHARACTERS PER WORD
18580ASC     ARG     27                  # OF BITS TO SHIFT
18590        OCT     000040040040        BLANK FILL
18600        OCT     40                  THE TALLYB BIT
18610        DEC     4                   FOUR CHARACTERS PER WORD
18620TAL1    BSS     1                   WORD INTO WHICH "FROM" TALLY IS BUILT
18630TAL2    BSS     1                   DITTO, "TO" TALLY
18640X6      ARG     ASC                 POINTER TO BCD/ASCII PARAMETERS
18650        END
18660$       GMAP                                                        DIMXXX
18670        LBL     DIMXXX,  ALLOCATE STORAGE FOR DISTORT
18680        TTL          ALLOCATE STORAGE FOR DISTORT
18690        SYMDEF  ......              ENTER THE BATCH PORTION THRU HERE
18700        SYMREF  .FBAD.
18710HOLE    EQU     31                  LOCATION OF AVAIL CORE WORD
18720...... LDA      1,DL                OPEN FILE 01 AS INPUT
```

```
        STA     .FBAD.
        CALL    .FOPEN(=0)       IN ORDER TO ALLOC:TE BUFFERS
        LDA     2,DL             DO THE SAME FOR FILE 02
        STA     .FBAD.
        CALL    .FOPEN(=1)       EXCEPT OUTPUT
        LDA     6,DL             FILE 06 TOO
        STA     .FBAD.
        CALL    .FOPEN(=1)       ALSO OUTPUT
        LXLO    HOLE             NOW THAT OTHER DEMANDS HAVE BEEN SATISFIED
        SBXO    HOLE             COMPUTE AVAILABLE CORE LEFT
        EAQ     0,0              DIVIDE BY 2 AND STORE FOR COMPARE
        QRL     18               SHIFT TO LOWER HALF
        SBLQ    1600,DL          SAVE SPACE FOR RANDOM FILE
        QRL     1                DIVIDE BY 2
        STQ     AVAIL
        LDQ     DIM              GET TRIAL ARRAY DIMENSION
        MPY     DIM              SQUARE IT
        CMPQ    AVAIL            AND CHECK TO SEE IF IT'S TOO BIG
        TRC     *+3              IT IS
        AOS     DIM              IT'S NOT.  INCREASE AND TRY AGAIN
        TRA     *-5
        LCA     1,DL             DECREASE DIM BY 1 SO IT FITS
        ASA     DIM
        LDXO    HOLE             GET LOC OF AVAIL
        STXO    CALL+3           STORE IT IN CALLING SEQ
        STXO    CALL+4
        LDQ     DIM              SQUARE ARRAY DIMENSION
        MPY     DIM              TO GET EXTENT OF EACH ARRAY
        QLS     18               SHIFT TO QU
        ASQ     CALL+4           AND ADD TO OFFSET SECOND ARRAY
        QLS     1                MULT DIM**2 BY 2
        ASQ     HOLE             AND UPDATE CORE USED
        MME     GESNAP           SNAP CALL FOR DEBUGGING
        ZERO    CALL+3,4
CALL    CALL    MAIN(**,**,DIM)
DIM     DEC     1                DIMENSION OF ARRAYS
AVAIL   BSS     1                AVAIL CORE / 2
        END
$       FORTRAN                                                               USAMAI
*
CMAIN         ***MAIN PROGRAM***
*
*
        MAIN PROGRAM
*
*
      SUBROUTINE MAIN(C1,C2,IDIM)
       COMMON X,Y,Z,X1,Y1,Z1,HO,NS,FP,AK,IG,NN,NB,AC,BC,IT,AZ,NA,RNG
     1 ,GREL,TRANEL,ANTEL,DTA,BTA,DBM,IFP,IFT,IRG
       DIMENSION C1(IDIM,IDIM),C2(IDIM,IDIM)
        DIMENSION HO(200),RA(200),X(200),Y(200),Z(200),AV(200),
     1 BV(200),AZ(361),L1(200),NS(200),IF(200),IZ(200),AC(500),
19250      2 RNG(6).FA(40).AP(5).BP(5).ZZ(5).AO(5).BQ(5).LR(500),
19260      3 IT(20).C5(3.3),BC(500).A3(13).B3(13).C3(5.16).C4(5,16),
19270      4 FR(10).IV(10).C6(3,3).BZ(200).X1(200).Y1(200).Z1(200),
19280      5 S3(5).EE(4).FF(4),NP(20),ALT(6)
19290        CALL RANSIZ(03,9)
19300        N5=11;N6=13
19310        AN=N5-1
19320        CN=N6-1
19330*
19340*     ***READ :PART OF FILE 01
19350*
19360        READ(1,10) NB,NN,KW,II2,GP,CL
19370        IG=0
19380        IF(GP.GT..5) IG=1
19390        IC=0
19400        IF(CL.GT..5) IC=1
19410        READ(1,12) (HO(I),I=1,NN)
19420        READ(1,14) (RA(I),I=1,NN)
19430        READ(1,12) (X(I),I=1,NN)
19440        READ(1,12) (Y(I),I=1,NN)
19450        READ(1,12) (Z(I),I=1,NN)
19460        DO 4 I=1,NN
19470 4      BZ(I)=0.
19480        IF(II2.NE.0) READ(1,16) (BZ(I),I=1,NN)
19490 17     FORMAT(4F8.1,F9.6,2F6.1)
19500        READ(1,18) IFP,IFT,IDB,IRG,ATHE,APHI,AINT,AINP,DPER
19510        IF(IRG.EQ.1) READ(1,17) GREL,TRANEL;ANTEL,DTA,BTA,DBM,APIN
19520        IF(IRG.EQ.1) READ(1,12) (FA(I),I=1,NB)
19530        IF(IRG.EQ.1) READ(1,15) NA,(ALT(I),I=1,NA)
19540        DDD=DBM
19550        DBM=1.E-3*10.**(DBM*.1)
19560 15     FORMAT(I3,(6F8.1))
19570*
19580*     ***SPECIFY SUB-ANTENNA POSITIONS (X,Y,Z) AND (X1,Y1,Z1)
19590*
19600        CALL PXYZ
19610 10     FORMAT(3I4,I2,2F3.0)
19620 12     FORMAT(9F7.3)
19630 14     FORMAT(8F8.5)
19640 16     FORMAT(6E11.3)
19650 18     FORMAT(4I3,4F7.2,F6.1)
19660 20     FORMAT(I5,15I4)
19670 3      FORMAT(3I4,2I3,F7.2)
19680        DO 100 KV=1,KW
19690*
19700*     ***READ:THE REST OF FILE 01
19710*
19720        READ(1,3) MM,NZ,IV(KV),II4,II6,FR(KV)
19730        HX=.015
19740        ALAM=300./FR(KV)
19750        IW=IV(KV)
19760        DO 6 I=1,NN
```

```
19770          BV(I)=0.
19780 6        AZ(I)=0.
19790          READ(1,12) (AV(I),I=1,NN)
19800          IF(II4.NE.0) READ(1,12) (BV(I),I=1,NN)
19810          IF(II6.NE.0) READ(1,16) (AZ(I),I=1,NN)
19820          READ(1,20) (NS(I),I=1,NN)
19830          READ(1,20) (L1(I),I=1,NN)
19840          READ(1,20) (IF(I),I=1,NN)
19850          IF(IC.EQ.1) READ(1,20) (IZ(I),I=1,NZ)
19860*          IF(NB.NE.0) WRITE(2,320) IV(KV),FR(KV)
19870         IF(NB.NE.0) WRITE(2) 1,2,IV(KV),FR(KV)
19880 320      FORMAT(//' *** ANT# (FED)=',I3,1X,' FREQ (MHZ)=',F7.2,' ***')
19890          FP=FR(KV)*1.E6
19900          IF(NB.NE.0) WRITE(6,320) IV(KV),FR(KV)
19910*          IF(IRG.EQ.1) WRITE(2,305) DDD,APIN
19920          IF(IRG.EQ.1) WRITE(6,305) DDD,APIN
19930 305      FORMAT(//' RECEIVER SENSITIVITY (DBM)=',F6.1/
19940        1 ' TRANSMITTER POWER (WATT)  =',F6.1//)
19950 310      A4=0.
19960          B5=0.
19970          DO 24 I=1,MM
19980          LR(I)=0
19990          DO 24 J=1,MM
20000          C1(I,J)=0.
20010 24       C2(I,J)=0.
20020          DO 19 I=1,5
20030          DO 19 J=1,16
20040          C3(I,J)=0.
20050 19       C4(I,J)=0.
20060          EF=0.
20070          EG=0.
20080          AK=2.*3.141593
20090          CW=AK*.25/CN
20100          EPS=1.E-9/(18.*AK)
20110          XMU=2.E-7*AK
20120          OME=AK*FP
20130          OMP=OME*EPS
20140          XK=OME/3.E8
20150          XL=XK*XK
20160*
20170*     ***CALCULATE THE GENERALIZED IMPEDANCE MATRIX C1,C2
20180*
20190          K2=2
20200          IF(IG.EQ.0) K2=1
20210          BLO=ALOG(2.)
20220          II=0
20230          K3=0
20240          K=0
20250          NX=NB
20260          IF(NB.EQ.0) NX=NN
20270          DO 325 KS=1,NX
20280          MK=0
```

```
          NK=1
          NP(KS)=0
          IF(NB.EQ.0) GO TO 8
          IU=-1
          IF(IT(KS).EQ.1097) NK=5
          IF(IT(KS).EQ.197) NK=26
          IF(NK.EQ.26.AND.IW.EQ.KS) MK=1
    8     DO 324 KQ=1,NK
          K=K+1
*
*     ***MK=1: #197 FEED AND CURRENTS ARE SYMMETRICAL ABOUT AXIS
*     ***KQ:SUB-ANTENNA NUMBER
*
          IF(MK.EQ.1.AND.KQ.GT.3.AND.KQ.NE.15) GO TO 324
          L=0
*
*     ***USE THE SYMMETRY PROPERTY OF #197 ANTENNA
*     ***ABOUT THE AXIS ,WHEN IT IS FED,FOR KQ=1,2
*
          MV=0
          K9=K-1
          KG=1
          IF(IG.NE.0.AND.ABS(Z(K)).LT.1.E-5) KG=0
          AS=NS(K)+KG
          IF(NK-26) 298,290,298
    290   IF(KQ-2) 295,298,292
    292   IF(KQ-14) 295,295,294
    294   MV=2
          AS=AS-1.
          GO TO 298
    295   MV=1
          AS=AS-1.
    298   DK=HO(K)/AS
          NP(KS)=NP(KS)+NS(K)
          IF(K.EQ.1) GO TO 26
    182   II=II+NS(K9)
    26    NSK=NS(K)
          J=0
          DO 80 LS=1,NX
          NL=1
          ML=0
          IF(NB.EQ.0) GO TO 9
          IF(IT(LS).EQ.1097) NL=5
          IF(IT(LS).EQ.197) NL=26
          IF(NL.EQ.26.AND.LS.EQ.IW) ML=1
    9     DO 80 LQ=1,NL
          L=L+1
          L9=L-1
          K8=0
          KG=1
          IF(K.NE.L.OR.RA(L).LT..02*ALAM) KG=0
          IF(IG.NE.0.AND.ABS(Z(K)).LT.1.E-5) KG=0
```

```
       LG=1
       IF(IG) 126,128,126
126    IF(Z(L)) 128,127,128
127    LG=0
128    AT=NS(L)+LG
       MW=0
       IF(NL-26) 288,281,288
281    IF(LQ-2) 285,288,282
282    IF(LQ-14) 285,285,284
284    MW=2
       AT=AT-1.
       GO TO 288
285    MW=1
       AT=AT-1.
288    DL=HO(L)*.5/AT
       IF(K.NE.L.OR.RA(L).LT..02*ALAM) LG=0
       ALP=DL*.5
       XKD=XK*ALP
       AP1=1.005*ALP
       AKL=AK*ALP
       XD=XKD*XKD
       AW=2.*ALP/AN
       IF(L9) 28,28,180
180    IF(ML.EQ.0) GO TO 134
       IF(LQ.GT.3.AND.LQ.NE.15) GO TO 28
134    J=J+NS(L9)
28     NSL=NS(L)
       DO 460 KK=1,K2
       N1=1
       G=3-2*KK
       IF(KK-1) 162,146,162
146    IF(NB) 148,164,148
148    IF(KS-LS) 440,430,440
430    IF(LS.EQ.1) GO TO 440
       IF(IU.EQ.0) GO TO 440
       IF(IU.GT.0) GO TO 460
       L8=26
       IF(IT(LS).EQ.197.AND.LS.NE.IW) GO TO 435
       IF(IT(LS).NE.1097) GO TO 440
       IF(LS.EQ.IW.AND.IC.EQ.0) GO TO 440
       L8=5
435    LX=LS
       K1=LS-1
       IU=0
       PO 438 JX=1,K1
       IF(L8-10) 445,445,443
443    IF(JX.EQ.IW.OR.IT(JX).NE.197) GO TO 438
       GO TO 448
445    IF(IT(JX).NE.1097.OR.(JX.EQ.IW.AND.IC.EQ.0)) GO TO 438
448    IU=IU+1
       KX=JX
438    CONTINUE
```

```
       IF(IU.EQ.0) GO TO 440
       CALL QZP(K1,L8,II,L9,MX,MZ,KX,NP)
       DO 450 I=1,MZ
       MP=MX+I
       KM=II+I
       DO 450 JX=1,MZ
       MQ=MX+JX
       LN=II+JX
       C1(KM,LN)=C1(MP,MQ)
       C2(KM,LN)=C2(MP,MQ)
450    CONTINUE
       GO TO 460
440    IF(K9) 164,164,138
138    IF(K-L) 164,139,164
139    IF(NL.EQ.26.OR.IT(KS).EQ.1000) GO TO 164
       DO 152 K5=1,K9
       RR=RA(K5)-RA(K)
       HH=HO(K5)-HO(K)
       HN=NS(K5)-NS(K)
       RH=RR*RR+HH*HH+HN*HN
       IF(RH-1.E-6) 150,152,152
150    DO 156 K7=1,K3
       K8=K7
       IF(LR(K7)-K5) 156,154,156
156    CONTINUE
154    GO TO 164
152    CONTINUE
       GO TO 164
162    IF(Z(L)) 164,158,164
158    N1=2
164    KP=(KK-1)**2+(K-L)**2
       DO 455 M=1,NSK
       KM=II+M
       IF(KP) 23,22,23
22     IF(NL.EQ.26) GO TO 23
       N1=M
23     ZK=M-1+KG
       PZK=Z(K)+DK*ZK
       IF(IG.NE.0.AND.ABS(Z(K)).LT.1.E-5) ZK=ZK-1
       DO 78 N=N1,NSL
       LN=J+N
       IF(IT(KS).EQ.1000) GO TO 31
       IF(KK-1) 31,25,31
25     IF(NL.EQ.26.OR.NK.EQ.26) GO TO 32
       IF(L-K) 78,27,31
27     IF(N1-1) 29,29,72
29     IF(K9) 173,31,173
173    IF(K8) 31,31,174
174    CC=C3(K8,N)
       CD=C4(K8,N)
       GO TO 76
32     IF(KS.NE.LS) GO TO 31
```

```
21850          IF(KK.EQ.2) GO TO 11
21860          IF(ML.EQ.1) GO TO 11
21870          IF(KQ.LE.3.AND.LQ.LE.3) GO TO 31
21880          IF(LQ.NE.3.AND.LQ.NE.15) GO TO 36
21890          IF(KQ.GT.3.AND.KQ.NE.15) GO TO 37
21900          GO TO 31
21910 36       IF(KQ.EQ.3.OR.KQ.EQ.15) GO TO 31
21920          JP=0
21930          JQ=0
21940          IF(KQ.GE.4.AND.KQ.NE.15) JP=NS(K)
21950          IF(LQ.GE.4.AND.LQ.NE.15) JQ=NS(L)
21960          GO TO 40
21970 11       IF(KQ.GT.2) GO TO 31
21980          IF(LQ.LE.3.OR.LQ.EQ.15) GO TO 31
21990          CC=C5(M,N)
22000          CD=C6(M,N)
22010          GO TO 76
22020 31       ZL1=N-1+LG
22030          CC=0.
22040          CD=0.
22050          MC=4
22060          IF(K.NE.L.OR.RA(L).LT..02*ALAM) MC=1
22070          DO 69 MN=1,MC
22080          IF(MC.EQ.1) GO TO 35
22090          IF(N-N1) 34,35,34
22100 34       IF(MN-3) 64,35,35
22110*
22120*      ***SIN EXPANSION AND PULSE TESTING FOR
22130*      ***RADIUS <0.02 OF WAVELENGTH
22140*
22150 35       AMN=MN-1
22160          ZL=ZL1+ZL1-1.5+AMN
22170          IF(MC.EQ.4) GO TO 329
22180          ZL=ZL1+.5
22190          IF(IG.NE.0.AND.ABS(Z(L)).LT.1.E-5) ZL=ZL-1
22200          JQ=11-(KK-1)*4
22210          IF(NL.LT.6) GO TO 108
22220          IF(ML.EQ.0) JQ=7-(KK-1)*2
22230          IF(ML.EQ.0.AND.IC.EQ.0) JQ=5
22240 108      IF(KS.NE.LS) JQ=3
22250          QJ=JQ-1
22260          KT=JQ/2+1
22270          QK=KT-1
22280          JY=1
22290          IF(N.EQ.N1.OR.JQ.EQ.3) GO TO 361
22300          JY=2
22310          A3(1)=A3(JT)
22320          B3(1)=B3(JT)
22330 361      JT=JQ
22340          DO 360 JJ=JY,JT
22350          AJ=JJ-1
22360          DL=HO(L)/AT
```

```
      IF(JJ.LE.KT) DD=DL/QJ
      DE=DL/QJ
      ZD=(ZL+AJ/QJ)*DL
      RQ=RA(L)
      T1=X1(L)-X(L)
      T2=Y1(L)-Y(L)
      T3=Z1(L)-Z(L)
      R1=SQRT(T1*T1+T2*T2+T3*T3)
      IF(JJ.LE.KT.OR.MW.NE.1) GO TO 363
      IF(N.NE.NSL) GO TO 363
      DL=HX
      DE=DL/QJ
      RQ=.002
      T1=0.
      T2=0.
      T3=1.
      R1=1.
      PXL=X1(L)
      PYL=Y1(L)
      PZL=Z1(L)+(AJ-QK)*DE
      GO TO 370
363   IF(JJ.GT.KT.OR.MW.NE.2) GO TO 366
      IF(N.NE.1) GO TO 366
      DL=HX
      DD=DL/QJ
      RQ=.002
      T1=0.
      T2=0.
      T3=1.
      R1=1.
      ZD=-DL*.5+AJ*DD
366   ZE=ZD/R1
      PXL=X(L)+ZE*T1
      PYL=Y(L)+ZE*T2
      PZL=Z(L)+ZE*T3
370   A3(JJ)=0.
      B3(JJ)=0.
      IF(JQ.EQ.3.AND.JJ.NE.2) GO TO 360
      DO 350 JX=1,2
      AJX=JX
      U1=X1(K)-X(K)
      U2=Y1(K)-Y(K)
      U3=Z1(K)-Z(K)
      RO=SQRT(U1**2+U2**2+U3**2)
      DK=HO(K)/AS
      IF(JX-1) 352,352,355
352   IF(MV.NE.2.OR.M.NE.1) GO TO 353
      DK=HX
      U1=0.
      U2=0.
      U3=1.
      RO=1.
      PXK=X(K)
      PYK=Y(K)
      PZK=Z(K)-DK
      GO TO 354
353   DKR=DK*ZK/RO
      PXK=X(K)+DKR*U1
      PYK=Y(K)+DKR*U2
      PZK=Z(K)+DKR*U3
354   DR=DK/RO
      PX1=PXK+DR*U1
      PY1=PYK+DR*U2
      PZ1=PZK+DR*U3
      GO TO 358
355   IF(MV.NE.1.OR.M.NE.NSK) GO TO 356
      DK=HX
      U1=0.
      U2=0.
      U3=1.
      RO=1.
      PXK=X1(K)
      PYK=Y1(K)
      PZK=Z1(K)
      GO TO 357
356   PXK=PX1
      PYK=PY1
      PZK=PZ1
357   DR=DK/RO
      PX1=PXK+DR*U1
      PY1=PYK+DR*U2
      PZ1=PZK+DR*U3
358   DX=XK*DK
      CS=COS(DX)
      SN=SIN(DX)
      P1=PXL-PXK
      P2=PYL-PYK
      P3=PZL*G-PZK
      P4=PXL-PX1
      P5=PYL-PY1
      P6=PZL*G-PZ1
      ZH=(P1*U1+P2*U2+P3*U3)/RO
      IF(ABS(ZH).LT.1.E-15) ZH=0.
      C=P1**2+P2**2+P3**2+RO**2
      IF((C-ZH**2)-RO**2) 666,666,667
666   SD=RO; GO TO 668
667   SD=SQRT(C-ZH**2)
668   DX=PXK+ZH*U1/RO
      DY=PYK+ZH*U2/RO
      DZ=PZK+ZH*U3/RO
      WX=(PXL-DX)/SD
      WY=(PYL-DY)/SD
      WZ=(PZL*G-DZ)/SD
      R4=SQRT(C)
      R5=SQRT(P4**2+P5**2+P6**2+RQ**2)
      XR=XK*R4
      SR=SIN(XR)
      CR=COS(XR)
      XT=XK*R5
      ST=SIN(XT)
      CT=COS(XT)
      ZI=ZH-DK*(2.-AJX)
      IF(ABS(ZI).LT.1.E-15) ZI=0.
      WS=ZI*ZI
      IF(JX-1) 380,380,384
380   WD=R5*R5
      WR=XT*WD
      E1=30.*((-SR/R4+CS*ST/R5)/SN-ZI*(ST-XT*CT)/WR)
      E2=30.*((-CR/R4+CS*CT/R5)/SN-ZI*(CT+XT*ST)/WR)
      E3=30.*((ZH*SR/R4-ZI*CS*ST/R5)/SN-(XT*WS*CT+(WD-WS)*ST
     1 )/WR)/SD
      E4=30.*((ZH*CR/R4-ZI*CS*CT/R5)/SN-((WD-WS)*CT-XT*WS*ST
     1 )/WR)/SD
      GO TO 386
384   WD=R4*R4
      WR=XR*WD
      E1=30.*((-ST/R5+CS*SR/R4)/SN+ZI*(SR-XR*CR)/WR)
      E2=30.*((-CT/R5+CS*CR/R4)/SN+ZI*(CR+XR*SR)/WR)
      ZI=ZH-DK
      E3=30.*((-ZH*CS*SR/R4+ZI*ST/R5)/SN+(XR*WS*CR+(WD-WS)*SR
     1 )/WR)/SD
      E4=30.*((-ZH*CS*CR/R4+ZI*CT/R5)/SN+((WD-WS)*CR-XR*WS*SR
     1 )/WR)/SD
386   EX=E3*WX+E1*U1/RO
      FX=E4*WX+E2*U1/RO
      EY=E3*WY+E1*U2/RO
      FY=E4*WY+E2*U2/RO
      EZ=E3*WZ+E1*U3/RO
      FZ=E4*WZ+E2*U3/RO
      A3(JJ)=((EX*T1+EY*T2)*G+EZ*T3)/R1+A3(JJ)
350   B3(JJ)=((FX*T1+FY*T2)*G+FZ*T3)/R1+B3(JJ)
360   CONTINUE
      IF(KS-LS) 362,364,362
362   EF=A3(2)*(DD+DE)
      EG=B3(2)*(DD+DE)
      GO TO 368
364   CALL FUN(DD,DE,A3,EF,JQ)
      CALL FUN(DD,DE,B3,EG,JQ)
368   AQ(MN)=-EF
      BQ(MN)=-EG
      AM=1.
      GO TO 57
*    ***4 PULSE EXPANSION AND IMPULSE TESTING FOR
*    ***RADIUS  NOT LESS THAN 0.02 WAVE LENGTH
*    ***----FOR Z-DIRECTED WIRE ONLY
329   PZL=Z(L)+DL*ZL
```

```
          PL=1.
          IF(G) 330,335,335
    330   PL=-1.
    335   R33=DK*.5
          R43=DL*.5
          XX=X(K)-X(L)
          YY=Y(K)-Y(L)
          S3(1)=PZK-PZL*G
          S3(2)=S3(1)+R33+R43
          S3(3)=S3(1)-R33-R43
          S3(4)=S3(1)+R33-R43
          S3(5)=S3(1)-R33+R43
          OX=OME*XMU*DK*DL
          DO 60 I=1,5
          ZZ(I)=S3(I)
          C=XX**2+YY**2+ZZ(I)**2+RA(L)**2
          ZJ=ZZ(I)*ZZ(I)
          AA=C-ZJ
          B=SQRT(C)
          ABZ=ABS(ZZ(I))
*
*      ***DISTANCE BETWEEN 2 PULSES NOT LESS THAN
*      ***10*0.25*LENGTH OF SUBSECTION ?
*
          IF(B-10.*ALP) 42,38,38
*
*      ***THIN WIRE APPROXIMATION
*
    38    ACOS=COS(XK*B)/(2.*AK*B)
          BSIN=-SIN(XK*B)/(2.*AK*B)
          ZR2=ZJ/C
          IF(ZR2.LT.1.E-10) ZR2=0.
          ZR4=ZR2*ZR2
          DR2=ALP*ALP/C
          H=(-1.+3.*ZR2)/6.+(3.-30.*ZR2+35.*ZR4)/40.*DR2
          A2=-ZR2/6.-DR2*(1.-12.*ZR2+15.*ZR4)/40.
          A2=A2*1.E5
          XZR=1.E5*XD*ZR4/120.
          PSI1=1.+DR2*H+XD*(A2+XZR)*1.E-5
          PSI2=XKD*(H+XD*(3.*ZR2-5.*ZR4)/60.)*ALP/B
          GO TO 50
*
*      ***CONSTANT CURRENTS ARE ASSUMED OVER THE SURFACE
*      ***OF THE CYLINDER INSTEAD OF OF A FILAMENT
*
    42    C=AA*2.+ZJ
          B=SQRT(C)
          ACOS=COS(XK*B)/AK
          BSIN=-SIN(XK*B)/AK
          A0=XL*B*.5
          A1=(1.-XL*C*.5)/AKL
          A2=-XL*.5/AKL
```

```
24450          BO=XK*(-.5+XL*(C*.25+(ALP*ALP/3.+C+AA)/12.))
24460          B1=XK*B*(1.-XL*C/6.)/AKL
24470          B2=-XK*XL*B*.5/AKL
24480          DO 45 MZ=1,N5
24490          BN=MZ-1
24500          ZP=ALP*((BN+BN)/AN-1.)
24510          ZX=ZZ(I)-ZP
24520          Z3=4.*AA+ZX*ZX
24530          Z2=SQRT(Z3)
24540          P=4.*AA/Z3
24550          B3(MZ)=ELE(P)*Z2
24560          IF(ABZ-AP1) 45,45,39
24570 39       A3(MZ)=ELK(P)/Z2
24580 45       CONTINUE
24590          CALL FUN(AW,AW,B3,B5,N5)
24600          IF(ABZ-AP1) 46,46,44
24610 44       CALL FUN(AW,AW,A3,A5,N5)
24620          GO TO 48
24630 46       AI=1.
24640          IF(ABZ-1.E-5) 52,53,53
24650 52       AI=2
24660 53       DO 47 MZ=1,N6
24670          BN=MZ-1
24680          TP=AK*.25*BN/CN
24690          S=SIN(TP)
24700          Y3=ALP*ALP+AA*S*S*AI*AI
24710          Y2=ALP+SQRT(Y3)
24720          Y7=Y2/(AI*RA(L))
24730 47       A3(MZ)=ALOG(Y7)
24740          CALL FUN(CW,CW,A3,A4,N6)
24750          A5=AI*(.5*AK*BLO+2.*A4)*.5
24760 48       PSI1=A0+A1*A5+A2*B5
24770          PSI2=B0+B1*A5+B2*B5
24780*
24790*    ***CALCULATE THE FIELD AT THE POSITIONS OF WIRES FROM
24800*    ***FOUR PULSE CURRENT EXPANSIONS
24810*
24820 50       AP(I)=ACOS*PSI1-BSIN*PSI2
24830          BP(I)=BSIN*PSI1+ACOS*PSI2
24840 60       CONTINUE
24850 56       AQ(MN)=-OX*BP(1)+(BP(4)+BP(5)-BP(2)-BP(3))/OMP
24860          BQ(MN)=OX*AP(1)-(AP(4)+AP(5)-AP(2)-AP(3))/OMP
24870          GO TO 68
24880 64       AQ(MN)=AQ(MN+2)
24890          BQ(MN)=BQ(MN+2)
24900 68       AM=.75
24910          IF(MN-1) 55,55,54
24920 54       IF(MN-4) 57,55,55
24930 55       AM=.25
24940*
24950*    ***CALCULATE TOTAL FIELD FROM 4 PULSE OR 2 HALF SIN
24960*    ***CURRENTS (CC,CD)
```

```
24970*
24980 57      CC=AQ(MN)*AM+CC
24990 69      CD=BQ(MN)*AM+CD
25000         IF(KK.EQ.2) GO TO 76
25010         IF(KS.NE.LS.OR.ML.EQ.0) GO TO 41
25020         IF(KQ.GT.2) GO TO 41
25030         IF(LQ.EQ.3.OR.LQ.EQ.15) GO TO 49
25040 41      IF(NL.EQ.26.OR.IT(KS).EQ.1000) GO TO 76
25050         IF(KP) 76,160,76
25060 160     IF(NB) 161,76,161
25070 161     IF(N-1) 167,166,167
25080 166     K3=K3+1
25090 167     C3(K3,N)=CC
25100         C4(K3,N)=CD
25110         LR(K3)=K
25120         GO TO 76
25130 49      C5(M,N)=CC
25140         C6(M,N)=CD
25150         GO TO 76
25160 37      I3=15
25170         IF(KQ.LT.15) I3=3
25180         I1=KM-NS(K)*(KQ-I3)
25190         J1=LN+NS(L)*(12+I3-KQ)
25200         GO TO 43
25210 40      I1=KM-JP
25220         J1=LN-JQ
25230 43      C1(KM,LN)=C1(I1,J1)
25240         C2(KM,LN)=C2(I1,J1)
25250         GO TO 78
25260 72      K1=KM-1
25270         L2=LN-1
25280         C1(KM,LN)=C1(K1,L2)
25290         C2(KM,LN)=C2(K1,L2)
25300 79      C1(LN,KM)=C1(KM,LN)
25310         C2(LN,KM)=C2(KM,LN)
25320         GO TO 78
25330*
25340*     ***CALCULATE THE MATRIX ELEMENTS OF (C1,C2)
25350*
25350 76      C1(KM,LN)=CC+C1(KM,LN)
25370         C2(KM,LN)=CD+C2(KM,LN)
25380         IF(KK.EQ.1.AND.(NL.NE.26.AND.NK.NE.26)) GO TO 79
25390 78      CONTINUE
25400 455     CONTINUE
25410 460     CONTINUE
25420 80      CONTINUE
25430 324     CONTINUE
25440 325     CONTINUE
25450*
25460*     ***ADD THE LOAD AZ,BZ
25470*
25480         DO 90 M=1,NN
```

```
25490        I=L1(M)
25500        C1(I,I)=C1(I,I)+AZ(M)
25510 90     C2(I,I)=C2(I,I)+BZ(M)
25520*
25530*   ***INVERT THE MATRIX C1,C2
25540*
25550        MY=MM
25560 98     DO 105 I=1,MY
25570 105    LR(I)=I
25580        DO 118 M=1,MY
25590        K=M
25600        DO 102 I=M,MY
25610        A1=C1(I,M)*C1(I,M)+C2(I,M)*C2(I,M)
25620        A2=C1(K,M)*C1(K,M)+C2(K,M)*C2(K,M)
25630        IF(A1-A2) 102,102,106
25640 106    K=I
25650 102    CONTINUE
25660        LS=LR(M)
25670        LR(M)=LR(K)
25680        LR(K)=LS
25690        STOR1=C1(K,M)
25700        STOR2=C2(K,M)
25710        SD=STOR1*STOR1+STOR2*STOR2
25720        DO 107 J=1,MY
25730        STO1=C1(K,J)
25740        STO2=C2(K,J)
25750        C1(K,J)=C1(M,J)
25760        C2(K,J)=C2(M,J)
25770        C1(M,J)=(STO1*STOR1+STO2*STOR2)/SD
25780 107    C2(M,J)=(STO2*STOR1-STO1*STOR2)/SD
25790        C11=C1(M,M)
25800        C1(M,M)=(C1(M,M)*STOR1+C2(M,M)*STOR2)/SD
25810        C2(M,M)=(C2(M,M)*STOR1-C11*STOR2)/SD
25820        DO 118 I=1,MY
25830        IF(I-M) 112,118,112
25840 112    ST1=C1(I,M)
25850        ST2=C2(I,M)
25860        C1(I,M)=0
25870        C2(I,M)=0
25880        DO 110 J=1,MY
25890        C1(I,J)=C1(I,J)-C1(M,J)*ST1+C2(M,J)*ST2
25900        C2(I,J)=C2(I,J)-C2(M,J)*ST1-C1(M,J)*ST2
25910 110    CONTINUE
25920 118    CONTINUE
25930        DO 109 J=1,MY
25940 114    LRJ=LR(J)
25950        DO 113 I=1,MY
25960        T1=C1(I,LRJ)
25970        T2=C2(I,LRJ)
25980        C1(I,LRJ)=C1(I,J)
25990        C2(I,LRJ)=C2(I,J)
26000        C1(I,J)=T1
```

```
113   C2(I,J)=T2
      L=LR(J)
      LR(J)=LR(LRJ)
      LR(LRJ)=L
      IF(J-LR(J)) 114,109,114
109   CONTINUE
300   IPIN=0
*
*   ***CALCULATE THE CURRENT AC,BC
*
314   DO 65 I=1,MM
      AC(I)=0
      BC(I)=0
      DO 51 J=1,NN
      ABV=AV(J)**2+BV(J)**2
      IF(ABV.LT.1.E-8) GO TO 51
      K=IF(J)
      AC(I)=AC(I)+C1(I,K)*AV(J)-C2(I,K)*BV(J)
      BC(I)=BC(I)+C1(I,K)*BV(J)+C2(I,K)*AV(J)
51    CONTINUE
65    CONTINUE
*
*   ***CALCULATE THE INPUT POWER PIN
*
      PIN=0.
      MX=1
      AA=1
      DO 246 I=1,NN
      K=IF(I)
      IF(NB.NE.0) GO TO 244
      IF(I.EQ.1) GO TO 242
      II=I-1
      MX=MX+NS(II)
242   IF(K.EQ.MX.AND.ABS(Z(I)).LT.1.E-5.AND.IG.EQ.1) AA=.5
244   PIN=PIN+AA*(AC(K)*AV(I)+BC(K)*BV(I))
246   CONTINUE
      IF(IRG.EQ.0) APIN=PIN
      IF(IRG.EQ.0) GO TO 316
      IPIN=IPIN+1
      IF(IPIN.GT.1) GO TO 316
      PSQ=SQRT(APIN/PIN)
      DO 312 I=1,NN
      AV(I)=AV(I)*PSQ
312   BV(I)=BV(I)*PSQ
      IF(IPIN.EQ.1) GO TO 314
316   IF(IC.EQ.0) GO TO 70
*
*   ***CALCULATE AND WRITE OUT THE COUPLING COEFFICIENT
*
*      WRITE(2,63)
      WRITE(6,63)
63    FORMAT(//' COUPLING COEFFICIENT')
```

```
26530*        WRITE(2,120)
26540        WRITE(6,120)
26550 120    FORMAT(/' ANTENNA NO.       POWER RECEIVED (DB)')
26560        M=0
26570        DO 140 I=1,NB
26580        NX=1
26590        IF(IT(I).EQ.197) NX=26
26600        IF(IT(I).EQ.1097) NX=5
26610        PRE=0.
26620        DO 130 K=1,NX
26630        M=M+1
26640        IF(NX.EQ.26.AND.K.GT.2) GO TO 130
26650        J=L1(M)
26660        PRE=PRE+AZ(M)*(AC(J)*AC(J)+BC(J)*BC(J))
26670 130    CONTINUE
26680        IF(I.EQ.IW.OR.IT(I).EQ.1000) GO TO 140
26690        AA=PRE/PIN
26700        ADB=-100.
26710        IF(AA.GT.1.E-10) ADB=10.*ALOG10(AA)
26720*        WRITE(2,125) I,ADB
26730        WRITE(6,125) I,ADB
26740 125    FORMAT(I6,13X,F10.2)
26750 140    CONTINUE
26760 70     IF(IFP.EQ.0) GO TO 82
26770*
26780*    ***CALCULATE  AND WRITE OUT THE VERTICAL RADIATION PATTERN
26790*
26800        WRITE(6,71)
26810*        WRITE(2,71)
26820 71     FORMAT(//' VERTICAL PATTERN')
26830*        WRITE(2,73) APHI
26840       WRITE(2) 4,1,APHI
26850        WRITE(6,73) APHI
26860 73     FORMAT(/' PHI=',F7.1)
26870 74     FORMAT(/' THETA     NMAG      NMAG(DB)')
26880        AIT=181.
26890        IF(IG.EQ.1) AIT=90.
26900        AMAX=0.
26910        ATH=0.
26920        IY=1
26930        CALL PATT(APHI,ATH,IW,AIT,AINT,AMAX,APIN,ALT,IY,FA)
26940*        WRITE(2,74)
26950        WRITE(6,74)
26960        CALL NPAT(AMAX,AIT,AINT,NFP)
26970 82     IF(IFT.EQ.0) GO TO 83
26980*
26990*    ***CALCULATE AND WRITE OUT THE HORIZONTAL RADIATION PATTERN
27000*
27010        WRITE(6,201)
27020*        WRITE(2,201)
27030 201    FORMAT(//' HORIZONTAL PATTERN')
27040*        WRITE(2,81) ATHE
```

```
        WRITE(2)5,1,ATHE
        WRITE(6,81) ATHE
  81    FORMAT(/' THETA=',F7.1)
  85    FORMAT(/'  PHI      NMAG      NMAG(DB)')
        AMAX=C.
        IY=2
        AIP=360.
        APH=0.
        CALL PATT(APH,ATHE,IW,AIP,AINP,AMAX,APIN,ALT,IY,FA)
*        WRITE(2,85)
        WRITE(6,85)
        CALL NPAT(AMAX,AIP,AINP,NFP)
*
*   ***CALCULATE AND WRITE OUT THE PATTERN DISTRIBUTION
*
        IF(DPER.GT.1.E-3) CALL PATD(DPER,IDB,NFP)
  83    IF(IRG.EQ.0) GO TO 100
*
*   ***READ DATA IN SCREEN-FILE
*   ***CALCULATE AND WRITE OUT THE COMMUNICATION RANGE CONTOUR
*
*        WRITE(2,86)
        WRITE(6,86)
  86    FORMAT(//' COMMUNICATION RANGE CONTOUR')
*        WRITE(2,99) (ALT(I),I=1,NA)
        WRITE(6,99) (ALT(I),I=1,NA)
  87    FORMAT(/' PHI(DEG)',12X,'RANGE(NM)')
*        WRITE(2,87)
        WRITE(6,87)
  99    FORMAT(/'  ALT(FT)',6F8.1/(9X,6F8.1))
        IY=3
        CALL PATT(APHI,ATHE,IW,AIT,AINP,AMAX,APIN,ALT,IY,FA)
  100   CONTINUE
        STOP
        END
$       FORTRAN                                                      USAPAT
*
*   ***SUBROUTINE PATT - TO CALCULATE RADIATION PATTERN
*
        SUBROUTINE PATT(APHI,ATHE,IW,AIT,AINT,AMAX,PIN,ALT,IY,FA)
        COMMON X,Y,Z,X1,Y1,Z1,HO,NS,FP,AK,IG,NN,NB,AC,BC,IT,AZ,NA,RNG
       1 ,GREL,TRANEL,ANTEL,DTA,BTA,DBM,IFP,IFT,IRG
        DIMENSION X(200),Y(200).Z(200).HO(200).NS(200).AC(500).RNG(6).
       1 FA(40).X1(200),Y1(200),Z1(200).IT(20).AZ(361).BC(500).
       2 ALT(6).ATH(6).GX(3).GY(3).GZ(3)
        IMG=IG
        BK=AK*FP/3.E8
        R2=2.*3437.76*6080.
        R3=R2*.5*.3048*4./3.
        FX=1.E-7*AK*FP
        J1=0
  70    J1=J1+1
```

```
         IF(IFP.EQ.1.AND.IY.EQ.1) GO TO 31
         IF(IFT.EQ.1.AND.IY.EQ.2) GO TO 34
         IF(IRG.EQ.1) GO TO 38
*
*   ***VERTICAL PATTERN- CONSTANT PHI, VARY THETA
*
 31      IF(J1.NE.1) ATHE=ATHE+AINT
         GO TO 36
*
*   ***HORIZONTAL PATTERN-CONSTANT THETA,VARY PHI
*
 34      IF(J1.NE.1) APHI=APHI+AINT
*
*   ***FIND PHI, THETA IN RADIANS(PHI,THE)
*
 36      PHI=APHI/57.29578
         THE=ATHE/57.29578
         GO TO 39
*
*   ***COMMUNICATION RANGE-READ LINE OF SIGHT DATA
*
 38      READ(3'J1) ZZ,ELANG,DIST,RNG
         IF(ZZ.GT.9998./57.29578) GO TO 100
         AX=DTA*SIN(BTA)
         AY=DTA*COS(BTA)
         SX=DIST*SIN(ZZ)*.3048
         SY=DIST*COS(ZZ)*.3048
         ATS=SQRT((SX-AX)**2+(SY-AY)**2)
*
*   *** FIND PHI,THETA IN RADIANS
*
         PHI=ATAN2((SX-AX),(SY-AY))
         IF(PHI.LT.0.) PHI=PHI+2.*3.1415927
         HSCRN=DIST*.3048*SIN(ELANG+DIST/R2)/COS(ELANG+DIST/R2)+TRANEL
         TTH=ATAN2(HSCRN-ANTEL-FA(IW),ATS)-ATS*.75/(R2*.3048)
         THE=3.141593*.5-TTH
 39      M=0
*
*   ***CALCULATE COORDINATES
*
         CP=COS(PHI)
         CT=COS(THE)
         SP=-SIN(PHI)
         ST=SIN(THE)
         AE=0.
         BE=0.
         NX=NB
         IF(NB.EQ.0) NX=NN
         I=0
         DO 20 KS=1,NX
         NK=1
         MK=0
```

```
          IF(NB.EQ.0) GO TO 3
          IF(IT(KS).EQ.1097) NK=5
          IF(IT(KS).EQ.197) NK=26
          IF(NK.EQ.26.AND.KS.EQ.IW) MK=1
3         DO 20 KQ=1,NK
          MW=0
          I=I+1
          IF(NK.NE.26) GO TO 56
          IF(KQ-14) 50,50,54
50        IF(KQ.EQ.2) GO TO 56
          MW=1
          GO TO 56
54        MW=3
56        IF(I-1) 10,10,45
45        IF(MK.EQ.0) GO TO 5
          IF(KQ.GT.3.AND.KQ.NE.15) GO TO 10
5         JJ=I-1
          M=M+NS(JJ)
10        NSI=NS(I)
          NG=1
          IF(IG) 12,14,12
12        IF(Z(I)) 14,13,14
13        NG=0
14        AN=NSI+NG
          IF(KQ.GT.14) NG=0
          IF(NK.EQ.26.AND.KQ.NE.2) AN=AN-1.
          DO 20 J=1,NSI
          A1=J-2+NG
          N=M+J
          DO 20 L=1,3
          AX=L
          KK=1
          A=A1+AX*.5
          IF(L.NE.2) GO TO 32
          A2=1.
          IF(MW.EQ.1.AND.J.EQ.NSI) KK=2
          IF(MW.EQ.3.AND.J.EQ.1) KK=2
32        BB=KK
          DO 20 K=1,KK
          IF(J.NE.1.AND.L.EQ.1) GO TO 35
          T1=X1(I)-X(I)
          T2=Y1(I)-Y(I)
          T3=Z1(I)-Z(I)
          DK=HO(I)/AN*.5
          IF(MW.NE.1.OR.J.NE.NSI) GO TO 60
          IF(KK.NE.2.AND.L.NE.3) GO TO 60
58        T1=0.
          T2=0.
          T3=1.
          DK=.0075
          GO TO 65
60        IF(MW.NE.3.OR.J.NE.1) GO TO 65
```

```
28610          IF(KK.EQ.1.AND.L.NE.3) GO TO 58
28620 65       RO=SQRT(T1*T1+T2*T2+T3*T3)
28630          DA=DK*A*2./RO
28640          PX=X(I)+DA*T1
28650          PY=Y(I)+DA*T2
28660          PZ=Z(I)+DA*T3
28670          IF(MW.EQ.1.AND.J.EQ.NSI) PX=X1(I)
28680          IF(MW.EQ.1.AND.J.EQ.NSI) PY=Y1(I)
28690          IF(MW.EQ.1.AND.J.EQ.NSI) PZ=Z1(I)+(AX-2.)*DK
28700          SS=DK*.5*3.141593
28710          IF(L.NE.2) A2=.5
28720          DD=A2*DK/BB
28730          DT=(T1*CT*CP+T2*CT*SP)/RO
28740          DS=-T3*ST/RO
28750          PS=BK*(PX*CP+PY*SP)*ST
28760          P3=COS(PS)
28770          P4=SIN(PS)
28780          PQ=BK*CT*PZ
28790          P5=SIN(PQ)
28800          P6=COS(PQ)
28810 35       P1=P3*AC(N)-P4*BC(N)
28820          P2=P3*BC(N)+P4*AC(N)
28830          AGTD=0.
28840          BGTD=0.
28850          GX(L)=PX
28860          GY(L)=PY
28870          GZ(L)=PZ
28880*
28890*      ***FIND WHETHER IMAGE FIELDS EXIST (IMG=1), OR NOT (IMG=0)
28900*
28910          IF(L.NE.3.OR.K.NE.1) GO TO 24
28920*
28930*         CALCULATE COORDINATE GX,GY,GZ OF THESEGMENTS FOR GTD
28940*      ***  (FOR FUTURE USE)
28950*
28960          GX(1)=2.*GX(1)-GX(2)
28970          GY(1)=2.*GY(1)-GY(2)
28980          GZ(1)=2.*GZ(1)-GZ(2)
28990          GX(3)=2.*GX(3)-GX(2)
29000          GY(3)=2.*GY(3)-GY(2)
29010          GZ(3)=2.*GZ(3)-GZ(2)
29020*
29030*      ***GTD IS NOT CONSIDERED
29040*      ***FIND DIFFRACTION FIELDS BY GTD (AGTD,BGTD)
29050*      ***   (FOR FUTURE USE)
29060 24       CONTINUE
29070          AA=IMG
29080          IF(J-1) 17,15,17
29090 15       IF(Z(I)) 17,16,17
29100 16       AA=0.
29110 17       AP=1.+AA
29120          AM=1.-AA
```

```
      QS=P5*(DT*AP+DS*AM)
      QC=P6*(DT*AM+DS*AP)
*
*  ***CALCULATE RADIATION PATTERN E(AE,BE)
*
      AE=AE-(P1*QC-P2*QS)*DD+AGTD
      BE=BE-(P1*QS+P2*QC)*DD+BGTD
20    CONTINUE
      BB=AE*AE+BE*BE
      AZ(J1)=FX*SQRT(BB)
*
*  ***COMMUNICATION RANGE CONTOUR-IY=3,IRG=1
*
     IF(IRG.EQ.1.AND.IY.EQ.3) GO TO 72
     IF(AZ(J1).GT.AMAX) AMAX=AZ(J1)
     IF(IFP.EQ.1.AND.IY.EQ.1.AND.ATHE.LE.AIT+.001-AINT) GO TO 70
     IF(IFT.EQ.1.AND.IY.EQ.2.AND.APHI.LE.AIT+.001-AINT) GO TO 70
      GMAX=AMAX*AMAX/PIN/30.
      GDB=10.*ALOG10(GMAX)
*
*  ***WRITE RADIATION PATTERN AMAX,GMAX,GDB
*
*      WRITE(2,280) AMAX,GMAX,GDB
      WRITE(6,280) AMAX,GMAX,GDB
     WRITE(2)2,3,AMAX,GMAX,GDB
280   FORMAT(/' EMAX=',F8.3,' GAIN=',F7.3;' GAIN(DB)=',F6.2)
      GO TO 100
*
*  ***CALCULATE RANGES FROM THE ANTENNA ,RNG
*
72    RRR=AZ(J1)/(BK*SQRT(120.*DBM))
      DO 80 I=1,NA
      UU=RNG(I)*1852./R3
      VV=ALT(I)*.3048+GREL+R3
      WW=ANTEL+FA(IW)+R3
      RR=SQRT(VV**2+WW**2-2.*WW*VV*COS(UU))
      ALP=1.
      IF(RRR.GT.VV-WW) ALP=(WW**2+VV**2-RRR**2)/(2.*WW*VV)
      PPP=ARCOS(ALP)
      IF(RRR.LT.RR) RNG(I)=R3*PPP/1852.
      RG=1.E-5
      IF(RNG(I).GT.1.E-8) RG=RNG(I)*1852.
      QQ1=(ALT(I)*.3048+GREL-ANTEL-FA(IW))/RG
      QQQ=ATAN(QQ1)-RG/(R3*2.)
      IF(RR.LT.RRR) QQQ=TTH
      ATH(I)=QQQ*57.29578
80    CONTINUE
      APHI=FHI*57.29578
      ATHE=90.-THE*57.29578
*      WRITE(2,74) APHI,(RNG(I),I=1,NA)
      WRITE(6,74) APHI,(RNG(I),I=1,NA)
74    FORMAT(F7.2,1X,6F8.1/(6F8.1))
```

```
29650         GO TO 70
29660 100     RETURN
29670         END
29680$        FORTRAN                                                  USAFUN
29690*
29700*    ***SOUBROUTINE FUN - NUMERICAL INTEGRATION
29710*
29720         SUBROUTINE FUN(DD,DE,Y9,SUM,NDIM)
29730         DIMENSION Y9(13)
29740         SUM=0
29750         NDH=NDIM/2+1
29760         DO 10 I=2,NDIM
29770         HH=.5*DE
29780         IF(I.LE.NDH) HH=.5*DD
29790         SUM=SUM+HH*(Y9(I)+Y9(I-1))
29800 10      CONTINUE
29810         RETURN
29820         END
29830$        FORTRAN                                                  USAELE
29840*
29850*    ***FUNCTION ELE - ELIPTIC INTEGRAL OF SECOND KIND
29860*
29870         FUNCTION ELE(P)
29880         IF(P-1.) 3,5,4
29890 4       WRITE(6,100) P
29900         STOP
29910 100     FORMAT(10X,'ILLEGAL ARGUMENT FOR ELE. ARG=',E13.4)
29920 5       ELE=1.
29930         RETURN
29940 3       AM1=1.-P
29950         ELE=1.+AM1*(.4630151+.1077812*AM1-(.2452727+.0412496
29960       X *AM1)*ALOG(AM1))
29970         RETURN
29980         END
29990$        FORTRAN                                                  USAELK
30000*
30010*    ***FUNCTION ELK - ELIPTIC INTEGRAL OF FIRST KIND
30020*
30030         FUNCTION ELK(P)
30040         IF(P-1.) 3,4,4
30050 4       WRITE(6,100) P
30060         STOP
30070 100     FORMAT(10X,'ILLEGAL ARGUMENT FOR ELK. ARG=',E13.4)
30080 3       AM1=1.-P
30090         ELK=1.386294+AM1*(.1119723+.0725296*AM1)-(.5+AM1*
30100       X (.1213478+.0288729*AM1))*ALOG(AM1)
30110         RETURN
30120         END
30130$        FORTRAN                                                  USAPXY
30140*
30150*    ***SUBROUTINE PXYZ - TO CALCULATE POSITION (X,Y,Z), (X1,Y1,Z1)
30160*
```

```
30170         SUBROUTINE PXYZ
30180         COMMON X,Y,Z,X1,Y1,Z1,HO,NS,FP,AK,IG,NN,NB,AC,BC,IT,AZ,NA,RNG
30190      1 ,GREL,TRANEL,ANTEL,DTA,BTA,DBM,IFP,IFT,IRG
30200         DIMENSION HO(200),X(200),Y(200),Z(200),X1(200),Y1(200),
30210      1 Z1(200),NS(200),AC(500),BC(500),IT(20),AZ(361),RNG(6)
30220         NX=NN
30230         IF(NB) 2,2,1
30240*
30250*     ***ENTER ANTENNA TYPE FOR THE SIMPLE PROGRAM ANTTENA
30260*
30270 1       IF(NB.NE.0) READ(1,20) (IT(I),I=1,NB)
30280         NX=NB
30290         THO=150.
30300         PHO=0.
30310         THX=THO*.0174533
30320         C=COS(THX)
30330         SIT=SIN(THX)
30340 2       L=0
30350         DO 319 LS=1,NX
30360         NK=1
30370         IF(NB.EQ.0) GO TO 318
30380         IF(IT(LS).EQ.1097) NK=5
30390         IF(IT(LS).EQ.197) NK=26
30400 318     DO 315 LQ=1,NK
30410         L=L+1
30420         ALS=LQ
30430         IF(NB.EQ.0) GO TO 305
30440         IF(IT(LS).NE.197) GO TO 305
30450         IF(LQ-2) 305,305,303
30460 303     IF(LQ-14) 312,312,308
30470*
30480*     ***CALCULATE ANTENNA POSITION (X,Y,Z),(X1,Y1,Z1)FOR GENERAL
30490*     ***PROGRAM ANTENNA AND SUB-ANTENNA WIRE #1,2 OF
30500*     ***SIMPLE PROGRAM ANTENNA
30510*
30520 305     X1(L)=X(L)
30530         Y1(L)=Y(L)
30540         Z1(L)=Z(L)+HO(L)
30550         GO TO 315
30560*
30570*     ***CALCULATE (X1,Y1,Z1) FOR SUB-ANTENNA #15-26
30580*
30590 308     PH1=30.*(ALS-15.)+PHO
30600         PHX=PH1*.0174533
30610         X1(L)=X(L)+HO(L)*COS(PHX)
30620         Y1(L)=Y(L)+HO(L)*SIN(PHX)
30630         Z1(L)=Z(L)
30640         GO TO 315
30650*
30660*     ***CALCULATE (X1,Y1,Z1),(X,Y,Z) FOR SUB-ANTENNA#3-14
30670*
30680 312     PH1=30.*(ALS-3.)+PHO
```

```
30690          PHX=PH1*.0174533
30700          X1(L)=X(L)
30710          Y1(L)=Y(L)
30720          Z1(L)=Z(L)
30730          X(L)=X1(L)+HO(L)*SIT*COS(PHX)
30740          Y(L)=Y1(L)+HO(L)*SIT*SIN(PHX)
30750          Z(L)=Z1(L)+HO(L)*C
30760 315      CONTINUE
30770 319      CONTINUE
30780 20       FORMAT(I5,15I4)
30790          RETURN
30800          END
30810$         FORTRAN                                                        USAQZP
30820*
30830*     ***SUBROUTINE QZP - TO CALCULATE SIZE OF MATRIX USABLE AGAIN
30840*
30850          SUBROUTINE QZP(K1,L8,II,L9,MX,MZ,KX,NP)
30860          COMMON X,Y,Z,X1,Y1,Z1,HO,NS,FP,AK,IG,NN,NB,AC,BC,IT,AZ,NA,RNG
30870        1 ,GREL,TRANEL,ANTEL,DTA,BTA,DBM,IFP,IFT,IRG
30880          DIMENSION X(200),Y(200),Z(200),HO(200),NS(200),AC(500),
30890        1 BC(500),X1(200),Y1(200),Z1(200),IT(20),NP(20),AZ(361),
30900        2 RNG(6);J=0
30910          DO 20 I=KX,K1
30920          J=J+NP(I)
30930 20       CONTINUE
30940          MX=II-J
30950          MZ=0
30960          M1=L9+1
30970          M2=L9+L8
30980          DO 50 J=M1,M2
30990          MZ=MZ+NS(J)
31000 50       CONTINUE
31010          RETURN
31020          END
31030$         FORTRAN                                                        USANPA
31040*
31050*     ***SUBROUTINE NPAT - TO CALCULATE NORMALIZED RADIATION PATTERN
31060*
31070          SUBROUTINE NPAT(AMAX,AIT,AINT,I)
31080          COMMON X,Y,Z,X1,Y1,Z1,HO,NS,FP,AK,IG,NN,NB,AC,BC,IT,AZ,NA,RNG
31090        1 ,GREL,TRANEL,ANTEL,DTA,BTA,DBM,IFP,IFT,IRG
31100          DIMENSION X(200),Y(200),Z(200),X1(200),Y1(200),Z1(200),
31110        1 HO(200),NS(200),AC(500),BC(500),AZ(361),IT(20),RNG(6)
31120*
31130*     ***INCREASE ATH FROM 0 TO AIT, THE ANGLE RANGE OF RADIATION
31140*     ***PATTERN CONCERNED
31150*
31160          I=0
31170          ATH=-AINT
31180 10       ATH=ATH+AINT
31190          I=I+1
31200*
```

```
31210*     ***NORMALIZE THE RADIATION PATTERN
31220*
31230         FM=AZ(I)/AMAX
31240         IF(FM.LT.1.E-5) GO TO 252
31250         FDB=20.*ALOG10(FM)
31260         GO TO 254
31270 252     FDB=-1000.
31280*
31290*     ***PRINT OUT THE NORMALIZED RADIATION PATTERN
31300*
31310 254     WRITE(6,88) ATH,FM,FDB
31320          WRITE(2)3,3,ATH,FM,FDB
31330*254    WRITE(2,88)ATH,FM,FDB
31340         IF(ATH.LE.AIT+.001-AINT) GO TO 10
31350 88      FORMAT(F5.0,F10.4,F10.2)
31360         RETURN
31370         END
31380$        FORTRAN                                                    USAPAD
31390*
31400*     ***SUBROUTINE PATD - TO CALCULATE PATTERN DISTRIBUTION
31410*
31420         SUBROUTINE PATD(DPER,IDB,NFP)
31430         COMMON X,Y,Z,X1,Y1,Z1,HC,NS,FP,AK,IG,NN,NB,AC,BC,IT,AZ,NA,RNG
31440       1 ,GREL,TRANEL,ANTEL,DTA,BTA,DBM,IFP.IFT.IRG
31450         DIMENSION X(200),Y(200),Z(200),HO(200),NS(200),AC(500),
31460       & RNG(6),X1(200),Y1(200),Z1(200),IT(20),AZ(361),BC(500)
31470         NX=NFP-1
31480*
31490*     ***RE-ORDER AZ IN INCREASING MAGNITUDE
31500*
31510         DO 20 I=1,NX
31520         BMAX=0.
31530         K=I
31540         DO 10 J=I,NX
31550         IF(AZ(J).LE.BMAX) GO TO 10
31560         BMAX=AZ(J)
31570         K=J
31580 10      CONTINUE
31590         AZ(K)=AZ(I)
31600         AZ(I)=BMAX
31610 20      CONTINUE
31620         NDIV=100./DPER
31630         ANDIV=NX-1
31640         ND=NDIV+2
31650         AND=ND-2
31660*        WRITE(2,25)
31670         WRITE(6,25)
31680 25      FORMAT(///' PATTERN DISTRIBUTION'//)
31690*
31700*     ***IDB=-1 :X,MAG
31710*     ***IDB=0 :X,MAG, DB
31720*     ***IDB=1 :X,DB
```

```
31730*
31740        IF(IDB.EQ.-1) WRITE(6,55)
31750*        IF(IDB.EQ.-1) WRITE(2,55)
31760        IF(IDB.EQ.0) WRITE(6,65)
31770*        IF(IDB.EQ.0) WRITE(2,65)
31780        IF(IDB.EQ.1) WRITE(6,75)
31790*        IF(IDB.EQ.1) WRITE(2,75)
31800 55     FORMAT('    %         MAG')
31810 65     FORMAT('    %         MAG        (DB)')
31820 75     FORMAT('    %         (DB)')
31830*
31840*    ***CALCULATE PERCENTAGE P, MAGNITUDE G , DB H
31850*
31860        PP=DPER*AND
31870        IF(PP.GT.99.99) ND=ND-1
31880        DO 40 J=1,ND
31890        AJ=J-1
31900        A=DPER*ANDIV*.01*AJ+1.
31910        NP=A
31920        NQ=NP+1
31930        B=NQ
31940        P=DPER*AJ
31950        IF(P.GT.100.01) P=100.
31960        F=(AZ(NP)-AZ(NQ))*(B-A)+AZ(NQ)
31970        G=F/AZ(1)
31980        IF(P.GT.100.01) G=0.
31990        IF(J.EQ.1) G=1.
32000        IF(G.GE.1.E-5) GO TO 35
32010        H=-100.
32020        GO TO 38
32030 35     H=20.*ALOG10(G)
32040 38     IF(IDB.EQ.-1) WRITE(6,50) P,G
32050*        IF(IDB.EQ.-1) WRITE(2,50) P,G
32060        IF(IDB.EQ.0) WRITE(6,60) P,G,H
32070*        IF(IDB.EQ.0) WRITE(2,60) P,G,H
32080        IF(IDB.EQ.1) WRITE(6,70) P,H
32090*        IF(IDB.EQ.1) WRITE(2,70) P,H
32100 40     CONTINUE
32110 50     FORMAT(F6.1,F9.3)
32120 60     FORMAT(F6.1,F9.3,F10.2)
32130 70     FORMAT(F6.1,F10.2)
32140        RETURN
32150        END
32160$       ENDEDIT
```

APPENDIX II

Program cross-reference

FILEDIT NAME	DECK NAME	SYMDEF
LNKMNA	LINKMN	LINKMN
LNKM1A	LINKM1	LINKM1
LNKM2A	LINKM2	LINKM2
LNKM3A	LINKM3	LINKM3
DSTRTP	.....	originally ..... changed to DSTPLT
DSTRP1	PPLOT	PPLOT
Y.LABL	LABEL	LABEL
YSFUTL	YSFUTL	CALLTS and others
PACK	PACK	UNPACK and others
DIMXXX	DIMXXX	......
USAMAI	MAIN	MAIN
USAPAT	PATT	PATT
USAFUN	FUN	FUN
USAELE	ELE	ELE
USAELK	ELK	ELK
USAPXY	PXYZ	PXYZ
USAQZP	QZP	QZP
USANPA	NPAT	NPAT
USAPAD	PATD	PATD

APPENDIX III - GLOSSARY

Adjustable dimensions

A facility of the FORTRAN language which allows the size of an array to be defined in a given routine and then passed to subordinate routines via the argument list, eliminating the need to define the size of the array again in each of the subordinate routines. Thus the dimensions of an array in a subroutine may be varied without actually modifying the subroutine itself.

DD02, p. 2-16 thru 2-17

ASCII

A particular standardized system which assigns certain binary bit patterns as representations for a set of graphic and control characters; these bit patterns are used internal to data processing equipment when the strings of symbols they represent are to be stored, transmitted, or manipulated. ASCII is one of the two native character sets of the Honeywell 600/6000 series computers (see also BCD) and may be accepted by programs written in the FORTRAN language on these machines. There are, however, certain pieces of software associated with FORTRAN which exist in two mutually exclusive forms, each designed to be compatible with a particular character set. As a result, it is sometimes necessary to specify which of the two character sets is to be used; it may not be possible to use both character sets within a given program. There are, in addition, some routines designed to automatically convert characters of one set to those of another set; it may be possible or necessary to specify the use of these routines for certain applications.

DD02, p. 3-28 thru 3-29, A-1 thru A-3

DD07, sections I and IX

DD20, sections II and III

BATCH

One of the two major environments in which a program may execute on these computers (see also timesharing). A batch job generally does not interact directly with a user at a remote terminal; it is not monitored or serviced by the TSS Executive; and it is not subject to certain constraints imposed upon jobs which do execute under the TSS Executive. Since batch programs are monitored and serviced by GCOS rather than by the TSS Executive, some functions which can be performed in the timesharing environment cannot be performed in the batch environment, and vice versa; in addition, the way in which certain functions are accomplished varies depending upon the environment. As in the case of character set differences (see ASCII above), there exist two mutually exclusive versions of certain pieces of software, each designed to work in a particular environment. This may again necessitate specification of the environment in which the program will operate; it may not operate in the same fashion in a different environment.

DD07, sections I and IX

DD20, sections II and III

BCD

A system by which a set of graphic symbols may be represented as a string of binary bits. This is one of the two representation systems native to the Honeywell 600/6000 series computers (see also ASCII).

DD02, p. 3-28 thru 3-29, A-1 thru A-3

DD07, sections I and IX

DD20, sections II and III

Binary file

A file which has been created by a binary write statement or other process which does not employ format conversion. Such files will often contain information in its internal binary representation, and do not usually contain character information such as headings, textual messages, or spaces between data items. Binary files are used primarily where data will be used as input to another program and will not ordinarily be examined by a human being. The information is in a form which is readily machine-readable and which does not require conversion from its external format to the internal binary format of the machine; it is not readily human-readible (see also formatted file).

DD02, p. 5-14

DD20, p. 2-4 thru 2-8

Binary number

Numbers are generally converted from their external, human-readable form (a string of characters) into a binary form which is more convenient for internal arithmetic manipulations. The format of this internal representation is detailed in the reference.

BN86, p. 1-20 thru 1-28

Buffer

In order to improve the efficiency of I/O operations, data transmitted to or from an I/O device is accumulated in an arbitrarily-chosen area of memory referred to as a buffer. When data is being transferred to a file (a WRITE operation), data is accumulated in the buffer until the buffer becomes full; the data is then written to the I/O device (called "flushing" the buffer) and the process repeats. Because of this buffering process, it is sometimes necessary to take explicit action to "flush" a buffer or "close" a file, a process which includes flushing the buffer and marking the end of the data just written on the I/O device. These flushing and closing operations transfer the last portion of data from the buffer to the I/O device, so that the information of the I/O device is complete and up to date.

DD07, section VI

DD20, p. 2-37 thru 2-38, 2-44 thru 2-50

CALLSS

CALLSS is a mnemonic for a function performed by the TSS Executive. The mnemonic is used in conjunction with the DRL machine operation, which causes a transfer of control to the TSS Executive, which then preserves the state of the program which executed the DRL CALLSS and invokes the specified timesharing subsystem on the program's behalf. Upon termination of the subsystem, the TSS Executive reinstates the suspended program.

DD17, p. 3-9

CARDIN

A timesharing subsystem which facilitates spawning of batch jobs from timesharing.

DD21, section II

DOMDECK

A compressed source deck. The format is basically that of the corresponding source deck, but with most of the blanks removed. Control information is added to allow the deleted blanks to be reinserted; identification and error detection data is also added.

BN86, p. 2-5 thru 2-6

Command Loader

A timesharing subsystem capable of restoring and placing in execution properly prepared programs in

H* format. The command loader is invoked whenever a system-selection or build-mode command is not recognized as being valid by the TSS Executive.

DD17, p. 5-1 thru 5-3

COMMON

Blank COMMON will normally be loaded just above the batch slave prefix, starting at location 100 octal. If, as in the case of a job which is to execute in timesharing, blank COMMON must be relocated, several methods are available. These are discussed in the loader manual. Labeled COMMON, on the other hand, is loaded in the same area as the program instructions. It is not usually necessary to take extraordinary action to specify the size or location of labeled COMMON (but the loader has facilities for this should it be required).

DD02, p. 4-13 thru 4-14

DD10, sections II thru IV

Control Card

See JCL

Deck Name

The name punched in columns 73 thru 80 of a COMDECK or an OBJECT DECK. This name comes from the LBL pseudo-operation of a GMAP program, or from the initial COMMENT card of a FORTRAN program or from its initial SYMDEF, in the absence of an initial SYMDEF, in the absence of an initial COMMENT. This name appears at the top of each page of a FORTRAN compilation listing following the word LABEL, and appears on the $ OBJECT card of the object deck resulting from the compilation or assembly. This $ OBJECT card may be printed out in subsequent activities which use the object deck.

BN86, p. 6-12

DD02, p. 2-4

DRL — (Pronounced DERAIL.) A machine instruction which causes a program interruption. When this instruction is executed by a timesharing program, the TSS Executive gains control of the CPU and interprets the interruption as a request for service. The exact nature of the service is specified by information accompanying the DRL instruction.

BN86, p. 1-15 thru 1-20, 4-118

DD17, p. 3-6 thru 3-54

DRLSAV — A particular function associated with the DRL instruction. Saves a program on a disk file in H* format.

DD17, p. 3-23 thru 3-24

Entry point — A point at which execution of instructions within a particular routine is to begin. A given routine may have one or more entry points, usually defined by a SYMDEF. When a program is loaded by GELOAD, CALL statements are linked with the corresponding entry points, and a single entry point for the program as a whole is determined. This will usually be the beginning of the main program, which has an implied entry name of six dots (......). When an H* file is created, the entry point for the element that is saved on the file is written to the file, so that execution can be initiated at the proper location when the element is restored.

BN86, p. 6-27 thru 6-27, 6-63 thru 6-67

DD02, p. 4-30, 4-37 thru 4-39, 4-60 thru 4-61

DD10, p. 4-1

DD31, p. 2-39

FDUMP A timesharing subsystem which allows the user to examine or modify a disk file in its entirety, regardless of the content, format, or characteristics of the file. The contents are displayed in octal.

DD21, section IV

File Control Block A group of words located in an arbitrary section of the user's memory space, used by GFRC to record the characteristics and status of a file on an I/O device. There is one file control block for each file used by a given program.

DD07, section IV

DD10, section XI

DD20, p. 2-37 thru 2-38, 2-44 thru 2-50

FILEDIT — A batch mode system program, capable of invoking various compilers, for creating, maintaining, listing and retrieving source and object program libraries.

BJ71, all sections

FILFDIT Name — The name which appears in columns 73 thru 80 of the $ language card ($ GMAP, $ FORTRAN, etc.) preceding each source program deck. This is the name used by the source file editor (FILEDIT) to access source programs (the object file editor portion of FILEDIT uses the DECK NAME from the $ OBJECT card). Note that the FILEDIT NAME, DECK NAME, and SYMDEF of a program are not necessarily the same; the FILEDIT NAME is used throughout this document to refer to the individual programs of the DISTORT system.

BJ71, sections VII thru XIII

Formatted file — A file which is subjected to format conversion, in contrast to a BINARY FILE. A formatted file contains information in the form of a string of characters; before this information can be used by the computer, it must undergo format conversion, in which numeric data, represented on the file as a string of numeric characters, decimal points, and exponent designators, is converted to an equivalent internal binary representation (see BINARY NUMBER).

The opposite conversion must be performed when the data is written to the file, along with the addition of information to control the carriage of the line printer or remote terminal printer.

DD02, p. 5-14

DD20, p. 2-12 thru 2-16, 2-39 thru 2-40

FORTRAN Library Routine

Any of the numerous subroutines, called either explicitly or implicitly, which perform various services for the FORTRAN program. Many of these routines in turn make use of library routines which make up GFRC, or they may, directly or indirectly, utilize the services of the operating system (GCOS and/or the TSS Executive).

DD02, p. 6-11 thru 6-49

DD20, all sections

GCOS

An operating system which monitors the operation of, and provides services to, all jobs operating in the batch environment. In the case of the RADC machine, GCOS is subordinate to another operating system called MULTICS. MULTICS monitors and services GCOS, much as GCOS monitors and services batch programs, and in so doing provides better security measures than those afforded by GCOS alone. MULTICS is not used on the AFCS H6060.

DD19, all sections

GELOAD

The main loader for Honeywell 600/6000 series machines. It is invoked in batch by the $ EXECUTE card; its input consists of object decks and control cards which modify the loading process or pass information to the loader; its primary output is a program, linked and relocated, ready for execution. GELOAD also produces a memory map to indicate where various programs have been loaded in core; GELOAD may be used to create H* files.

DD10, all sections

General Loader

See GELOAD.

GERSTR

A particular function associated with the MME instruction. Causes a program saved on a file in H* format to be restored (placed back in core).

DD19, p. 4-52

GESAVE

A particular function associated with the MME instruction. Saves a program on a disk file in H* format.

DD19, p. 4-53

GFRC

A set of routines which interfaces with the operating system (GCOS or TSS Executive) to provide basic services to a program with respect to I/O.

DD07, all sections

H* file

A random disk file upon which a program has been written (saved) in contemplation of placing that program in core and executing it at some later time. A program which resides on an H* file has already been loaded, relocated, and linked by GELOAD; thus the program appears on the H* file as an exact image of the program as it will appear in core. The file contains only the minimum amount of control information necessary to restore the program (place it in core) and initiate its execution. An H* file is created by a MME GESAVE or DRL DRLSAV; GELOAD can be used to accomplish the MME GESAVE. The file upon which the links of a linked or overlayed program are saved is an H* file.

BS18, Appendix B, *Random Storage Format*

DD10, section VIII

HIGHLOAD — A mode of operation wherein GELOAD places the instructions and data in the highest address locations of the memory segment specified by the $ LIMITS control card. Any unused space will then be at the lower address locations. This is the normal mode of operation for GELOAD.

DD10, section II

JCL — Job control language. This is a term borrowed from IBM, and refers to the card images which define the attributes of a batch job to the operating system and to any other system programs which require this information. On the Honeywell 600/6000 series computers, these card images are defined by the presence of the character $ in the first column; blanks in the second thru seventh; a verb beginning in column 8; and any applicable options beginning in column 16.

DD31, all sections

Linking loader

See GELOAD. So called because separate subroutines are linked together during the loading process, wherever a correspondence exists between the name of an entry point and the name used in a function or subroutine call.

DD10, all sections

Loader

See GELOAD

LOWLOAD

A mode of operation wherein GELOAD places the instructions and data in the lower address locations of the memory segment specified by the $ LIMITS card. See also HIGHLOAD.

DD10, section II

Master Mode

A mode of operation available to the Honeywell 600/6000 series CPU wherein a program can exercise maximum control over the system hardware, accessing any portion of memory physically present and operating any I/O device. This mode of operation is used only by the operating systems and a restricted set of system programs; user programs operate in SLAVE MODE, wherein the segment of memory which can be accessed is limited, and certain machine instructions relating to I/O and system control are rendered inoperable. This protects the operating system and other user's programs and data from inadvertent or malicious misuse or destruction.

BN86, p. 1-5

DD19, p. 1-4 thru 1-5, section IV

Memory layout

Reference shows the physical position of various items in the user's memory space.

DD10, section II

DD19, section V

MME

A machine instruction which causes a program interruption. When this instruction is executed by a batch program, GCOS gains control of the CPU and interprets the interruption as a request for service. The exact nature of the service is specified by the information accompanying the MME instruction.

BN86, p. 1-15 thru 1-20, 4-117

DD19, section IV

Object deck

A card deck, or image of that deck on a disk or tape file, which is the primary output of a compiler or assembler. Consists of machine language instructions and data which make up a program, along with identification and error detection information, and control information which allows GELOAD to properly load, relocate, link, and initiate execution of the program.

BN86, p. 2-4 thru 2-7

DD10, sections II and III

PSEUDO — A particular function associated with the DRL instruction. Causes data to be copied from the user's memory into the keyboard input buffer maintained by the TSS Executive, thereby simulating input from the keyboard of a remote terminal.

DD17, p. 3-20

Random library — A random disk file containing a collection of frequently used programs. A random library is created by the program RANLIB and is used as input to GELOAD. The random library contains essentially the same information as the collection of object decks from which it was created, but in a slightly different format and with the addition of a directory to aid in locating the required object deck. When a random library is supplied to GELOAD, the directory of the library file is searched at certain times to find programs which are required but which have not been loaded previously.

DA97, section III

DD10, section V

RANLIB
A program which generates random libraries.
DD42, section III

RESTOR
A particular function associated with the DRL instruction. Causes a program saved on a file in H* format to be restored (placed back in core).
DD17, p. 3-21 thru 3-22

Save name
The name used to save or restore a program on an H* file. This name must be specified whenever a MME GESAVE, MME GERSTR, DRL DRLSAV, or DRL RESTOR is used; some other means of restoring a program from an H* file, such as the COMMAND LOADER, may or may not require the use of a save name. The save name is used primarily to distinguish between program elements when more than one program element is saved on a single H* file.
DD02, p. 3-14, 3-22 thru 3-25, 6-43
DD10, section VII
DD17, p. 3-21 thru 3-24, 5-1 thru 5-5
DD19, p. 4-3 thru 4-4, 4-52 thru 4-53
DD20, p. 3-11 thru 3-12
DD31, p. 2-130

SELECTA
The $ SELECTA card image causes CARDIN to insert the contents of the file specified in the variable field into the JCL stream at that point, when a

batch job is spawned from timesharing via CARDIN. Note that this is somewhat different from the $ SELECT card.

DD21, p. 2-16

Slave prefix

An area in locations 0 thru 77 octal (0 thru 143 octal in timesharing) of the user's memory space which is reserved for the use of the operating system. It is used to communicate between the user program and the operating system, and portions of this area may be used as temporary storage by the operating system.

DD10, section II

DD17, p. 3-1 thru 3-3.2

DD19, section V

Slave program

A program which operates in slave mode as opposed to master mode. User programs operate in slave mode.

DD19, p. 1-4 thru 1-5, 4-1

SYMDEF

Symbol definition. A SYMDEF defines the name and location of an entry point or data area, so that this location can be referenced by a separately compiled or assembled program. SYMDEFs are generated primarily by compilers and assemblers, and are placed in object decks to enable GELOAD to perform the linking process.

BN86, p. 6-27 thru 6-28, 6-63 thru 6-67

DD10, p. 4-1

DD31, p. 2-39, 2-196 thru 2-197

System standard format — A particular format used by GFRC for recording data on an I/O device. Most sequential files are recorded in system standard format.

DD02, p. 3-29

DD07, p. 3-1 thru 3-3

Timesharing — A system which allows interactive use of a computer by several users at once; see also TSS Executive and BATCH.

DD22, all sections

TSS Executive — A sub-operating system which operates subordinate to GCOS, performing basically the same functions as GCOS, but tailored to a timesharing environment.

DD17, all sections

DD22, all sections

Unformatted file — See BINARY FILE.

APPENDIX IV

References

Publication		Date	Title
BJ71	Rev 0	Oct 71	Source and Object Library Editor
	Add A	Jun 72	
	Add B	Dec 72	
	Add C	May 73	
BN86	Rev 2	Mar 73	Macro Assembler Program (GMAP)
	Add A	May 73	
BS18	Rev 1	May 72	System Library Editor
	Add A	May 73	
DA97	Rev 0	Jan 72	Service routines
	Add A	Jun 72	
	Add B	Jan 73	
DD02	Rev 0	Jan 75	FORTRAN
DD07	Rev 0	Apr 74	File and Record Control
	Add A	Jan 75	
DD10	Rev 0	Mar 74	General Loader
DD17	Rev 0	Mar 74	TSS System Programmer's Reference Manual
	Add A	Jan 75	
DD19	Rev 0	Apr 74	General Comprehensive Operating Supervisor (GCOS)
	Add A	Mar 75	
DD20	Rev 0	May 75	FORTRAN Subroutine Libraries
DD21	Rev 0	Apr 74	TSS Terminal/Batch Interface
	Add A	Jan 75	

Publication		Date	Title
DD22	Rev 0	Apr 74	Time Sharing System General Information Manual
	Add A	Jan 75	
	Add B	Apr 75	
DD31	Rev 0	Mar 74	Control Cards Reference Manual
	Add A	Jan 75	

MISSION
of
Rome Air Development Center

RADC plans and conducts research, exploratory and advanced development programs in command, control, and communications (C^3) activities, and in the C^3 areas of information sciences and intelligence. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.