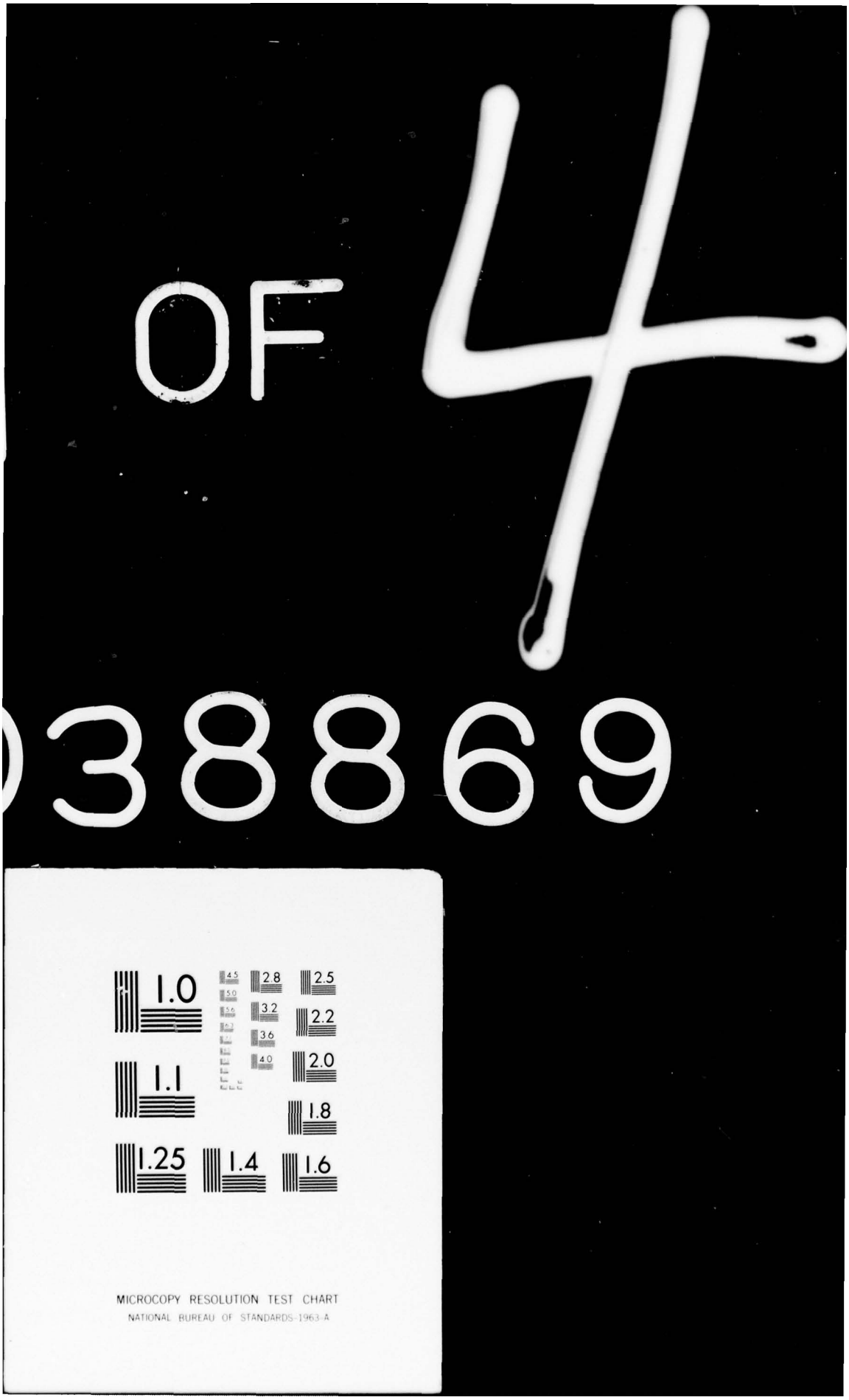
AD-A038 869 PARKE MATHEMATICAL LABS INC CARLISLE MASS F/G 20/14
ANALYSIS AND SYNTHESIS OF MODEL IONOGRAMS USING 3D RAY TRACKING--ETC(U)
FEB 77 B LANGWORTHY, T BARRETT, D BANDES F19628-76-C-0029
UNCLASSIFIED RADC-TR-77-60 NL

1 OF 4
AD
A038869




OF 4
038869
1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD A038869

RADC-TR-77-60
Technical Report
February 1977

Analysis and Synthesis of Model Ionograms Using 3D Ray Tracing Techniques

Parke Mathematical Laboratories, Inc.
One River Road, Carlisle, Mass. 01741

DDC
RECEIVED
APR 28 1976
C

Approved for public release;
distribution unlimited.

ROME AIR DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
GRIFFISS AIR FORCE BASE, NEW YORK 13441

DDC FILE COPY

AIR FORCE (1) MARCH 31, 1977--231

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.

APPROVED:

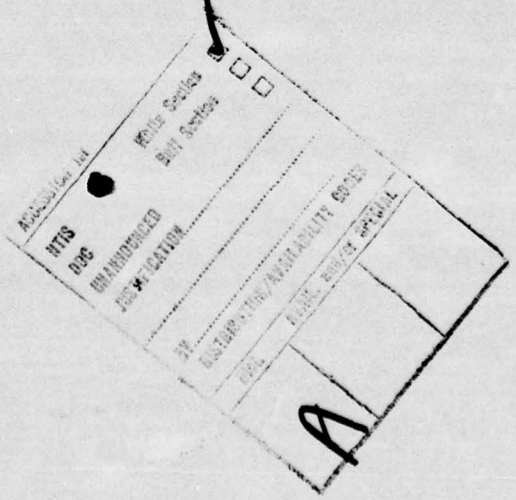
MING S. WONG
Contract Monitor
Electromagnetic Sciences Division

APPROVED:

TERENCE J. ELKINS
Chief, Ionospheric Radio
Physics Branch
Electromagnetic Sciences Division

APPROVED:

ALLAN C. SCHELL
Acting Chief
Electromagnetic Sciences Division



John P. Huss

Do not return this copy. Retain or destroy.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: RADC-TR-77-60	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): Analysis and Synthesis of Model Ionograms Using 3D Ray Tracing Techniques.	5. TYPE OF REPORT & PERIOD COVERED: Final Report. Aug 75-Oct 76, 8/75 - 10/76
	6. PERFORMING ORG. REPORT NUMBER: Final Report - 0029
7. AUTHOR(s): Barbara Langworthy, Theodore Barrett, Dean Bandes, Lorenzo Calabi	8. CONTRACT OR GRANT NUMBER(s): F19628-76-C-0029 new
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Parke Mathematical Laboratories, Inc. One River Road, Carlisle, Mass. 01741	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 61102F 56311601 17 16
11. CONTROLLING OFFICE NAME AND ADDRESS: Deputy for Electronic Technology (RADC/ETEI) Hanscom AFB, Massachusetts 01731 Contract Monitor: Ming Wong	12. REPORT DATE: February 1977
	13. NUMBER OF PAGES: 357p.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release: distribution unlimited

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

18. SUPPLEMENTARY NOTES

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

HF RAY TRACING
HF IONOGRAM SYNTHESIS
HF BACKSCATTER SIMULATION
MODEL IONOSPHERE ANALYSIS

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

A system of digital computer programs for use with the Jones-Stephenson ray trace program is described. Using these programs it is possible to: (1)

- produce complex ionospheres with simple input techniques; (2)
- trace rays through very large ionospheres by the use of disk storage and word packing techniques; (3) → (cont on p 1473 B)

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

276450

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(cont fr p. 173A)

- calculate power loss due to absorption, spreading, backscatter and antenna patterns; (4)
- produce leading edge ionograms, a dominant feature of full backscatter ionograms; (5)
- produce probable model ionosphere parameters from information obtained from the leading edge ionograms; and (6)
- produce many forms of ray trace information displays including various synthesized backscatter ionograms.

EVALUATION

F19628-76-C-0029 - Parke Mathematical Laboratories, Inc.

1. This report is the Final Report on the contract. It covers the work performed on computer simulation of backscatter-radar ionograms during the fourteen-month period from 1 August 1975 to 30 September 1976. The objective of this work is to develop an integrated system of computer programs, to be used with the ionospheric ray-tracing program at RADC/ETEI, for simulating swept-frequency ionograms as observed in over-the-horizon backscatter radars operating in the polar region.

2. The above work is of value since it provides a method for representing complex ionospheric structure with simple input techniques, a capability of computing rays through very large ionospheres by the use of disk storage and word-packing techniques, and a procedure for generating backscatter ionograms. It will be used by the USAF for interpreting the swept-frequency ionograms obtained at backscatter radar sites.

Ming S. Wong

MING S. WONG
Contract Monitor
Ionospheric Radio Physics Branch
Electromagnetic Sciences Division

Foreword

This report was prepared by the following scientists and programmers who worked on contract F19628-76-0029.

Dean Bandes

Theodore B. Barrett

Lorenzo Calabi

Barbara Langworthy

The authors wish to thank the contract monitor, Ming Wong (RADC/ETEI), for his many helpful suggestions and guidance during this contract. In our electron density processing program, we have incorporated his scheme of representation as follows. The electron density varies with height as a sequence of sine-square segments, each with phase angles from 0 to 90, or from 90 to 180 degrees. The maximum and minimum electron density in each segment, as well as the heights of the maximum and minimum, all vary with colatitude in sine-square segments. Each of these segments has an amplitude and end points which vary with longitude in additive sine-square segments.

He also suggested that for simplified ray tracing, the vertical plane containing the propagation direction be appropriately subdivided into cells bounded by constant-height and constant-distance contours. Within each cell, electron density varies only with height and as a sequence of two quasi-parabolic functions, each of which has vanishing height derivative at either the bottom or top boundary of the cell.

PRECEDING PAGE BLANK-NOT FILMED

TABLE OF CONTENTS

Foreword	v
PART I - DESCRIPTION	1
Chapter I - Introduction	2
Chapter II - Ionospheric and Atmospheric Determination	7
Section 2.1 - Ionospheric Pre-processing Program	7
Section 2.1.1 - Output from S2PPR Without Word Packing	12
Section 2.1.2 - Word Packing and Unpacking	12
Section 2.2 - Ionospheric Display Programs	15
Section 2.3 - Collision Frequency Pre-processing Program	16
Chapter III - Ray Tracing	18
Section 3.1 - Hamiltonian Ray Tracing Improvements and Modifications	18
Section 3.1.1 - The Electron Density Subroutine	18
Section 3.1.2 - Conditions of Perpendicularity to the Magnetic Field Lines	19
Section 3.1.3 - Establishment of Data Files for Ionogram Computation	20
Section 3.1.4 - Absorption Calculations	21
Section 3.1.5 - Other Modifications to the Ray Tracing Program	25
Section 3.2 - Two-Dimensional Ray Tracing with Quasi-Parabolic Model	25
Section 3.2.1 - Formulae for Quasi-Parabolic Models	26
Section 3.2.2 - Fitting Quasi-Parabolic Layers Together	26
Section 3.2.3 - Preliminary Results	27
Chapter IV - Backscatter Synthesis	28
Section 4.0 - Introduction	28
Section 4.1 - General Discussion	28
Section 4.2 - The Backscatter Algorithm	31
Section 4.3 - Ray Processing	34
Section 4.4 - Minimum Group Path Backscatter Synthesis	35
Section 4.5 - Complete Ionogram Synthesis	37

PRECEDING PAGE BLANK-NOT FILMED

TABLE OF CONTENTS (cont'd)

Chapter V - Backscatter Analysis 45

References 46

Figures:

1.1 Sequence of Events in Producing Backscatter Ionograms 5

2.1 Vertical Plasma Frequency Profile at 18° Dipolar Colatitude 8

2.2 Vertical Plasma Frequency Profile at 25° Dipolar Colatitude 9

2.3 Plasma Frequency of Layers as a function of Accurate Geomagnetic Colatitude 10

2.4 Height of Layers as a Function of Accurate Geomagnetic Colatitude 11

4.1 Sample Leading Edge Ionogram 36

4.2 Coded Print Ionogram 40

4.3 Ionogram Sample 41

4.4 Ionogram Sample with Calibration and Normalization Curve 42

4.5 Information Flow Diagram for Ionogram Synthesis, etc. 44

Tables:

3.1 Losses due to Absorption in Decibels at 5 MHz., Azimuth angle = 30°, for VAR2 data. 23

3.2 Losses due to Absorption at 5 MHz., Azimuth angle = 31°, for VAR2 electron density data 24

PART II - USER'S GUIDES 49

PML 127 E14994 Revision 1 - Mass Storage Electron Density 51

PML 127 E14994 Revision 2 - Mass Storage Electron Density 63

PML 136 S2PPR Sine Square Pre-processing Program 67

PML 136 S2PPR Revision 1 - Sine Square Pre-processing Program 95

PML 138 S2PLT Plasma Frequency Contour Plotting Program 97

PML 139 COLPP Collision Frequency Pre-processing Program for the Ray Trace Program 115

PML 140 COLAF1 Collision Frequency Subroutine for Tables Produced by COLPP 131

TABLE OF CONTENTS (cont'd)

PML 141	COLPLT	Collision Frequency Profile Plotting Program	137
PML 142	TRNSGN	Antenna Gain for a Logarithmic Antenna	143
PML 143	RCVRGN	Antenna Gain for a Rhombic Antenna	153
PML 144	GPMT	Minimum Group Path Trace Selection and Plotting	181
PML 145	POWER	Power Calculations Along Minimum Group Path Trace	207
PML 146	S2POBL	Plasma Frequency Contour Plotting Program	223
PML 150	AURLAT	Northern Hemisphere Auroral Zone Accurate Geomagnetic Latitude Function	231
PML 155	MLRA	Multiple Linear Regression Analysis	233
PML 157	CRRSET	RAYSET (elements) to RAY (bundles)	243
PML 158	BIBSUB1	File Sort	259
PML 159	BIBSUB2	Sequential to Word Addressable Transformer	269
PML 160	TRANS	Translation of Character String Data	275
PML 161	DEN	Ray Density Calculations and Flux Tube Generation	281
PML 161	DEN1	Revision 1 - Ray Density Calculations and Flux Tube Generation	297
PML 162	FLUXPR	Flux Tube Report Generator	301
PML 163	BACSCA1	Backscattered Power Calculations	307
PML 164	PLTDEV	Azimuth Deviation Plot/Print	319
PML 165	PRINT3	Automatic Formatted Output	329
PML 168	CRT1	CRT (intensity) Display	337
PML 169	BIN	Binning of Power for Ionogram Displays	347
PML 170	PRINT1	Printer Display of a Function of Two Variables	355

PART I
DESCRIPTION

PRECEDING PAGE BLANK-NOT FILMED

Chapter I. INTRODUCTION

This final report both summarizes work which was accomplished under contract F19628-76-0029, and provides a condensed source of information about the use of the various software packages which were developed for various ray tracing applications. The first part of the report consists of 4 chapters which discuss and describe in a logical sequence various aspects of the "ray trace system" as it was developed under this contract. The second part of the report is a collection of program "user's guides". These guides, which are frequently referred to in Part I, provide a concise description of what the program or subroutine does including a brief algorithmic description of the subroutine. It also is a convenient source of information on how to use the particular subroutine.

First the objectives of the contract:

a) Develop an electron density preprocessing program which accepts a combination of sine-square profiles and hand drawn contours and which permits variations of density with colatitude and longitude.

b) Develop a program for transforming electron density data and profiles in geographic coordinates, earth-centered dipolar coordinates, or accurate geomagnetic coordinates.

c) Develop a technique for minimizing the number of rays which must be traced to acquire all significant propagation effects of a particular ionospheric model.

d) Automate the generation of swept-frequency traces of minimum group path from systematically varying electron density distributions and then determine trial electron density distributions given experimentally observed backscatter-radar ionograms.

e) Develop a technique for handling large electron density data arrays without exceeding reasonable core memory requirements for ray-trace execution.

f) Develop techniques for determining radio signal strength by computing ray density, ionospheric radiowave absorption, and backscattering by the earth's surface and geomagnetic field-aligned irregularities; and for suitably displaying this information.

g) Develop a technique for extracting and displaying key parameters in the ray-trace computations.

h) Provide a system of documentation for the above efforts.

i) Optimize the ray-trace program for finding minimum group path rays vs frequency for a specified pair of neighboring take-off azimuths.

j) Develop a method for packing and unpacking sample electron density values within the ray trace system in order to permit ray tracing to be done with less external mass storage.

These objectives fall naturally into 4 categories.

1) Generation of the model medium through which rays are to be traced (includes electron density, collision frequency, earth's magnetic field).
2) Ray tracing (including development of techniques for optimizing ray tracing according to specific objectives)
3) Information extraction (including "synthesis" of various systems displays)
4) "Inversion" of ray tracing i.e. given some of the extracted information (e.g. ionograms) develop a best fit model. For lack of a better term this is called analysis.

Looking at the list of objectives, items a, b, e and j fall into category 1; items c, g (partially), and i fall into category 2; items d (1st part), f and g (partially) fall into category 3; item d (2nd part) falls into category 4. Item h is essentially accomplished through the user's guides which constitute Part II of this report.

Categories 1, 2, and 3 combined achieve the aim of synthesizing backscatter ionograms. The general approach used to achieve this objective is illustrated in Figure 1.1. Numbers given in parentheses refer to the section numbers of detailed descriptions. The sequence of steps follows:

a. Graphs of the critical frequency and height of the valleys and peaks of the vertical profiles are presented as a

function of accurate geomagnetic colatitude and longitude.

b. These curves are then introduced into an electron density preprocessing program which finds plasma frequency as a function of height, dipolar colatitude, and dipolar longitude along specified grid points.

c. This information is then put on mass storage files to be used as input to the ray tracing program.

d. The ray tracing program is the 1974 version of the Jones and Stephenson program with modifications for the special application. The program produces an output tape similar in function to the "raysets" developed by Croft. This output tape is then used in a variety of plotting programs.

e. Leading edge ionograms are produced from this tape and points along the leading edge are then used to calculate more rays to study the power along the leading edge.

f. Power calculations are made at coarser intervals for all available rays in the rayset and an amplitude modulated ionogram is produced.

g. Other display programs available for use with the ray trace output tape include most of the display variations produced by Stephenson and Georges (1969). The most useful of these in dealing with our problem is the display of azimuth deviation versus elevation angle for fixed azimuth and frequency. All displays are available for both earth landing points and points where the ray is perpendicular to the magnetic field lines.

The following chapters reflect or contain information according to this classification of the objectives and work into four categories. It should be noted here that the objectives on the whole are very broad in scope, running as they do from ray trace optimization to the inversion or "analysis" problem, a very difficult problem indeed even when looked at from a fairly restrictive view point (e.g. minimum group path length vs. frequency data).

It also should be pointed out in the introduction that due to exigencies that arose during the contract, emphasis was shifted into

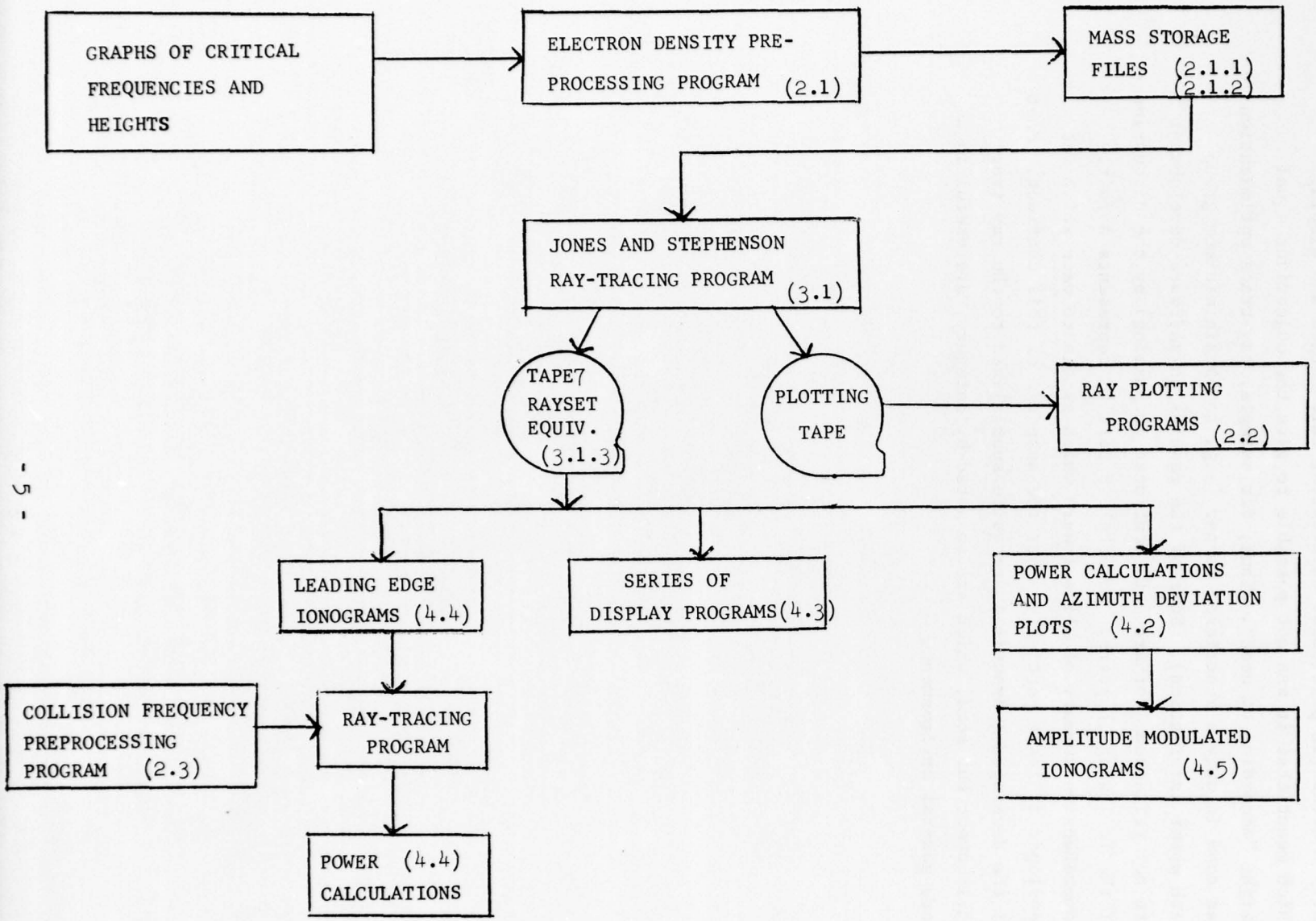


Figure 1.1 Sequence of Events in Producing Backscatter Ionograms

certain areas (in particular backscatter calculation and display) which meant that it was not possible to give the objectives equal weight "according to need". Thus, for example, ray trace optimization was done in only a restricted manner (e.g. to obtain minimum group path power information). Some of the results in software development are not reflected (or are only mentioned in passing) by the discussion of the following chapters. Nonetheless Part II represents a fairly comprehensive summary of all software which exists to date which was developed at least partially under the auspices of this contract. Most of the developed subroutines are quite specialized to the ray trace environment but some, which are so noted by category, are useful in a more general environment.

Chapter II IONOSPHERIC AND ATMOSPHERIC DETERMINATION

To achieve the purpose of predicting the electron density from given backscatter ionograms, it is necessary to construct backscatter ionograms from controlled, but complex ionospheres. Changes due to shifts in layer heights, thicknesses and widths must be identifiable in constructed backscatter ionograms to enable one to identify the best ionogram characteristics for measuring these shifts. For the purposes of the absorption calculation, however, it is necessary only to know within a few decibels what the absorption level is. For this reason established global models of atmospheric characteristics are used.

2.1 Ionospheric Pre-processing program

An ionospheric pre-processing program was developed which would allow for a flexible but convenient method of specifying a fairly complex plasma frequency distribution in three dimensions. Vertical profiles are composed of up to five sine-squared segments. Variations in latitude and longitude are also sine-squared in nature. Once a model is established it can be further varied by additive or multiplicative sine-squared perturbations. Some examples of vertical and horizontal plasma frequency variations produced by the preprocessing program which is called S2PPR are given in Figures 2.1 to 2.4. In the examples given, the model was specified in accurate geomagnetic coordinates, but they can also be given in geographic or dipolar geomagnetic coordinates. The use of the sine-squared layers was at the suggestion of the contract monitor, Wong (1975). The program S2PPR is based on modifications to program SMOD2 which was written by Arcon Inc.

Improvements to the pre-processing procedures which are incorporated in S2PPR include more sine-squared layers in a vertical profile, longitudinal variations, variations in the height increment of the plasma frequency output so that finer steps may be used in the

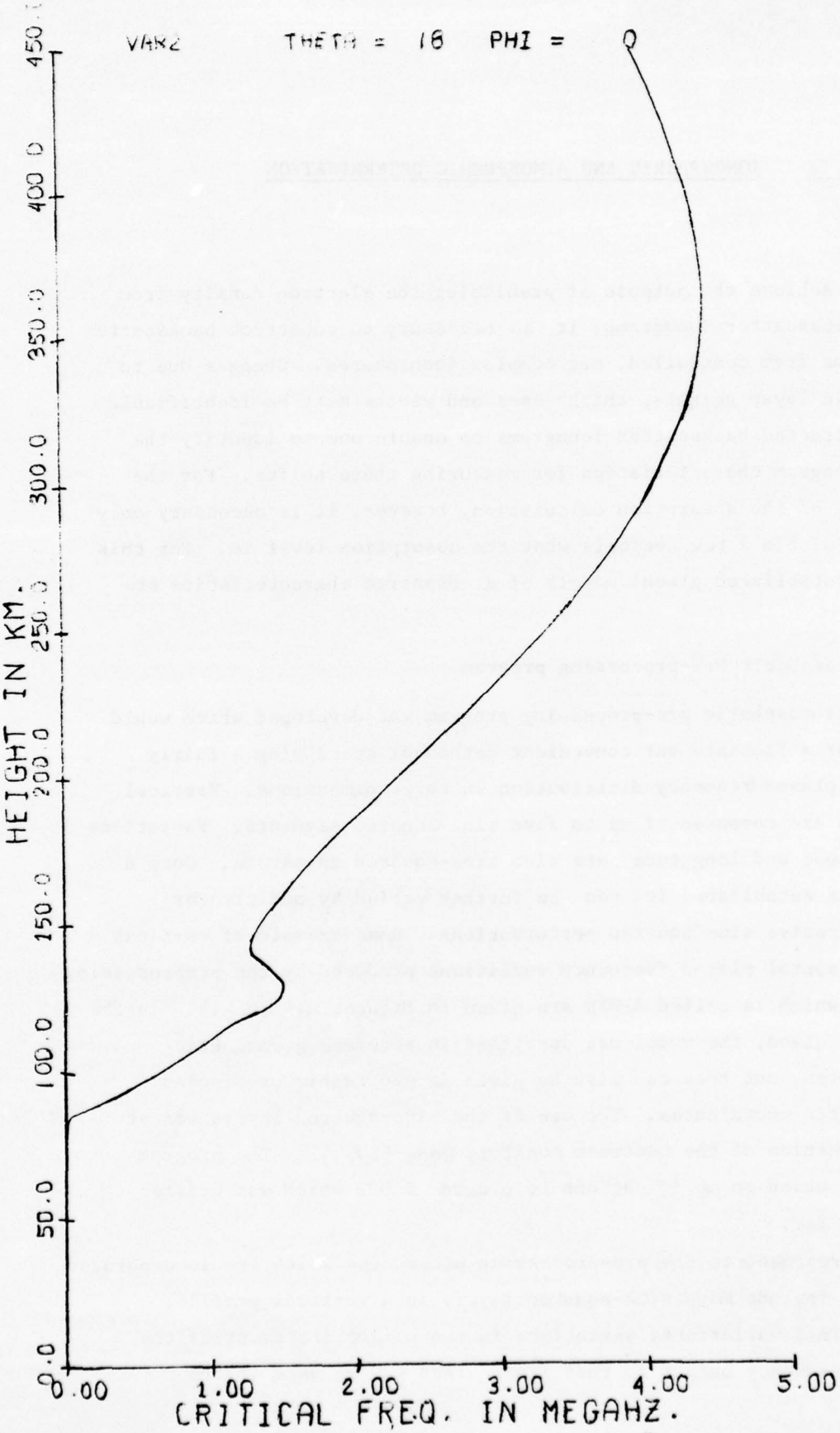


Figure 2.1 Vertical Plasma Frequency Profile at 18° Dipolar Colatitude

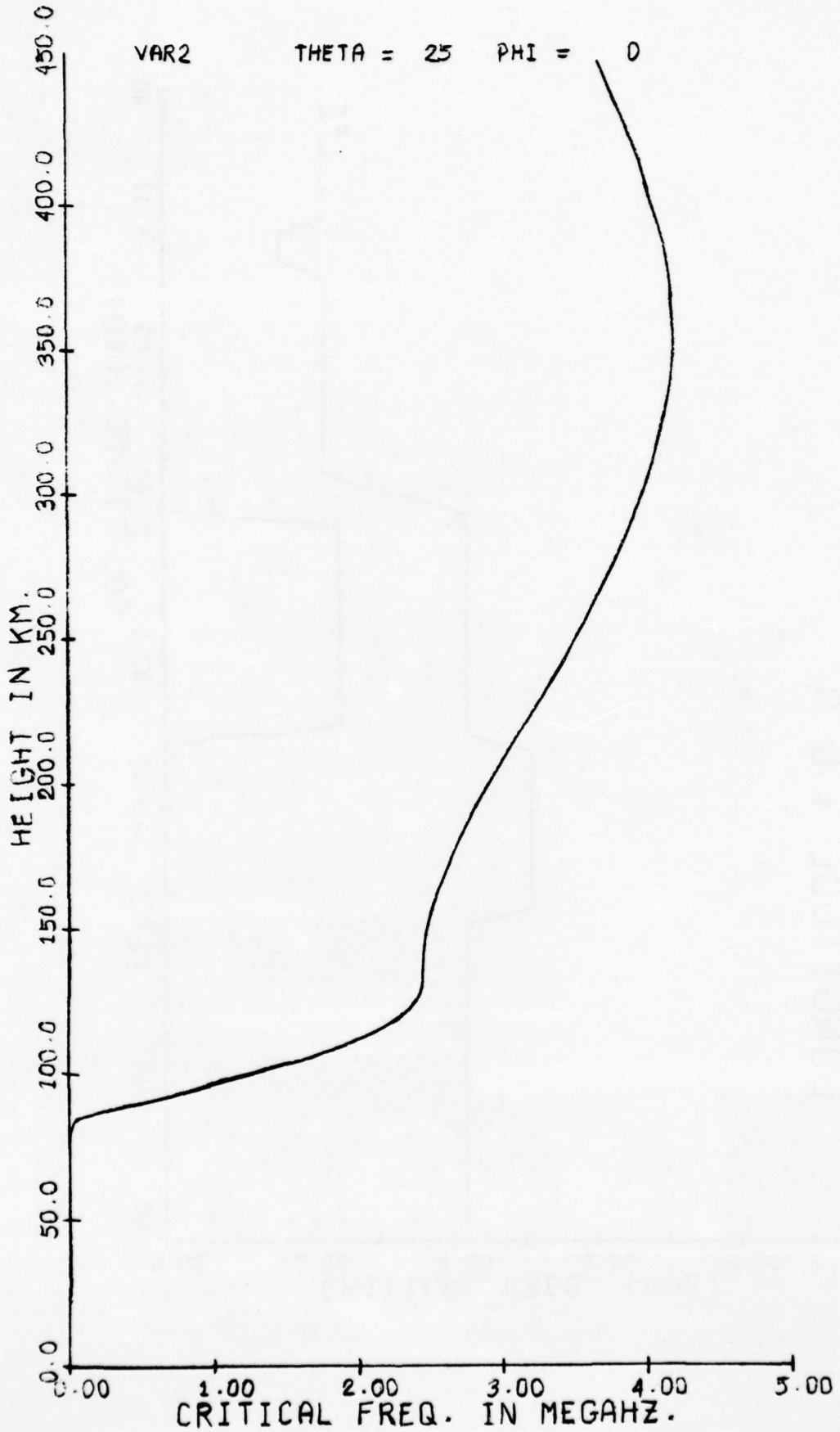


Figure 2.2 Vertical Plasma Frequency Profile at 25° Dipolar Colatitude

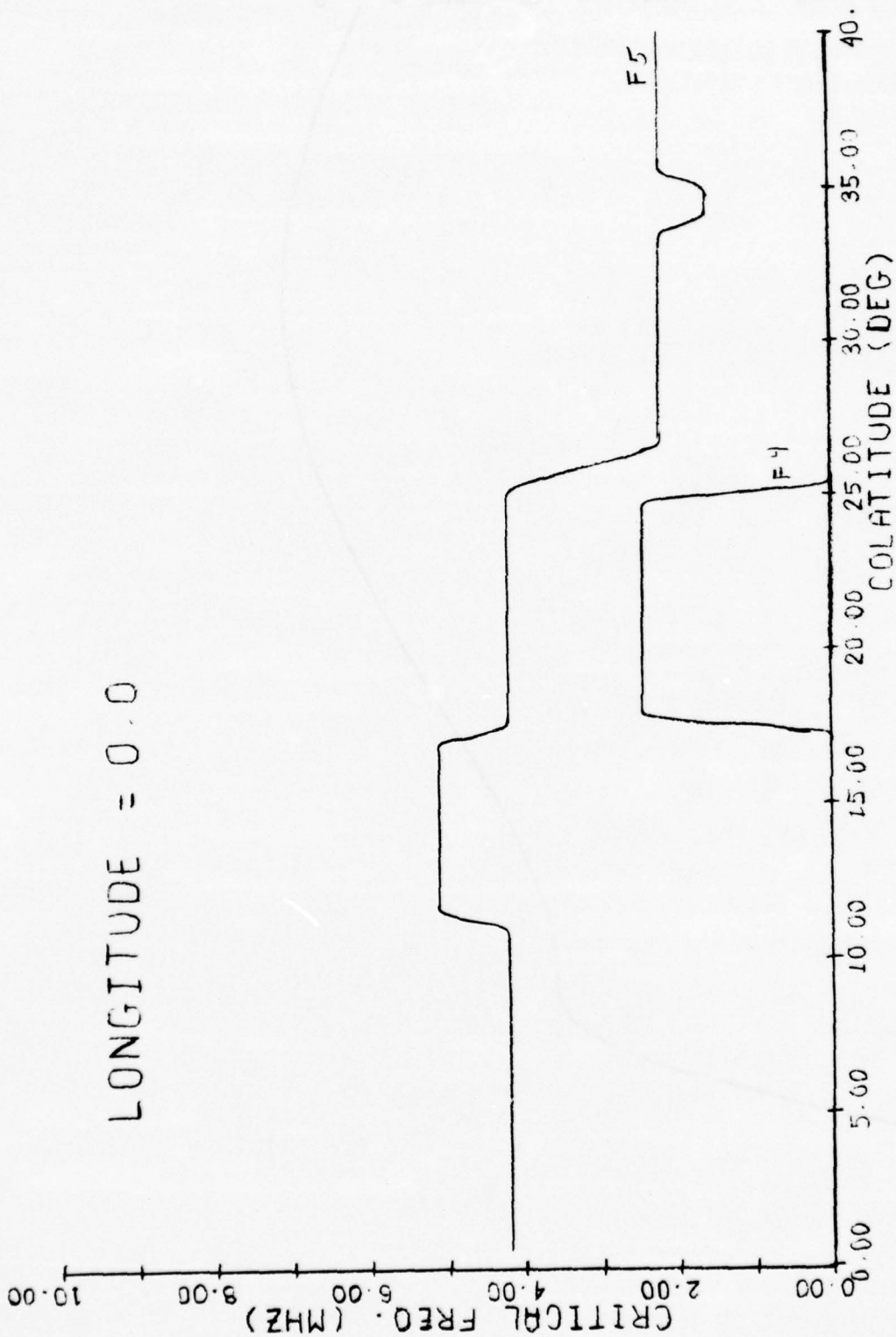


Figure 2.3 Plasma Frequency of Layers as a Function of Accurate Geomagnetic Colatitude

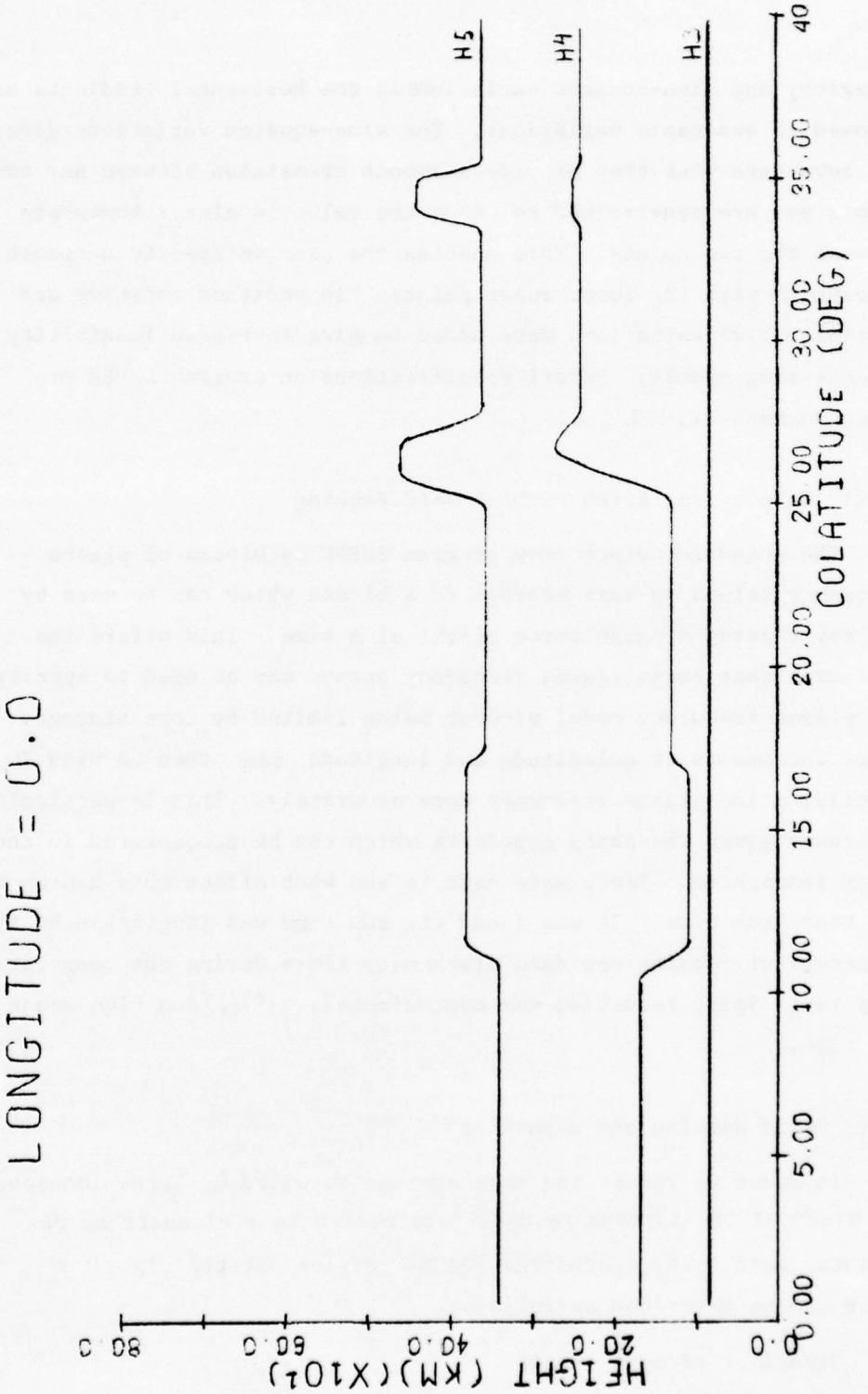


Figure 2.4 Height of Layers as a Function of Accurate Geomagnetic Colatitude

E-region, and sine-squared variations in the horizontal gradients as opposed to quadratic variations. The sine-squared variations give the advantage that they provide a smooth transition between any two points and are constrained so that the value is always somewhere between the two points. This enables the user to specify a smooth ionosphere with far fewer input points. In addition additive and multiplicative variations were added to give increased flexibility in determing models. Detail specifications on program S2PPR are given in Part II, PML 136.

2.1.1 Output from S2PPR without Word Packing

The standard output from program S2PPR is blocks of plasma frequency values in mass storage data blocks which can be used by the ray tracing program three blocks at a time. This offers the advantage that large plasma frequency arrays may be used to specify the plasma frequency model without being limited by core storage. Finer increments in colatitude and longitude can then be used in specifying the plasma frequency more accurately. This is particularly important given the sharp gradients which can be encountered in the polar ionosphere. Tests were made to see what effect this had on the ray trace run time. It was found the run time was lengthened by the procedure of reading new data blocks many times during the computation of a ray. Speed reduction was approximately 33%. (Run time means CPU time.)

2.1.2 Word packing and unpacking

In order to reduce the mass storage required by large ionospheres, the arrays of the ionosphere model are packed four elements to one computer word. The subroutine FPACK4 carries out the algorithm which can be described as follows:

Given four array elements

$$x_1 = f(i,4j + 1,k) \qquad x_2 = f(i,4j + 2,k)$$

$$x_3 = f(i,4j + 3,k) \qquad x_4 = f(i,4j + 4,k) ,$$

let $\bar{x}_i$ be the integer obtained by multiplying x_i , i = 1,2,3,4, by 1000. and by rounding:

$$\bar{x}_i = \text{integral part of } (1000\, x_i + .5),$$

and let y_i be the integer

$$y_1 = \bar{x}_1 \quad y_2 = \bar{x}_2 - \bar{x}_1 \quad y_3 = \bar{x}_3 - \bar{x}_2 \quad y_4 = \bar{x}_4 - \bar{x}_3 \quad .$$

It is assumed that

$$0 \le x_1 < 32.768 \qquad 0 \le y_1 < 32768$$

$$|x_i - x_{i-1}| < 16.384 \qquad |y_i| < 16384 \qquad (i = 2,3,4).$$

Let

$$z_1 = y_1,\ z_i = y_i, \text{ if } y_i \ge 0$$

and

$$z_i = y_i + 16384 \text{ if } y_i < 0 \quad (i = 2,3,4) \quad .$$

The z_i can be represented with 15 bits and stored as

$$z_1 \cdot 2^{45} + z_2 \cdot 2^{30} + z_3 \cdot 2^{15} + z_4$$

in one 60-bit computer word.

(Observe that the assumptions $0 \le x_1 < 65.536$ and $|x_i - x_{i-1}| < 1.024$ allow five elements to be stored in one word using the formula

$$z_1 \cdot 2^{44} + z_2 \cdot 2^{33} + z_3\, 2^{22} + z_4\, 2^{11} + z_5 \ .)$$

The ray tracing program which uses the ionosphere calls subroutine FUNP4 to reverse the packing process in a straight forward manner. Notice that, since the x_i and not their differences are rounded, the unpacking algorithm yields the actual values $\bar{x}_i$ and hence ultimately values of the x_i rounded in the third decimal position (e.g. the original 12.34567 is given back as 12.346).

It is not expected that modifications in the ionosphere of less than .0005 will have detectable effects on the ray tracing. (If needed, four decimal places could be stored by multiplying x_i by 1000 to obtain $\bar{x}_i$; and assume e.g. $0 \le x_1 < 26.2144$ and $|x_i - x_{i-1}| < .8192$.)

The reduction in external storage achieved by the procedure just described implies also a reduction in transfer time from disk to internal storage and hence a reduction in the run time of the ray tracing program. On the other hand, the unpacking does need time. There are two possible strategies. The whole section of compacted ionosphere which is in core can be unpacked, or only those few elements needed for interpolation can be unpacked as required. A first analysis seems to indicate that the second strategy is preferable. It has also the advantage of reducing core requirement.

The modification to the ray-tracing program of unpacking only those elements needed for interpolation proved to be well worth the effort by cost comparison runs. Rays were traced with

Frequency of 5 MHz,
Azimuth of 0^{o} and
Elevations from 0^{o} to 30^{o} at 1^{o} intervals.

The accuracy of the output was well within required tolerances. Most values printed were exactly the same. The maximum range difference occurred at 22^{o} elevation where "ground reference" of the original run was at 1611.08 KM and that of the modified program with local unpacking was at 1607.87 KM.

Because of the time required for unpacking, CPA time is slightly greater in the modified program. However, since the information read from mass storage is packed into 1/4 as many words, IO time is considerably shortened. Total run-time was reduced to 52.5 °/o. Storage requirements are decreased from 142,000 octal to 72,000 octal. With both time and space reduced, the total cost of the runs decreased to 37.1 °/o. The following table show the actual values.

	Original Program		Revision 1	
CPA	74.659 sec.	$1.530	103.141 sec.	$2.114
IO	241.885 sec.	1.620	63.201 sec.	.423
CM	13927.915 KWs	15.320	3917.858 KWs	4.309
Cost of Job		$18.471		$6.847

2.2 Ionospheric Display Programs

The ionospheric model can be displayed a number of ways. Several of the displays used herein were developed under contract F19628-73-C-0307 and are described in report AFCRL-TR-75-0319. The first method of display is the vertical profile. The vertical profile can be displayed for an arbitrary set of colatitudes and longitudes once the ionospheric model has been established. Figures 2.1 and 2.2 are examples of this type of display; details may be found in Appendix F of the aforementioned report.

To display the model in the vertical plane over a range of points, several options are available and have been used in the course of this contract. First the layer specifications can be displayed in the input coordinate system to S2PPR. Layer plasma frequencies and heights can be displayed as a function of the input colatitude. Figures 2.3 and 2.4 are examples of this type of display. In the figures layer parameters are displayed as a function of accurate geomagnetic colatitude. The program which produces these plots, S2PLT, uses the same data deck as program S2PPR. Details on S2PLT are given in Part II, PML 138. Using program S2PLT before an ionospheric model is generated ensures that the data deck will produce the desired model.

A companion program to S2PLT is program S2POBL which is described in Part II, PML 146. This program again uses the data deck for program S2PPR. S2POBL can produce plots of layer parameters similar to those of S2PLT but offers the advantage that plots are of arbitrary ionospheric cross-sections which may be specified by giving a starting point, an azimuth, and a range. Plots of this type are useful in comparing backscatter ionograms for a particular azimuth with the plasma frequency distribution along that azimuth.

A third method of displaying the vertical ionosphere is program RAYPLOT-2 which is a companion program to the ray trace program. Once rays are traced they may be plotted in the vertical plane along with the contours of constant plasma frequency. This procedure is time consuming since it requires the evaluation of the plasma frequency at many points by the ray tracing electron density subroutine. Details on this program can be found in Appendix C of Report AFCRL-TR-75-0319.

Another option of program RAYPLOT-2 is the contour display of plasma frequency in the horizontal plane. This also requires that at least some rays be traced, but at the present time is the only display program available for the horizontal contours.

2.3 Collision Frequency Pre-Processing Program

In the backscatter synthesis rays are computed using an index of refraction with magnetic field but with no collisions. The collision frequency is required to compute the absorption along the ray. This is important especially with low angle rays since excessive absorption may cause rays not to appear on a real backscatter ionogram. The collision frequency, ν', in collision/second/cc. has been specified by the following equations:

$$\nu' = \nu'_{en} + \nu'_{ei}$$

where ν'_{ei} is the collision frequency with ionized molecules in collisions/sec/cc. and ν_{en} is the collision frequency with neutral molecules in collisions/sec/cc.

$$\nu_{en} = 3.6 \times 10^{-10} N_n (\alpha T_n)^{1/2}$$

where N_n is the number of molecules/cc

α is the ratio of electron to neutral temperature

T_n is the neutral temperature in ^{o}K.

$$\nu_{ei} = \frac{5.5 N_e}{(\alpha T_n)^{3/2}} \log_e \left\{ \frac{220. (\alpha T_n)}{N_e^{1/3}} \right\}$$

where N_e is the electron density in electrons per cc.

The quantities N_n and T_n are supplied by the subroutine CCIA described in report AFCRL-72-0171. N_n and T_n are functions of position in r,θ,ϕ, the time of the day, the date (gives the declination angle of the sun), the 10.7 cm solar flux, the average 10.7 cm solar flux, and the planetary index K_p. Details of this dependence can be found in the aforementioned report and in CIRA 1972 report pp 227-257.

The quantity α has been modeled to vary with season, time of day, and solar cycle based on data given by C.M. Rush and T.J. Elkins, (1975). A variation with accurate geomagnetic latitude is also given.

The electron density N_e is supplied during the course of a ray execution by the desired electron density subroutine.

The final collision frequency is not computed until rays are traced since the plasma frequency is required. In this way many plasma frequency models can be used with one collision frequency model. The quantities which are computed by the pre-processing program, COLPP, are N_n, T_n, and α as a function of position, date, and universal time. Tables of ν_{en} and αT_n are then generated for use in the ray trace program. Details of program COLPP can be found in Part II, PML 139.

To display the collision frequency a program was written which plots the collision frequency profile for an arbitrary point in dipolar geomagnetic coordinates. This program requires that program COLPP has been previously run and that a plasma frequency subroutine to the ray trace program is available. New collision frequency profiles should be plotted whenever the electron density subroutine is changed. Plots are labeled with both the collision frequency model and the electron density model used. Details of the collision frequency plotting program, COLPLT, are given in Part II, PML 141.

Chapter III RAY TRACING

Due to the complex nature of the polar trough region of the ionosphere, three dimensional ray tracing was recognized by the contract monitor as the only approach which would allow proper evaluation of the diverse phenomena of this region. This is the only way of evaluating the effects of off-great-circle deviations and of evaluating the effects of the magnetic field. The execution time of the Hamiltonian three-dimensional ray tracing program with magnetic field is, however, very time consuming. As an alternate means of synthesizing backscatter ionograms, a two dimensional quasi-parabolic approach is also being tried. If results from this two-dimensional approach show consistancy with the three-dimensional method, this may be used in regions where the trough gradients are not too great or as a first pass at verifying or modifying predicted electron densities from backscatter ionograms. Any simplification involved in the two-dimensional approach must be tested against the full ray tracing program.

3.1 Hamiltonian Ray tracing Improvements and Modifications

The ray tracing program used in the current investigation is the Jones ray tracing program as modified under contract F19628-73-C-0307 and described in Appendix A of report AFCRL-TR-75-0319. More details on the ray tracing approach are given in Jones and Stephenson (1975). For the current utilization of this program, more specific information had to be extracted than was available in the RAYSET form. A specialized electron density was added, and special magnetic field considerations were implemented. Since a different approach to the computation of absorption was used, these differences caused program modification. Finally the computation of coordinate transformations in the print routine was simplified saving both time and storage.

3.1.1 The Electron Density Subroutine

The electron density subroutine currently used in the ray tracing subroutine is a modification of a routine developed by Arcon Inc. (Vanguri, 1973). Plasma frequency values are read in at discrete grid

points and values of the plasma frequency and its derivatives with respect to height, colatitude, and longitude in the geomagnetic dipolar system are interpolated. Interpolation is quadratic in height and colatitude for plasma frequency values and linear in the longitudinal direction. Derivatives are computed using central finite differences of the four closest points to maintain continuity.

Modifications to this routine were to allow for variations in the height increments. This meant that a finer gird size could be used in the E-region where sharp horizontal gradients occur. Methods of interpolation and derivative computation remained consistent with the previous program.

The other major modification to the Arcon subroutine was to convert the data storage to blocks of data which are read into the computer a section at a time as a particular ray enters that section. In this way much larger data arrays may be used to specify the ionosphere allowing both wider coverage and finer increments in data. The finer increments were necessary particularly in colatitude and longitude to faithfully reproduce the sharp trough wall structure.

At the present time plasma frequency distributions of up to four million data points are in use. Distributions are stored on disk packs for speed in processing. While this method does involve increased computer time, the overall speed reduction runs about 33 %. This is, however, necessary if rays are to accurately represent the trough conditions. Details of this subroutine can be found in Part II, PML 127, Revision 1.

3.1.2 Conditions of Perpendicularity to the Magnetic Field Lines

In the polar region ionization is frequently enhanced along magnetic field lines. For this reason backscatter occurs from magnetic field lines when rays approach perpendicularity to them. It is , therefore, necessary to know the angle between the magnetic field and the ray direction, which is referred to as Ψ . Since the dipolar magnetic field model is not a good approximation in the particular

polar region of interest, it was necessary to determine this angle, Ψ, with a more accurate magnetic field model. For the purposes of ray computation, however, the dipolar model was found to be adequate.

This meant that ray integration was performed using the dipolar model; and when an integration step was completed, Ψ was computed using the accurate magnetic field model. Integration was controlled so that points at which the magnetic field and the ray direction were perpendicular (or Ψ was 90°) were interpolated and printed out. These points then became the basis for the field aligned backscatter component of the ionogram.

The accurate magnetic field model was developed by Arcon, Inc. in 1973.

3.1.3 Establishment of Data Files for Ionogram Computation

As rays are run certain key points arestored for future access in creating backscatter ionograms and making power computations. The points of particular interest are points where the ray is perpendicular to magnetic field lines and points where the ray impacts the earth. All the information about these points is saved in a single record along with an identfying code such as "90 DEG." or "RCVR". Other points which give additional insight to the rays are saved such as the transmitter point, the apogee point, and points past the 90° point which are within 6° of perpendicularity. The information was selected to be comprehensive but not too cumbersome to be easily stored. For each ray trace run which requires a new W-array, the W-array is also stored. All subsequent processing programs on the rays with the exception of program RAYPLOT-2 use this file. The data file has logical file name TAPE7 and is generally stored on multi-file tapes. TAPE7 in effect replaces the "raysets" developed by Croft. Detailed descriptions of the contents of TAPE7 can be found in several program write-ups included in this report. See pages 9-11 of PML 144 in Part II or pages 11-13 of PML 145.

3.1.4 Absorption Calculations

The absorption computation was necessary for some rays since the power level of the ray had to be considered to know if it would appear in a backscatter ionogram. For this reason the collision frequency and the imaginary part of the complex index of refraction had to be computed. There was no need, however, to do complex ray tracing to determine the ray position since collisions had little effect on the ray path. For this reason the complex index of refraction was computed only for the absorption calculations.

The ionospheric radiowave absorption is integrated along a given ray. Its derivative with respect to group path, the independent variable, is

$$\dot{A} = \frac{10}{\log_e 10} \frac{\omega}{c} \frac{\text{imag}(\frac{\omega^2}{c^2} n^2)}{K_r^2 + K_\theta^2 + K_\phi^2} \dot{P} \qquad (1)$$

where A is the absorption in decibels

ω is the angular wave frequency

c is the speed of electromagnetic waves in free space

K_r, K_θ, K_ϕ are the components of the propagation vector

P is the phase path length

and n is the phase refractive index.

All of the above quantities except n are defined in the normal sense as used by Jones ray trace program.

Two methods of computing n^2 were tried. The first and more simple approach was to compute n^2 taking collisions into account but ignoring the magnetic field. In this case

$$n^2 = 1 - \frac{X-iZ}{1-iZ}$$

where X is the refractivity due to the electron density

and Z is the refractivity due to the collision of electrons with neutral and ionized molecules.

$$Z = \frac{\nu}{2\pi f \times 10^6}$$

where ν is the collision frequency in collisions/sec.
and f is the transmission frequency in MHz.

We then have

$$\text{imag}\left(\frac{\omega^2}{c^2} n^2\right) = -\frac{\omega^2}{c^2} \frac{X \cdot Z}{1 + Z^2} \quad .$$

Results of the absorption calculation using this formula can be found in Tables 3.1 and 3.2 under the heading w/o Field.

The second approach was to use the imaginary part of the full Appleton-Hartree formula with magnetic field and with collisions. Results using this approach are given in Tables 3.1 and 3.2 under the heading w/Field. As can be seen from the tables there is a significant difference in the absorption loss over 1 and 2 hops using formulas with and without magnetic field when the elevation angle is 10^o and under. Since the inclusion of magnetic field represents an average computer run time increase of 3%, the absorption calculations now include the magnetic field effects

Ray trace runs should include the absorption calculation only when this information is specifically required. The computer run time in the sample cases in Tables 3.1 and 3.2 represents an average increase of 36%. Moreover, 31,000 octal words of additional core storage were required. While the sample runs in Tables 3.1 and 3.2 were made with automatic error control on the integration step size, no significant discrepancy in absorption losses was noted when a fixed integration step size of 10 km was used.

The collision frequency was computed using collision frequency subroutine COLAF1 which is described in PML 140, Part II. Subroutine COLAF1 is dependent on a pre-processing program COLPP which is described in Section 2.3.

This subroutine makes use of N_e, the electron density, as it is computed along the ray path. Interpolation in the tables provided by COLPP is linear in r, θ, and ϕ for the quantity αT_n and linear in θ and ϕ for ν_{en}. Interpolation for ν_{en} in the height direction is done by

	Hop 1								Hop 2							
	Apogee				Receiver				Apogee				Receiver			
	Linear		Logarithmic		Linear		Logarithmic		Linear		Logarithmic		Linear		Logarithmic	
	w/o Field	w/	w/o Field	w/	w/o Field	w/	w/o Field	w/	w/o Field	w/	w/o Field	w/	w/o Field	w/	w/o Field	w/
El = 0	6.3	6.2	5.2	5.1	12.1	11.7	10.0	9.6	12.2	11.8	10.0	9.7				
1	6.3	6.7	5.4	5.3	12.5	12.1	10.0	9.7	18.8	18.3	14.9	14.5				
2	4.8	4.7	3.8	3.7	10.0	9.2	8.1	7.5	17.8	16.9	14.3	13.6	23.3	21.9	19.0	17.9
3	1.8	1.8	1.7	1.7	7.0	6.2	6.1	5.3	13.0	11.8	11.2	10.2	18.2	16.3	15.6	14.0
4	.8	.8	.8	.8	5.9	5.0	5.2	4.5	11.2	10.9	9.7	8.6	16.2	14.2	14.2	12.5
5	.6	.6	.6	.6	5.4	4.5	4.7	4.0	10.5	9.1	9.2	8.1	15.4	13.4	13.4	11.7
6	.4	.4	.4	.3	5.1	4.2	4.4	3.7	9.9	8.6	8.7	7.5	14.7	12.6	12.8	11.0
7	.3	.3	.3	.3	4.7	3.9	4.1	3.4	9.4	7.9	8.2	6.9	13.8	11.7	12.0	10.2
8	.2	.2	.2	.2	4.4	3.6	3.9	3.1	8.8	7.3	7.6	6.4	13.0	10.8	11.3	9.4
9	.2	.2	.2	.2	4.2	3.3	3.6	2.9	8.2	6.7	7.2	5.9	12.2	9.9	10.6	8.7
10	.2	.2	.2	.2	3.9	3.1	3.4	2.7	7.8	6.2	6.8	5.4	8.2	6.6	7.2	5.8
11	.1	.1	.2	.2	.8	.7	.7	.7								
12	.2	.2	.2	.2	.4	.4	.5	.4								
13	.2	.2	.2	.2	.4	.4	.4	.4								
14	.2	.2	.2	.2	.3	.3	.3	.3								
15	.2	.2	.2	.2	.3	.3	.3	.3								
16	.2	.2	.2	.2	.3	.3	.3	.3	.3	.3	.3	.3	.4	.4	.4	.4
17	.2	.2	.2	.2	.3	.3	.3	.3	.3	.3	.3	.3	.3	.3	.3	.3
18	.2	.2	.2	.2	.3	.3	.3	.3	.3	.3	.3	.3				
19	.2	.2	.2	.2	.3	.3	.3	.3	.3	.3	.3	.3				
20	.2	.2	.2	.2	.3	.3	.3	.3								
21	.2	.2	.2	.2	.3	.3	.3	.3								
22	.3	.3	.3	.3												
23																

Table 3.1 - Losses due to Absorption in decibels at 5 MHz., Azimuth angle = 30°, for VAR2 data.

	Hop 1								Hop 2							
	Apogee				Receiver				Apogee				Receiver			
	Linear		Logarithmic		Linear		Logarithmic		Linear		Logarithmic		Linear		Logarithmic	
	w/o Field	w/	w/o Field	w/	w/o Field	w/	w/o Field	w/	w/o Field	w/	w/o Field	w/	w/o Field	w/	w/o Field	w/
El = 0	6.3	6.2	5.2	5.1	12.1	11.7	10.0	9.6	15.8	15.4	13.0	12.6	19.4	18.9	15.8	15.4
1	8.2	8.0	4.8	4.7	12.9	12.4	10.0	9.6	21.1	20.6	15.3	14.8	26.2	25.5	20.5	19.9
2	3.6	3.6	2.8	2.8	9.2	8.4	7.3	6.7	16.1	15.1	12.8	12.1	22.2	20.7	17.8	16.5
3	1.9	1.9	1.8	1.8	7.8	7.0	6.9	6.3	13.9	12.6	12.3	11.2	19.9	18.0	17.5	16.0
4	.7	.7	.8	.8	5.7	4.9	5.1	4.3	10.9	9.5	9.6	8.5	16.0	14.0	14.0	12.3
5	.4	.5	.5	.5	5.3	4.5	4.7	4.0	10.4	9.0	9.1	7.9	15.3	13.3	13.3	11.6
6	.3	.3	.3	.3	5.0	4.1	4.4	3.6	9.8	8.4	8.6	7.4	14.5	12.4	12.6	10.9
7	.3	.3	.3	.3	4.6	3.8	4.1	3.3	9.2	7.8	8.0	6.8	13.6	11.5	11.8	10.0
8	.2	.2	.2	.2	4.3	3.5	3.8	3.0	8.6	7.1	7.5	6.2	12.8	10.6	11.1	9.2
9	.2	.2	.2	.2	4.1	3.2	3.5	2.8	8.0	6.5	7.0	5.7	10.5	8.7	9.3	7.7
10	.2	.2	.2	.2	.8	.7	.8	.7								
11	.1	.1	.1	.1	.4	.4	.4	.4								
12	.2	.2	.2	.2	.4	.4	.4	.4								
13	.2	.2	.2	.2	.3	.3	.3	.3								
14	.1	.1	.2	.2	.3	.3	.3	.3								
15	.2	.2	.2	.2	.3	.3	.3	.3	.3	.3	.3	.3	.4	.4	.4	.4
16	.1	.1	.1	.1	.2	.3	.3	.3	.3	.3	.3	.3	.3	.3	.3	.3
17	.1	.1	.1	.1	.2	.2	.2	.2	.3	.3	.3	.3				
18	.2	.2	.2	.2	.2	.2	.2	.2	.2	.3	.3	.3				
19	.2	.2	.2	.2	.2	.2	.2	.2	.3	.3	.3	.3				
20	.2	.2	.2	.2	.2	.2	.2	.2								
21	.2	.2	.2	.2	.3	.3	.3	.3								
22	.3	.3	.3	.3												
23																

Table 3.2 Loses due to Absorption at 5 MHz., Azimuth angle = 31°, for VAR2 electron density data

linear interpolation on the logarithm of ν_{en} since ν_{en} is rapidly decreasing in the lower regions of the ionosphere where absorption is greatest. Comparison runs were done using linear interpolation for ν_{en} in height and interpolating on the logarithm of ν_{en}. Results of these runs can be found in Tables 3.1 and 3.2.

3.1.5 Other Modifications to the Ray Tracing Program

Other modifications to the ray tracing program include changes to the print routine. Ray position in dipolar geomagnetic coordinates was added to the print out at each selected position. Also added was the quantity PSI which is the angle between the magnetic field and the wave normal in degrees.

Computation of several quantities which were printed out was changed. They are now computed directly by spherical geometry formulas instead of by matrix transformation. In this way both computation time and storage are reduced. Values were cross checked and no discrepancies found.

3.2 Two-dimensional Ray Tracing with Quasi-Parabolic Model

Three-dimensional ray tracing with the Jones program may prove too slow for applications such as real-time ionogram inversion. Where speed is important, a two-dimensional ray trace may be necessary. The program RAY2 does a two-dimensional ray trace through a model ionosphere made of rectangular cells within each of which the electron density varies quasi-parabolically with height and is constant in the horizontal direction.

The quasi-parabolic model of electron density, introduced by Croft and Hoogasian (1968), is very much like a parabola. It has the advantage that it admits exact equations for ray-path parameters. A single integration, which may be carried out analytically, gives the exact results of transmitting a ray through a quasi-parabolic layer. Quasi-parabolic layers may be stacked one above another to make a piecewise quasi-parabolic profile over a point or region, and rays over that

region may be traced with one integration for each quasi-parabolic layer. If the ionosphere over that region were perfectly horizontally stratified (spherically symmetrical) with a true quasi-parabolic profile, that ray trace would be exact. To deal with horizontal variation, recall that any function may be approximated by a step function. The horizontal extent of regions in RAY2 is limited to the distance where the ionosphere is approximately constant. By the same token, the program is not limited to a fixed integration step nor even to a strategy of varying integration steps depending on local conditions; it can look ahead and determine in advance the region over which conditions are reasonably constant.

3.2.1 Formulae for Quasi-Parabolic Models

The basic formulae for tracing rays through quasi-parabolic layers have been given by Croft and Hoogasian (1968). The integral they give for ground range has six closed forms depending on radicand parameters (see Pierce (1929), pp 16 and 25-26). RAY2 includes subroutines for the inverses of those six functions as well as for the functions, because the quasi-parabolic cells are limited in horizontal as well as vertical extent. Let a cell extend from height H_0 to H_1 and let the ground range of a ray traversing that cell be

$$R = \int_{H_0}^{H_1} f(h)\ dh = F(H_1) \quad .$$

If R is greater than the horizontal extent R_0 of the cell, it is necessary to invert F to compute $H_* = F^{-1}(R_0)$, the height which the ray reaches at the far end of the cell (equivalently, H_* is the upper limit of integration such that

$$R_0 = \int_{H_0}^{H_*} f(h)\ dh) \quad .$$

3.2.2 Fitting Quasi-Parabolic Layers Together

RAY2 accepts the same data for ionosphere description as the three-dimensional ray trace, but ignores perturbations. It fits two

quasi-parabolic cells between successive H-levels (critical points of electron density as a function of height). The quasi-parabolic layers are chosen to fit the input data as to slope and electron density at the critical points. Further, the layer with higherelectron density agrees with the sine-squared model as to value at its low-value side; the other layer matches the slope of the former where they meet. This results in a slight discontinuity in electron density between successive quasi-parabolic layers, but the derivative of electron density is continuous.

3.2.3 Preliminary Results

Preliminary test runs show agreement within 5% for ray apogee height, apogee range, and one-hop ground range between the two- and three-dimensional ray trace programs for most low-angle rays, with some disagreements around 8%. Agreement is poorer for rays which have initial elevations above 12 degrees.

Comparison of program time is difficult. The two-dimensional ray trace does not yet compute group path, phase path, and a few other ray parameters computed by the three-dimensional ray trace. Inclusion of any of those computations will slow the two-dimensional trace down somewhat. On the other hand, the two-dimensional trace includes a pre-processing step which is done in a separate program in the three-dimensional job; this makes the comparison more favorable to the two-dimensional trace when total time from data input to completed rays is of interest. At any rate, preliminary timings, not corrected for the above-mentioned effects, indicate that the two-dimensional trace is running almost ten times as fast as the three-dimensional one.

Much more testing is needed to see if the minimum trace plots generated by this program show the same general features as those from the three-dimensional ray trace. The preliminary plots of ground range versus elevation angle doshow the same general features, so the two-dimensional ray trace appears promising.

Chapter IV BACKSCATTER SYNTHESIS

4.0 Introduction

A wealth of information is produced as a result of tracing rays through a complicated 3 dimensional medium represented by an inhomogeneous, anisotropic ionosphere. In point of fact, each computed point on each ray and the environment at this point is a potential source of useful information. To be really useful however, this mass of data must be sorted out, possibly be transfomed into other information and finally displayed in a fairly easily digestible form. For this purpose several "display" programs have been written. They are described in Part II. Here, however, a particular set of information transformations - display subroutines is described. The displays are "synthesized" backscatter plots of various sorts including a full ionogram (backscatter intensity vs. group path delay and frequency), a minimum group path "ionogram", and others. These displays closely emulate work which has been reported on previously (Georges and Stephenson, 1969). However, these previously reported techniques have been broadened and restructured to suit the specific objectives of the contract and to provide a set of more generalized software which can be adapted to varying needs.

4.1 General Discussion

Ideally one can imagine that a given set of rays which have been traced represents an adequate sample of the continuum of rays which nature uses in the real world (assuming, of course, that the ray approximation to radio wave propagation is valid). From a backscatter standpoint, the information which each sample should provide is:

1) group path length (to give the time delay to the backscatter region)
2) "effective distance" to the backscatter region (in order to compute R^{-4} losses)
3) power absorbed from the ray due to electronic collisions

4) various geometric quantities such as backscatter region coordinate, ray direction, etc. for purposes of computing backscatter cross sections.

This information is normally supplied by the ray trace program on TAPE7 (Section 3.1.3).

Assuming that an adequate sample is available for a given frequency, the received power within a particular range (group path length) gate, Δg, can be computed from

$$P_r^i = \sum_{\text{all samples}} p_j \qquad \begin{pmatrix}\text{such that group path length of the sample is within the } g_i,\ g_i + \Delta g \\ \text{range "bin"}\end{pmatrix}$$

The quantity p_j is the computed incremental backscatter power due to the j^{th} sample ray. It is assumed that the rays have been suitably normalized so that their incremental outgoing powers sum to the total transmitted power.

Unfortunately it appears to be difficult to obtain a good measure of "effective distance" (or ray focusing) although Bennett (1969, 1973) alludes to the possibility of calculating this quantity for a single ray. An intuitively appealing alternative to this ideal single ray is a small set of rays which can define a "flux tube". Assuming things go well (the set of rays do not get all "tangled" or go wandering off some place), an effective R can be calculated from $R = \sqrt{A_p/S}$ where A_p is the cross sectional area of the flux tube at the backscatter region (hereafter often referred to as "landing point") and S is the "size" of the flux tube at the transmitter in steradions. Thus one is confronted with the problem of how many rays constitute an adequate set for cross-sectional area calculations. Obviously at least 3 are needed. It is equally obvious that if the rays happen to "land" along a line that cross-sectional area cannot be defined. The algorithms used here require 4 rays to define a set suitable for cross-sectional area calculations. Four rays can land on or close to a line just as can 3 rays but less likely so. In any case, with enough sample ray sets, it probably does not make much difference how many rays are used to define a flux tube.

The strategy described in Section 4.2 for computing "ray density" and hence backscatter power is to compute the projected area A_p for the

pertinent landing points between all neighboring rays which have been traced and which land, using 4 neighboring rays to define a flux tube. This strategy was also used by Georges and Stephenson (1969). Imagine a Cartesian frame of take-off elevation and azimuth where elevation typically runs from 0 to 30 degrees and azimuth over same interval (20 to 30 degrees) which suitably models a radar antenna pattern relative to the electron density model and transmitter location. The rays are usually traced at equal intervals of .5 or 1 degrees. Taking a reference ray at (a_i, e_i) (azimuth, elevation) then the 3 neighboring rays which define a sample flux tube are at $(a_i, e_i - \Delta e)$, $(a_i - \Delta a, e_i)$, $(a_i - \Delta a, e_i - \Delta e)$. Thus, except at the borders of the sample region, each ray may be used 4 times to define 4 flux tubes - providing of course that all rays in a given flux tube quartet arrive at the appropriate landing point.

Subroutine DEN, which is described briefly in the next section and in Part II, PML 161, accepts a ray sampling procedure which is slightly more general than that described above in that neighboring elevation angles need not be the same. For minimum group path length (g-p-l) power calculations, ray density is calculated around only those rays which are found to be minimum g-p-l rays, rather than for all rays. This is discussed in section 4.4.

In addition to the problem of computing the effective R to be used in the radar equation, one is faced with the problem of computing radar backscatter cross sections for the various regions where such backscatter is liable to contribute significant power. The two regions (landing points) used herein are the earth's surface and certain regions in the auroral zone where the rays intersect the earth's magnetic field at ninety degrees. These regions often experience enhanced ionization in the form of field aligned "irregularities" which cause partial reflection of electromagnetic energy provided the perpendicularity condition is satisfied. As far as the rays themselves are concerned, the irregularities are assumed to be so localized that the ray paths are not affected. A brief discussion on radar cross-section calculation is also included in the next section.

4.2 The Backscatter Algorithm

The received power, P_r due to a "target" with total radar cross-section, σ, at distance R from the transmitter may be approximated by the "radar equation" (Skolnik, 1962).

$$P_r = \frac{P_t G_t A_r \sigma}{(4\pi R^2)^2}$$

where

$A_r = \frac{\lambda^2 G_r}{4\pi}$ is the receive antenna effective aperature,

G_t, G_r are the transmit, receive antenna power gains (over isotropic antennas),

P_t is the transmitted power,

λ is the free space wavelength of the electromagnetic waves.

This equation makes several assumptions such as a "point" target, a non-absorbing medium, etc. For extended targets such as the earth's surface or large areas of enhanced ionization it is generally necessary to consider the contributions to received power from many small flux tubes for which σ and R remain approximately constant. Furthermore, for pulsed radars which are considered here, it is frequently desirable to sum the received power into various group path (time delay) "bins". These bins are usually representative of a given radar system (range resolution) and/or the sampling intervals used during ray tracing. Thus consider the enhanced radar equation, including losses, for a single flux tube

$$p_{r,i} = \frac{P_t G_t A_r \sigma_i}{(4\pi R_i^2)^2} \cdot (\text{abs})_i \cdot (\text{refl})_i$$

where subscript i refers to the i^{th} flux tube,

p_r is the incremental received power,

(abs) is the absorption loss factor,

(refl) is the reflection loss factor for multi-hop,

and the other terms are defined above.

Let the area which the i^{th} flux tube cuts out of the landing point be denoted by A_i and the angle of arrival be denoted by ψ_i such that $A_{p,i} = A_i \sin\psi_i$, then the above expression for incremental received power can be written as:

$$p_{r,i} = P_t \left(\frac{\lambda^2}{4\pi}\right) G_t G_r \left(\frac{S_i}{4\pi A_i \sin\psi_i}\right)^2 \sigma_i \, (\text{abs})_i (\text{refl})_i$$

This is essentially the form used by Georges and Stephenson (1969) except they include a further loss term due to pulse spread and do not include losses due to absorption and ground reflection. The pulse spreading results merely because the four rays defining a given flux tube have different group path lengths to the landing point which in turn results in varying delay factors in the return of the pulse leading and trailing "edges". It is an approximate factor which is calculated here as

$$(\text{spread})_i = \frac{\tau}{2\sigma_{t,i}}$$

where τ is the transmitted pulse width (in units commensurate with group path "length") and $\sigma_{t,i}$ is the sample standard deviation of group path lengths of the quartet of rays defining the i^{th} flux tube.

Note that σ_i is the total radar cross section which can be computed from $\sigma_i^o A_i$ where σ_i^o is the (non-dimensional) radar cross-section per unit area and A_i is the illuminated area at the landing point.

The above expression for $p_{r,i}$ is written in a form which expresses the various loss terms in a meaningful way. Operating on the terms in brackets (and the spread loss term) with $10 \log_{10}$, the following loss terms (and their units) are obtained:

$$G_1 = 10 \log_{10} \frac{\lambda^2}{4\pi} = \text{"wave length loss"} \quad (\text{db-km}^2)$$

$$G_2 = 10 \log_{10} G_t = \text{transmit antenna gain} \quad (\text{db})$$

$$G_3 = 10 \log_{10} G_r = \text{receive antenna gain} \quad (\text{db})$$

$$G_4 = 20 \log_{10} \left(\frac{S_i}{4\pi A_i \sin\psi_i}\right) = \text{distance gain} \quad (\text{db-km}^{-4})$$

$G_5 = 10 \log_{10} \sigma_i^o \cdot A_i$ = cross-section gain (db-km^2)

$G_6 = 10 \log_{10} \frac{\tilde{\tau}}{2\sigma_{t,i}}$ = pulse spread gain (db)

$G_7 = 10 \log_{10}(abs)_i$ = absorption (db)

$G_8 = 10 \log_{10}(refl)_i$ = multi-hop reflection gain (db)

The following is a brief summary of how these factors are computed.

G_1 is computed directly from knowledge of the specified transmitted frequency.

G_2 is computed through the use of various appropriate function subroutines. (For an example see Part II, PML 142.)

G_3 is computed in a similar manner as G_2. For an example see Part II, PML 143.

G_4 is computed from data returned by subroutine DEN which among other things computes the quantities S_i, A_i and ψ_i. A fairly complete description of this subroutine appears in Part II, PML 161.

G_5 is computed through the use of appropriate backscatter cross-section function subroutines. As examples refer to Part II, PML 163. PML 163 is a user's guide to subroutine BACSCA1 which computes the incremental power from one flux tube and one landing point.

G_6 is computed from information supplied by DEN and the system pulse width (expressed in kilometers)

G_7 must be computed within the ray trace program since absorption is a quantity which is calculated by integration along the ray path. Refer to Chapter 2, part 2.3 and Chapter 3, part 3.1.4 for more details.

G_8 is computed by assuming a constant db loss per ground reflection. This is taken to be around 1 db per reflection independent of angle of incidence and frequency.

It should be pointed out here that no account is taken of the fact that power is extracted from the flux tube due to scattering as it goes from landing point to landing point. In particular, even though

scattering from field aligned irregularities may partially shield the ground backscatter region, this is considered to be a second order effect.

Finally something about "flux density" should be mentioned. A measure of "flux density" (e.g. lines/km^2) is given by the inverse of the cross-sectional area of the flux tube at a given point. This measure is valid only if flux conservation is observed i.e. as long as all rays which start out within the boundaries of the flux tube remain there.

The boundaries of the flux tube, on the other hand must change in shape as it wanders through the ionosphere. In particular it is extremely unlikely that it maintains the straight sides which it starts out with. Flux tube area is computed assuming straight sides. Hopefully, on the whole, this will provide a good approximation. Subroutine DEN is designed to "flag" those instances when it appears likely that, because of distortions in the relative geometry of the flux tube quartet, the area estimate is a poor one. Flux tube density is not really a measure of ray focusing unless geometric path length is available. This quantity can be computed by integration of geometric path along the ray during ray tracing. A measure of flux tube focusing or defocusing is the ratio of effective distance R (given by $\sqrt{A_p/S}$) and geometric path R_e. More exactly, focusing gain is given by $40 \log_{10} R_e/R$. This quantity is calculated and available for display provided R_e has been computed.

4.3 Ray Processing

The ray trace program produces several useful outputs which includes a printed digest of information about each ray, a file of ray coordinates which is useful for plotting purposes and finally a "rayset" which is a file of information which provides enough "raw data" for further calculations and displays. (See Langworthy, Barrett, 1975.) For a description of the "rayset" file see Part II, PML 144 or PML 157. This file of "raysets" (called TAPE7 within the ray trace program) is processed through subroutines CRRSET (Part II, PML 157) and DEN (Part II, PML 161) to produce the file of "flux tubes" which are finally used to provide information for ionogram synthesis and other similar displays. A detailed description of this file processing is given there and will not be repeated here.

4.4 Minimum Group Path Backscatter Synthesis

In Section 4.5 the procedure for full ionogram synthesis is outlined. A useful simplification is the "leading edge ionogram", which for a fixed azimuth frequently shows the dominant features of the full ionogram (Wong, 1975). The leading edge ionogram usually consists of several disconnected traces which represent regions of enhanced backscatter return because of ray focusing effects and the minimum distance "effect". This synthesized ionogram should follow the bottom line of enchanced power offull ionograms.

This leading edge ionogram is obtained by looking at the variation in group path length withelevation angle for a particular azimuth and transmission frequency. All relative minima in group path are considered to be leading edge points. Also considered a leading edge point is the last elevation angle before ray penetration if the group path is decreasing with elevation angle.

The leading edge points are considered to be the first returns of any power; as a rule the minimum path points suffer less power loss. When spread losses were calculated, such pointswere indeed found to give minimum spread losses. Final ionograms cannot be constructed until all power considerations are known. For this reason, leading edge ionograms are constructed in two stages.

First, information on minimum points is extracted by program GPMT from TAPE7 as described in Section 3.1.3. Details of program GPMT can be found in Part II, PML 144. Sample output from this program, a leading edge ionogram, is shown in Figure 4.1. For any given azimuth, points are collected for all available rays and joined on the basis of similar elevation angles.

If frequency spacing is too large, points may not be connected properly. The leading edge ionogram program is iterative in nature so that if additional frequencies are needed to clarify some fine structure, just those additional rays are computed. Leading edges are computed for both 90^{o} points and 1-hop receiver points, giving both field aligned and ground backscatter.

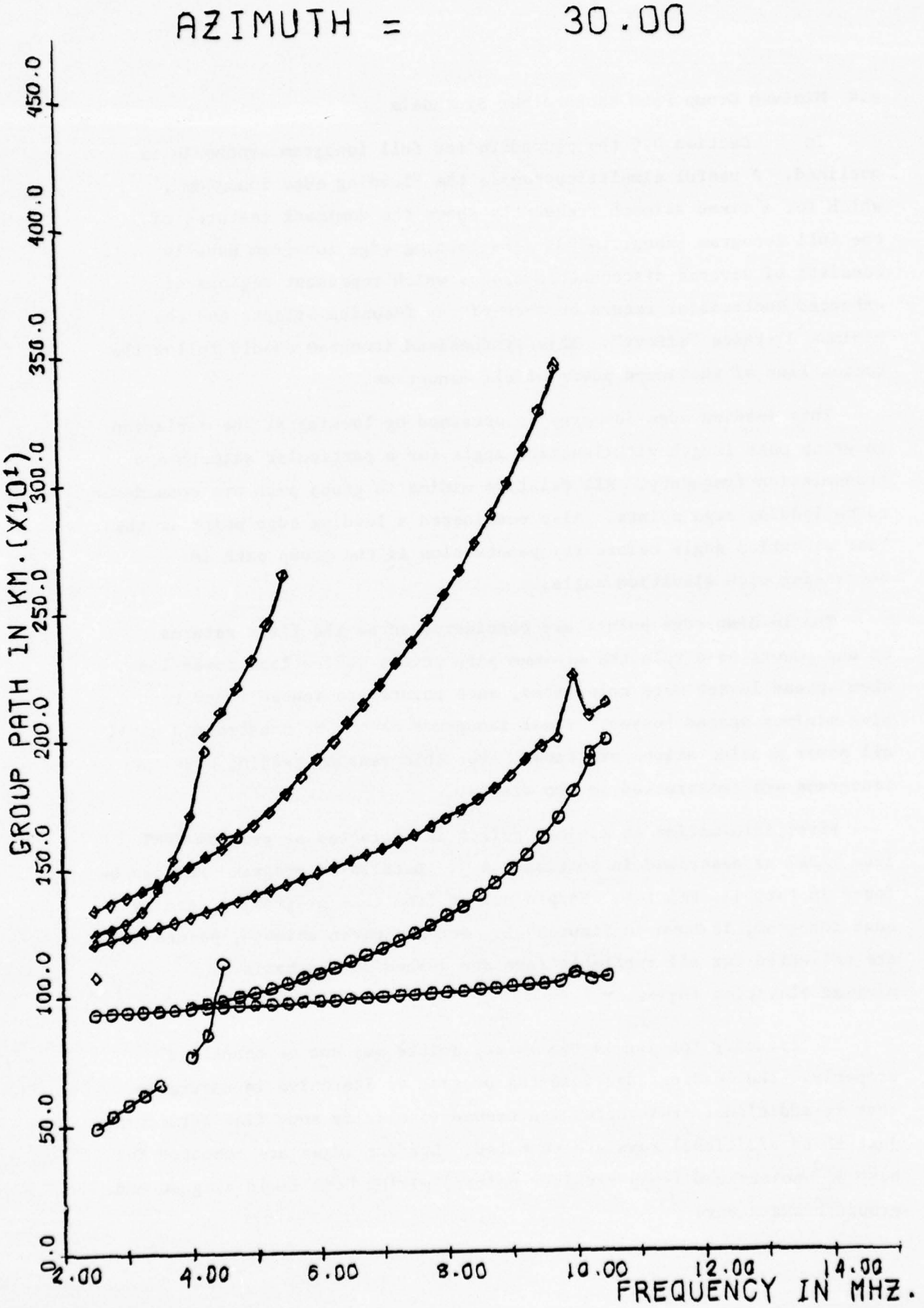


Figure 4.1 Sample Leading Edge Ionogram

Once rays of the leading edge ionogram are selected, power for these rays is then calculated. Program GPMT creates special ray trace runs which will compute the absorption and provide rays at fine increments to compute spread losses. The ray trace information along with other output tables from program GPMT are used as input to program POWER which uses the formula for power at the receiver, P_r, as given in Section 4.2.

4.5 Complete Ionogram Synthesis

A complete ionogram is synthesized by utilizing all of the pertinent ray trace data which, hopefully, provides an adequate sample for the continuum of rays which could be traced. From a systems standpoint it is sometimes convenient to think of ray tracing (or the results thereof) as a mapping from one set of "observables", frequency, initial azimuth, initial elevation, (f,a,e) to a second set of "observables", backscatter intensity, group path length (or range), (P,G). Usually this mapping is not 1 - 1 since, for a given (f_1,a_1,e_1) there may be several (P_iG_i) due to multiple backscatter regions. Thus one should consider the map $(f,a,e) \longrightarrow (P_1,G_1,\cdots P_nG_n)$ where n is the number of backscatter regions.

There are several displays of information which can be considered to follow from the mapping. They correspond to various displays (or ionograms) which are described in Stephenson, Georges (1969) and are as follows: (note that in general these may be "3D" displays with the dependent variable indicated by grey scale.)

1) P vs $(e,a)_{\text{const f}}$

   f is held constant and a summation over all P_i is done; G_i is ignored i.e. the map

   $(a,e)_{\text{const f}} \longrightarrow \sum P_i$ is displayed.

2) P vs $(G,e)_{\text{const f}}$

   here consider two steps

   a) $(e)_{\text{const f}} \longrightarrow$ curves in each (P_i,G_i) plane

   b) for each curve (parametrized by e) sum all values of P (over all P_iG_i planes) for a given value of G

3) P vs $(G,a)_{\text{const f}}$

same as for (2) but the curve parameter becomes a, i.e. change e to a in (2).

4) P vs G,f

here consider three steps

a) for each value of f and e (say) $(a)_{f,e}$ $\rightarrow$ curves in each (P_i,G_i) plane (for each f obtain curves parametrized by e)

b) for each value of G sum all values of P (over all curves parametrized by e and over all P_i,G_i planes) (thus we get a curve in the P, G plane parametrized by f)

c) repeat a) and b) for all values of f to obtain a family of curves. Then given any value of G and f obtain a unique value of P.

From a more operational (and computer oriented) point of view imagine the f, a, e region divided up into resolution cells (of size Δf, Δa, Δe) and the P, G region divided into G bins of width ΔG (here consider only one of the P, G planes). The various displays are then generated according to how P is binned. More specifically consider three types of binning denoted by CAA, CIA and AII.

CAA binning

One of the 3 variables in the f, a, e region is held constant and the G-bin is infinitely wide, i.e. the only summation of P is over the various P_i,G_i regions (irregardless of value of G). This type binning is typified by

P vs $(e,a)_{\text{const f}}$

The significance of the symbol CAA is that 1 variable in the f,a,e region is held constant and the other two are arguments (independent variables).

CIA binning

One of the 3 variables in the f, a, e region is held constant; bin P over another one of these variables into G-bins for each value of the third variable. This type binning is typified by

P vs $(G,e)_{\text{const f}}$

C and A stand for Constant and Argument as for CAA: the I stands for "integration".

AII binning

Bin P over two of the f, a, e variables into G-bins for each value of the third (independent) variable. This type binning is typified by

P vs G,f

A subroutine called BIN (Part II, PML 169) is used to do this binning. However, BIN is more general than is typified by the examples in that there are no preferred variables in the f, a, e region. Any may be C or A or I, provided of course, that the data allows them to be.

Once the flux tubes have been "binned" in the desired fashion, the resulting information can be displayed. For example, the usual ionogram which belongs to display type 4 (P vs G,f) can be produced. At present there are available two ways of making such displays. The first is produced on a printer wherein backscatter power, P, is printed as either a single code symbol which represents the value of P or as a floating point number. The position of the P-value on the printed page gives its corresponding group path delay, G, and frequency, f, values. The code symbol scheme is preferable for a qualitative look at the data, for an example see figure 4.2. The floating point number scheme gives quantitative, but difficult to digest, results. This type display is produced using subroutine PRINT1, (Part II, PML 170).

The other type display is a true intensity display using a CRT plotter to produce intensity variations proportional to P as a function of position representing G and f, for example. This type display is produced by subroutine CRT1 (Part II, PML 168). For examples see figures 4.3 and 4.4. The algorithm for producing the intensity variations is the same as that reported on in Stephenson, Georges (1969).

There are several ways of normalizing (or distorting) the values of P depending on the system which is being simulated. In Figure 4.4 the P values are normalized to the largest P value (the largest value becomes 1) at each frequency. The various discrete intensity levels are

BEST AVAILABLE COPY

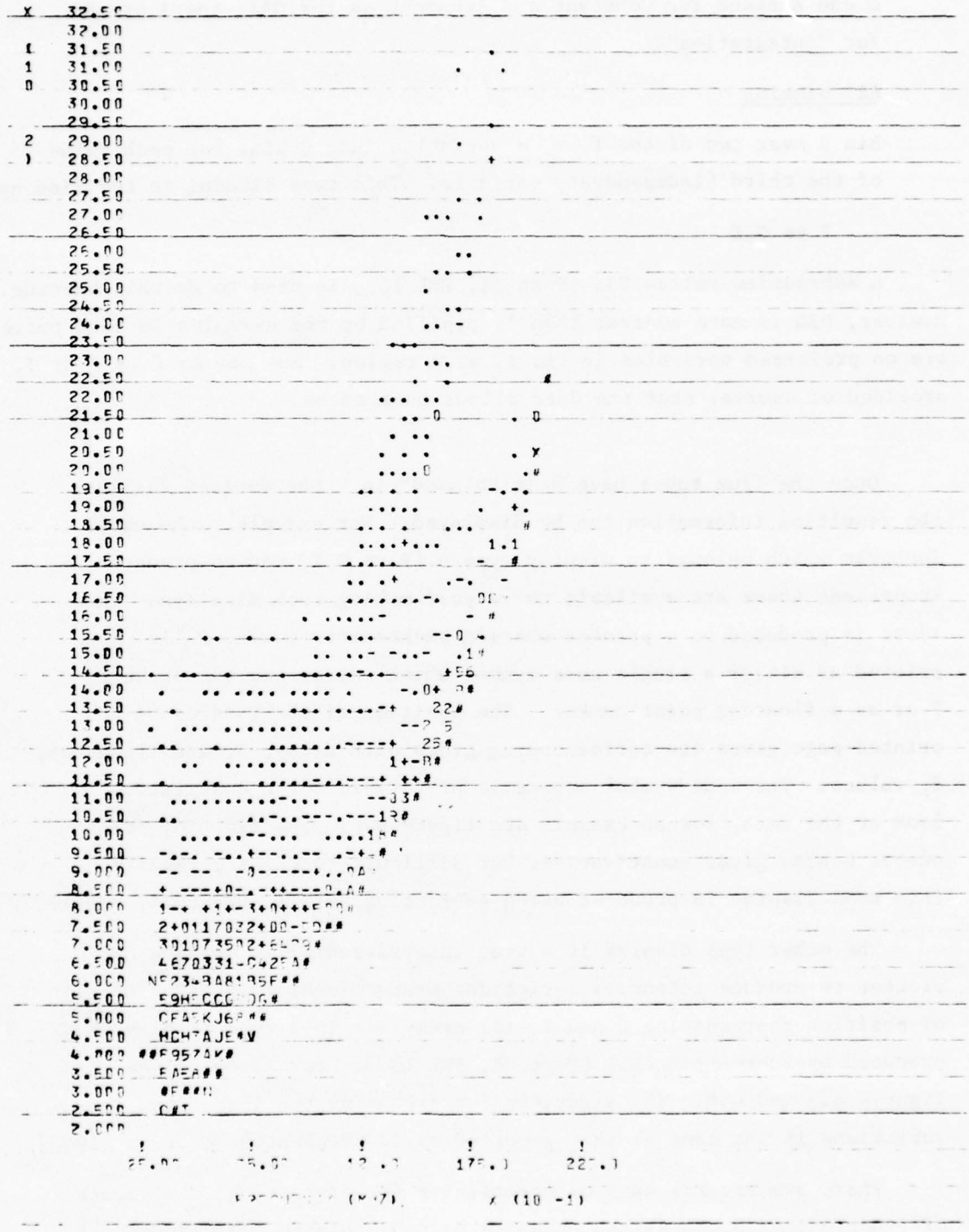


Figure 4.2 Coded Print Ionogram

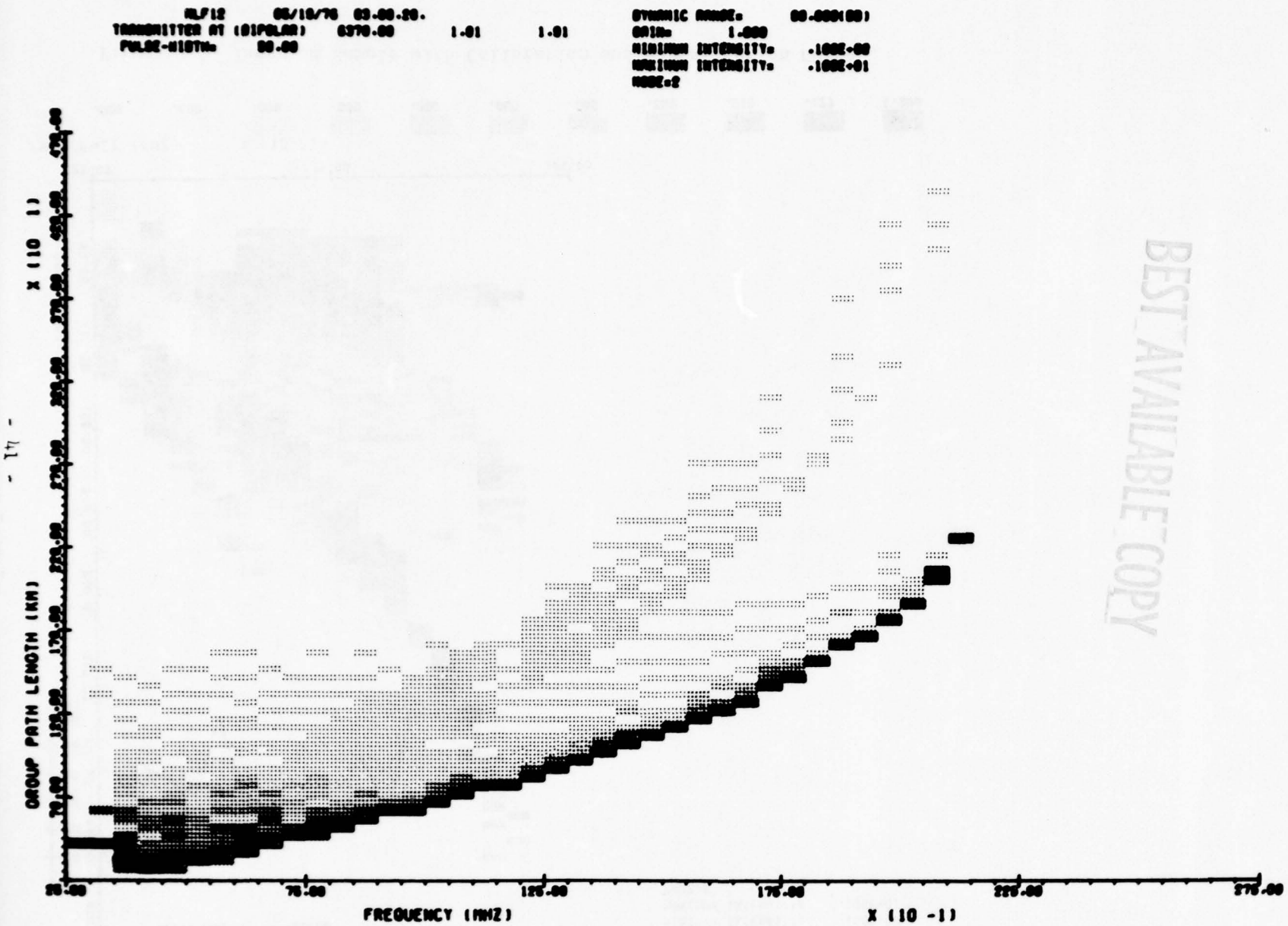


Figure 4.3 Ionogram Sample

BEST AVAILABLE COPY

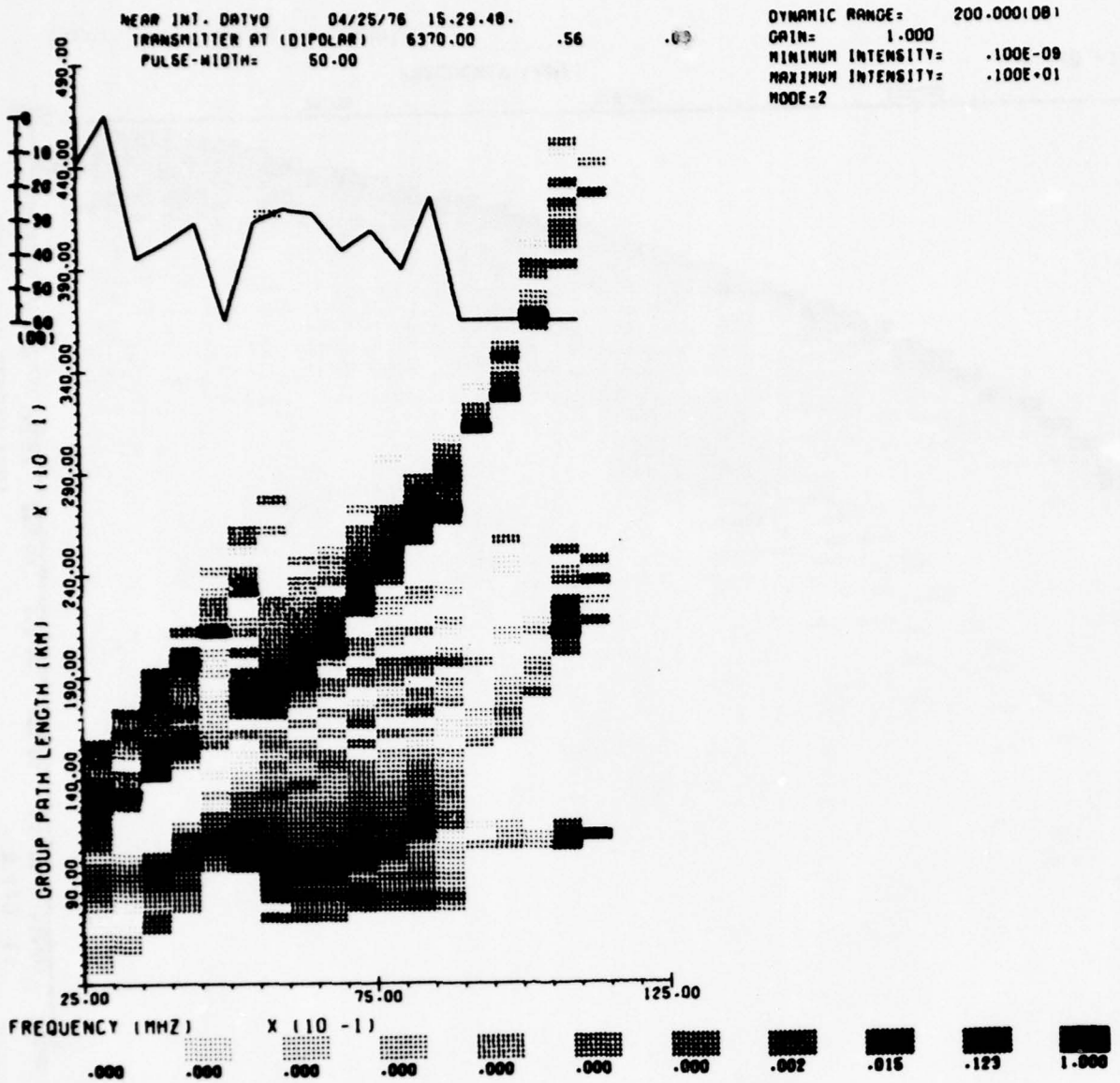


Figure 4.4 Ionogram Sample with Calibration and Normalization Curve

related to one another in logarithmic fashion. For example, if I_{max} is the highest intensity value, the other values are given by

$$I_i = I_{max} f^{-i} \quad , i = 1, N$$

where N is the number of intensity levels and f is a factor (>1) to be specified by the user. At present, the number of intensity levels which can be displayed is 11 going from white to black. The P value to be displayed is, of course "digitized" into the appropriate intensity level. If P falls below the minimum intensity level given by

$$I_{min} = I_{max} f^{-N} ,$$

the P value is not displayed (appears as white). In figure 4.4, a curve showing the relative values of the maximum P values for each frequency is superimposed on the intensity plot. Also shown is a calibration scale.

In addition to the various "ionograms" which can be produced through a suitable choice of BIN values (see PML 169 for details), several other display subroutines have been developed for the extraction of information from "raysets" and/or "flux tubes". Refer to Part II, PML 162 and PML 164 for examples. The following diagram shows the information flow and processing required for various displays (Figure 4.5).

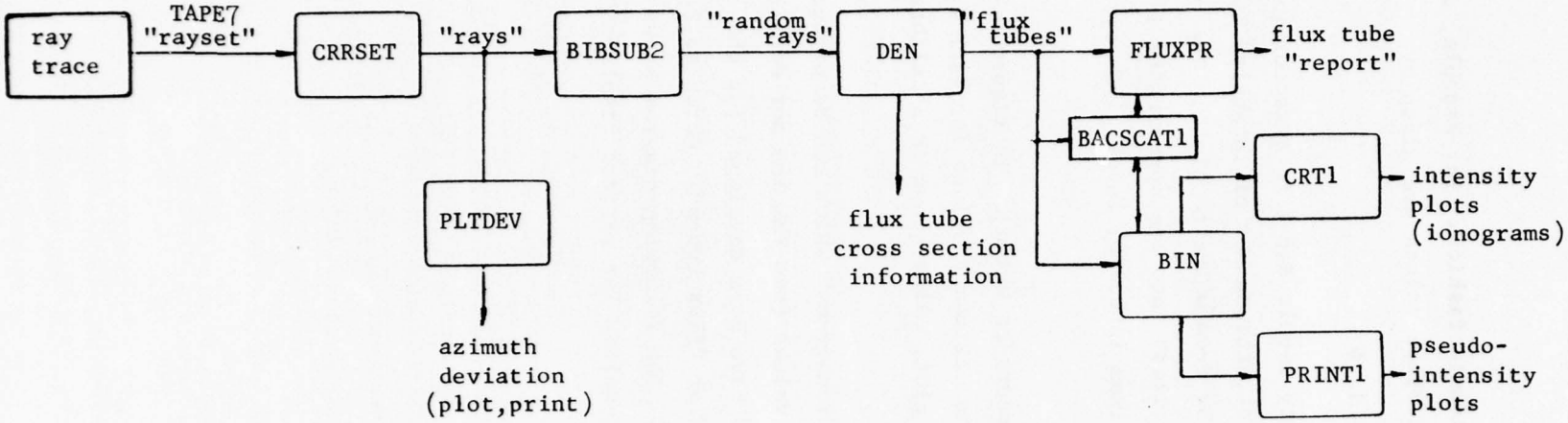


Figure 4.5 Information Flow Diagram for Ionogram Synthesis, etc.

Chapter V BACKSCATTER ANALYSIS

Once backscatter ionograms have been computed they are correlated with the electron densities which produced them to find out which ionogram characteristics are good indicators of particular electron density characteristics. For the present, group path minimum trace ionograms are being used for this purpose; but eventually characteristics of the complete backscatter ionograms will also be used.

To determine which ionogram characteristics should be used in predicting electron density characteristics, a multiple linear regression analysis program was written. This program MLRA also computes a standard error of estimate and the coefficient of multiple correlation. Using the coefficient of multiple correlation, the effect of an individual term can be measured. Details of program MRLA can be found in PML 155, Part II.

Electron density characteristics will be determined for any given azimuth from plots produced by program S2POBL.

At the present time only limited use has been made of program MRLA, but it will become a part of a system of programs designed to determine a polar trough ionosphere given a backscatter ionogram. Ionograms will be broken down into general classes based on large scale features. A trial electron density will then be predicted using the regression coefficients for that particular class of ionogram. The electron density will then be used to produce a synthetic backscatter ionogram using either the two- or three-dimensional ray tracing approach. (For sharp gradients the three-dimensional approach must be used.) Based on the synthesized ionogram, the electron density model will be modified in proportion to the regression coefficients and the process will be repeated.

References

Bennett, J.A. : (1969) On the Application of Variation Techniques to the Ray Theory of Radio Propagation. Radio Science 4, 667

Bennett, J.A. : (1973) Variations of the Ray Path and Phase Path: A Hamiltonian Formulation. Radio Science 8, 737

CIRA (1972) COSPAR International Reference Atmosphere, Compiled by the Committee for the COSPAR International Reference Atmosphere, COSPAR Working Group 4, Akademie Verlag, Berlin, 227-257

Croft, T.A. and H. Hoogasian : (1975) Exact Ray Calculations in a Quasi-Parabolic Ionosphere With No Magnetic Field. Radio Science 3, 69-74

Georges, T.M. and Stephenson, J.J. : (1969) HF Radar Signatures of Traveling Ionospheric Irregularities, 3D Ray-tracing Simulation. Radio Science 4, 679

Jones, R.M. and J.J. Stephenson : (1975) A Versatile Three-Dimensional Ray Tracing Computer Program for Radio Waves in the Ionosphere. U.S. Dept. of Commerce, OT Report 75-76.

Langworthy, B.M. and T.B. Barrett : (1975) Implements and Techniques for the Analysis of Radio Propagation Trough in the Artic Ionosphere using Three-Dimensional Ray Tracing. AFCRL-TR-75-0319, Hanscom AFB, Mass.

Pierce, B.O. : (1929) A Short Table of Integrals. Boston: Ginn and Co.

Rush, C.M., and Elkins, T.J: (1975) An Assessment of the Magnitude of the F-Region Absorption. Telecommunications Journal 42, 476-488

Skolnik, M.I. : (1962) Introduction to Radar Systems. New York: McGraw-Hill

Stephenson, J.J. and Georges T.M.: (1969) Computer Routines for Synthesizing Ground Backscatter from Three-Dimensional Raysets. ESSA Tech. Report ERL120-ITS84

References (cont'd)

Vanguri, K-S. : (1973) A Three Dimensional Ray Tracing Computer Program - ARCON Version - I. ARCON Corp., Wakefield, Mass.

Wong, M.S. : (1975) Iterative Ray Tracing Simulation of Minimum Group-Path Traces in Swept-Frequency Backscatter Ionogram. AFCRL-TR-75-0241, Environmental Research Papers, No. 509.

PART II

USER'S GUIDES

NAME: E14994, revision 1, subroutine PML 127
CATEGORY: Electron Density Subroutine for Ray Tracing Program
TITLE: Mass Storage Electron Density
LANGUAGE: CDC extended Fortran - version 4
PROGRAMMER: B. M. Langworthy, Parke Mathematical Laboratories, Inc.
DATE: December 31, 1975

DESCRIPTION

The electron density subroutine, E14994, has been revised to accept mass storage data records directly from preprocessing program S2PPR (See PML 136) instead of reformating them to convert to mass storage. The mass storage record has been dimensioned in a more efficient manner and two mass storage records are needed per three-dimensional electron density block to circumvent a CDC system problem encountered in using many random access storage records. This subroutine can handle an arbitrarily large data base although the largest that has been used so far is 1.8 million words. This new version of E14994 when used in conjunction with program S2PPR eliminates the need for program RETAPE (PML 133).

Provisions for unequally spaced height entries have been made.

PRECEDING PAGE BLANK-NOT FILMED

INSTRUCTION SET

The use of the revised subroutine E14994 differs from the original only in that the random access mass storage is refered to as TAPE6. TAPE7 is no longer needed. Details of the structure of TAPE6 will be given under FILE DESCRIPTIONS.

STORAGE REQUIRED

This revision of subroutine E14994 uses more core storage than the original. This is mainly due to the increased size in the electron density array brought in for each block. This was seen as necessary from an overall efficiency of running standpoint. In the original configuration because of the required overlap between regions, only 40.5 percent of the array brought in represented new electron density values. In the present configuration this has been increased to 54.5 percent. This increased efficiency was needed since we are now dealing with arrays of a size that can fill working disk storage. The increase in core storage required is about 20,000 octal words.

ALGORITHM

The interpolation computation is the same as used in the original version except that provision has been made for varying the increment used in height. This means that in the height direction provision was made to use Lagrangian interpolation with unequally spaced data. This has caused considerable change to subroutine INTER3. Due to those changes and the fact that the existing equations in INTER3 have not been documented, the complete equation set will be given here.

From a three-dimensional table of plasma frequency, the value of the plasma frequency and its partial derivatives in each of the three dimensions is to be calculated. In two of these dimensions, θ and ϕ, data is given at evenly spaced intervals. In the height direction, h, the data is given unequally. The interpolation form used is Lagrangian. The plasma frequency is interpolated linearly in ϕ, and quadratic in h and θ. Derivatives are computed using finite differences of four points except that only three points are used near the boundary of an array. The derivatives are then interpolated linearly in the remaining two dimensions.

Definition of terms:

f_p	=	computed plasma frequency
f_H	=	table values of plasma frequency
h	=	height above the earth's surface
θ	=	dipolar colatitude
ϕ	=	dipolar longitude
j	=	index which references height i.e., $h_j \leq h < h_{j+1}$
k	=	index which references longitude i.e., $\phi_k \leq \phi, < \phi_{k+1}$
m	=	index which references colatitude i.e., $\theta_m \leq \theta \leq \theta_{m+1}$
J	=	number of entries in height
K	=	number of entries in longitude
L	=	number of entries in colatitude.
$\Delta\theta$	=	difference between equally spaced colatitude entries
$\Delta\phi$	=	difference between equally spaced longitude entries
f, f_1, f_2,		and partial derivatives are intermediate computational functions and are only temporarily defined in the equation.

Computation of the plasma frequency, f_p, at (h, θ, ϕ)

$$f_p(h,\theta,\phi) = \frac{1}{\Delta\phi}\left\{\frac{(h-h_j)(h-h_{j+1})}{(h_{j-1}-h_j)(h_{j-1}-h_{j+1})} f(h_{j-1},\theta,\phi) + \frac{(h-h_{j-1})(h-h_{j+1})}{(h_j-h_{j-1})(h_j-h_{j+1})} f(h_j,\theta,\phi)\right.$$

$$\left. + \frac{(h-h_{j-1})(h-h_j)}{(h_{j+1}-h_{j-1})(h_{j+1}-h_j)} f(h_{j+1},\theta,\phi)\right\}$$

where

$$f(h_j,\theta,\phi) = (\phi-\phi_k)[f_1(h_j,\theta,\phi_{k+1}) - f_1(h_j,\theta,\phi_k)] + \Delta\phi \cdot f_1(h_j,\theta,\phi_k)$$

and similarly for $f(h_{j-1},\theta,\phi)$ and $f(h_{j+1},\theta,\phi)$

where

$$f_1(h_j,\theta,\phi_k) = \frac{(\theta-\theta_m)(\theta-\theta_{m-1})}{2(\Delta\phi)^2} f_H(h_j,\theta_{m-1},\phi_k)$$

$$- \frac{(\theta-\theta_{m-1})(\theta-\theta_{m+1})}{(\Delta\phi)^2} f_H(h_j,\theta_m,\phi_k)$$

$$+ \frac{(\theta-\theta_{m-1})(\theta-\theta_m)}{2(\Delta\phi)^2} f_H(h_j,\theta_{m+1},\phi_k)$$

and similarly for $f_1(h_j,\theta,\phi_{k+1})$ and variations using h_{j-1} and h_{j+1}.

- 4 -

Computation of the partial derivative of plasma frequency with respect to colatitude at the point (r, θ, ϕ).

$$\frac{\partial f_p}{\partial \theta} = \frac{1}{2\,\phi(\Delta\theta)^2}\left[\frac{\partial f}{\partial \theta}(\theta,h_f,\phi) + \frac{h-h_j}{h_{j+1}-h_j}\left(\frac{\partial f}{\partial \theta}(\theta,h_{j+1},\phi) - \frac{\partial f}{\partial \theta}(\theta,h_j,\phi)\right)\right]$$

where

$$\frac{\partial f}{\partial \theta}(\theta,h_j,\phi) = \Delta\phi\,\frac{\partial f_1}{\partial \theta}(\theta,h_j,\phi_k) + (\phi-\phi_k)\left(\frac{\partial f_1}{\partial \theta}(\theta,h_j,\phi_{k+1}) - \frac{\partial f_1}{\partial \theta}(\theta,h_j,\phi_k)\right)$$

where

$$\frac{\partial f_1}{\partial \theta}(\theta,h_j,\phi_k) = \Delta\theta\,\frac{\partial f_2}{\partial \theta}(\theta_m,h_j,\phi_k) + (\theta-\theta_m)\left(\frac{\partial f_2}{\partial \theta}(\theta_{m-1},h_j,\phi_k) - \frac{\partial f_2}{\partial \theta}(\theta_m,h_j,\phi_k)\right)$$

$\frac{\partial f_1}{\partial \theta}$ is defined similarly for h_{j+1} and ϕ_{k+1}

where

$$\frac{\partial f_2}{\partial \theta}(\theta_m,h_j,\phi_k) = f_H(\theta_{m+1},h_j,\phi_k) - f_H(\theta_{m-1},h_j,\phi_k) \quad \text{if } m \geq 2$$

$$= -3f_H(\theta_1,h_j,\phi_k) + 4f_H(\theta_2,h_j,\phi_k) - f_H(\theta_3,h_j,\phi_k) \quad \text{if } m = 1$$

$$\frac{\partial f_2}{\partial \theta}(\theta_{m+1},h_j,\phi_k) = f_H(\theta_{m+2},h_j,\phi_k) - f_H(\theta_m,h_j,\phi_k) \quad \text{if } m < L-1$$

$$= f_H(\theta_{L-2},h_j,\phi_k) - 4f_H(\theta_{L-1},h_j,\phi_k) + 3f_H(\theta_L,h_j,\phi_k) \quad \text{if } m = L-1$$

$\frac{\partial f_2}{\partial \theta}$ defined similarly for h_{j+1} and ϕ_{k+1} .

This method amounts to establishing differences either side of the point and then doing a three dimensional linear interpolation.

Computation of the partial derivative of plasma frequency with respect to longitude at the point (r, θ, ϕ)

$$\frac{\partial f_p}{\partial \phi} = \frac{1}{2\Delta\theta(\Delta\phi)^2}\left[\frac{\partial f}{\partial \phi}(\theta,h_j,\phi) + \frac{h-h_j}{h_{j+1}-h_j}\left(\frac{\partial f}{\partial \phi}(\theta,h_{j+1},\phi) - \frac{\partial f}{\partial \phi}(\theta,h_j,\phi)\right)\right]$$

where

$$\frac{\partial f}{\partial \phi}(\theta,h_j,\phi) = \Delta\theta\,\frac{\partial f_1}{\partial \phi}(\theta_m,h_j,\phi) + (\theta-\theta_m)\left(\frac{\partial f_1}{\partial \phi}(\theta_{m+1},h_j,\phi) - \frac{\partial f_1}{\partial \phi}(\theta_m,h_j,\phi)\right)$$

$\frac{\partial f}{\partial \phi}$ is defined similarly for $\frac{\partial f}{\partial \phi}(\theta,h_{j+1},\phi)$

where

$$\frac{\partial f_1}{\partial \phi}(\theta_m,h_j,\phi) = \Delta\phi\,\frac{\partial f_2}{\partial \phi}(\theta_m,h_j,\phi_k) + (\phi-\phi_k)\left(\frac{\partial f_2}{\partial \phi}(\theta_m,h_j,\phi_{k+1}) - \frac{\partial f_2}{\partial \phi}(\theta_m,h_j,\phi_k)\right)$$

$\frac{\partial f_1}{\partial \phi}$ is defined similarly for θ_{m+1} and h_{j+1}

where

$$\frac{\partial f_2}{\partial \phi}(\theta_m,h_j,\phi_k) = f_H(\theta_m,h_j,\phi_{k+1}) - f_H(\theta_m,h_j,\phi_{k-1}) \quad \text{if } k \geq 2$$

$$= -3f_H(\theta_m,h_j,\phi_1) + 4f_H(\theta_m,h_j,\phi_2) - f_H(\theta_m,h_j,\phi_3) \quad \text{if } k = 1$$

$$\frac{\partial f_2}{\partial \phi}(\theta_m,h_j,\phi_{k+1}) = f_H(\theta_m,h_j,\phi_{k+2}) - f_H(\theta_m,h_j,\phi_k) \quad \text{if } k < K-1$$

$$= f_H(\theta_m,h_j,\phi_{K-2}) - 4f_H(\theta_m,h_j,\phi_{K-1}) + 3f_H(\theta_m,h_j,\phi_K) \quad \text{if } k = K-1$$

$\frac{\partial f_2}{\partial \phi}$ is defined similarly for h_{j+1} and θ_{m+1}.

This process is analogous to that followed for the partial derivative with respect to colatitude.

Computation of the partial derivative of plasma frequency with respect to height.

$$\frac{\partial f_p}{\partial h} = \frac{1}{\Delta\theta \cdot \Delta\phi} \left[\Delta\phi \frac{\partial f}{\partial h} (\theta, h, \phi_k) + (\phi - \phi_k) \frac{\partial f}{\partial h} (\theta, h, \phi_{k+1}) \right]$$

where

$$\frac{\partial f}{\partial h} (\theta, h, \phi_k) = \Delta\theta \frac{\partial f_1}{\partial h} (\theta_m, h, \phi_k) + (\theta - \theta_m) \left(\frac{\partial f_1}{\partial h} (\theta_{m+1}, h, \phi_k) - \frac{\partial f_1}{\partial h} (\theta_m, h, \phi_k) \right)$$

$\frac{\partial f}{\partial h}$ is defined similarly for $\frac{\partial f}{\partial h} (\theta, h, \phi_{k+1})$

where

$$\frac{\partial f_1}{\partial h} (\theta_m, h, \phi_k) = \frac{\partial f_2}{\partial h} (\theta_m, h_j, \phi_k) + \frac{h - h_j}{h_{j+1} - h_j} \left(\frac{\partial f_2}{\partial h} (\theta_m, h_{j+1}, \phi_k) - \frac{\partial f_2}{\partial h} (\theta_m, h_j, \phi_k) \right)$$

$\frac{\partial f_1}{\partial h}$ is defined similarly for θ_{m+1} and ϕ_{k+1}

where

$$\frac{\partial f_2}{\partial h} (\theta_m, h_j, \phi_k) = \frac{1}{h_{j+1} - h_{j-1}} \left(f_H(\theta_m, h_{j+1}, \phi_k) - f_H(\theta_m, h_{j-1}, \phi_k) \right) \quad \text{if } j \geq 2$$

$$= -\left(\frac{1}{h_j - h_{j-1}} + \frac{1}{h_{j+1} - h_{j-1}} \right) f_H(\theta_m, h_1, \phi_k)$$

$$+ \frac{h_{j+1} - h_{j-1}}{(h_{j+1} - h_j)(h_j - h_{j-1})} f_H(\theta_m, h_2, \phi_k)$$

$$- \frac{h_j - h_{j-1}}{(h_{j+1} - h_{j-1})(h_{j+1} - h_j)} f_H(\theta_m, h_3, \phi_k) \quad \text{if } j = 1$$

$$\frac{\partial f_2}{\partial h}(\theta_m, h_{j+1}, \phi_k) = \frac{1}{h_{j+2}-h_j}\left(f_H(\theta_m, h_{j+2}, \phi_k) - f_H(\theta_m, h_j, \phi_k)\right) \quad \text{if } j < J-1$$

$$= \frac{h_{j+1}-h_j}{(h_j-h_{j-1})(h_{j+1}-h_{j-1})} f_H(\theta_m, h_{J-2}, \phi_k) - \frac{h_{j+1}-h_{j-1}}{(h_{j+1}-h_j)(h_j-h_{j-1})} f_H(\theta_m, h_{J-1}, \phi_k)$$

$$+ \left(\frac{1}{h_{j+1}-h_j} + \frac{\quad}{h_{j+1}-h_{j-1}}\right) f_H(\theta_m, h_J, \phi_k) \quad \text{if } j = J-1$$

$\frac{\partial f_2}{\partial h}$ is defined similarly for θ_{m+1} and ϕ_{k+1}.

This is analogous to the previous partial derivatives except that provisions were made for unequally spaced height data.

SPECIAL CAUTIONS AND FEATURES

The total number of height entries must not exceed 200 entries. The total number of blocks of electron density data must not exceed 98 because of the dimension limit of 199 random access mass storage records. All values of θ and ϕ are assumed to be computational or dipolar coordinates.

TIMING

This revision of E14994 will take slightly less execution time in most cases due to the increased data block size. Since there are fewer colatitude entries per block, ray traced at azimuth angles heading either due north or south will take slightly longer. If the fineness of the grid size is increased the runs will take more time since the electron density data block will have to be changed more often.

ERROR MESSAGES

No error messages are included in either E14994 or INTER3; however, since E14994 is the only subroutine of RAYTRACEBL or RAYPLOT-2 to use mass storage "reads" and "writes", any system messages dealing with mass storage will refer to actions taken by E14994.

SUBROUTINES

INTER3 This subroutine with entry point INTER interpolates between table entries for values of electron density and its derivatives with respect to r, θ, and ϕ in the manner described in the ALGORITHM section.

ACCURACY

Runs on the new version of E14994 were verified against runs on the older version.

FILE DESCRIPTIONS

E14994 reads data from TAPE6 which is a mass storage file of up to 199 records. This file is created by the electron preprocessing program, S2PPR, as described in PML 136. The program never writes on TAPE6.

TAPE6 This file consists of up to 199 random access records of length ranging from 12 to 13223 words and containing the following information:

Record 1:

LATMX, LONMX, LHTMX, THZO, PHIZO, HTZO, DLTH, DLPH, DLHT, NAME, NLAT, NLON

- LATMX - The number of latitude entries.
- LONMX - The number of longitude entries.
- LHTMX - The number of height entries. ($\leq$ 200)
- THZO - The initial colatitude entry in radians. (This is the lowest colatitude for which data is available.)
- PHIZO - The initial longitude entry in radians.
- HTZO - The initial height entry in km.
- DLTH - The increment in colatitude entries in radians.
- DLPH - The increment in longitude entries in radians.
- DLHT - The increment in height entries in km. (Not used.)
- NAME - An alphanumeric identifier to be printed on output.
- NLAT - The number of data blocks required to cover the latitude values.
- NLON - The number of data blocks required to cover the longitude values.

NOTE: NLAT * NLON should not exceed 98.

Record 2: (HGT(I), I=1,200)

This represents the table of height values. 200 values are read in although only the first LHTMX of them represent legitimate height values.

Record 3: Not used or generated.

Record 4: (THETA(I), I=1,11), (PHI(J), J=1,12), (((FH(I,K,J), I=1,11), K=1,200), J=1,6) or a total of 13,223 words.

THETA - an array of colatitude values in radians.

PHI - an array of longitude values in radians.

FH - an array of plasma frequencies in MHz.

Record 5: (((FH(I,K,J), I=1,11), K=1,200), J=7,12) or 13200 words. This with record 4 represents the first data block.

Subsequent pairs of records such as records 6 and 7, 8 and 9, etc. will be repeats of records 4 and 5 for subsequent data blocks. TAPE6 can have a total of 98 blocks or up to 199 records. The total number of blocks available is NLAT*NLON. Blocks are numbered according to the following scheme.

Ascending ϕ ⟶

Ascending θ ↓

Block 1 Records 4 and 5	Overlap areas	Block NLAT + 1 Records 2*NLAT+4 and 2*NLAT+5	. . .	Block (NLON-1)*NLAT + 1 Records 2*(NLON-1)*NLAT+4 and 2*(NLON-1)*NLAT+5
Block 2 Records 6 and 7	Overlap areas	Block NLAT + 2 Records 2*NLAT+6 and 2*NLAT+7	. . .	Block (NLON-1)*NLAT + 2 Records 2*(NLON-1)*NLAT+6 and 2*(NLON-1)*NLAT+7
. . .	Overlap areas	. . .	. . .	. . .
Block NLAT Records 2*NLAT+2 and 2*NLAT+3		Block 2*NLAT Records 4*NLAT+2 and 4*NLAT+3	. . .	Block NLON*NLAT Records 2*NLON*NLAT+2 and 2*NLON*NLAT+3

DECK SET UP

This is the deck set up for running the raytracing program with E14994 and plotting the results using RAYPLOT-2 (PML 126).

```
BARRT,T300,CM100000.                                PROB#      NAME
ATTACH(TAPE6,VAR2DATAX3693818,ID=LANGW)
ATTACH(BMAIN,BMAINX3693818,ID=LANGW)
ATTACH(BINDEX,BINDEX3693818,ID=LANGW)
ATTACH(BSUBS,BSUBSX3693818,ID=LANGW)
REQUEST(TAPE3,*PF)
LDSET(PRESET=ZERO)
LOAD(BMAIN)
SLOAD(BINDEX,AHWFNC)
SLOAD(BSUBS,E14994,INTER3,ELECT1,DIPOLY)
EXECUTE(NITIAL)
CATALOG(TAPE3,PLOTTAPEX3693818,ID=LANGW,RP=999)
REWIND (TAPE3)
REWIND (TAPE6)
RETURN(BMAIN,BINDEX)
ATTACH(BPLT,BRAYPLT2X3693818,ID=LANGW)
ATTACH(PEN,ONLINEPEN)
LIBRARY(PEN)
LDSET(PRESET=ZERO)
LOAD(BPLT)
SLOAD(BSUBS,E14994,INTER3,ELECT1)
EXECUTE(RAYPLT2)
DISPOSE(PLOT,*OL)
EXIT.
DISPOSE(PLOT,*OL)
7/8/9
    Data for raytrace
7/8/9
    Data for plotting
6/7/8/9
```

NAME:	E14994, revision 2, subroutine PML 127
CATEGORY:	Electron Density Subroutine for Ray Tracing Program
TITLE:	Mass Storage Electron Density
LANGUAGE:	CDC Extended Fortran - version 4
PROGRAMMER:	B.M. Langworthy, Parke Mathematical Laboratories, Inc.
	Revision 2 by L. Calabi and I.B. Jarvis
DATE:	January 4, 1977

DESCRIPTION

This program has been modified to utilize the array for the ionospheric model which has been packed to 4 elements per computer word as described in the program write-up, S2PPR, revision 1. Only those elements required by the interpolation subroutine, INTER3, are unpacked by a new subroutine called FUNP4.

ALGORITHM

Given a packed word

$$z = z_1 \cdot 2^{45} + z_2 \cdot 2^{30} + z_3 \cdot 2^{15} + z_4$$

it is easy to obtain each z_i. Algebraically one could compute, for instance,

$z_4 = z - [z/2^{15}]*2^{15}$ where brackets indicate the integral part.

On the CDC6600 we have, however, two very fast operations which yield the z_i, in the order needed, in a loop: rotation and masking Thus, by rotating (left) z by 15 bits and masking all but the 15 right-most bits we obtain z_i. Repetition of the same sequence of operations yields z_2, z_3 and z_4 in sequence.

We then set

$y_1 = z_1/1000.$

$y_i = z_i/1000.$ if $z_i < 16384$ $\qquad i > 1$

$= (z_i - 16384)/1000.$ otherwise

and finally

$$x_1 = y_1$$

$$x_i = x_{i-1} + y_i \qquad i > 1$$

In order to determine which packed word z = KOMP(L,I,K) to use, remember that KOMP(L,I,K) has been obtained by packing FH(L, 4I +1, K), FH(L, 4I + 2, K), FH(L, 4I +3, K) and FH(L, 4I + 4, K).

The subroutine FUNP4, when called with arguments LL,JJ,KK, does not unpack the whole array KOMP but only a few elements.

When an interpolation is required around elements whose indices are LL,JJ,KK, at most 4 values of each index (1 before and 2 after) would be required for the interpolation. However since 4 values of the JJ index are stored per computer word, it might be necessary to unpack 2 of these words for each value of the other 2 indices to obtain the required values on either side of the JJ index.

Let JJ_0 = integral part of $\frac{JJ-2}{4} + 1$

and JJ_1 = integral part of $\frac{JJ+1}{4} + 1$.

Let LM = LL-2 unless LL < 2, then LM = 0

KM = KK-2 unless KK < 2, then KM = 0.

Then we unpack only KOMP (L,I,K) with

$$LM + 1 \leq L \leq LM + 4$$

$$JJ_0 \leq I \leq JJ_1$$

$$KM + 1 \leq K \leq KM + 4 ,$$

storing the unpacked values in an array GH of dimensions (4,8,4). If we set

$$JM = (\text{integral part of } \frac{JJ-2}{4}) * 4$$

then

$$GH(A,B,C) = FH(A-LM,B-JM,C-KM)$$

in the interpolation subroutine. All references to the old FH array are changed to GH and the indices L, J and K must be changed to L-LM, J-JM and K-KM.

The only other modifications made to this program are:

1) Remove the FH array from COMMON/DENX/ and add the GH array, in subroutines E14994 and INTER3.

2) Change the number of words to read from mass storage in E14994 to 6623.

3) CALL FUNP(LL,JJ,KK) before any interpolation is carried out, even when it is not necessary to read a new block of mass storage.

STORAGE REQUIRED

The total storage required for this revision used with the ray trace program is 72000 octal words of core storage.

TIMING

CP time for rays traced for 2 hops in the test run varied from 1.9 seconds to 6.2 seconds.

Because of the time required for unpacking, CPA time is slightly greater in the modified program. However, since the information read from mass storage is packed into 1/4 as many words, IO time is considerably shortened. Total run-time was reduced to 52.5 %. Storage requirements are decreased from 142,000 octal to 72,000 octal. With both time and space reduced the total cost of the runs decreased to 37.1 %. The following table shows the actual values.

	Original Program		Revision 1	
CPA	74.659 sec.	$1.530	103.141 sec.	$2.114
IO	241.885 sec.	1.620	63.201 sec.	.423
CM	13927.915 KWS	15.320	3917.858 KWS	4.309
Cost of job		$18.471		$6.847

NAME: S2PPR, revision 0, program, PML 136
CATEGORY: Preprocessing program for a ray-trace ionospheric model
TITLE: Sine Square Preprocessing Program
LANGUAGE: CDC Extended Fortran - Version 4
PROGRAMMER: B. M. Langworthy, Parke Mathematical Laboratories, Inc.
DATE: December 31, 1975

DESCRIPTION

This program is designed to replace programs SMOD2 and COMBSNC written by ARCON Corp. in 1972 and to extend the capabilities in modeling an arctic ionosphere usable by the AFCRL ionospheric ray-trace program. S2PPR generates a three-dimensional array of plasma frequencies for an arbitrarily defined range of heights, dipolar colatitudes, and dipolar longitudes. This array is generated from a table of specifications of profiles composed of up to 5 sine square segments with a Chapman layer composing the upper ionosphere. The sine square segments are specified by a plasma frequency and the height at which this frequency occurs. Profiles can be specified for up to 25 values of colatitude in the dipolar magnetic, accurate geomagnetic, or geographic coordinate system. Variations with longitude are then given by tables of additive and/or multiplicative constants on an 8 x 8 grid in colatitude and longitude. In addition the same data may be given for a perturbation model to be superimposed on the existing sine square model. Transition between tabular values is also taken to be sine square in nature.

The size of the three-dimensional array of plasma frequencies is no longer limited by the amount of core storage available but can be extended up to 4 million points since the program has been adapted to generate random access permanent file data in blocks of a size that can be handled by the ray-trace program in a more modest core area than the previously mentioned programs. Height specifications need not be at fixed intervals over the entire array, but may be varied so that three unique height increments may be used on any given vertical. This feature allows a finer height increment in the auroral E layer.

PRECEDING PAGE BLANK-NOT FILMED

INSTRUCTION SET

To use S2PPR the user must specify the size of the three-dimensional plasma frequency array to be generated. Sample input is given in Attachment 1 to correspond to the description given below. This information is supplied on the first two data cards which are read. The first card is read with the format (3I5,A5,6F10.3) and contains the following information:

Card 1

Cols 1 - 5 NLONMX - The number of values of dipolar geomagnetic longitude desired. (I5)

Cols 6 - 10 NLATMX - The number of values of dipolar geomagnetic colatitude desired. (I5)

Cols 11 - 15 NHTMX - The number of values of height desired. (I5)

Cols 16 - 20 NAME - A 5 character name by which the data set is to be known. (A5) This name will eventually appear on ray-trace printout to specify the electron density model used.

Cols 21 - 30 PHZO - Tne starting value of dipolar geomagnetic longitude in degrees. (F10.3)

Cols 31 - 40 THZO - The starting value of dipolar geomagnetic colatitude in degrees. (F10.3)

Cols 41 - 50 HTZO - The starting value of height above the earth's surface in km. (F10.3) This value is no longer used since the feature of variable height increments has been added.

Cols 51 - 60 DLPH - The increment in dipolar geomagnetic longitude in degrees. (F10.3)

Cols 61 - 70 DLTH - The increment in dipolar geomagnetic colatitude in degrees. (F10.3)

Cols 71 - 80 DLHT - The increment in height in km. (F10.3) This value is no longer used since the feature of variable height increments has been added.

Card 2 This card is for data specifying the variable height increments and is read with the format (6F6.2).

Cols 1 - 6	HBOT(1)	- The starting value of height in km.
Cols 7 - 12	HINC(1)	- The increment in height in km. to be used above HBOT(1). (F6.2)
Cols 13 - 18	HBOT(2)	- The value of height at which the height increment is to be changed in km. (F6.2)
Cols 19 - 24	HINC(2)	- The increment in height in km. to be used above HBOT(2). (F6.2)
Cols 25 - 30	HBOT(3)	- A quantity analogous to HBOT(2).
Cols 31 - 36	HINC(3)	- A quantity analogous to HINC(2).

Next the user must specify the model of the ionosphere to be used in generating the array. This is done by entering tables of values for colatitude versus plasma frequency or altitude. Separate tables are entered for the variation of plasma frequency of a certain layer and the altitude at which that plasma frequency occurs. Tables are also entered for h_0, the bottom of the ionosphere, and h_{max}, the upper limit of the ionosphere. This is the "background" model for all longitudes which may then be modified to give a variation in longitude by additional input. The first card needed in this effort is a card telling the number of sine square segments to be used and the coordinate system in which the data is given.

Card 3

Cols 1 - 6	NSIN	- The number of sine square segments to be used to model the area below the F-layer maximum. This number must be between 1 and 5. (I6)
Cols 7 - 12	NTHET	- This number specifies the coordinate system in which the input tables are given. If NTHET = 1, data is in the dipolar geomagnetic system = 0, data is in the accurate geomagnetic system = -1, data is in the geographic system.

The layer parameters themselves are then supplied by tables composed of a card which tells the number of table entries followed by cards containing the table entries themselves. Tables are entered in the order of: plasma frequency for the 1st sine square layer, plasma frequency for the 2nd sine square layer, etc.; then height for the base of the 1st sine square layer, height for the top of the 1st sine square layer, height for the top of the 2nd sine square layer, etc; and finally height beyond which the model is undefined. Plasma frequencies are given in MHz and heights are given in km. Card sets for each layer parameter have the following format where I represents the Ith layer parameter.

1st card in set

Cols 1 - 6 LATMAX(I) - The number of table entries used to specify the Ith layer parameter (I6). Must be no larger than 25.

2nd card in set

Cols 1 - 6 THETA(1,I)- The 1st colatitude entry in degrees for the Ith layer parameter table. (F6.2)

Cols 7 - 12 FH(1,I) - The value of plasma frequency or height at the previous colatitude. (F6.2)

Cols 13 - 18 THETA(2,I)- The 2nd colatitude entry in degrees for the Ith layer parameter table. (F6.2)

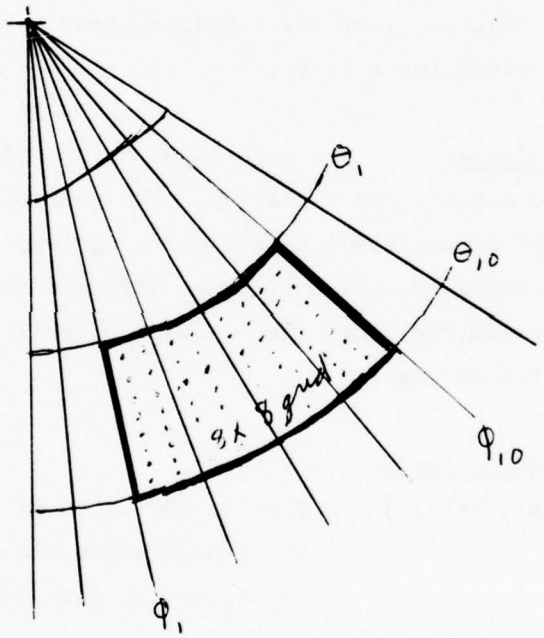
Cols 19 - 24 FH(2,I) - The value of plasma frequency or height at the previous colatitude. (F6.2)

These entries are repeated up to 6 entries per card for as many cards as are required for LATMAX(I) entries.

NOTE: Be sure to include a decimal point in all numbers except those with I formats.

A note on the values which can be used for FH and H: A zero value may be given to any plasma frequency and two consecutive layers may have the same value (which effectively gives a constant layer), but no values of height except H_o may be zero and consecutive layer height values must have an increase over the next lower layer although the increase may be small. Values given in the tables represent the ends of a sine square curve which will have a zero derivative.

This tabular data can then be varied with longitude using an 8 x 8 grid of additive modifying values and/or 8 multiplicative modifying values for each quantity to be varied. Interpolation between modifying values is sine square in nature. While there are only 8 entries in the addition and multiplication tables, 10 values of colatitude and longitude must be specified to give the boundaries of the modified area where additive values become zero and multiplicative values become 1. The figure below illustrates a modification region.



The first card to be inputed for the longitudinal variation is a signal card to tell which quantities are to be varied.

Next Card This is a signal card to tell which layer quantities are to be varied with longitude and what kind of a variation this is to be.

Cols 1 - 60 (NON(I), I=1,2*NSIN+2) where values are read in the format (12I5) or a layer signal every 5 columns, right justified.
If NON(I) = 0, no Ø variation is to be given for the Ith entry.
= 1, the Ø variation is to be given by an additive 8 x 8 grid for the Ith entry.

= 2, the $\emptyset$ variation is to be given by an additive 8 x 8 grid and 8 multiplicative values for the Ith entry.

= 3, the $\emptyset$ variation is to be given by 8 multiplicative values for the Ith entry.

The signals are entered in the order in which input is given in the original table that is: f_1, f_2, ..., f_{NSIN}, h_0, h_1, h_2, ..., h_{NSIN}, h_{MAX} where NSIN is the total number of sine square segments to be used. For each non zero entry of NON a set of layer modification cards must follow. These sets must be in the order of their signals, i.e. layer modification cards for f_{NSIN} comes before layer modification cards for h_0, etc.

Layer Modification Cards For each layer to be modified, a set of cards must be included to specify the variation. The variation may be made by adding a quantity (positive or negative) to the existing layer value, by multiplying the existing layer value by a factor, or by adding a quantity to the existing layer value and then multiplying. The details of this operation are given in the ALGORITHM section on page 9.

Layer modification card 1 (10F8.0)

(PHENT(K,I), K=1,10) - This is the table of $\emptyset$ values in degrees for the Ith additive and multiplicative variation. The first and last values represent the boundaries where the additive variation returns to a value of zero and the multiplicative variation returns to a value of one. All ten entries must be supplied.

Layer modification card 2 (10F8.0) Used only if an additive variation is desired i.e. NON(I) = 1 or 2.

(THENT(L,I), L=1,10) - This is the table of θ values in degrees for the Ith additive variation. As above, the first and last values represent the boundaries where the additive variations returns to a value of zero. This card is supplied only if an additive variation is desired since the multiplicative variation has no θ dependence. All ten entries must be supplied.

Layer modification card 3 (8F10.0) Only for additive variation. Omit if NON(I) = 3.

(AD(L,2,I), L=2,9) - This is the additive variation for the second Ø entry and the 2nd through the 9th Θ entries. (It is assumed to be zero for L = 1 and L = 10.) Entries will be in MHz or km depending on whether the Ith quantity is a plasma frequency or a height. All eight values must be supplied.

Next 7 layer modification cards (8F10.0) Only for additive variation. Omit if NON(I) = 3.

(AD(L,K,I), L=2,9) - This is the same additive variation as above but for the Kth Ø entry. K varies from 3 to 9 in the 7 cards. All 8 of these cards must be supplied to give the 8 x 8 additive variation.

Layer modification card 11 (8F10.0) Only for the multiplicative variation. Omit if NON(I) = 1.

(FM(K,I), K=2,9) - FM is the multiplicative factor for the Ith layer. The 8 entries correspond to the Ø values for K = 2,9. The multiplicative factor is assumed to be one for K = 1 and K = 10. The quantity FM is dimensionless. All eight values must be supplied.

A set of layer modification cards must be supplied for each non zero NON. To summarize:

If NON =	1	2	3
the following cards must be supplied	Ø variation (card 1) Θ variation (card 2) 8 Additive cards (Cards 3-10)	Ø variation (card 1) Θ variation (card 2) 8 Additive cards (Cards 3-10) Multiplicative card (Card 11)	Ø variation (card 1) Multiplicative card (card 11)

Interpolation between all layer modification table entries will be sine square.

Perturbation Model Input

Input for the perturbation is identical to the input required for the original model. The first card is analogous to Card 3.

Cols 1 - 6 NSIN2 - The number of sine square segments to be used to model the perturbation below the F-layer maximum. This number must be between 1 and 5. (I6)

NOTE: There is no NTHET entry here as there is in Card 3. The perturbation must be given in the same coordinate system as the original model.

The tables of colatitude versus plasma frequency or height for each of the 2*NSIN2 + 2 layer parameters for the perturbation model are then entered in the same manner as the original model. For each table the first card gives the number of entries - LATMX2(I) for the Ith layer parameter. This is then followed by the table itself with up to 6 entries per card. The order in which the tables are entered is: f_1, f_2, f_3, ..., f_{NSIN2}, h_0, h_1, h_2, h_3, ..., h_{NSIN2}, h_{MAX}.

These tables must then be followed by a signal card to tell which parameters of the perturbation model are to be given a $\emptyset$ variation and if it is to be additive, multiplicative, or both. Those parameters to be modified must then have a set of layer modification cards supplied as described on pages 6-7.

The input deck will then be composed of the following cards:

Card 1 which specifies the desired data output file.
Card 2 which gives the height variation for the output array.
Card 3 which gives NSIN, the number of sine square segments and the coordinate system of the model to be inputted.
2*NSIN + 2 card sets which give the number of table entries and the tables for each layer parameter.
A signal card to tell which model parameters are to be modified.
Layer modification cards.
A card which gives NSIN2, the number of sine square segments in the perturbation model.
2*NSIN2+2 card sets which give the number of table entries and the tables for each layer parameter of the perturbation model.
A signal card to tell which perturbation model parameters are to be modified.
Layer modification cards for the perturbation.

Output The printed output from program S2PPR consists of a listing of the table values inputted to the program for the model and the perturbation and the grid parameters of the three-dimensional output array. A message is written to indicate that the coefficients for the transformation from geographic to accurate geomagnetic coordinates has been read in. The values for the outputted height array are listed so that subsequent plasma frequency data can be located for a particular height. Some printout on the transformation from dipolar to accurate geomagnetic coordinates is given for checking purposes, but printout would become too lengthy if all transformations were given. The plasma frequency profile with height is given for occasional values of θ and ϕ. The θ and ϕ values listed with these profiles is dipolar. Plasma frequency is given in MHz.

Any other output found is probably an error message and is described on page 13. Sample printed output is given in Attachment 2.

The principal output of S2PPR is the plasma frequency data file for use by electron density subroutine E14994 of the ray-trace program. (See PML 127, revision 1). The data file which is referred to as TAPE2 is a random access mass storage file and is described under FILE DESCRIPTIONS.

STORAGE REQUIRED

The storage required for S2PPR is 200000 octal words of core storage. Almost half of this storage is needed to store one data block.

ALGORITHM

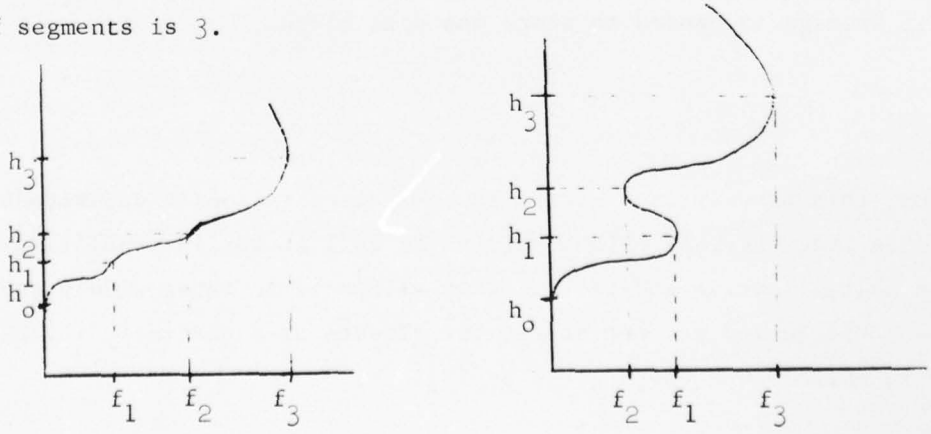
Unlike S2MOD, this program uses a sine square interpolation in colatitude between plasma frequency and height entries as well as for interpolating between additive and multiplicative modifiers. This allows us to enter widely separated values of colatitude and yet not suffer the effects of a quadratic interpolation. The manner in which S2PPR operates is as follows:

A particular grid point (θ,ϕ) is given in dipolar coordinates. This is then transformed to accurate geomagnetic coordinates or geographic coordinates or it is left in dipolar coordinates depending on which system the original model is specified in. The transformed coordinates are then used to make a sine square interpolation on the values of f_1, f_2, f_3, etc., and the corresponding height values. These values are then used to generate plasma frequency for all heights at this particular (θ,ϕ) point. The next (θ,ϕ) point is then transformed and the same operation is repeated. When the plasma frequencies for all height and colatitudes have been generated for a particular dipolar longitude, this two-dimensional array is then written out and the next value of longitude is dealt with. The flow chart in figure 1 explains the sequence of the interpolations.

The conversion to accurate geomagnetic coordinates is that written by J. Dryden, Geophysical Institute, Univ. of Alaska, Oct. 1970.

The height profile generated for a particular (θ,ϕ) value is composed of sine square segments below the F2 layer and a Chapman layer above this point. Since the equations used therein have not been documented, we have included them here for completeness.

The plasma frequency profile is composed of up to 5 sine square segments the last of which is assumed to be the F2 layer. These segments may cause an increase or a decrease in plasma frequency as shown in the following figure where the number of segments is 3.



In the following equations the total number of sine square segments is NSIN. As noted above, the values for f_1, f_2, ..., h_0, h_1, h_2, etc. are values that have already been interpolated for particular values of (θ,ϕ) in dipolar coordinates, h indicates the present value of the height above the earth's surface.

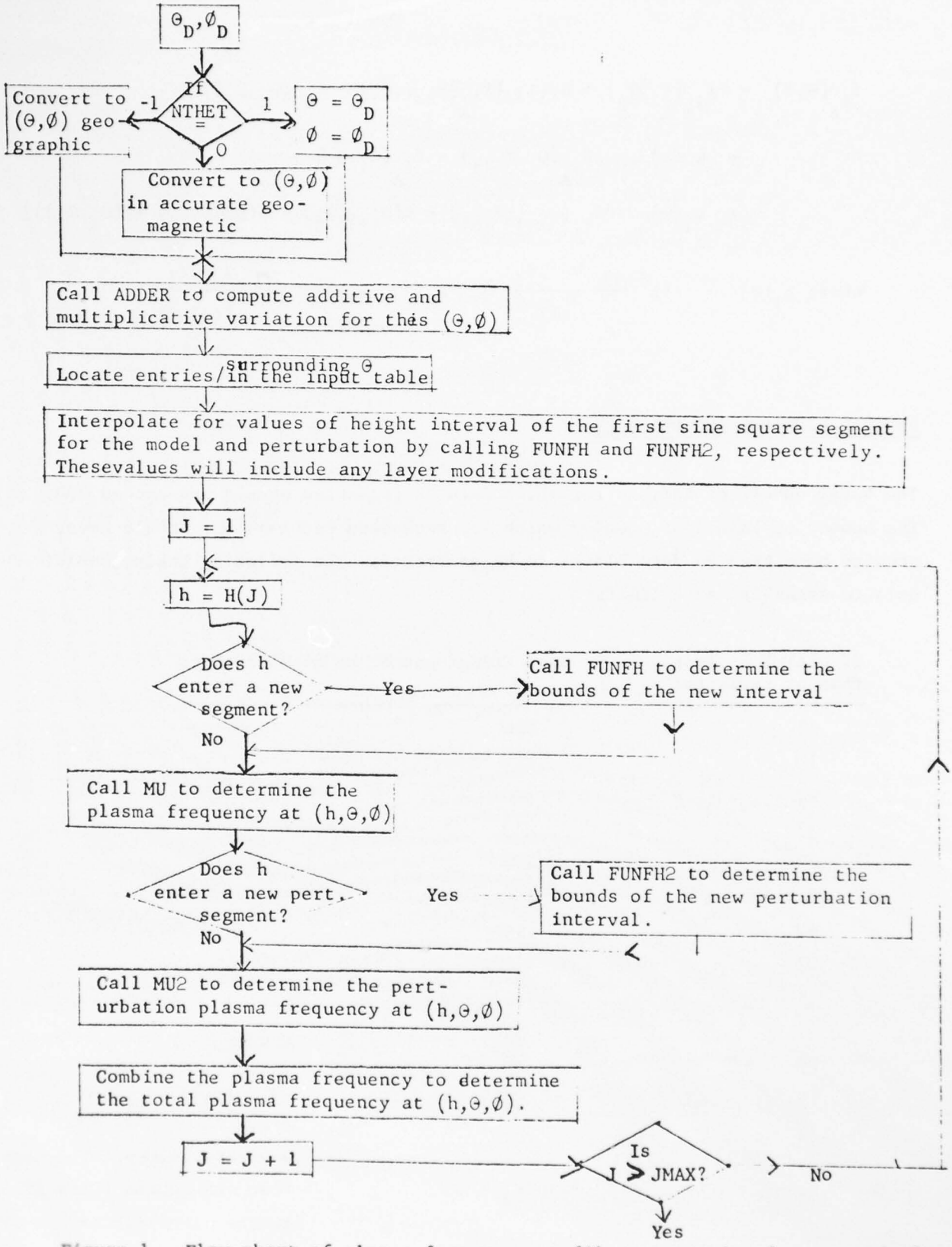


Figure 1. Flow chart of plasma frqquency profile computation for a particular point $(\Theta_D,\emptyset_D)$ in geomagnetic dipolar coordinates.

$$f_{1_A}(\theta,\phi) = f_{1_A}(\theta_i,\phi_j) + s_i(\theta).(f_{1_A}(\theta_{i+1},\phi_j) - f_{1_A}(\theta_i,\phi_j))$$
$$+ s_2(\phi) \cdot [f_{1_A}(\theta_i,\phi_{j+1}) - f_{1_A}(\theta_i,\phi_j)$$
$$+ s_1(\theta) \cdot (f_{1_A}(\theta_{i+1},\phi_{j+1}) - f(\theta_{i+1},\phi_j) - f(\theta_i,\phi_{j+1}) + f(\theta_i,\phi_j))]$$

where $s_1(\theta) = \sin^2(\frac{\pi}{2} \frac{\theta - \theta_i}{\theta_{i+1} - \theta_i})$, $s_2(\phi) = \sin^2(\frac{\pi}{2} \frac{\phi - \phi_j}{\phi_{j+1} - \phi_j})$.

SPECIAL CAUTIONS AND FEATURES

The total number of heights for which data is requested should not exceed 200. The number of values of θ and ϕ which are requested can vary but should never require more than 99 data blocks to be generated. The following table should help to establish such limits:

If NLATMX is less than or equal to:	then NLONMX cannot be greater than:
11	894
19	444
27	300
35	219
43	174
51	156
59	129
67	111
75	102
91	84
99	75
115	66
131	57
155	48
195	39
267	30
355	21
795	12

NOTE: The program will not generate values for latitudes that are beyond those given in the input tables but will stop and print an error message.

TIMING

The timing of S2PPR varies with the size of the output array desired. For 18 data blocks the execution time was 101 seconds or approximately 5 1/2 seconds per data block.

ERROR MESSAGES

PARAMETERS OUTSIDE LIMITS, NUMBER OF SINE SQUARE CURVES = XXX XXX

This indicates the number of sine square segments specified for either the model or the perturbation is not between 1 and 5. The first number given is for the model and the second is for the perturbation. Check to see that these numbers are punched in the correct columns or that the input deck is set up in the correct order. This will cause a program stop.

LAYER XX HAS XXX ENTRIES. MODEL = XXX PERTURB. = XXX.

This indicates that the number of table entries from the Ith layer of either the model or the perturbation is not between 1 and 25. The additional print out allows one to tell if it is the model or the perturbation which is out of range. Check to see that these numbers are punched in the correct columns or that the input deck is in the correct order. This will cause a program stop.

ANGLES OUTSIDE RANGE OF IONOSPHERE

THETA = xxx.x PHI = xxx.x INDICES ARE xxx xxx

This message is printed out when the θ value of the model or the perturbation is not in the given table range. The values of θ and ϕ given here will have been converted from the dipolar system to the system of the input tables. The two indices given are for the block index of the output array -- the first indicating the 1st, 2nd, etc. block in the θ direction and the second indicating the 1st, 2nd, etc. block in the ϕ direction.

HEIGHT LAYER INDEX JJ = xxx HL = xxx.xxx HMAX = xxx.xxx

This message will be printed out twice with the first information from the model and the second set of information from the perturbation. JJ indicates the height layer index, HL indicates the height for which information is being sought, and HMAX indicates the maximum height for which the model exists at this particular value of (θ,ϕ). This message will be printed out whenever information is sought outside of the height values for which plasma frequency information is available. If this larger height is desired, simply increase the value of HMAX.

ERROR LATITUDE,LONGITUDE xxx.xxx xxx.xxx

This is an error message from the subroutine which converts geographic coordinates to accurate geomagnetic coordinates. The message appears if the inputted latitude is not between $+90^{o}$ and -90^{o} or if the longitude is less than zero. The first quantity listed is the latitude and second quantity listed is the longitude. This message should not appear since subroutine DICOORD should check for this before CORRGM is called. The subroutine will try to correct the longitude by adding 360^{o} but will proceed anyway.

NO TRIANGLE FOR XLAT,XLONG = xxx.xxx xxx.xxx

This may occur when the point for which the accurate geomagnetic coordinates are desired is in the region of the accurate magnetic pole. The region is divided into triangles having the pole as one vertex. If a triangle cannot be located, it is assumed that this is the accurate magnetic pole and the above message will appear giving the coordinates in the geographic system in degrees. No value is assigned to the accurate geomagnetic longitude in this case. Computation then proceeds as normal.

Any error messages referring to random access mass storage will refer to the WRITMS calls given in S2PPR.

SUBROUTINES

S2PPR — This is the main program. It reads in all data cards, sets up the θ, and ϕ values for the array and sequences through h,θ, and ϕ to generate the output data block. All mass storage writes take place in S2PPR. S2PPR calls ADDER, CORRGM2, FUNFH, FUNFH2, MTOG, MU, and MU2.

ADDER — Given a particular value for θ and ϕ, this subroutine will determine any quantities that should be added or multiplied by the original model to give a ϕ variation. Corrections to all layer specification e.g. f_1, f_2, h_0, h_1, etc. are determined for the (θ,ϕ) point inputted so that adder is only called once for a profile. ADDER is called by S2PPR.

FUNFH — This is a function which gives the necessary layer parameters when given the height and colatitude indices. This function is used by S2PPR and MU.

MU — This subroutine determines the value of the plasma frequency using the sine square segments or the Chapman profile for a particular point (r, θ, ϕ) given the layer parameters as a function of θ and ϕ. This subroutine is called by S2PPR and uses function FUNFH.

DICOORD — This subroutine through entry point MTOG converts a point in colatitude and longitude from dipolar geomagnetic to geographic coordinates. This subroutine is called by S2PPR.

CORRGM2 — This subroutine takes points in the geographic coordinate system and converts them to accurate geomagnetic coordinates. A call to CORRGM2 causes the transforming array to be read in. For points to be converted, a call must be made to CORRGM. This subroutine is called by S2PPR.

FUNFH2 — This is a function which gives the necessary layer parameters for the perturbation when given the height and colatitude indices. This function is used by S2PPR and MU2.

MU2 — This subroutine determines the value of the perturbation plasma frequency using sine square segments or the Chapman profile for a particular point (r,θ,ϕ) given the layer parameters as a function of θ and ϕ. This subroutine is called by S2PPR and uses function FUNFH2.

ACCURACY

The question of accuracy here applies mainly to the way in which the data will be used. It should be kept in mind in selecting the grid parameters for the data file that interpolation on this data will be quadratic in height and colatitude and linear in longitude.

COMMENTS ON USAGE

The data file created by S2PPR will typically be very large in terms of Permanent Record Units (PRUS). Files created so far have ranged from 7000 to 40,000 PRUS. To keep these permanently in the system permission must be obtained from Mr. Gosselin and they must be accessed frequently. This means that large files should be created only when extensive production work is to be done. If only one run is desired on a particular data set, the file should be created in that run and not made permanent.

FILE DESCRIPTIONS

The program card for this program is as follows:

PROGRAM S2PPR(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE2,TAPE1)

INPUT and TAPE5 — This is the system input file under normal operation. It consists of the cards described on pages 2 - 8.

OUTPUT TAPE6 — This is the system output file under normal operation. It contains the information described on page 9 as printed output plus any error messages from the program or the system.

TAPE1 — This is the permanent file which contains the transformation array for going from geographic to accurate geomagnetic coordinates. It is stored in both systems I and II under the file name MJDATA with ID=LOGICON.

TAPE2 — This is the major output of S2PPR, a mass storage file to be used by the electron density subroutine of the ray trace program. (See PML127, Revision 1.)

This file consists of up to 30 random access records of length ranging from 12 to 8823 words and containing the following information:

Record 1:

LATMX, LONMX, LHTMX, THZO, PHIZO, HTZO, DLTH, DLPH, DLHT, NAME, NLAT, NLON

- LATMX - The number of latitude entries.
- LONMX - The number of longitude entries.
- LHTMX - The number of height entries. ($\leq$ 200)
- THZO - The initial colatitude entry in radians. (This is the lowest colatitude for which data is available.)
- PHIZO - The initial longitude entry in radians.
- HTZO - The initial height entry in km.
- DLTH - The increment in colatitude entries in radians.
- DLPH - The increment in longitude entries in radians.
- DLHT - The increment in height entries in km. (Not used.)
- NAME - An alphanumeric identifier to be printed on output.
- NLAT - The number of data blocks required to cover the latitude values.

NLON - The number of data blocks required to cover the longitude values.

NOTE: NLAT * NLON should not exceed 98.

Record 2: (HGT(I), I=1,200)
This represents the table of height values. 200 values are read in although only the first LHTMX of them represent legitimate height values.

Record 3: (THETA(I), I=1,11),(PHI(J), J=1,12),
(((FH(I,K,J),I=1,11),K=1,200),J=1,4) or a total of 8823 words.
THETA - An array of colatitude values in radians.
PHI - An array of longitude values in radians.
FH - An array of plasma frequencies in MHz.

Record 4: (((FH(I,K,J), I=1,11),K=1,200),J=5,8) or 8800 words.

Record 5: (((FH(I,K,J), I=1,11),K=1,200),J=9,12) or 8800 words.

Records 3, 4, and 5 represent the first data block.

Subsequent sets of 3 records such as 6,7, and 8, 9,10, and 11, etc. will be repeats of records 3,4, and 5 for subsequent data blocks. TAPE6 can have a total of 98 blocks or up to 299 records. The total number of blocks available is NLAT * NLON. Blocks are numbered according to the following scheme.

Ascending ∅ ⟶

Ascending θ ⟶

Block 1 Records 3, 4, and 5	Overlap areas	Block NLAT + 1 Records 3*NLAT+3 3*NLAT+4 and 3*NLAT+5	. . .	Block (NLON-1)*NLAT + 1 Records 3*(NLON-1)*NLAT+3 3*(NLON-1)*NLAT+4 and 3*(NLON-1)*NLAT+5
Block 2 Records 6, 7, and 8	Overlap areas	Block NLAT + 2 Records 3*NLAT+6 3*NLAT+7 and 3*NLAT+8	. . .	Block (NLON-1)*NLAT + 2 Records 3*(NLON-1)*NLAT+6 3*(NLON-1)*NLAT+7 and 3*(NLON-1)*NLAT+8
.	Overlap areas	.	.	.
.		.	.	.
.		.	.	.
Block NLAT Records 3*NLAT 3*NLAT+1 and 3*NLAT+2		Block 2*NLAT Records 6*NLAT 6*NLAT+1 and 6*NLAT+2	. . .	Block NLON*NLAT Records 3*NLON*NLAT 3*NLON*NLAT+1 and 3*NLON*NLAT+2

SAMPLE DECK SETUP

```
Job Card with CM200000.
ATTACH(SOURCE,S2PPRX3693818,ID=LANGW)
FTN(I=SOURCE,B=BS2,PL=20000)
ATTACH(TAPE1,MJDATA,ID=LOGICON)
REQUEST(TAPE2,*PF)
BS2.
CATALOG(TAPE2,VAR2DATAX3693818,ID=LANGW,RP=999)
7/8/9
   Data cards for S2PPR
6/7/8/9
```

Attachment 1. Sample Input for S2PPR

Grid lines appear every 5 columns.

Output Grid Specification - Card 1

NLONMX	NLATMX	NHTMX	NAME	PHIZO	THZO	HTZO	DLPH	DLTH	DLHT
51	71	199	MAP4	-3.	14.	80.	1.50	.2	2.

This card gives the number of entries in longitude, colatitude, and height, their starting values, and their increments. NAME is the 5 character indentifier of the data set.

Variable Height Increment Specification - Card 2

HBOT(1)	HINC(1)	HBOT(2)	HINC(2)	HBOT(3)	HINC(3)
80.	4.	112.	1.	144.	2.

Model Specification - Card 3

NSIN	NTHET
5	0

NSIN tells the number of sine square segments which compose the lower ionosphere and NTHET which is zero indicates that the tables to follow are in accurate geomagnetic coordinates.

Plasma Frequency Model - Next 2*NSIN + 2 sets of cards.

2												Specification of F_1 - 2 entries
.5	.0001	50.	.0001									
2												Specification of F_2 - 2 entries
.5	0	50.	0.									
2												Specification of F_3 - 2 entries
.5	.0001	50.	.0001									
6												Specification of F_4 - 6 entries
.5	0.	17.2	0.	17.9	2.45	24.7	2.45	25.4	0.	50.	0.	
12												Specification of F_5 - 12 entries
.5	4.2	10.9	4.2	11.5	5.1	16.9	5.1	17.6	4.2	25.0	4.2	
26.8	2.2	33.4	2.2	34.1	1.6	34.9	1.5	35.6	2.2	50.0	2.2	
2												Specification of H_o - 2 entries
.5	81.	50.	81.									
2												Specification of H_1 - 2 entries
.5	82.	50.	82.									
2												Specification of H_2 - 2 entries
.5	83.	50.	83.									
2												Specification of H_3 - 2 entries
.5	84.	50.	84.									
13												Specification of H_4 - 13 entries
.5	150.	10.4	168.	11.6	108.	16.7	108.	17.4	130.	25.0	130.	
26.8	270.	28.0	240.	33.4	240.	34.1	250.	34.9	250.	35.6	240.	
30.0	240.											

14				← Specification of H_5 - 14 entries							
.5	340.	10.9	340.	11.5	380.	16.9	380.	17.6	356.	24.9	356.
26.0	460.	25.8	460.	28.0	360.	33.4	360.	34.1	440.	34.9	440.
35.6	360.	50.0	360.								
2				← Specification of H_{MAX} - 2 entries							
.5	501.	50.	501.								

Layer Modification Signal Card

Signal for:

f_1	f_2	f_3	f_4	f_5	h_o	h_1	h_2	h_3	h_4	h_5	h_{max}
0	0	0	0	1	0	0	0	0	2	0	0

This card indicates that f_5 is to have an additive variation and that h_4 is to have an additive and a multiplicative variation.

f_5 Additive Variation

Longitude values:

0.	5.	10.	15.	20.	25.	30.	35.	40.	45.

Colatitude values:

25.5	26.0	26.5	27.0	27.5	28.0	28.5	29.0	29.5	30.0

Plasma Frequency Values for Addition - 8 Cards

-0.	1.	1.	-0.	-1.	1.	1.	0.	
0.	1.	0.	-1.	-1.	0.	0.	0.	
0.	0.	-0.	0.	-0.	-0.	-0.	-0.	
0.	0.	0.	0.	.5	.7	1.	1.	
0.	-0.	0.	-.5	-.7	-1.	-1.2	-1.	
0.	0.	.5	.7	1.	1.2	1.	1.	
0.	0.	.5	.7	1.	1.2	1.	1.	
0.	0.	-.5	-.7	-1.	-1.2	-1.	-1.	

h_4 Additive and Multiplicative Variation

Longitude values:

0.	5.	10.	15.	20.	25.	30.	35.	40.	45.

Colatitude values:

25.5	26.0	26.5	27.0	27.5	28.0	28.5	29.0	29.5	30.0

Height Values for Addition - 8 Cards

-50.	-25.	0.	10.	-15.	-15.	-5.	-10.	
-50.	-25.	0.	5.	10.	15.	0.	0.	
-50.	-25.	0.	5.	10.	15.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	
10.	20.	40.	50.	-10.	-20.	-20.	-10.	
10.	20.	40.	50.	-10.	-20.	-20.	-10.	
0.	0.	0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	0.	0.	

Multipliers for Height Values

1.	1.	1.	1.	1.20	.8	1.5	.5

Perturbation Specification

NSIN2

5

NSIN2 is analogous to NSIN and gives the number of sine square segments in the perturbation model.

Plasma Frequency Perturbation Model - Next 2*NSIN2 + 2 sets of cards.

2	Specification of Perturb. F_1 - 2 entries
.5 .001 50. .001	
2	Specification of Perturb. F_2 - 2 entries
.5 0. 50. 0.	
6	Specification of Perturb. F_3 - 6 entries
.5 0. 16.9 0. 17.5 1. 18.4 1. 19.1 0. 50. 0.	
2	Specification of Perturb. F_4 - 2 entries
.5 0. 50. 0.	
2	Specification of Perturb. F_5 - 2 entries
.5 0. 50. 0.	
2	Specification of Perturb. H_0 - 2 entries
.5 81. 50. 81.	
2	Specification of Perturb. H_1 - 2 entries
.5 100. 50. 100.	
2	Specification of Perturb. H_2 - 2 entries
.5 112. 50. 112.	
2	Specification of Perturb. H_3 - 2 entries
.5 128. 50. 128.	
2	Specification of Perturb. H_4 - 2 entries
.5 144. 50. 144.	
2	Specification of Perturb. H_5 - 2 entries
.5 160. 50. 160.	
2	Specification of Perturb. H_{MAX} - 2 entries
.5 501. 50. 501.	

Perturbation Layer Modification Signal Card

Signal for:

f_1	f_2	f_3	f_4	f_5	h_o	h_1	h_2	h_3	h_4	h_5	h_{max}
0	0	0	0	0	0	0	0	0	0	0	0

In this case no layers are to be modified so this concludes the input data.

BEST AVAILABLE COPY

INPUT TABLES ARE GIVEN IN ADJUR.MAG.COORDINATES. BLOCKED DATA IS GENERATED IN DIPOLE MAGNETIC COORDINATES.

INPUT DATA FOR MODEL DF F1

COLAT	F1
.50	.00
50.00	.00

INPUT DATA FOR MODEL DF F2

COLAT	F2
.50	0.00
50.00	0.00

INPUT DATA FOR MODEL DF F3

COLAT	F3
.50	.00
50.00	.00

INPUT DATA FOR MODEL DF F4

COLAT	F4
.50	0.00
17.20	0.00
17.90	2.45
24.70	2.45
25.40	0.00
50.00	0.00

INPUT DATA FOR MODEL DF F5

COLAT	F5
.50	4.20
10.90	4.20
11.60	5.10
16.90	5.10
17.60	4.20
23.00	4.20
25.80	2.20
33.40	2.20
34.10	1.60
34.90	1.60
35.60	2.20
50.00	2.20

VARIATION WITH LONGITUDE

ADDITIVE VAR.	LONG. = THETA	0.00	5.00	10.00	15.00	20.00	25.00	30.00	35.00	40.00	45.00
	25.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	26.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	26.50	0.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	27.00	0.00	1.00	0.00	0.00	0.00	0.00	.50	.50	-.50	0.00
	27.50	0.00	0.00	-1.00	0.00	0.00	.50	.70	.70	-.70	0.00
	28.00	0.00	-1.00	-1.00	0.00	.50	.70	1.00	1.00	-1.00	0.00
	28.50	0.00	1.00	0.00	0.00	.70	1.00	1.20	1.20	-1.20	0.00
	29.00	0.00	1.00	0.00	0.00	1.00	1.20	1.00	1.00	-1.00	0.00
	29.50	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00	-1.00	0.00
	30.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

INPUT DATA FOR MODEL OF 40

COLAT	40
.50	81.00
50.00	81.00

INPUT DATA FOR MODEL OF 41

COLAT	41
.50	82.00
50.00	82.00

INPUT DATA FOR MODEL OF 42

COLAT	42
.50	83.00
50.00	83.00

INPUT DATA FOR MODEL OF 43

COLAT	43
.50	84.00
50.00	84.00

INPUT DATA FOR MODEL OF 44

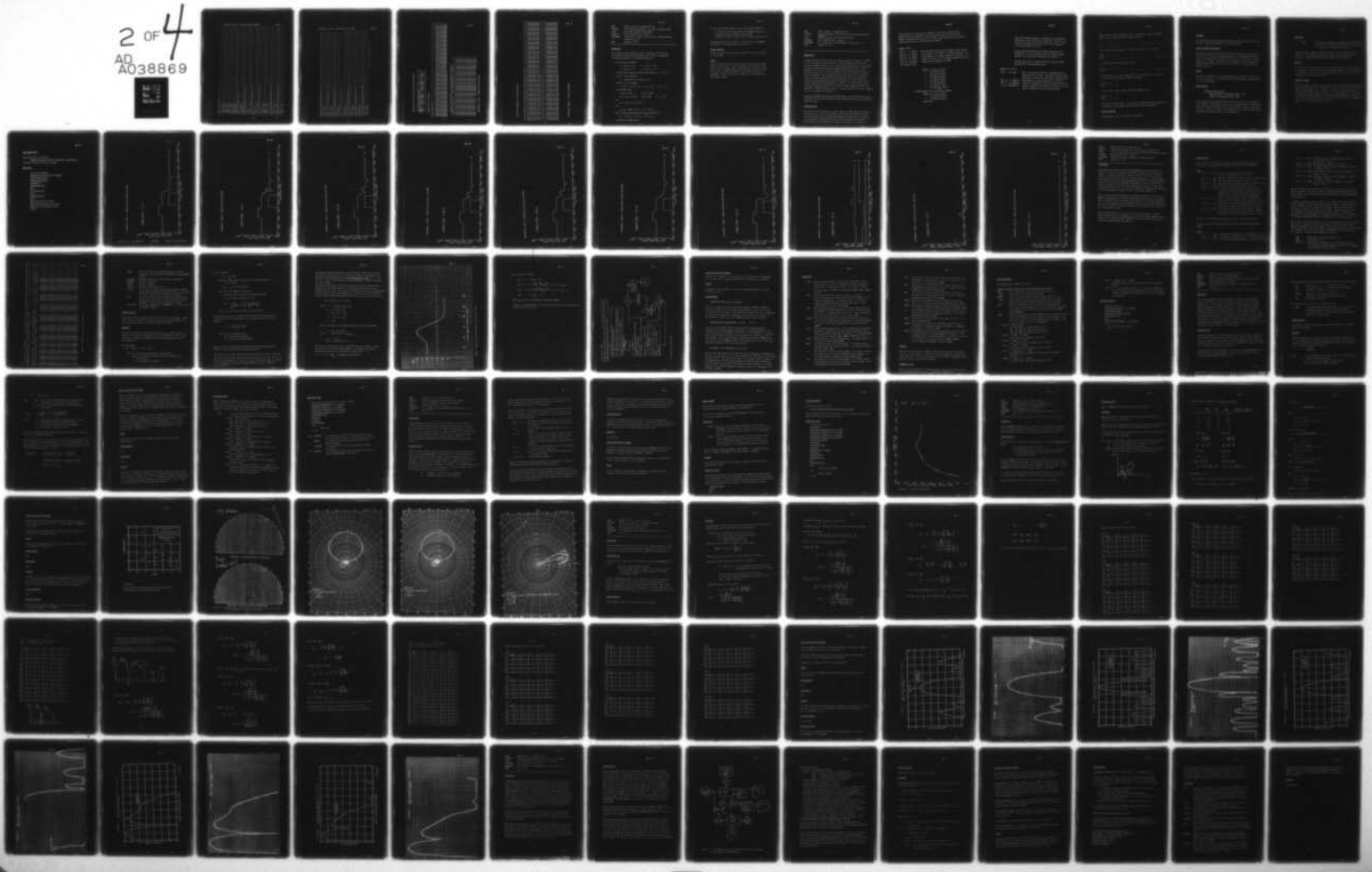
COLAT	44
.50	168.00
10.40	168.00
11.60	168.00
15.70	168.00
17.40	130.00
25.00	130.00
25.30	270.00
28.00	240.00
33.40	240.00
34.10	250.00
34.90	250.00
35.60	246.00
51.00	240.00

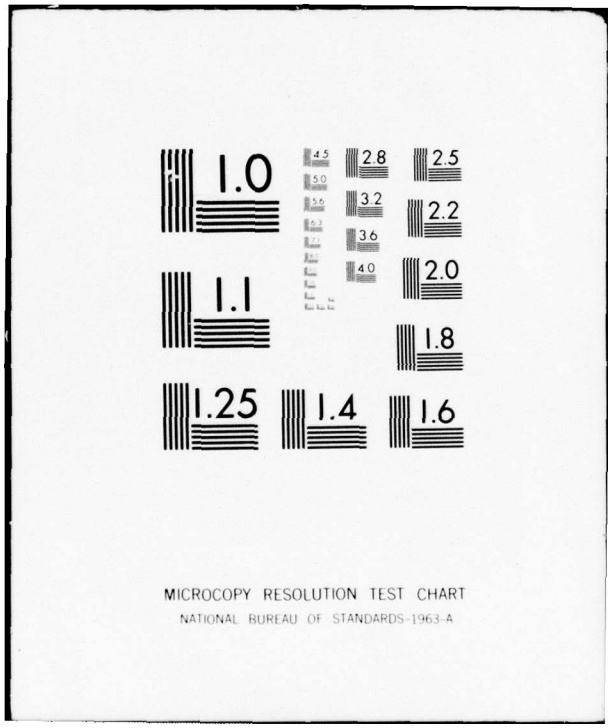
VARIATION WITH LONGITUDE

ADDITIVE VAR.	LONG. = THETA	1.00	5.00	10.00	15.00	20.00	25.00	30.00	35.00	40.00	45.00
	25.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	26.00	1.00	-50.00	-50.00	-50.00	0.00	10.00	10.00	0.00	0.00	0.00
	26.50	1.00	-25.00	-25.00	-25.00	0.00	20.00	20.00	0.00	0.00	0.00
	27.00	1.00	0.00	0.00	0.00	0.00	30.00	30.00	0.00	0.00	0.00
	27.50	1.00	10.00	5.00	5.00	0.00	50.00	50.00	0.00	0.00	0.00
	28.00	1.00	15.00	10.00	10.00	0.00	-10.00	-10.00	0.00	0.00	0.00
	28.50	1.00	15.00	15.00	15.00	0.00	-20.00	-20.00	0.00	0.00	0.00
	28.75	1.00	-5.00	0.00	0.00	0.00	-20.00	-20.00	0.00	0.00	0.00
	29.50	1.00	-10.00	0.00	0.00	0.00	-10.00	-10.00	0.00	0.00	0.00
	31.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
MULTIPLICATIVE VAR.		1.0000	1.0000	1.0000	1.0000	1.0000	1.2000	.8000	1.5000	.5000	1.0000

AD-A038 869 PARKE MATHEMATICAL LABS INC CARLISLE MASS F/G 20/14
ANALYSIS AND SYNTHESIS OF MODEL IONOGRAMS USING 3D RAY TRACKING--ETC(U)
FEB 77 B LANGWORTHY, T BARRETT, D BANDES F19628-76-C-0029
UNCLASSIFIED RADC-TR-77-60 NL

2 OF 4
AD A038869




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

INPUT DATA FOR MODEL OF 45

COLAT	H5
.50	340.00
10.90	340.00
11.00	340.00
13.90	350.00
17.50	355.00
24.90	355.00
25.00	430.00
28.00	300.00
33.40	350.00
34.10	440.00
34.90	640.00
35.60	300.00
50.00	360.00

INPUT DATA FOR MODEL OF 44AX

COLAT	H44AX
.50	301.00
50.00	301.00

INPUT DATA FOR PERTURBATION MODEL OF F1

COLAT	F1
.50	.00
50.00	.00

INPUT DATA FOR PERTURBATION MODEL OF F2

COLAT	F2
.50	0.00
50.00	0.00

INPUT DATA FOR PERTURBATION MODEL OF F3

COLAT	F3
.50	0.00
16.90	0.00
17.50	1.00
18.40	1.00
19.10	0.00
50.00	0.00

INPUT DATA FOR PERTURBATION MODEL OF F4

COLAT	F4
.50	0.00

```
 INPUT DATA FOR PERTURBATION MODEL OF  10

 COLAT  10
    .50   81.00
  50.00   81.00

 INPUT DATA FOR PERTURBATION MODEL OF  11

 COLAT  11
    .50  100.00
  50.00  100.00

 INPUT DATA FOR PERTURBATION MODEL OF  12

 COLAT  12
    .50  112.00
  50.00  112.00

 INPUT DATA FOR PERTURBATION MODEL OF  13

 COLAT  13
    .50  120.00
  50.00  120.00

 INPUT DATA FOR PERTURBATION MODEL OF  14

 COLAT  14
    .50  140.00
  50.00  140.00

 INPUT DATA FOR PERTURBATION MODEL OF  15

 COLAT  15
    .50  160.00
  50.00  160.00

 INPUT DATA FOR PERTURBATION MODEL OF  MAX

 COLAT  MAX
    .50  501.00
  50.00  501.00
```

Specification of output array of plasma frequency

REGULAR GRID PARAMETERS: VAR2

NLONMX =	14	NLATMX =	71	NHTMX =	199
PHZO =	-3.00	THZO =	14.00	HTZO =	80.00
DLPH =	1.50	DLTH =	.20	DLHT =	2.00

Height Table for Profiles

HEIGHT ARRAY IN KM.

80.0	84.0	88.0	92.0	96.0	100.0	104.0	108.0	112.0	116.0	120.0	124.0	128.0	132.0	136.0	140.0	144.0	148.0	152.0	156.0	160.0
164.0	168.0	172.0	176.0	180.0	184.0	188.0	192.0	196.0	200.0	204.0	208.0	212.0	216.0	220.0	224.0	228.0	232.0	236.0	240.0	244.0
248.0	252.0	256.0	260.0	264.0	268.0	272.0	276.0	280.0	284.0	288.0	292.0	296.0	300.0	304.0	308.0	312.0	316.0	320.0	324.0	328.0
332.0	336.0	340.0	344.0	348.0	352.0	356.0	360.0	364.0	368.0	372.0	376.0	380.0	384.0	388.0	392.0	396.0	400.0	404.0	408.0	412.0
416.0	420.0	424.0	428.0	432.0	436.0	440.0	444.0	448.0	452.0	456.0	460.0									

Conversions from dipolar to accurate geomagnetic coordinates

GEOMAGNETIC DATA READ IN. NAME =GUSTAFSSON REV.DATE = 5/13/73

T DIP =	73.20	P DIP =	-3.00	CONVERTS TO T ACC =	75.42	P ACC =	6.63
T DIP =	74.20	P DIP =	-3.00	CONVERTS TO T ACC =	74.52	P ACC =	6.15
T DIP =	73.20	P DIP =	-1.50	CONVERTS TO T ACC =	75.34	P ACC =	8.23
T DIP =	74.20	P DIP =	-1.50	CONVERTS TO T ACC =	74.44	P ACC =	7.75
T DIP =	73.20	P DIP =	0.00	CONVERTS TO T ACC =	75.26	P ACC =	9.81
T DIP =	74.20	P DIP =	0.00	CONVERTS TO T ACC =	74.36	P ACC =	9.36
T DIP =	73.20	P DIP =	1.50	CONVERTS TO T ACC =	75.16	P ACC =	11.33
T DIP =	74.20	P DIP =	1.50	CONVERTS TO T ACC =	74.25	P ACC =	10.89
T DIP =	73.20	P DIP =	3.00	CONVERTS TO T ACC =	75.05	P ACC =	12.86
T DIP =	74.20	P DIP =	3.00	CONVERTS TO T ACC =	74.13	P ACC =	12.44
T DIP =	73.20	P DIP =	4.50	CONVERTS TO T ACC =	74.97	P ACC =	14.38
T DIP =	74.20	P DIP =	4.50	CONVERTS TO T ACC =	74.06	P ACC =	13.98
T DIP =	73.20	P DIP =	6.00	CONVERTS TO T ACC =	74.88	P ACC =	15.91
T DIP =	74.20	P DIP =	6.00	CONVERTS TO T ACC =	73.96	P ACC =	15.53
T DIP =	73.20	P DIP =	7.50	CONVERTS TO T ACC =	74.79	P ACC =	17.44
T DIP =	74.20	P DIP =	7.50	CONVERTS TO T ACC =	73.88	P ACC =	17.08
T DIP =	73.20	P DIP =	9.00	CONVERTS TO T ACC =	74.70	P ACC =	18.92

Attachment 2, Page 5 - Sample Output from S2PPR

Listing of profiles at two (θ,Ø) points

THETA = 19.00 PHI = -3.00 HTZO = 60.0 DEL HT = 2.0

0.00	0.00	.33	.66	.98	1.27	1.55	1.79	2.00	2.05	2.10	2.16	2.21	2.26	2.31	2.37	2.41	2.46	2.50	2.54	2.57
2.60	2.62	2.63	2.64	2.64	2.64	2.63	2.62	2.60	2.59	2.57	2.55	2.54	2.52	2.50	2.49	2.48	2.47	2.47	2.47	2.48
2.48	2.50	2.50	2.51	2.53	2.54	2.55	2.56	2.58	2.59	2.61	2.62	2.64	2.66	2.67	2.69	2.71	2.73	2.75	2.77	2.79
2.81	2.83	2.86	2.88	2.90	2.92	2.95	2.97	2.99	3.02	3.04	3.06	3.09	3.11	3.14	3.16	3.19	3.21	3.23	3.26	3.28
3.31	3.33	3.35	3.38	3.40	3.43	3.45	3.47	3.50	3.52	3.54	3.56	3.59	3.61	3.63	3.65	3.67	3.69	3.72	3.74	3.76
3.79	3.79	3.81	3.83	3.85	3.87	3.88	3.90	3.92	3.94	3.95	3.97	3.98	4.00	4.01	4.02	4.04	4.05	4.06	4.07	4.08
4.09	4.10	4.11	4.12	4.13	4.14	4.15	4.15	4.16	4.17	4.17	4.18	4.18	4.19	4.19	4.19	4.20	4.20	4.20	4.20	4.20
4.20	4.20	4.20	4.19	4.19	4.19	4.18	4.18	4.17	4.17	4.16	4.16	4.15	4.14	4.13	4.12	4.11	4.10	4.09	4.08	4.07
4.06	4.05	4.04	4.03	4.01	4.00	3.98	3.97	3.96	3.94	3.93	3.91	3.90	3.88	3.87	3.85	3.83	3.82	3.80	3.78	3.76
3.75	3.73	3.71	3.69	3.67	3.65	3.64	3.62	3.60	3.58											

THETA = 19.20 PHI = -3.00 HTZO = 80.0 DEL HT = 2.0

0.00	0.00	.33	.66	.98	1.27	1.55	1.79	2.00	2.05	2.10	2.15	2.20	2.25	2.31	2.36	2.40	2.44	2.48	2.52	2.55
2.58	2.59	2.61	2.62	2.62	2.62	2.61	2.60	2.59	2.57	2.56	2.54	2.53	2.51	2.50	2.49	2.48	2.47	2.47	2.47	2.48
2.48	2.50	2.50	2.51	2.53	2.54	2.55	2.56	2.58	2.59	2.61	2.62	2.64	2.66	2.67	2.69	2.71	2.73	2.75	2.77	2.79
2.81	2.83	2.86	2.88	2.90	2.92	2.95	2.97	2.99	3.02	3.04	3.06	3.09	3.11	3.14	3.16	3.19	3.21	3.23	3.26	3.28
3.31	3.33	3.35	3.38	3.40	3.43	3.45	3.47	3.50	3.52	3.54	3.56	3.59	3.61	3.63	3.65	3.67	3.69	3.72	3.74	3.76
3.79	3.79	3.81	3.83	3.85	3.87	3.88	3.90	3.92	3.94	3.95	3.97	3.98	4.00	4.01	4.02	4.04	4.05	4.06	4.07	4.08
4.09	4.10	4.11	4.12	4.13	4.14	4.15	4.15	4.16	4.17	4.17	4.18	4.18	4.19	4.19	4.19	4.20	4.20	4.20	4.20	4.20
4.20	4.20	4.20	4.19	4.19	4.19	4.18	4.18	4.17	4.17	4.16	4.16	4.15	4.14	4.13	4.12	4.11	4.10	4.09	4.08	4.07
4.06	4.05	4.04	4.03	4.01	4.00	3.98	3.97	3.96	3.94	3.93	3.91	3.90	3.88	3.87	3.85	3.83	3.82	3.80	3.78	3.76
3.75	3.73	3.71	3.69	3.67	3.65	3.64	3.62	3.60	3.58											

NAME: S2PPR, revision 1, program, PML 136
CATEGORY: Preprocessing program for a ray-trace ionospheric model
TITLE: Sine Square Preprocessing Program
LANGUAGE: CDC Extended Fortran - Version 4
PROGRAMMER: Revision by L. Calabi and I.B. Jarvis, Parke Mathematical Laboratories, Inc.
DATE: December 6, 1976

DESCRIPTION

This program has been modified to pack 4 elements of the array for the ionospheric model into one computer word. This is accomplished in subroutine FPACK4 by the following algorithm:

Given four array elements

$x_1 = f(i,4j + 1,k)$ $\quad x_2 = f(i,4j + 2,k)$

$x_3 = f(i,4j + 3,k)$ $\quad x_4 = f(i,4j + 4,k)$,

let $\bar{x}_i$ be the integer obtained by multiplying x_i , i = 1,2,3,4, by 1000. and by rounding:

$\bar{x}_i$ = integral part of $(1000\ x_i + .5)$,

and let y_i be the integer

$y_1 = \bar{x}_1$, $y_2 = \bar{x}_2 - \bar{x}_1$, $y_3 = \bar{x}_3 - \bar{x}_2$, $y_4 = \bar{x}_4 - \bar{x}_3$.

It is assumed that

$0 \le x_1 < 32.768$ $\quad 0 \le y_1 < 32768$

$|x_i - x_{i-1}| < 16.384$ $\quad |y_i| < 16384$ $\quad (i = 2,3,4)$.

Let

$z_1 = y_1$, $z_i = y_i$, if $y_i \ge 0$

and

$z_i = y_i + 16384$ if $y_i < 0$ $\quad (i = 2,3,4)$.

The z_i can be represented with 15 bits and stored as

$z_1 \cdot 2^{45} + z_2 \cdot 2^{30} + z_3 \cdot 2^{15} + z_4$

in one 60-bit computer word.

The only modifications required to the main program (S2PPR) are:

1) Include the packed array KOMP in the COMMON storage area.
2) CALL FPACK before writing to mass storage.
3) Change the number of words to write on mass storage to 6623.

Corresponding modifications must of course be made to RAYTRACEBL as described in its program write-up, revision 1.

STORAGE REQUIRED

Storage required for this revision of S2PPR is 115000 octal words of core storage.

TIMING

166 CP seconds execution time was required for the test run which created 24 data blocks. This is approximately 7 seconds per data block. This is slightly slower than the previous version of S2PPR but results in smaller permanent file and disk pack storage requirements. There is also a core storage and I/O savings for the ray tracing program.

NAME:	S2PLT, revision 0, program, PML 138
CATEGORY:	Plotting program for use with electron density preprocessing program.
TITLE:	Plasma frequency contour plotting program.
LANGUAGE:	CDC extended Fortran - version 4
PROGRAMMER:	B. M. Langworthy, Parke Mathematical Laboratories, Inc.
DATE:	December 23, 1975

DESCRIPTION

Program S2PLT may be used to plot the contours of models to be given to S2PPR. It will plot models with or without the layer modifications and will also plot the perturbation models with or without layer modifications. This program has been separated from the preprocessing program S2PPR to keep it within reasonable core limits. Plots are composed of layer characteristics such as the plasma frequency of the various layers versus colatitude for any given value of longitude. The coordinate system will be that of the input model. Plots may contain more than one contour within the following limitations: height and plasma frequencies cannot be intermixed on one plot and the model cannot be mixed with the perturbation on one plot. A plot can be made up of the plasma frequency contours for all five sine square segments or any one or combination of several of these contours. Plots can deal with only one longitude value on a particular set of axes.

By separating S2PLT from the preprocessing program, layer modifications can be assessed before the plasma frequency array is generated thus saving considerable computation time.

INSTRUCTION SET

The input data set to S2PLT is the same as that for S2PPR (See PML 136) except that input cards 1 and 2 are omitted since these two cards specify the grid size and fineness of the output plasma frequency array which is not generated by this program. Additional input must follow the model and perturbation specifications which indicates the contours to be plotted. For each set of

plots desired two input cards are required: the first to specify which contours are to be included on the plot and the second to tell for which values of longitude the plots are to be made. These cards follow the complete contour specification as would be given to S2PPR.

Card 1 (12I5)

Cols 1 - 5 IPL(1)
Cols 6 - 10 IPL(2)
Cols 11 - 15 IPL(3)
Cols 16 - 20 IPL(4)
etc up to a maximum of 7 values of IPL

IPL is an array containing the numbers of the contours to be plotted. The number of a contour depends on the total number of sine square segments in the model. For instance if there are 3 sine square segments the contour numbers will correspond as follows:

Contour 1 is the f_1 contour
" 2 is the f_2 contour
" 3 is the f_3 contour
" 4 is the H_0 contour
" 5 is the H_1 contour
" 6 is the H_2 contour
" 7 is the H_3 contour
" 8 is the H_{max} contour

If there were to be 5 sine square segments:

Contour 4 is the f_4 contour
" 5 is the f_5 contour
" 6 is the H_0 contour
and so on.

Since only plasma frequencies or heights can be specified on a given plot, card 1 could have no more than 7 entries as in the five sine square case if H_0, H_1 thru H_5, and H_{max} were all to be contained on a plot.

To plot the plasma frequency and height contours for the perturbation, 20 must be added to the above contour number. Thus the f_3 contour would be specified as contour 23.

All IPL entries must be right justified in the five spaces provided and contain no decimal point.

Card 2 (I10,7F10.0)

Cols 1 - 10 NPHI	This is the number of values of longitude for which the plots are to be made. The number should be right justified to column 10 and should be no greater than 7.
Cols 11 - 20 PHPLT(1) Cols 21 - 30 PHPLT(2) etc up to PHPLT(NPHI)	where PHPLT is an array containing the values of longitude in degrees for which the contours are to be plotted. This value will appear on the top of the plot. Values of PHPLT must contain a decimal point.

These two cards may be repeated as often as necessary to obtain all desired sets of plots. Below is listed a sample input case:

		1	1	2	2	3	3	4	4	5	5	6	6	7	7	8
Cols	5	0	5	0	5	0	5	0	5	0	5	0	5	0	5	0

If the number of sine segment is 5 and we wish to plot f_3, f_4, and f_5:

Card 1 3 4 5
Card 2 7 0.0 3.5 10. 15. 24. 67.125 70.

If we wish also to plot H_0, H_3, H_4, and H_5:

Card 1 6 9 10 11
Card 2 7 0.0 3.5 10. 15. 24. 67.125 70.

If the number of sine square segments for the perturbation is 3 and we wish to plot f_2 and f_3 for only one value of longitude:

Card 1 22 23
Card 2 1 0.0

We also wish to plot H_2 and H_3 for the perturbation model.

Card 1 26 27
Card 2 1 0.0

The output of S2PLT consist of a listing of inputed tables for the model and perturbation and the desired plots which are given in Attachment 1.

STORAGE REQUIRED

S2PLT requires 55000 octal words of core storage.

ALGORITHM

The layer modification computations performed by S2PLT are the same as those in S2PPR. Please refer to PML 136 for this algorithm.

SPECIAL CAUTIONS AND FEATURES

Care must be taken to be sure that the contours which are to be plotted on a given plots are either all plasma frequencies or all heights since the axis system is selected based on whether the first contour specified is a plasma frequency or a height. Thus trying to plot height on a scale selected for plasma frequency in MHz. will cause the plot limits to be exceeded and the program will stop.

TIMING

The timing of S2PLT will vary becoming greater with increased complexity of the layer modification. A typical timing with modifications to several layers is 1.5 seconds per plot.

ERROR MESSAGES

PARAMETERS OUTSIDE LIMITS
MAXIMUM NUMBER OF COLATITUDE ENTRIES = XXXX
NUMBER OF SINE SQUARE CURVES = XXXXX

This indicates that the maximum number of colatitude entries has exceeded 25 or the number of sine square curves specified is greater than 5. Check to see that these numbers are punched in the correct columns or that the input deck is set up in the correct order. This message will be printed out when this is encountered in either the model or the perturbation.

SUBROUTINES

ADDER	This subroutine computes the additive and multiplicative variations to be made in the model and the perturbation as a function of colatitude and longitude.

Program S2PLT also requires some AFCRL plotting software subroutines. See Section 14 of the AFCRL User's Guide for details. The plotting subroutines called are: PLTID3, AXIS, ENDPLT, LINE, NUMBER, PLOT, and SYMBOL.

ACCURACY

The curves put out by S2PLT are made up of points calculated every 0.5 degrees in colatitude. If a particular sine square segment happens over a small interval in colatitude, it will tend to look squared off, but this will not affect the computational values put out by S2PPR.

COMMENTS ON USAGE

S2PLT is designed to be used as a companion program to S2PPR. When constructing models and modifying layers, S2PLT should be used to evaluate the layer modifications before S2PPR is run. S2PLT will only depict the characteristics of a particular sine square segment. If profiles of plasma frequency versus height are desired, S2PPR must first be run and then program PROPLT must be used with electron density subroutine E14994. See PML 129 for a description of PROPLT and PML 127, rev. 1, for a description of E14994.

FILE DESCRIPTIONS

The program card is as follows:

```
PROGRAM S2PLT(INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT)
```

The input to S2PLT is described on page 2.

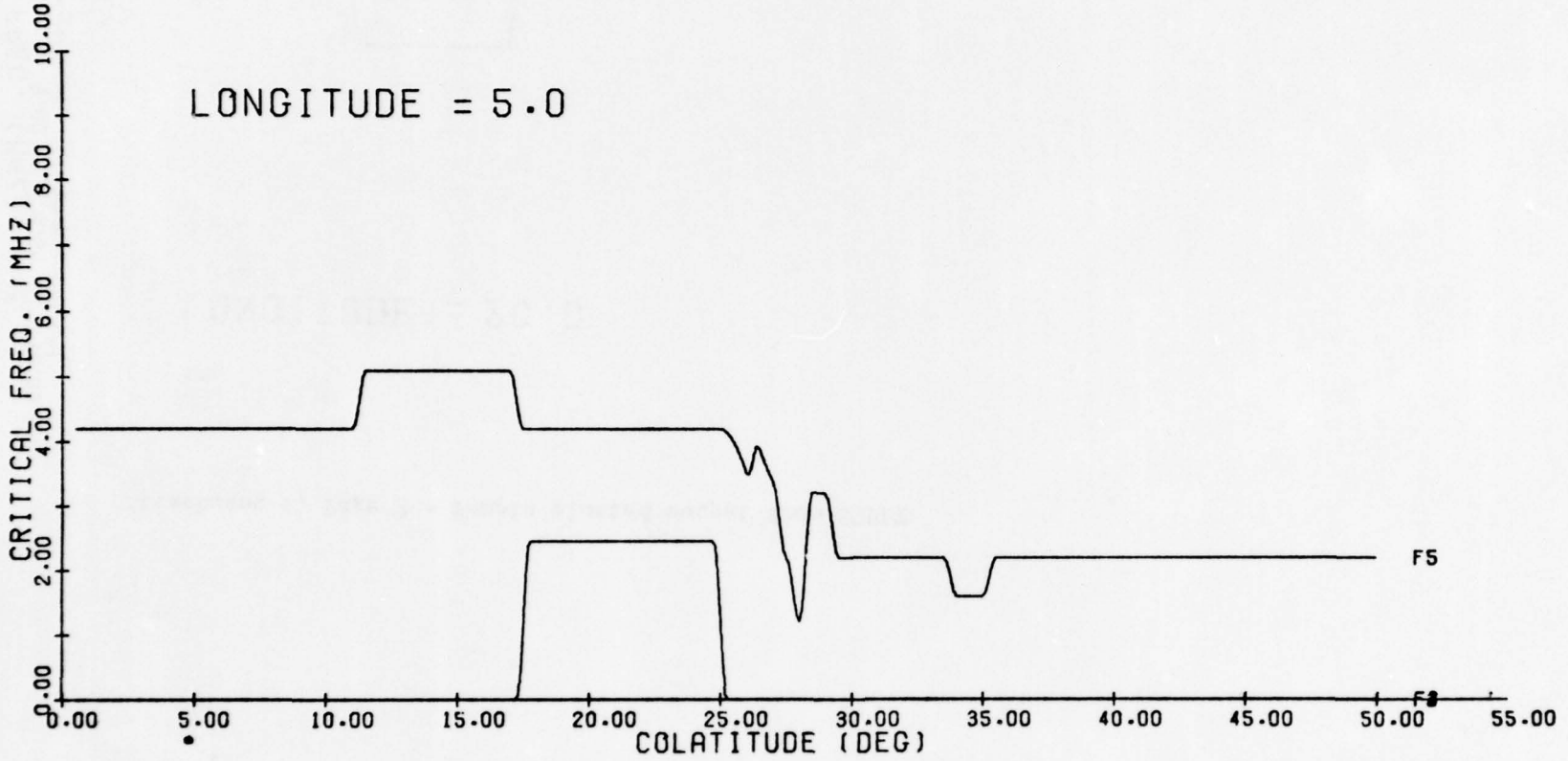
DECK SET UP

```
Job Card with CM60000.
ATTACH(SOURCE,S2PLTX3693818,ID=LANGW)
FTN(I=SOURCE,B=BS2P)
ATTACH(PEN,ONLINEPEN)
LIBRARY(PEN)
LDSET(PRESET=ZERO)
BS2P.
DISPOSE(PLOT,*OL)
EXIT.
DISPOSE(PLOT,*OL)
7/8/9
Model description data cards
Perturbation description data cards
Sets of 2 plot description cards
6/7/8/9
```

Attachment 1, Page 1 - Sample plotted output from S2PLT

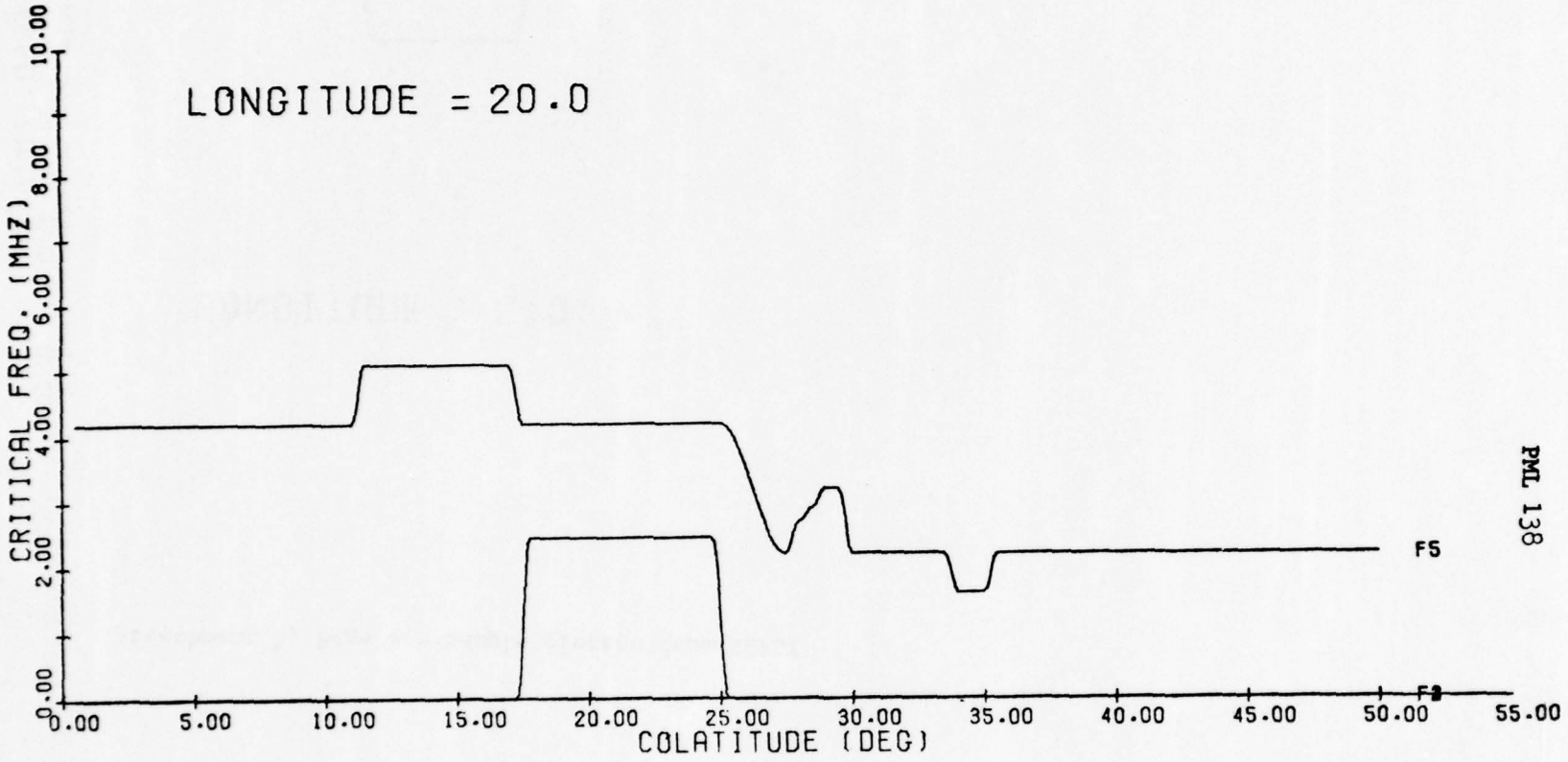
LANGMORTHY 2516 S53PLT 12/08/75 11.44.21.

Attachment 1, Page 2 - Sample plotted from S2PLT

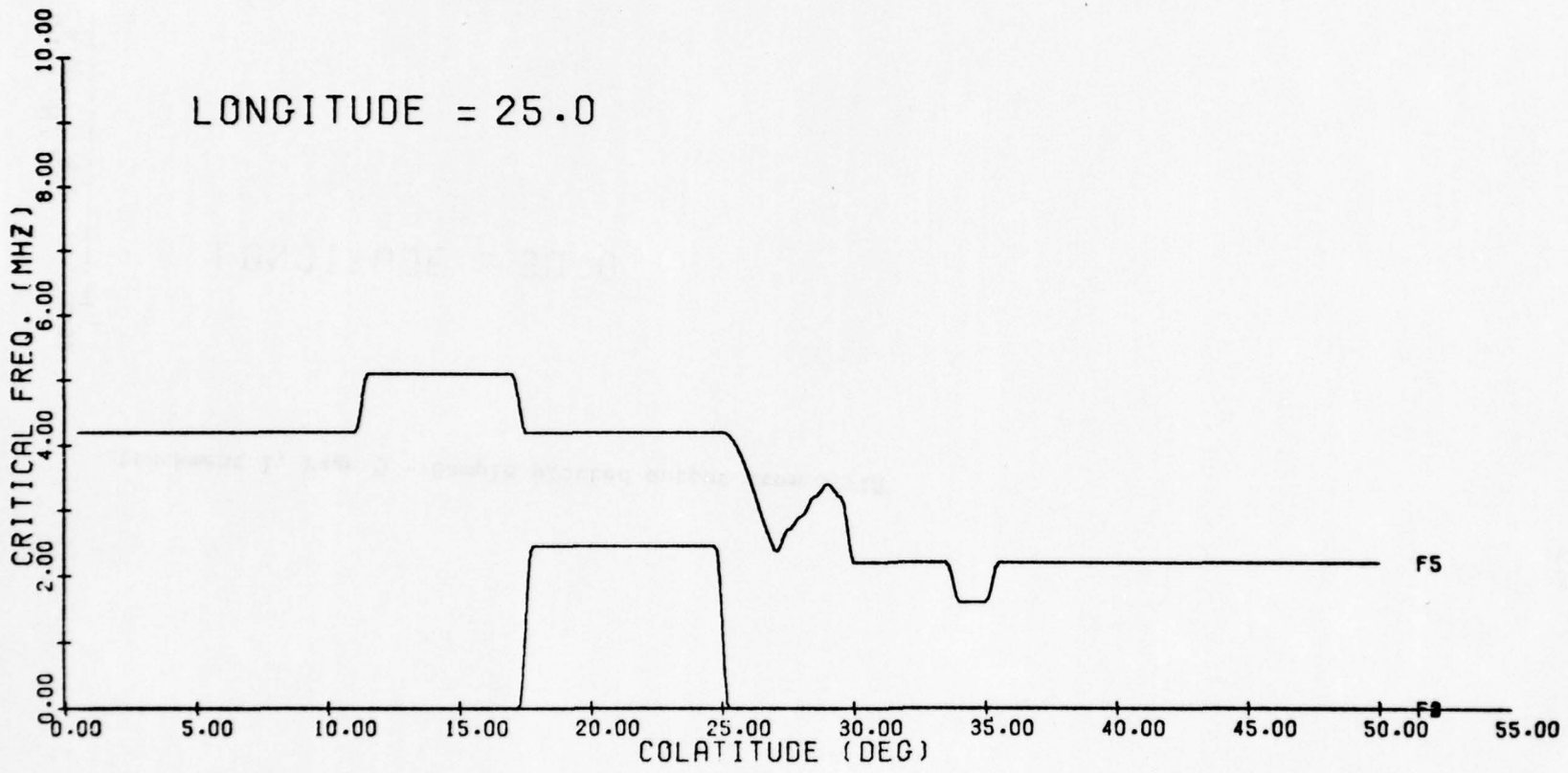


PML 138

Attachment 1, Page 3 - Sample plotted output from S2PLT

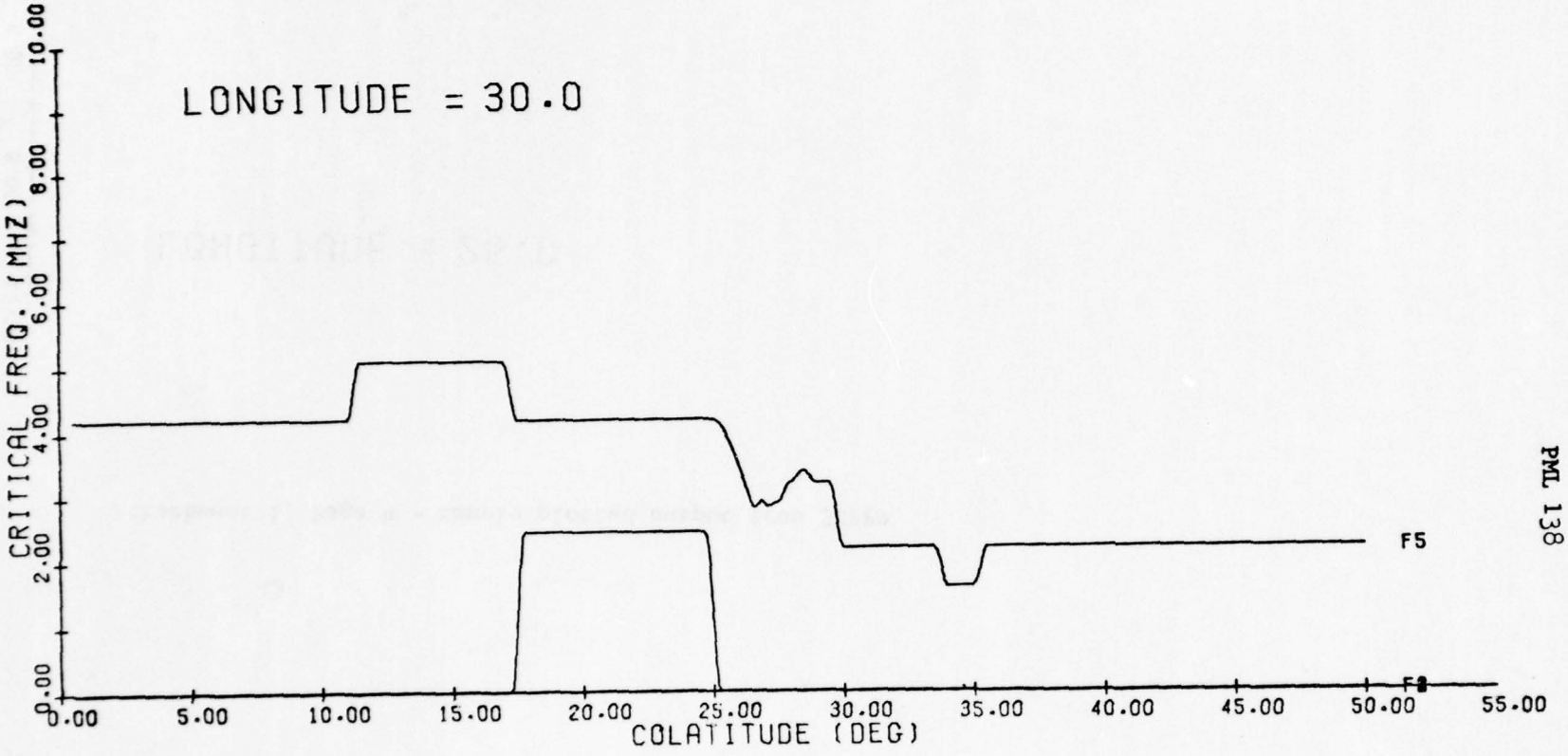


Attachment 1, Page 4 - Sample plotted output from S2PPR

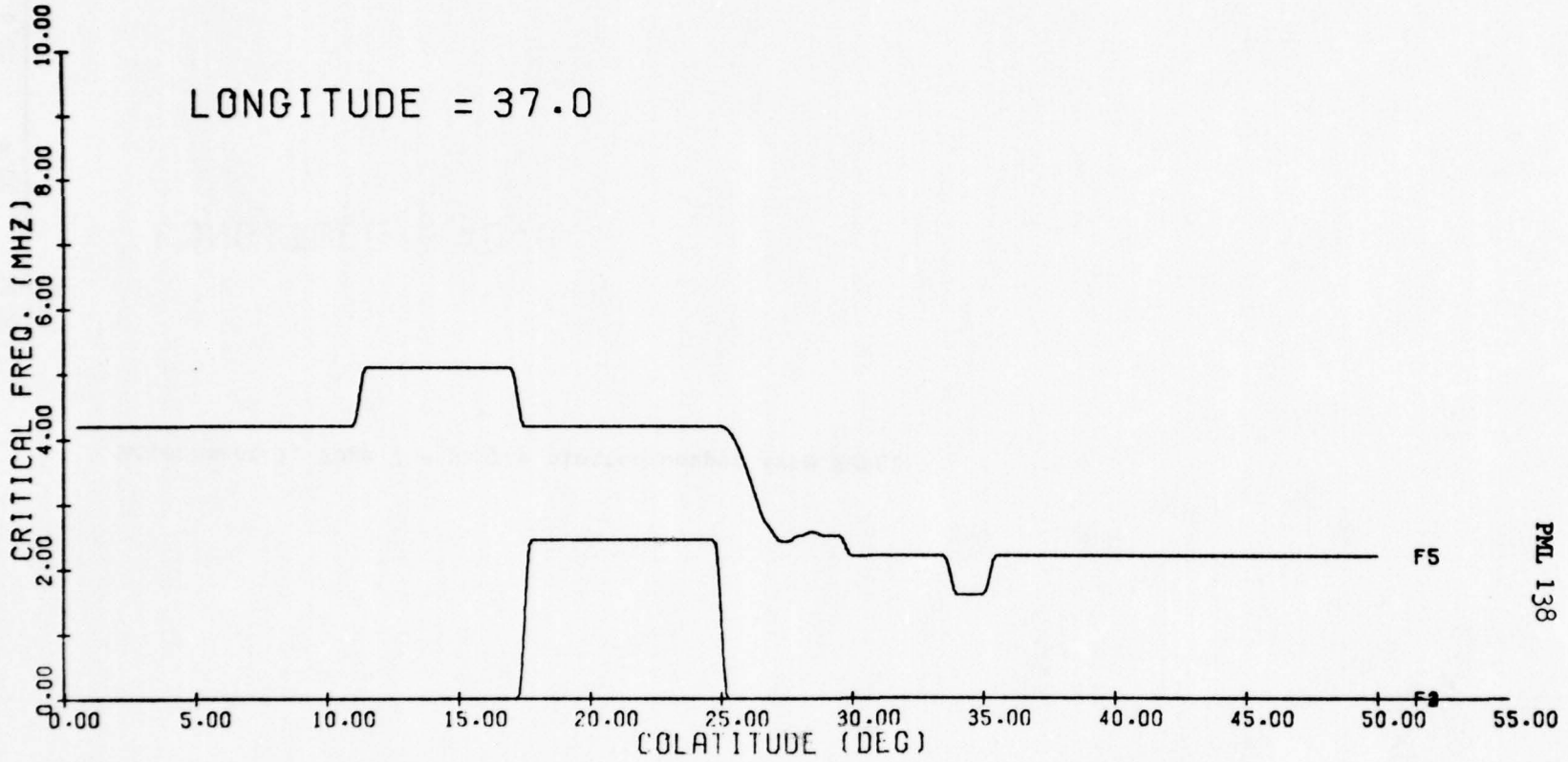


PML 138

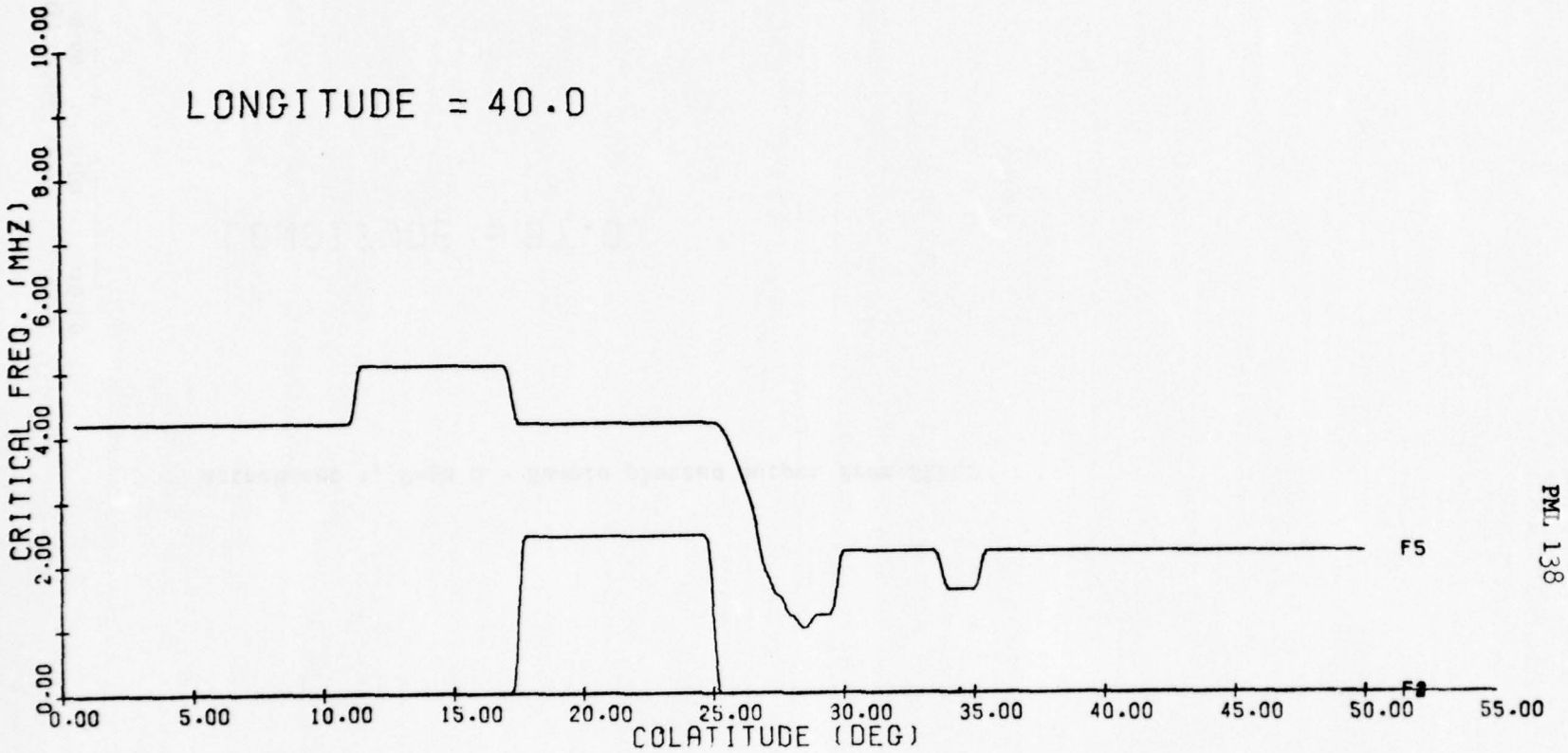
Attachment 1, Page 5 - Sample plotted output from S2PLT



Attachment 1, Page 6 - Sample plotted output from S2PLT

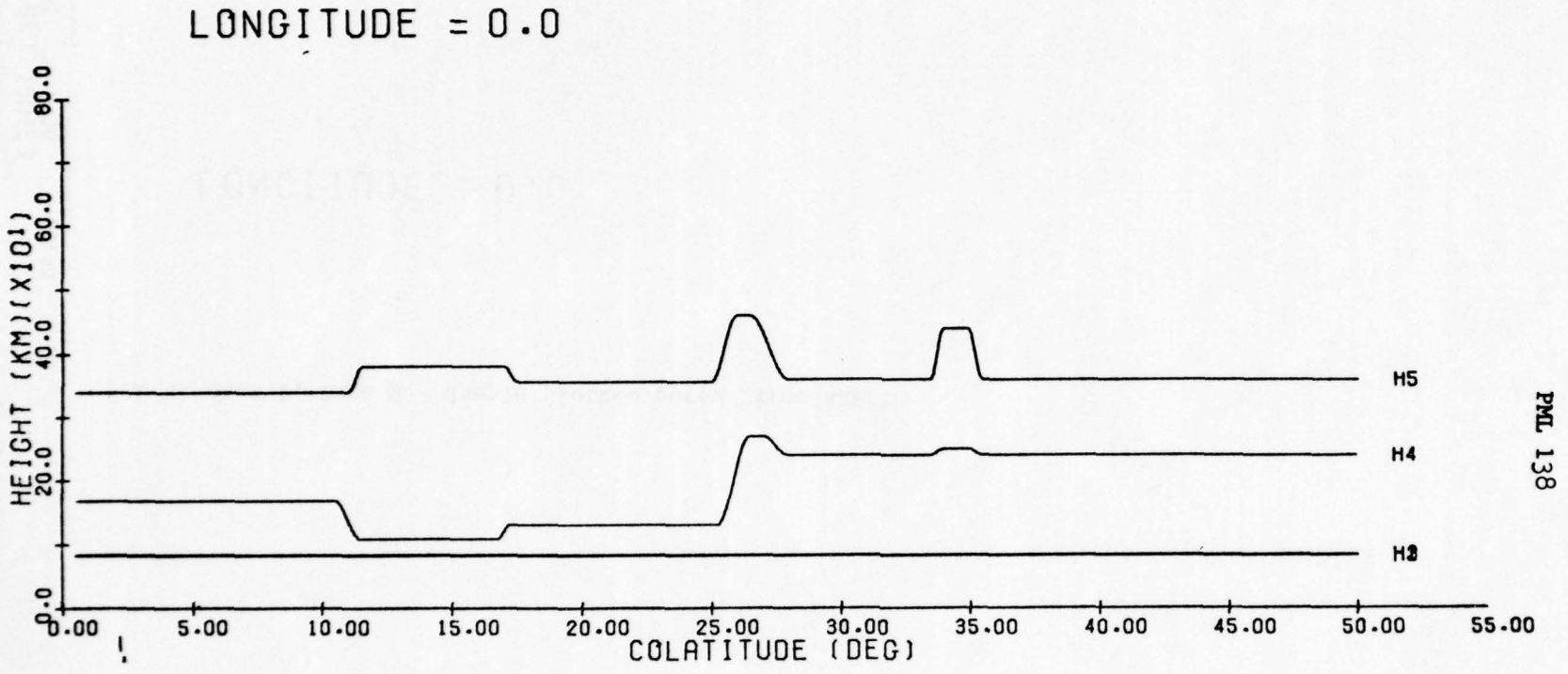


Attachment 1, Page 7 - Sample plotted output from S2PLT

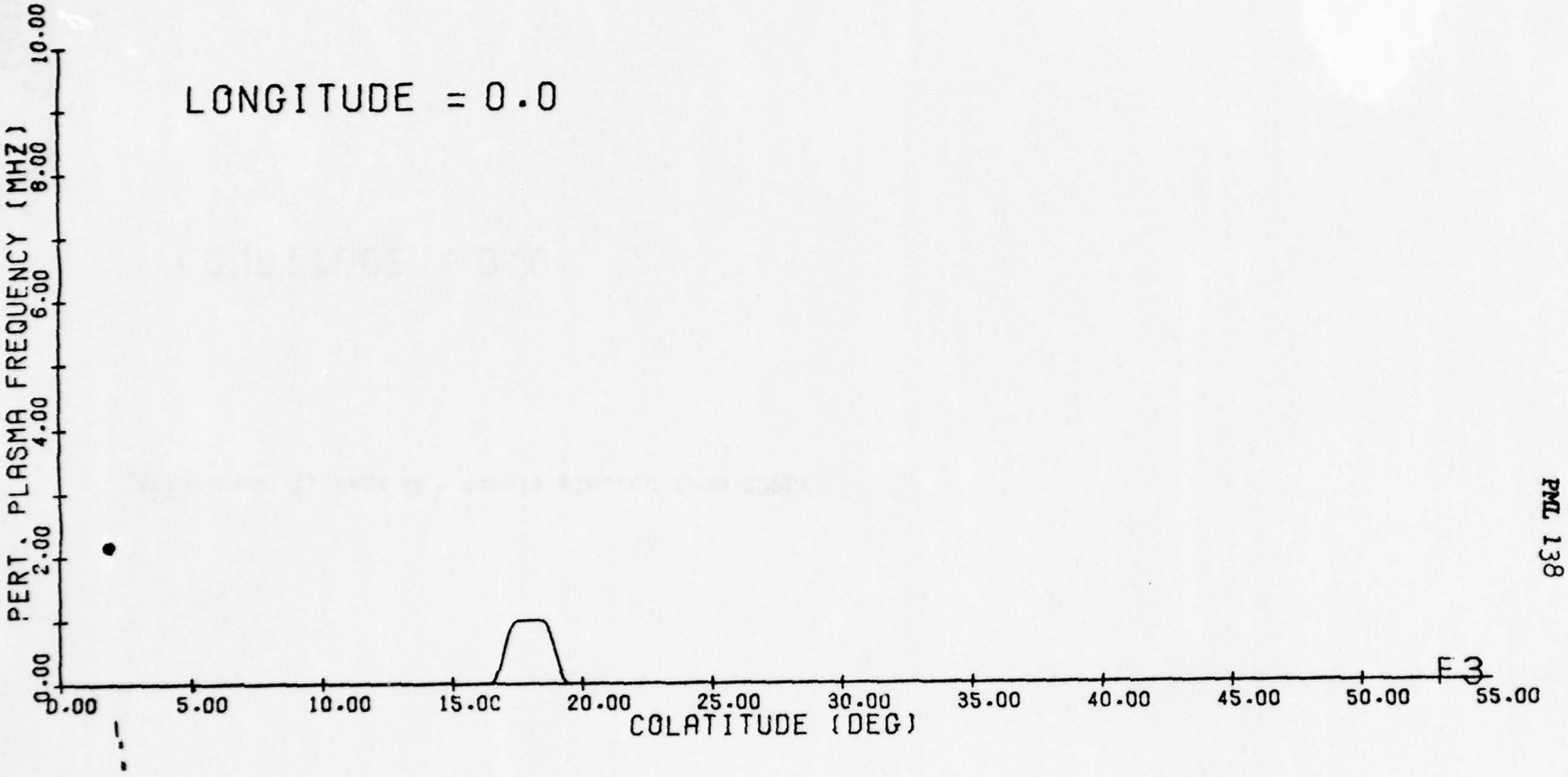


PML 138

Attachment 1, Page 8 - Sample plotted from S2PLT

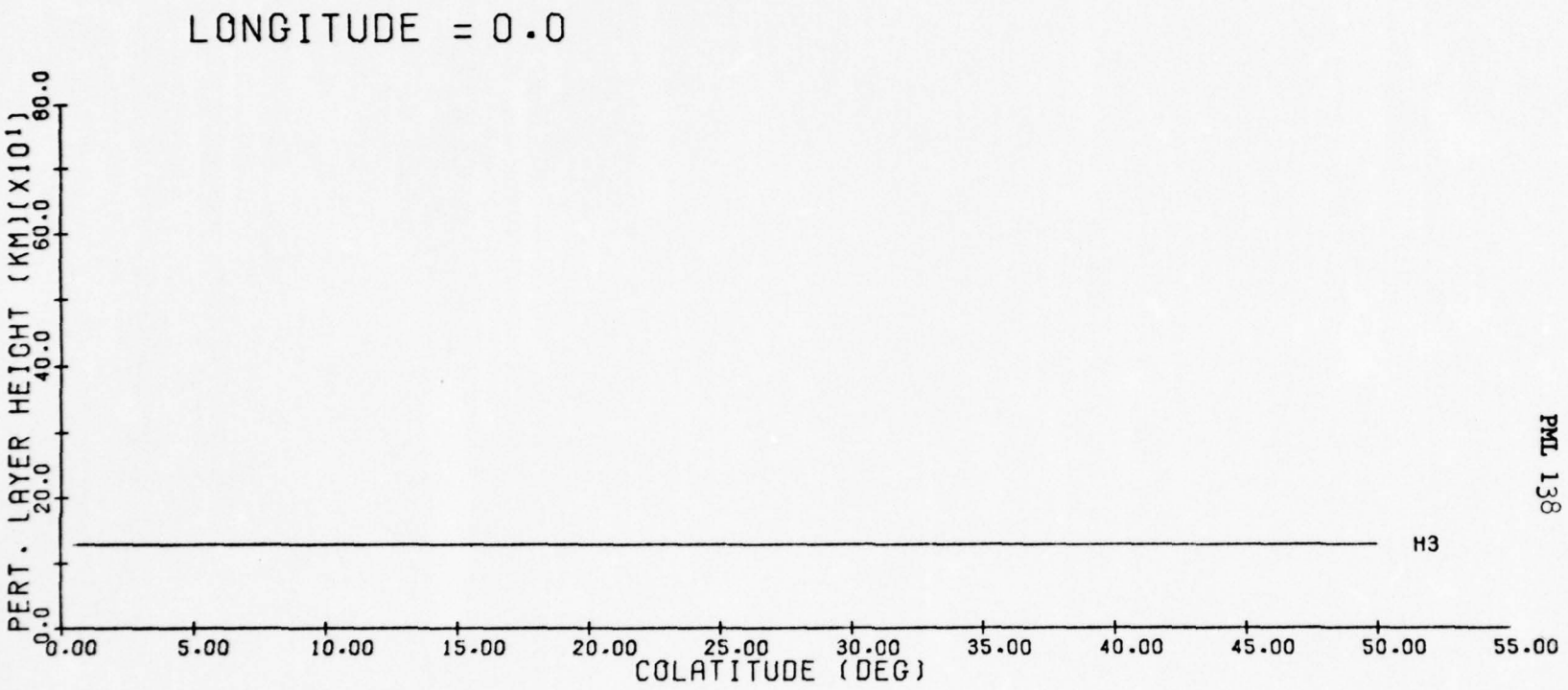


Attachment 1, Page 9 - Sample plotted output from S2PLT



PML 138

Attachment 1, Page 10 - Sample plotted output from S2PL



NAME: COLPP, Revision 0, program, PML 139
CATEGORY: Preprocessing program for collision frequency generation
TITLE: Collision Frequency Preprocessing program for the ray-trace program
LANGUAGE: CDC Extended Fortran - Version 4
PROGRAMMER: B.M.Langworthy, Parke Mathematical Laboratories, Inc.
DATE: February 20, 1976

DESCRIPTION

This program was written to take existing atmospheric and solar data and to generate, in tabular form magnetic and dipolar coordinates, those quantities needed to produce the collision frequency at any given location in (r,θ,ϕ) and for any given date and time. To accomplish this, parts of other programs developed by the Air Force Cambridge Research Laboratories were used extensively. A set of subroutines which provided the temperature, density, mean molecular weight, and number of molecules was obtained from Dr. Champion of LKB. This is the model developed by Dr. L. C. Jacchia of Smithsonian Astrophysical Observatory in 1971 and optimized for AFCRL use by Mr. John Kotelly, AFCRL-72-0171.

The solar declination was obtained from a subroutine SOL which is part of the CADNIP program developed for AFCRL by IBM Federal Systems Division of Burlington, Mass. Also modeled for this program was the ratio of the electron temperature and the neutral temperature from information supplied by Mr. Charles Rush of LII. Solar Flux and magnetic index data needed for the program have been supplied by Mrs. Isabel Hussey of SUYA.

Program COLPP creates a file of data on the collision frequency of neutral particles and the electron temperature as a function of height, magnetic dipolar colatitude, and magnetic dipolar longitude to be used by ray-trace collision frequency subroutine COLAF1 (See PML 140).

PRECEDING PAGE BLANK-NOT FILMED

INSTRUCTION SET

To use COLPP the user must specify some information about the desired day on which data is to be generated. The first card contains this information.

Card 1

Cols 1 - 3 MON - The month of the year for which the data is desired in numbers from 1 to 12. (I3)

Cols 4 - 6 IDAY - The day of the month for which the data is desired. (I3)

Cols 7 - 9 IYR - The year for which the data is desired. (I3)

Cols 11 - 20 UT - The universal time at which the data is desired. (F10.0)

Cols 21 - 30 TMJD - The date in modified Julian days. (F10.0)

Cols 31 - 40 SOLCY - The years into an eleven year solar cycle with 0 indicating a solar minimum and 5.5 indicating a solar maximum. (F10.0) This is used for the calculation of the ratio of electron to neutral temperature.

Cols 41 - 50 F7 - The 10.7 cm solar flux for a time 1.71 days earlier than the time for which the data is desired. (F10.0)

Cols 51 - 60 FB7 - The average 10.7 cm solar flux averaged over the four solar cycles centered on the desired time. (F10.0)

Cols 61 - 70 EKP - The planetary index K_p for a time .279 days earlier than that for which the table is desired. K_p should be entered as 3.333 or 2.667 and not 3+ or 3-. (F10.0)

Data for F7, FB7 and EKP can be obtained from Mrs. Isabel Hussey of SUYA.

To specify the size of the output table desired the following information is needed.

Card 2

Cols 1 - 3 NHT - The number of height entries in the table ($\leq$ 30) (I3)

Cols 4 - 6 NTH - The number of colatitude entries in the table ($\leq$ 10) (I3)

Cols 7 - 9 NPT - The number of longitude entries in the table ($\leq$ 12) (I3)

Cols 11 - 20 HTZO - The starting value of height above the earth's surface in km. (F10.0)

Cols 21 - 30 DLHT - The increment in height in km. (F10.0)

Cols 31 - 40 THZO - The starting value of dipolar geomagnetic colatitude in degrees. (F10.0)

Cols 41 - 50 DLTH - The increment in dipolar geomagnetic colatitude in degrees. (F10.0)

Cols 51 - 60 PHZO - The starting value of dipolar geomagnetic longitude in degrees. (F10.0)

Cols 61 - 70 DLPH - The increment in dipolar geomagnetic longitude in degrees. (F10.0)

The main objective of program COLPP is to produce a file of data for use by the ray-trace program. This file is refered to as TAPE8 by both COLPP and the ray-trace program. The contents of TAPE8 are described under the FILE DESCRIPTION section on Page 14.

In addition to TAPE8, a tabular output file is created to check the quantities stored on TAPE8. A sample output page is given in Figure 1. The first line of output is a listing of the input given to COLPP in card 1. Quantities are identified and listed in the order of their input. The second line of output is a statement to the effect that the coefficients needed to convert from geographic to accurate geomagnetic coordinates have been read in. It also gives the identifier contained on the file to identify the set of coefficients.

Next a table of values with height is given for each magnetic dipolar point (θ,ϕ). The dipolar coordinates of that point are given in radians and the accurate magnetic coordinates are given in degrees. The dipolar values given are colatitude and longitude, while the accurate magnetic coordinates are given in latitude and longitude. The table entries given are as follows:

HGTH - Height above the earth's surface in km.

TEMP - Temperature in ^{o}K.

TEMP INF - The exospheric temperature on which the data is based in ^{o}K. Note that this value changes at 200 km. due to a change in the formula for generating data.

DATA FOR 1/ 1/72 UNIV TIME = 4.0 MOD. JULIAN DATE IS 41317.0 SOLAR CYCLE = 5.00 F10.7 = 108.3 AV. F10.7 = 114.8 KP = 2.333

GEOMAGNETIC DATA READ IN. NAME =GUSTAFSSON REV.DATE = 5/16/73

DIPOLAR POSITION (RAD) .34907 0.00000 GEOGRAPHIC POSITION (DEG) 58.57 290.24 HOUR ANGLE = 170.24 ALPHA MUL = 1.02

HGHT	TEMP	TEMP INF	ALPHA	LOG PRESS	PRESSURE	DENSITY	NO. MOL	NU NEAV	NU NE	ALPHA TEMP
90.	183.0	799.6	1.00	-.720	.190E+00	.361E-08	.722E+14	.328E+06	.352E+06	183.3
100.	192.7	799.6	1.00	-1.477	.334E-01	.576E-09	.120E+14	.576E+05	.602E+05	193.1
110.	235.1	799.6	1.00	-2.117	.763E-02	.104E-09	.226E+13	.132E+05	.125E+05	235.9
120.	313.5	799.6	1.00	-2.593	.255E-02	.248E-10	.566E+12	.440E+04	.361E+04	314.6
130.	408.8	799.6	1.00	-2.935	.116E-02	.830E-11	.198E+12	.201E+04	.144E+04	410.4
140.	495.1	799.6	1.00	-3.194	.640E-03	.363E-11	.901E+11	.110E+04	.723E+03	497.2
150.	562.0	799.6	1.00	-3.407	.392E-03	.189E-11	.487E+11	.676E+03	.417E+03	564.6
160.	611.2	799.6	1.01	-3.591	.256E-03	.110E-11	.293E+11	.442E+03	.262E+03	614.7
170.	647.5	799.6	1.01	-3.757	.175E-03	.685E-12	.189E+11	.302E+03	.174E+03	651.2
180.	675.2	799.6	1.01	-3.910	.123E-03	.448E-12	.123E+11	.212E+03	.120E+03	679.3
190.	696.8	799.6	1.01	-4.053	.885E-04	.304E-12	.893E+10	.153E+03	.851E+02	701.4
200.	738.7	832.3	1.01	-4.179	.663E-04	.211E-12	.674E+10	.114E+03	.662E+02	744.0
210.	754.1	832.3	1.06	-4.304	.497E-04	.151E-12	.496E+10	.857E+02	.506E+02	802.1
220.	766.7	832.3	1.12	-4.423	.377E-04	.110E-12	.371E+10	.651E+02	.392E+02	856.8
230.	777.1	832.3	1.18	-4.539	.289E-04	.820E-13	.281E+10	.499E+02	.306E+02	914.4
240.	785.7	832.3	1.23	-4.650	.224E-04	.616E-13	.216E+10	.386E+02	.242E+02	969.0
250.	792.9	832.3	1.29	-4.758	.174E-04	.469E-13	.167E+10	.301E+02	.192E+02	1022.7
260.	799.9	832.3	1.35	-4.864	.137E-04	.363E-13	.130E+10	.236E+02	.154E+02	1078.5
270.	803.9	832.3	1.41	-4.967	.108E-04	.278E-13	.102E+10	.186E+02	.124E+02	1133.5
280.	808.0	832.3	1.47	-5.067	.856E-05	.217E-13	.809E+09	.148E+02	.100E+02	1187.9
290.	811.5	832.3	1.53	-5.167	.682E-05	.170E-13	.643E+09	.118E+02	.815E+01	1241.8
300.	814.4	832.3	1.59	-5.264	.545E-05	.134E-13	.513E+09	.940E+01	.665E+01	1295.2
310.	816.9	832.3	1.69	-5.360	.437E-05	.106E-13	.411E+09	.753E+01	.550E+01	1381.5
320.	819.0	832.3	1.79	-5.454	.351E-05	.847E-14	.330E+09	.606E+01	.456E+01	1467.5
330.	820.7	832.3	1.89	-5.548	.283E-05	.677E-14	.266E+09	.489E+01	.378E+01	1553.3
340.	822.2	832.3	1.99	-5.640	.229E-05	.542E-14	.216E+09	.395E+01	.314E+01	1639.0
350.	823.4	832.3	2.09	-5.731	.186E-05	.436E-14	.175E+09	.320E+01	.261E+01	1724.4
360.	824.5	832.3	2.13	-5.821	.151E-05	.351E-14	.142E+09	.260E+01	.214E+01	1752.7
370.	825.4	832.3	2.16	-5.911	.123E-05	.284E-14	.116E+09	.212E+01	.176E+01	1780.6
380.	826.2	832.3	2.19	-5.993	.102E-05	.233E-14	.946E+08	.175E+01	.145E+01	1808.4

Figure 1 - Sample output from program COLPP.

ALPHA - This is the ratio of electron temperature to neutral temperature. Its computation is described in the ALGORITHM section.

LOG PRESS - The log to the base 10 of the pressure in grams/cm^2.

PRESSURE - Pressure in grams/cm^2.

DENSITY - Density in grams/cc.

NO. MOL. - The number of molecules per cc.

NU NEAV - The collision frequency of the neutral molecules using the formula based on pressure (Valid below 200 km.) in collisions per second. (This quantity is supplied to COLAF1.)

NU NE - The collision frequency of the neutral molecules using the formula based on the number of molecules and the electron temperature. (This quantity is supplied to COLAF1.)

ALPHA TEMP - The electron temperature or ALPHA*TEMP in ^{o}K. (This quantity is supplied to COLAF1.)

STORAGE REQUIRED

The storage required for COLPP is 140000 octal words of core storage. A large part of this is required for storage of the output table and storage of the coefficients for conversion to accurate magnetic coordinates.

ALGORITHM

The output from COLPP is basically three quantities $\langle\nu_{en}\rangle$, ν_{en}, and $\propto T_n$, all a function of r,θ,ϕ and the statistics for the desired date and time. These quantities are used in the computation of the collision frequency in the following manner.

For the E region:

$$\nu = \langle\nu_{en}\rangle = 1.72517 \times 10^{6} P$$

where ν is the collision frequency in collision/sec

ν_{en} is the collisions frequency with neutral molecules in collisions/sec.

P is the pressure in grams/cm^2.

For the F region:

$$\nu = \nu_{en} + \nu_{ei}$$

where ν_{ei} is the collision frequency with ionized molecules in collisions/sec

$$\nu_{en} = 3.6 \times 10^{-10} N_n (\alpha T_n)^{1/2}$$

where N_n is the number of molecules/cc

α is the ratio of electron to neutral temperature

T_n is the neutral temperature in ^{o}K.

$$\nu_{ei} = \frac{5.5 N_e}{(\alpha T_n)^{3/2}} \log_e \left\{ \frac{220. (\alpha T_n)}{N_e^{1/3}} \right\}$$

where N_e is the electron density in electrons per cc.

The quantities N_n and T_n are supplied by the subroutine CCIA described in report AFCRL-72-0171. The pressure P can be computed using the following formula.

$$P = \frac{8.314 \times 10^6 \; T_n \rho}{\bar{M}}$$

where T_n is as described above

ρ is the density in grams/cc

$\bar{M}$ is the mean molecular weight.

The density ρ is also supplied by subroutine CCIA and the mean molecular weight is computed using a minor modification to CCIA.

The quantities N_n, T_n, ρ, and $\bar{M}$ are then functions of position in r,θ,Ø, the time of the day, the date (gives the declination angle of the sum), the 10.7 cm solar flux, the average 10.7 cm solar flux, and the planetary index K_p. Details of this dependence can be found in report AFCRL-72-0171 and in CIRA 1972 report pp 227-257.

The quantity α has been modeled to vary with season, time of day, and solar cycle based on data given by C.M. Rush and T.J. Elkins, An Assessment of the Magnitude of the F-Region Absorption, Telecommunication Journal, Vol. 42, Aug. 1975, pp. 476-488. A variation with accurate geomagnetic latitude is given by the curve in Figure 2.

The variation with season is done with a quartic which goes through the given value of α' for December, March, and June and has zero derivative at the December and June points. The value of α' indicates an α value for Massachusetts. Given values for the particular height, solar cycle, and time for each of the December, March, and June entries, call them α_D, α_M, and α_J, respectively,

$$\alpha' = ax^4 + bx^3 + cx^2 + dx + e$$

where $a = 1/2\,(2\alpha_M - \alpha_D - \alpha_J)$

$b = 1/4\,(\alpha_D - \alpha_J)$

$c = \alpha_D + \alpha_J - 2\alpha_M$

$d = 3/4\,(\alpha_J - \alpha_D)$

$e = \alpha_M$

where x is a measure of the season computed in the following manner.

$$x = (y - 91.25)/91.25$$

where $y = (\text{day of year} - 11)_{\text{modulo } 365}$.

if $y > 182.5$

then it is replaced by $365 - y$.

The variation of α with latitude is specified by the curve given in Figure 2. This variation is given in accurate magnetic colatitude and is used as a multiplier for that portion of α' which is greater than 1, i.e.

$\alpha = 1. + (\alpha' - 1)\alpha_M$ where α_M is the multiplier.

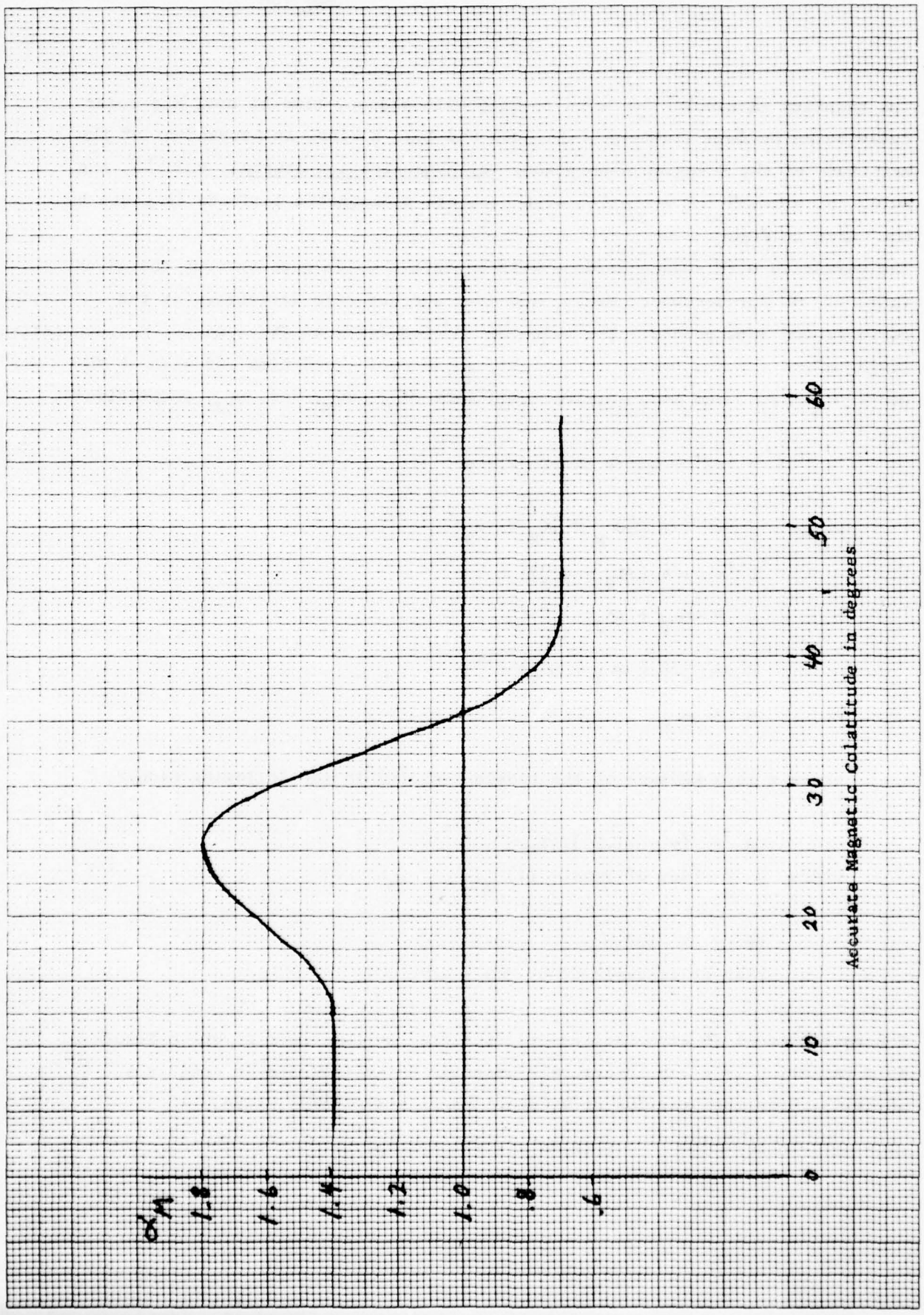


Figure 2 - α_M versus accurate magnetic colatitude in degrees.

α_M is given as follows.

$$\alpha_M = 1.4 \quad \text{if} \quad \theta_{AM} < 12.42$$

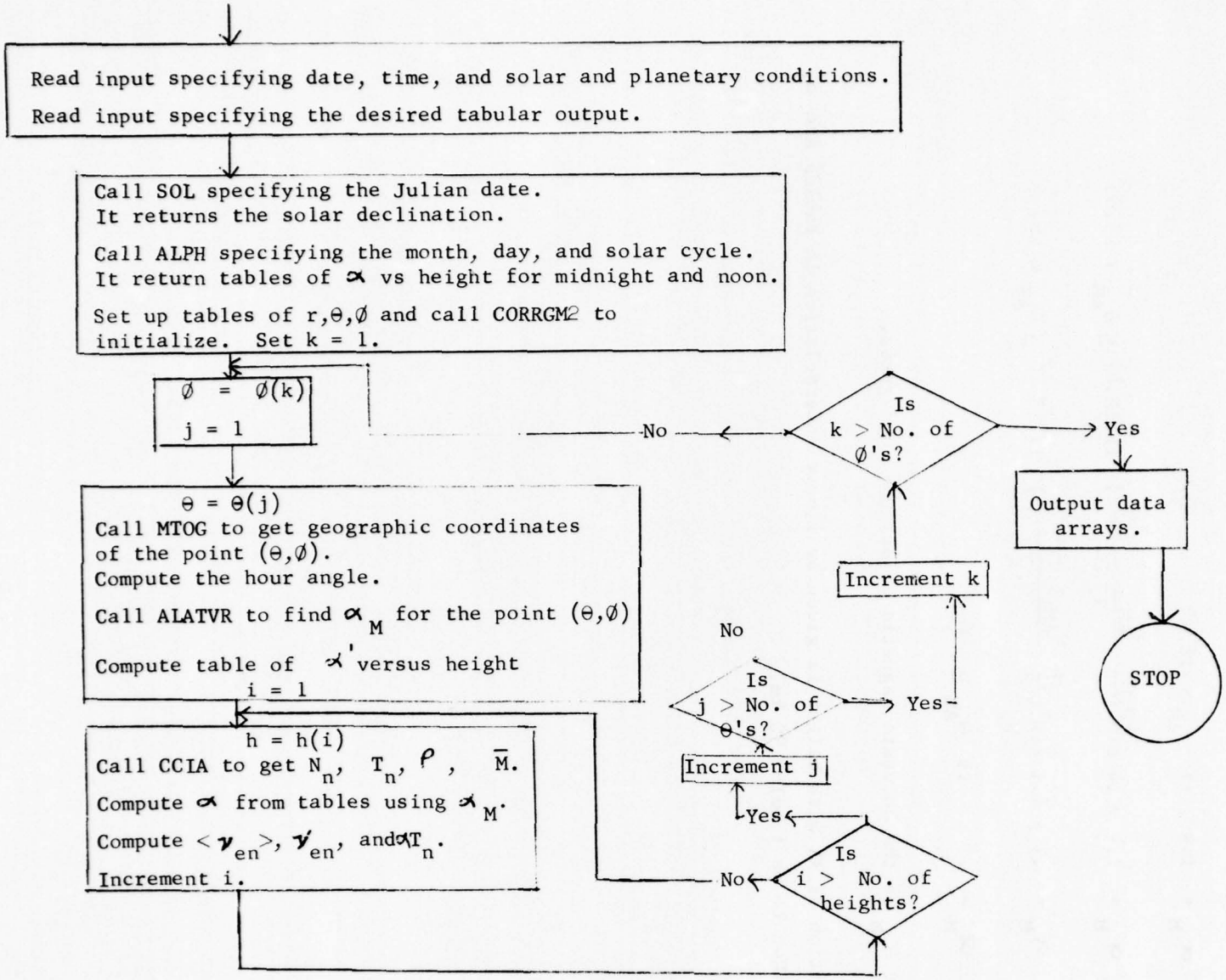
$$\alpha_M = 1.4 + .4 \sin^2\left(\frac{\pi}{2} \frac{\theta_{AM} - 12.42}{13.24}\right) \quad \text{if} \quad 12.42 \leq \theta_{AM} < 25.66$$

$$\alpha_M = .7 + 1.1 \sin^4\left(\frac{\pi}{2} \frac{\theta_{AM} - 46.36}{20.70}\right) \quad \text{if} \quad 25.66 \leq \theta_{AM} < 46.36$$

$$\alpha_M = .7 \quad \text{if} \quad \theta_{AM} \geq 46.36$$

where θ_{AM} is the accurate magnetic colatitude in degrees.

Variation of α with height is given by linear interpolation in tables and α is taken to be 1 below 50 km.



Flow Chart of Program COLPP.

SPECIAL CAUTIONS AND FEATURES

No more than 30 heights, 10 colatitudes, and 12 longitudes may be requested for the output array. Heights lower than 90. km. should not be requested.

TIMING

To generate an array of data 30 x 10 x 12, the program required 37 seconds of execution time.

ERROR MESSAGES

TOLERANCE NOT MET WITH 64*N DIVISIONS

This error message can be given from either subroutine SIMPER or S2MPER and occurs when evaluation of an integral using Simpson's rule does not converge to the proper error tolerance. This normally only happens when data is requested from outside the bounds of the atmospheric model validity, e.g. below 90 km. The program will proceed as if the integral had been properly evaluated.

XXXXERRORXXXX LATITUDE,LONGITUDE xxx.xxx xxx.xxx

This is an error message from the subroutine which converts geographic coordinates to accurate geomagnetic coordinates. The message appears if the inputted latitude is not between $+90^{\circ}$ and -90° or if the longitude is less than zero. The first quantity listed is the latitude and second quantity listed is the longitude. This message should not appear since subroutine DICOORD should check for this before CORRGM is called. The subroutine will try to correct the longitude by adding 360° but will proceed anyway.

NO TRIANGLE FOR XLAT,XLONG= xxx.xxx xxx.xxx

This may occur when the point for which the accurate geomagnetic coordinates are desired is in the region of the accurate magnetic pole. The region is divided into triangles having the pole as one vertex. If a triangle cannot be located, it is assumed that this is the accurate magnetic pole and the above message will appear giving the coordinates in the geographic system in degrees. No value is assigned to the accurate geomagnetic longitude in this case. Computation then proceeds as normal.

SUBROUTINES

COLPP This is the main program. It reads in all data cards, sets up the tables of h, θ, and ϕ and sequences through to compute the output tables of $\langle \nu_{en} \rangle$, ν_{en}, and T_n. A flow chart of COLPP is given on page 10. COLPP calls SOL, ALPH, CORRGM2, MTOG, ALATVR, and CCIA.

ALPH Given a particular month, day, and solar cycle indicator, subroutine ALPH generates tables of α', the ratio between electron and neutral temperature. It generates tables of α' versus height for midnight and noon of the particular day. It calls subroutine POLY. It is called by COLPP.

POLY This subroutine generates a quartic given three data points with the assumption that the derivative is to be zero at the first and third points. It assumes the x-value is -1, 0, and 1 respectively for each of the three points. It is called by ALPH.

ALATVR This subroutine computes the variation of α with latitude which we have called α_M when given the geographic latitude and longitude of a point. It is called by COLPP and in turn calls subroutine CORRGM.

CCIA This is the main subroutine of a set of subroutines which generate atmospheric information for any given date and time when certain solar and planetary information is supplied. It is called by COLPP and in turn calls functions BMZ, SIMPER, S2MPER, TZ1, and TZ2.

GRAV This function is part of the CCIA package. It computes gravity and is used by functions TGPN and TGRN.

BMZ The function is part of the CCIA package. It computes the mean molecular weight as a function of height. It is used by subroutine CCIA and function TGRN.

TZ1 This function is part of the CCIA package. It is used to compute the temperature as a function of height in certain regions. It is used by subroutine CCIA and functions TGRN and TGPN.

TZ2 This function is part of the CCIA package. It is used to compute the temperature as a function of height in certain regions. It is used by subroutine CCIA and function TGSN.

TGPN This function is used to calculate the gravity divided by the temperature in the region above 100 km. It is part of the CCIA package and is used by function S2MPER.

TGSN This function is used to calculate the gravity divided by the temperature in the region above 500 km. It is part of the CCIA package and is used by function S1MPER.

TGRN This function is used to calculate the gravity divided by the temperature in the region below 100 km. It is part of the CCIA package and is used by function S2MPER.

S1MPER This function calculates a definite integral using Simpson's rule. It is called by subroutine CCIA and uses function TGSN.

S2MPER This function calculates a definite integral using Simpson's rule. It is called by subroutine CCIA and uses functions TGPN and TGRN.

SOL This subroutine computes the solar declination angle and right ascension angle given the modified Julian date. It is called by COLPP and uses function ANGLE2.

ANGLE2 This function reduces an angle to the range 0 to 2π. It is used by subroutine SOL.

DICOORD This subroutine through entry point MTOG converts a point in colatitude and longitude from dipolar geomagnetic to geographic coordinates. It is used by COLPP.

CORRGM2 This subroutine takes points in the geographic coordinate system and converts them to accurate geomagnetic coordinates. A call to CORRGM2 causes the transforming array to be read in. For points to be converted, a call must be made to CORRGM. CORRGM2 is called by COLPP while CORRGN is called by ALATVR.

ACCURACY

The accuracy of the output of COLPP is limited by the accuracy of the models given in the CCIA subroutine. The size of the grid parameters chosen for the output table may in turn causes inaccuracies in the tabular interpolations when the data is used in the ray-trace program.

COMMENTS ON USAGE

The file of three dimensional collision frequency data should be stored by the user by either writing it on tape or requesting a permanent file.

FILE DESCRIPTIONS

The program card for COLPP is as follows:

PROGRAM COLPP(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE8,TAPE1)

INPUT / TAPE5 — This is the system input file under normal operation. It consists of the cards described on page 2.

OUTPUT / TAPE6 — This is the system output file under normal operation. It contains the information described on pages 3 and 4 as printed output plus any error messages from the program or the system.

TAPE1 — This is the permanent file which contains the transformation array for going from geographic to accurate geomagnetic coordinates. It is stored in both system I and II under the file name MJDATA with ID=LOGICON.

TAPE8 — This is the end product of program COLPP, a file of three-dimensional tables of information needed to compute collision frequency. This file is to be used by collision frequency subroutine COLAF1 of the ray trace program. This file is unformatted.

Record 1: NHT, NTH, NPH

NHT - The number of height entries (≤ 30)

NTH - The number of colatitude entries (≤ 10)

NPH - The number of longitude entries (≤ 12)

Record 2: (HT(I), I = 1,NHT)

This is the table of height values in km.

Record 3: (TH(J), J = 1,NTH)

This is the table of colatitude values in radians.

Record 4: (PH(K), K = 1,NPH)

This is the table of longitude values in radians.

Record 5: (GNN(I,1,1), I = 1,NHT)

where GNN is ν_{en}, the number of collisions/sec. with neutral molecules.

Record 6: (AT(I,1,1), I = 1,NHT)

where AT is T_M, the electron temperature in ^{o}K.

Record 7: (GEN(I,1,1), I = 1,NHT)

where GEN is $\langle \nu_{en} \rangle$, the collision frequency using the pressure formula.

Records 8; 9; 10 are repeats of records 5, 6, 7 except the subscript becomes (I,1,2) for the 2nd longitudinal entries. These three records are repeated until date is read in for all colatitudes and longitudes.

SAMPLE DECK SETUP

```
Job Card w/ CM140000.
ATTACH(SOURCE,COLPPX3693818,ID=LANGW)
FTN(I=SOURCE,B=BCOL)
ATTACH(TAPE1,MJDATA,ID=LOGICON)
REQUEST(TAPE8,*PF)
BCOL.
CATALOG(TAPE8,COLLDATAX3693818,ID=LANGW)
7/8/9
    Data cards for COLPP (page 2)
6/7/8/9
```

NAME:	COLAF1, revision 0, subroutine, PML 140
CATEGORY:	Subroutine for Ray Tracing Program
TITLE:	Collision frequency subroutine for tables produced by COLPP
LANGUAGE:	CDC extended Fortran - version 4
PROGRAMMER:	B. M. Langworthy, Parke Mathematical Laboratories, Inc.
DATE:	February 27, 1976

DESCRIPTION

The collision frequency subroutine, COLAF1, was written for compatibility with the 1974 version of the Jones raytracing program as used by AFCRL-LII. (See PML 121). It was written in conjunction with a preprocessing program which supplies tabular data on permanent file or tape. The preprocessing program is COLPP and is described in PML 139. COLAF1 differs from other collision frequency subroutines used by the ray trace program in that it does not compute the derivatives of the electron collision frequency. This requires modifications to the index of refraction subroutine as well. This subroutine was designed to be used with a specially modified version of the index of refraction subroutine AHWFNC in which the imaginary part of the index of refraction is used only for the evaluation of the absorption.

INSTRUCTION SET

Since the computation of the absorption requires a large amount of additional core storage and increases the computation time, subroutine COLAF1 should be kept on a separate file and loaded only when the computation of the absorption is desired. This will result in a warning that COLFRZ is an unsatisfied external, but no execution error will be encountered.

The data used by subroutine COLAF1 is described under FILE DESCRIPTIONS and is referenced by local file name TAPE8. This is the file produced by the preprocessing program COLPP referred to above.

PRECEDING PAGE BLANK-NOT FILMED

Subroutine COLAF1 uses the following information from the ray tracing program:

R(1) - radius in km.
R(2) - colatitude in radians
R(3) - longitude in radians
} Position for which the electron collision frequency is desired.

EARTHR - (Also W(2).) The radius of the earth in km.
F - (Also W(6).) The transmission frequency in MHz.
PIT2 - A constant with value 2π.

The subroutine then returns the following quantities.

MODZ - an alphanumeric identifier which gives the name of collision frequency, subroutine COLAF1 for print out purposes.
Z - the refractivity due to collisions with neutral and ionized molecules. (dimensionless)

STORAGE REQUIRED

Using COLAF1 to compute the absorption requires an additional 31,000 octal words of core storage.

ALGORITHM

Subroutine COLAF1 reads in tables of $\langle \nu_{en} \rangle$, ν_{en}, and $\propto T_n$ as a function of h, θ, and ϕ where h is the height above the earth's surface and θ and ϕ are colatitude and longitude in dipolar geomagnetic coordinates. The refractivity due to collisions is computed as follows:

$$Z = \frac{\nu}{2\pi f \times 10^6}$$

where Z is the refractivity due to the collision of electrons with neutral and ionized molecules.
ν is the collision frequency in collisions/sec.
f is the transmission frequency in MHz.

$$\nu = \nu_{en} + \nu_{ei}$$

where ν_{en} is the collision frequency with neutral molecules in collisions/sec. This is given by input tables.

ν_{ei} is the collision frequency with ionized molecules in collisions/sec.

$$\nu_{ei} = \frac{5.5 N_e}{(\alpha T_n)^{3/2}} \log_e \left\{ \frac{220. (\alpha T_n)}{N_e^{1/3}} \right\}$$

where N_e is the electron density in electrons per cc. This is obtained from the electron density subroutine.

α is the ratio of electron to neutral temperature

T_n is the neutral temperature in ^{o}K.

The electron temperature αT_n is obtained from the input tables. For details on how it is generated, see PML 139.

Table interpolation for αT_n is linear in h, θ, and ϕ while interpolation for ν_{en} is linear in θ and ϕ and logarithmic in h in the following sense. If we have interpolated in θ and ϕ, we have $\nu_{en}(h_i,\theta,\phi)$ and $\nu_{en}(h_{i+1},\theta,\phi)$. We then interpolate for $\nu_{en}(h,\theta,\phi)$ as follows:

$$\nu_{en}(h,\theta,\phi) = \exp\left\{ \log_e(\nu_{en}(h_i,\theta,\phi)) + \left(\frac{h - h_i}{h_{i+1} - h_i}\right) \cdot \right.$$

$$\left. (\log_e(\nu_{en}(h_{i+1},\theta,\phi)) - \log_e(\nu_{en}(h_i,\theta,\phi)) \right\}$$

where $h_i \leq h < h_{i+1}$.

SPECIAL CAUTIONS AND FEATURES

Subroutine COLAF1 is valid only in the range of height values for which the data is supplied by COLPP. An extrapolation is made using the first two table points for values of height greater than or equal to 80 km but less than the first table entry. Below this the collision frequency remains a constant. It is also assumed to be a constant equal to the last available table entry beyond the upper limits of height and the table limits in colatitude and longitude.

Since the computation of ν_{ei}, the collision frequency of electrons with ionized molecules, depends on knowing the electron density, the refractivity due to electron density must be computed prior to calling COLAF1. This is done in the ray tracing program but care should be taken to see that this is done in any other use of COLAF1 such as that collision frequency profile plotting program, COLPLT. (See PML 141.)

TIMING

Computation of absorption using COLAF1 increases the execution time by approximately 36%.

ERROR MESSAGES

None.

SUBROUTINES

None.

ACCURACY

The accuracy of COLAF1 is dependent on the interval chosen for the grid parameters of the input table. For the most part the input quantities ν_{en} and $\propto T$ are not subject to large variations in colatitude and longitude. The variation of ν_{en} with height is great especially in the region below 200 km. In this area the spacing at data points becomes more critical.

FILE DESCRIPTIONS

COLAF1 reads data from TAPE8 which is either a tape or a permanent file. Data is read in only once in the course of a program execution. This file is created by collision frequency preprocessing program, COLPP, as described in PML 139. Subroutine COLAF1 never writes on TAPE8.

TAPE8 - a file of three-dimensional tables of information needed to compute collision frequency. This file is unformatted.

Record 1: NHT, NTH, NPH

NHT - The number of height entries (≤ 30)

NTH - The number of colatitude entries (≤ 10)

NPH - The number of longitude entries (≤ 12)

Record 2: (HT(I), I = 1,NHT)

This is the table of height values in km.

Record 3: (TH(J), J = 1,NTH)

This is the table of colatitude values in radians.

Record 4: (PH(K), K = 1,NPH)

This is the table of longitude values in radians.

Record 5: (GNN(I,1,1), I = 1,NHT)

where GNN is ν_{en}, the number of collisions/sec. with neutral molecules.

Record 6: (AT(I,1,1), I = 1,NHT)

where AT is $\propto T_M$, the electron temperature in ^{o}K.

Record 7: (GEN(I,1,1), I = 1,NHT)

where GEN is $<\nu_{en}>$, the collision frequency using the pressure formula.

Record 8; 9; 10 are repeats of records 5, 6, 7 except the subscript becomes (I,1,2) for the 2nd longitudinal entries. These three records are repeated until data is read in for all colatitudes and longitudes.

SAMPLE DECK SETUP

```
Job Card with an additional 31000 words of storage
ATTACH(BRAY,BRAYTRACEX3693818,ID=LANGW)
ATTACH(BCOL,BCOLAF1X3693818,ID=LANGW)
ATTACH(TAPE6,ELECTDATAX3693818,ID=LANGW)
ATTACH(TAPE8,COLLDATAX3693818,ID=LANGW)
LDSET(PRESET=ZERO)
LOAD(BRAY)
LOAD(BCOL)
EXECUTE(NITIAL)
7/8/9
          Data Cards
6/7/8/9
```

where BRAYTRACE is a permanent file containing the binary version of the ray trace program without a collision frequency subroutine,

BCOLAF1 is a permanent file containing the binary version of subroutine COLAF1,

ELECTDATA is a permanent file containing any necessary electron density data,

and COLLDATA is a permanent file containing the information specified under FILE DESCRIPTIONS. (See above.)

NAME: COLPLT, revision 0, program, PML 141
CATEGORY: Plotting program for use with ray tracing program
TITLE: Collision frequency profile plotting program
LANGUAGE: CDC extended Fortran - version 4
PROGRAMMER: B. M. Langworthy, Parke Mathematical Laboratories, Inc.
DATE: March 1, 1976

DESCRIPTION

Program PROPLT may be used to plot collision frequency profiles for any collision frequency subroutine used by the ray tracing program. These plots are self scaling and need no input other than the (θ,ϕ) location for which the profile of collision frequency vs. height is desired and the input required by the collision frequency subroutine and the electron density subroutine if such a subroutine is required for collision frequency computation. The collision frequency subroutine and the (θ,ϕ) location will be indicated on the plot.

INSTRUCTION SET

Input to program COLPLT is made through the W-array in the same manner as for the ray trace program RAYTRACEBL. (See PML 121.) The program requires any input required by the collision frequency subroutine such as area W250-299 and TAPE8. It also requires any input needed for a companion electron density subroutine such as W100-149 and TAPE6 if such a subroutine is needed. To determine the input needed see Attachment 4 of PML 121. The program establishes the earth's radius at 6370 km., but this may be overridden by input to W2. The user must supply the following quantities:

W4 Geomagnetic colatitude of the transmitter
W5 Geomagnetic longitude of the transmitter

Please note that this differs from the definitions of W4 and W5 in the ray trace program, where W4 is the geographic latitude of the transmitter and W5 is the geographic longitude of the transmitter.

Data may be entered into the W-array in any order but each card may contain only one data entry. If two cards contain data for the same W-array item then the last card entered will take precedence. W-array cards should be entered in the following format.

Cols	1 - 3	- The number of the W-array entry (Do not enter "W")
Cols	4 - 17	- The value to be entered. (Be sure a decimal point is included.)
Col	18	- A "1" in this column indicates that the entry is in degrees.
Col	19	- A "1" in this column allows earth centered angles to be given in great circle distance along the ground in km. (For example if, instead of specifying a latitude, one could instead give a distance from the equator.)
Col	20	- A "1" in this column indicates that the data entry is in nautical miles instead of km.
Col	21	- A "1" in this column indicates that the data entry is in feet instead of km.
Cols	22 - 80	- May be used for comments.

Entries to the W-array should be followed by a card with Cols 1-3 blank to indicate that all data for the W-array has been entered.

For each profile desired new values may be entered to the W-array for the location (θ,ϕ) and the collision frequency subroutine but they must be followed by a blank card. If new values are not entered for a particular quantity then the previously assigned value is used. Do not input W6, the transmission frequency, as this must be set to 1.

Output in each case will be a profile plot of the log to the base 10 of the collision frequency in collision/sec. versus height above the earth's surface in km. If the value of the $\log_{10}$ of the collision frequency exceeds 20, the information will be printed instead of plotted. A sample profile plot is given in Attachment 1.

STORAGE REQUIRED

The amount of storage required varies with the collision frequency and companion electron density subroutine used. For collision frequency subroutine COLAF1 used with electron density subroutine E14994, the storage required is 200,000 octal words. This should represent a worst case.

ALGORITHM

Not applicable.

SPECIAL CAUTIONS AND FEATURES

The input to W4 and W5 differs from that used by RAYTRACEBL in that W4 is in geomagnetic colatitude instead of North geographic latitude and W5 is in geomagnetic longitude instead of geographic longitude.

No input should be made to W6, the transmission frequency, as this must be 1 MHz to obtain the proper collision frequency.

TIMING

Allow 2 seconds for load time and .4 seconds per plot depending on the complexity of the collision frequency subroutine.

ERROR MESSAGES

When print out of the table of height versus plasma frequency occurs, it is because the values exceeded allowable plotting bounds.

Any other error messages are from the AFCRL/CALCOMP plotting software or the collision frequency subroutine.

SUBROUTINES

COLFRZ — This is a collision frequency subroutine normally used by RAYTRACEBL. Given a particular value of (r,θ,ϕ) it will return the collision frequency and its derivatives via labeled common area /ZZ/.

ELECTX — This is called by COLPLT in case the collision frequency subroutine COLFRZ must be supplied the electron density. If a particular collision frequency subroutine does not need the electron density, then a dummy subroutine should be supplied.

Other subroutines called are AFCRL plotting software. Plotting subroutines used are: PLTID3, PLOT, ENDPLT, AXIS, LINE, SYMBOL, and NUMBER.

ACCURACY

Points are obtained every 5 km. in height. Straight line connections are plotted between points.

COMMENTS ON USAGE

This program was written for collision frequency subroutines which must have the electron density for their calculations. If an electron density calculation is not required for a particular collision frequency subroutine a dummy electron density subroutine must still be supplied. Such a subroutine would be:

```
SUBROUTINE ELECTX
RETURN
END
```

FILE DESCRIPTIONS

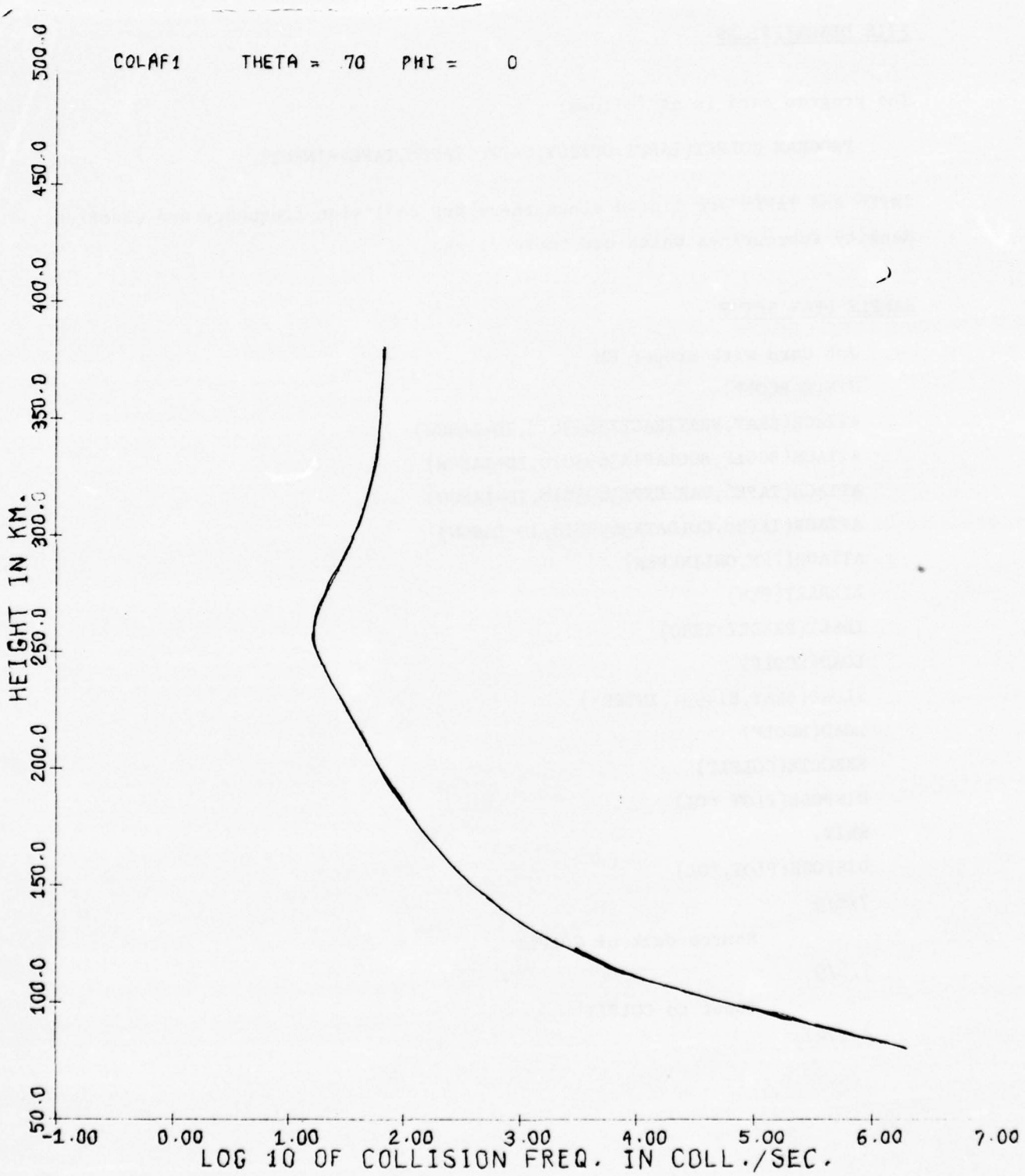
The program card is as follows:

```
PROGRAM COLPLT(INPUT,OUTPUT,TAPE6,TAPE8,TAPE4=INPUT)
```

TAPE6 and TAPE8 are listed since there are collision frequency and electron density subroutines which use these files.

SAMPLE DECK SETUP

```
Job Card with proper CM
FTN(B=BCOLP)
ATTACH(BRAY,BRAYTRACEX3693818,ID=LANGW)
ATTACH(BCOLF,BCOLAF1X3693818,ID=LANGW)
ATTACH(TAPE6,VAR2EXPX3693818,ID=LANGW)
ATTACH(TAPE8,COLDATX3693818,ID=LANGW)
ATTACH(PEN,ONLINEPEN)
LIBRARY(PEN)
LDSET(PRESET=ZERO)
LOAD(BCOLP)
SLOAD(BRAY,E14994,INTER3)
LOAD(BCOLF)
EXECUTE(COLPLT)
DISPOSE(PLOT,*OL)
EXIT.
DISPOSE(PLOT,*OL)
7/8/9
          Source deck of COLPLT
7/8/9
          Input to COLPLT
6/7/8/9
```



Attachment 1 - Sample plotted output.

NAME: TRNSGN, revision 0, function, PML 142
CATEGORY: Antenna gain simulation of a transmitting antenna
TITLE: Antenna Gain for a Logarithmic Antenna
LANGUAGE: CDC extended Fortran - version 4
PROGRAMMER: B. M. Langworthy, Parke Mathematical Laboratories, Inc.
DATE: June 7, 1976

DESCRIPTION

Function TRNSGN returns the gain in decibels of a logarithmic antenna when given an elevation angle, an azimuth angle, and a frequency of transmission. The antenna gain is modeled on the curves given in Attachments 1 and 2.

INSTRUCTION SET

To obtain the gain in decibels of the transmitter, use function TRNSGN(EL,AZ,F) where

EL is the elevation angle in radians.
AZ is the azimuth angle in radians in the dipolar coordinate system with boresight at azimuth angle 64.88° dipolar.
F is the frequency in MHz.

Values of TRNSGN range from +14 to -25 db. This function is only valid for frequencies between 8 and 28 MHz. Below 8 MHz the value of the gain is that for 8 MHz; and above 28 MHz the value of the gain is that for 28 MHz. Function TRNSGN has a common area /KEY/ with two variables: G and GELF.

G is the gain solely due to the azimuth pattern or Attachment 2. To obtain the figures given in Attachment 2, 15 decibels must be subtracted from G.

GELF is the gain solely due to the elevation pattern or Attachment 1.

STORAGE REQUIRED

Function TRNSGN requires 325 octal words of core storage.

ALGORITHM

Antenna gain (G) as a function of azimuth (α'), elevation (β), and frequency (f) for the transmitter.

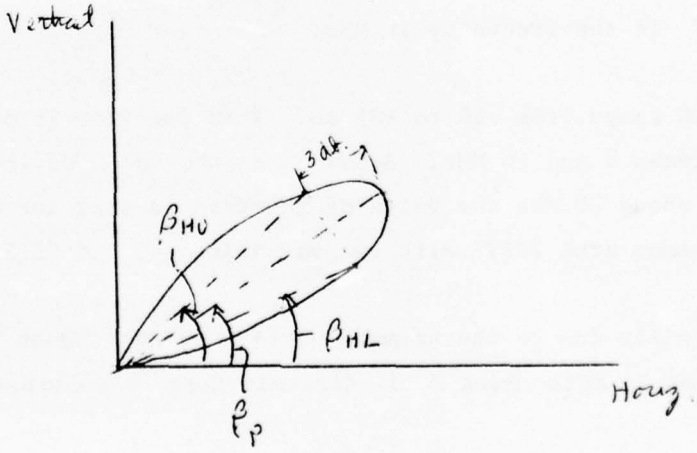
Specification of the antenna gain solely due to the elevation (G_E) is given below. This is then combined with the azimuthal variation to give G.

To determine G_E we must first interpolate with respect to frequency to find the quantities: β_{HL}, β_p, β_{HU}

where β_{HL} is the elevation angle at which power is half that of peak and which is lower than the elevation angle of peak power,

β_p is the elevation angle of the peak power,

and β_{HU} is the elevation angle at which power is half that of peak and which is higher than the elevation angle of peak power.



Below are given the tables of f vs β_{HL}, β_p, and β_{HU}.

f	β_{HL}	β_p	β_{HU}
8	6.5	13.5	21.5
12	4.6	9.1	13.7
16	3.5	7.0	11.2
20	2.8	5.6	8.0
24	2.65	4.6	6.5
28	1.9	4.0	5.6

Angles are in degrees; frequency in MHz.

If $\beta \leq \beta_p$

$$A = \frac{\cos^{-1}(.7)}{\beta_p - \beta_{HL}}$$

$$\beta_L = \beta_p - \frac{90.}{A}$$

If $\beta \leq \beta_L$

$$G_E = -10.$$

If $\beta_L < \beta \leq \beta_p$

$$G_E = 10*(\cos A(\beta_p - \beta)-1)$$

If $\beta > \beta_p$

$$A = \frac{\cos^{-1}(.7)}{\beta_{HU} - \beta_p}$$

$$\beta_U = \beta_p + \frac{90.}{A}$$

If $\beta \geq \beta_U$

$$G_E = -10.$$

If $\beta_p < \beta < \beta_U$

$$G_E = 10*(\cos A(\beta - \beta_p)-1.)$$

Otherwise $G_E = -10.$

We then compute G as follows, converting α' to degrees off of boresight, α

$\alpha = \alpha_D - 64.88$ where α_D is α' in degrees.

If $-80 < \alpha < 80$

$$G = (C_f \cdot \sqrt{\cos 1.125\alpha}\ ) + G_E) - 15.$$

where

$C_f = 29$ if $f \leq 13$.

$C_f = 29 + \frac{13-f}{7}$ if $13 < f < 20$

$C_f = 28$ if $f \geq 20$.

If $-98.66 \geq \alpha \geq -132$

$$G = (C_f \cdot \sqrt{\cos 2.7(132 + \alpha)}\ ) + G_E - 15.$$

where

$C_f = 7$ if $f \leq 13$

$C_f = 7 - 0.75\ \frac{(f-13)}{7}$ if $13 < f < 20$

$C_f = 6.25$ if $f \geq 20$.

If $-132 > \alpha \geq -180$.

$$G = (C_f \cos 1.7\ (132 + \alpha) + G_E) - 15.$$

where C_f is computed as above.

If $100 \leq \alpha \leq 140$

$$G = (C_f \cos 3\ (120 - \alpha) + G_E) - 15.$$

where

$C_f = 4$ if $f \leq 13$

$C_f = 4 - 2(\frac{f - 13}{7})$ if $13 < f < 20$

$C_f = 2$ if $f \geq 20$.

Otherwise $G = -15. + G_E$.

If $G < -15.$ or $G_E \leq -10.$, G is set to -15.

SPECIAL CAUTIONS AND FEATURES

While answers will be given beyond the modeled frequency range of 8 to 28 MHz, these answers will not be valid. Angle arguments to TRNSGN must be in radians.

The function has a labeled common area KEY which may be omitted if it conflicts with other labeled common.

TIMING

Execution time of TRNSGN will vary with the azimuthal region desired but is approximately 2 milliseconds.

ERROR MESSAGES

None,

SUBROUTINES

None.

ACCURACY

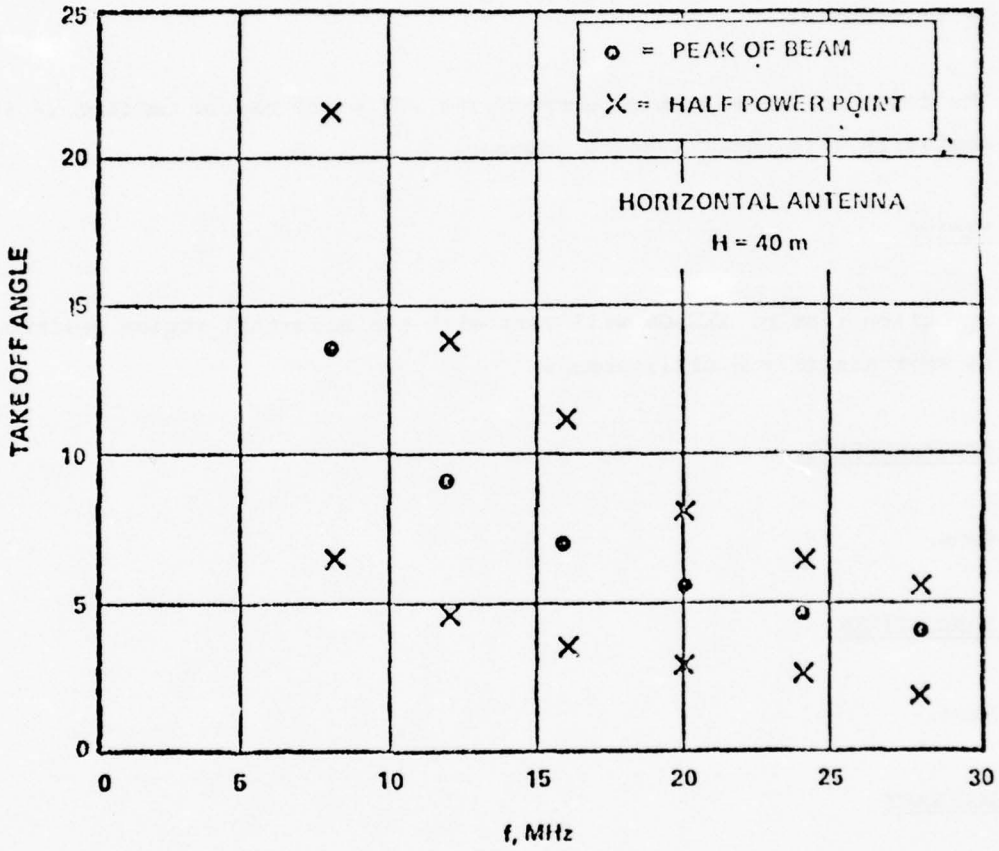
Attachments 3 and 4 give a comparison of the antenna pattern in the azimuthal plane with figures from Attachment 2 superimposed. In the elevation direction G_E is exactly zero at the peak power point and exactly -3 at the half power points. A typical elevation cross section is given in Attachment 5.

FILE DESCRIPTIONS

None.

SAMPLE DECK SETUP

The source deck of function TRNSGN is currently stored on permanent file in SYSI under LOGANTX3693818,ID=LANGW.



Attachment 1

Specification of the variation of power with frequency and elevation angle of the log periodic antenna.

BEST AVAILABLE COPY

Log Periodic

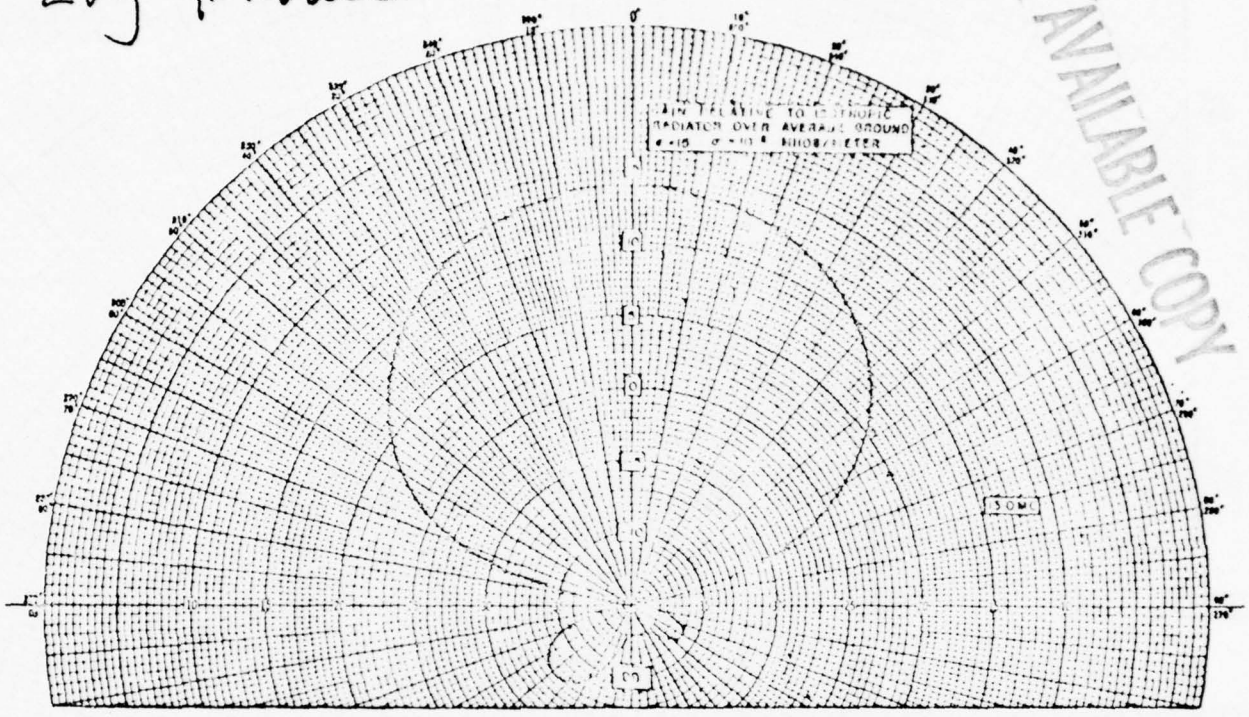


Figure 29A. Azimuth Plane Patterns, 237B Antennas

Freq Independent

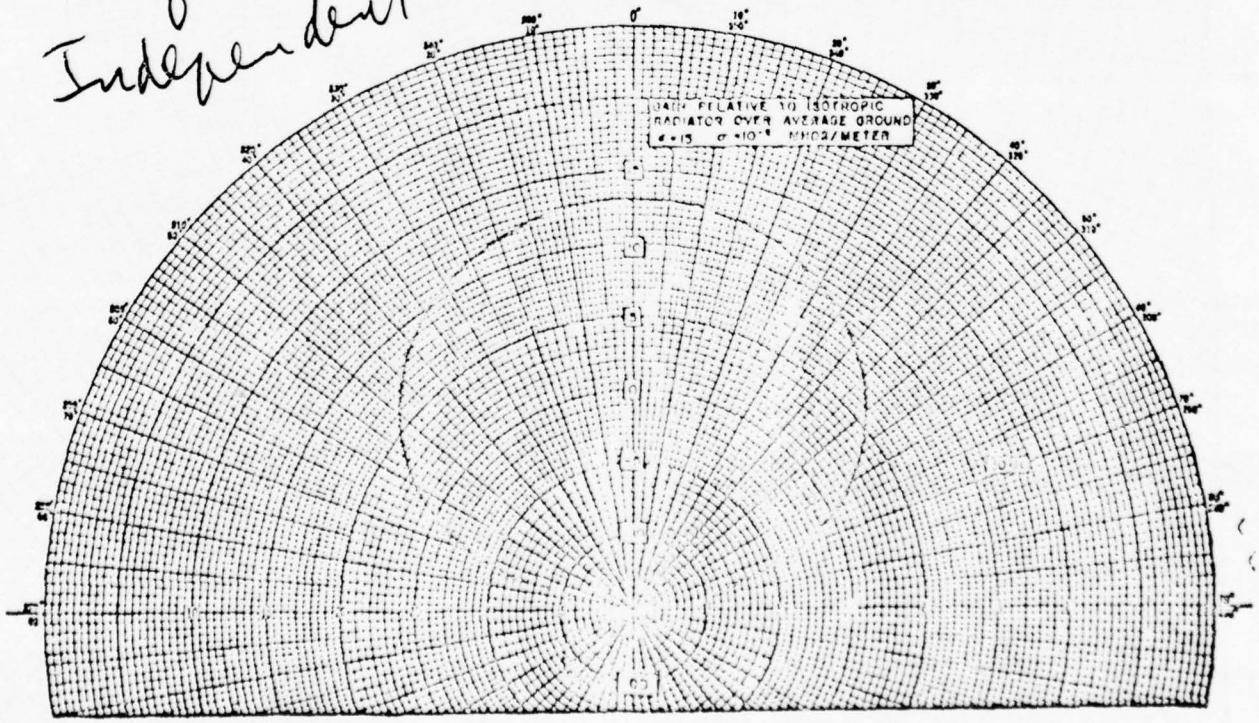
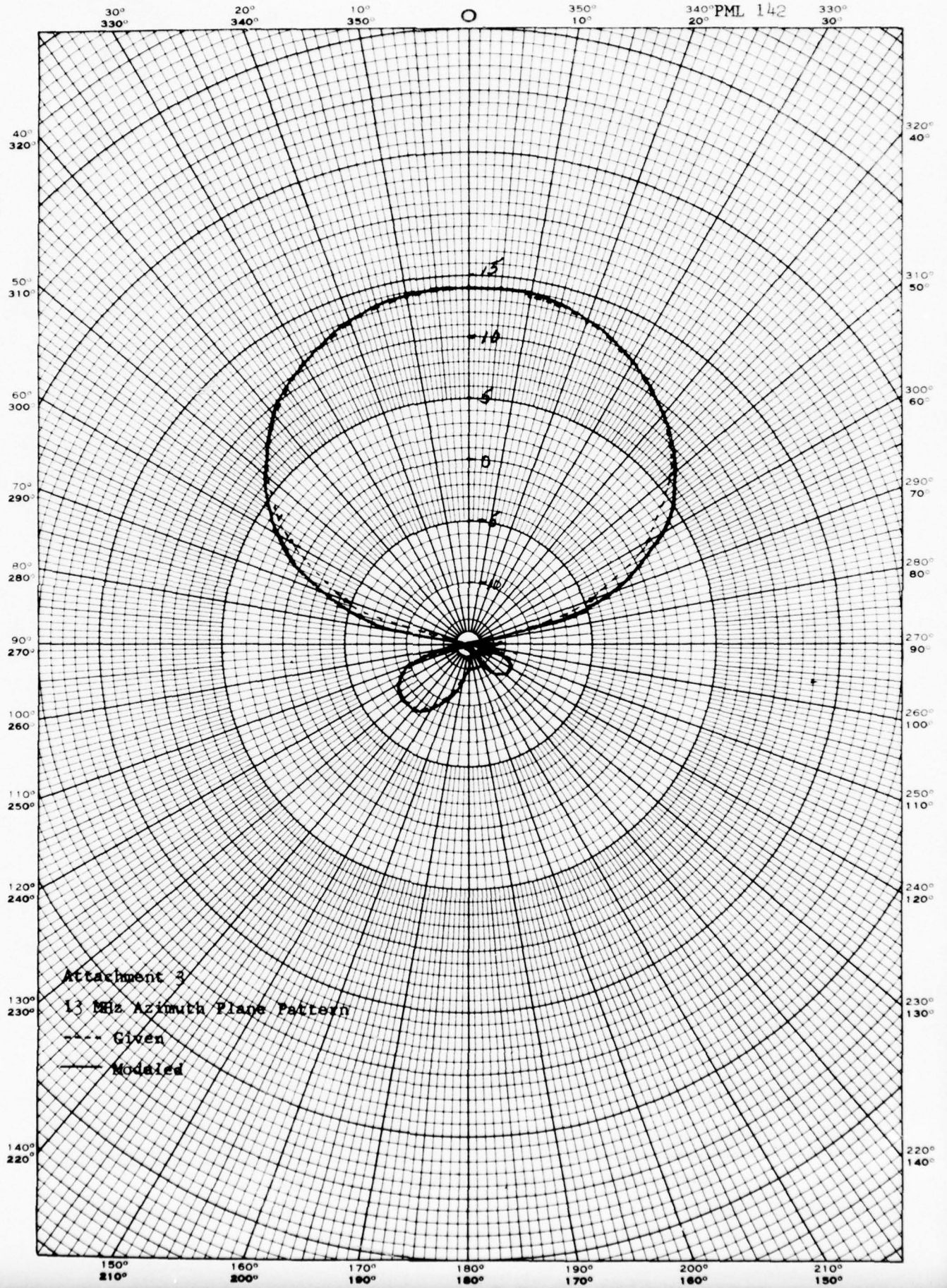
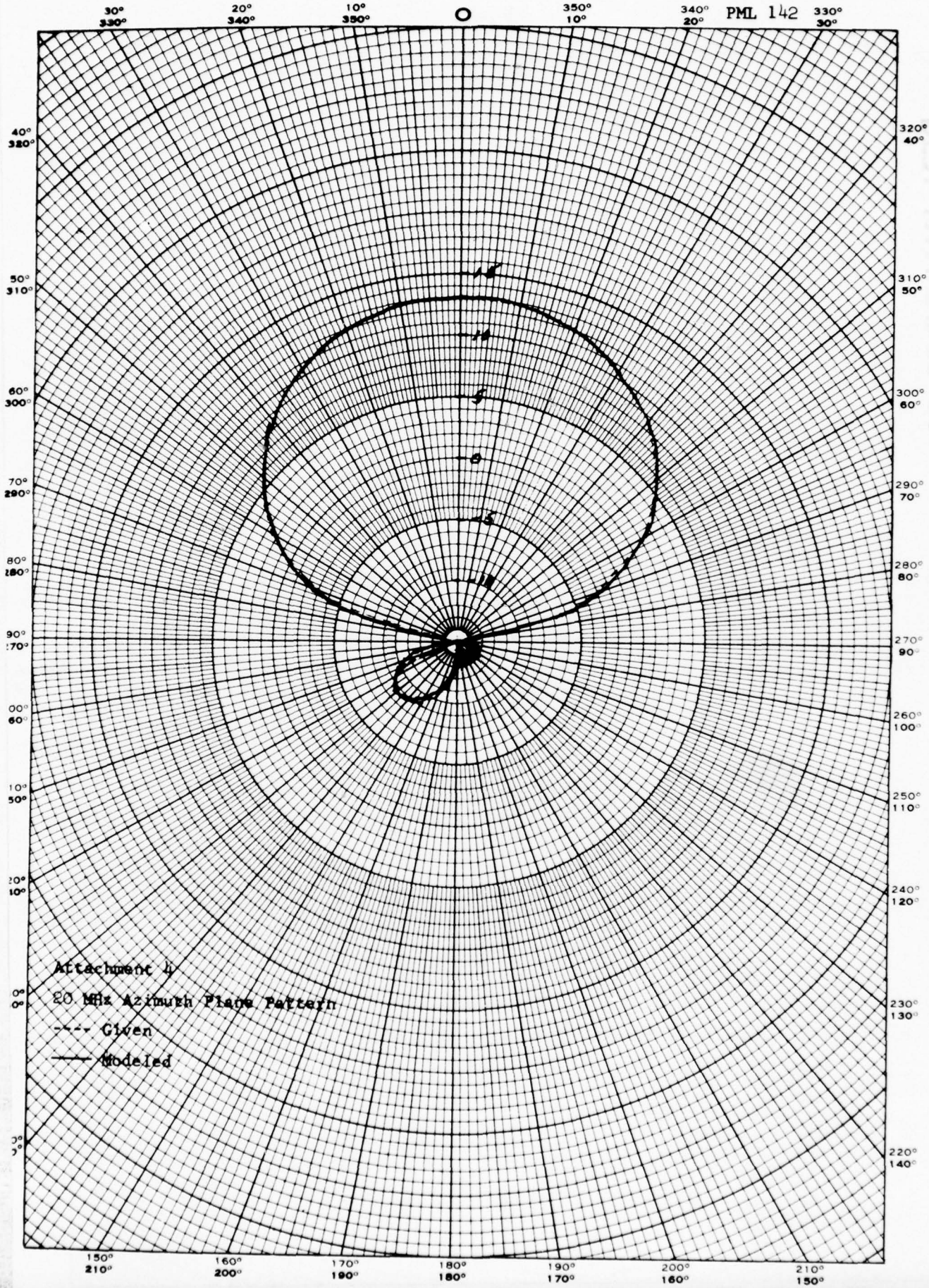
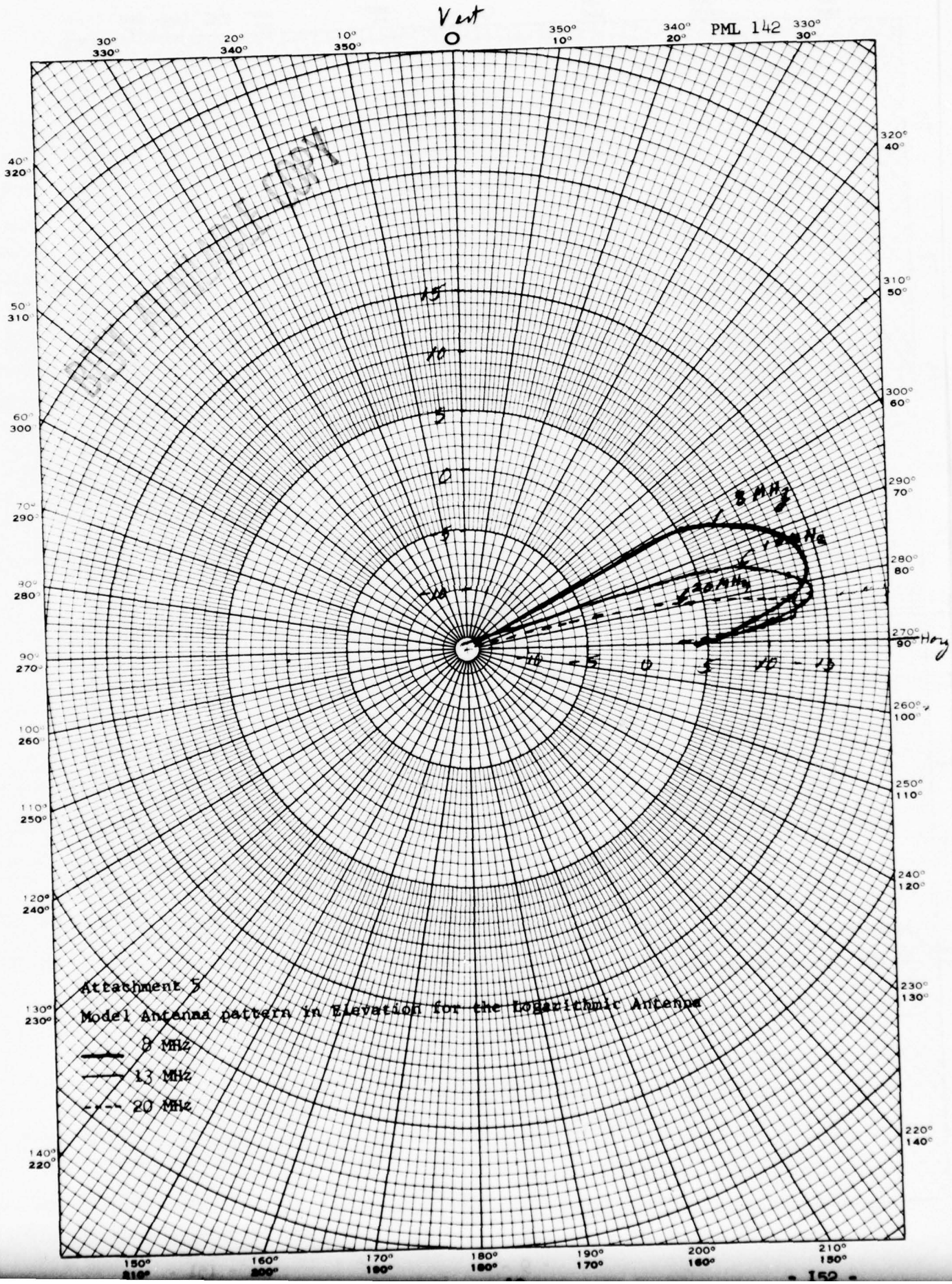


Figure 29B. Azimuth Plane Patterns, 237B Antennas

Attachment 2: Specification of the variation of power with frequency and azimuth off of boresight of the log periodic antenna.


PML 142
0°
10° 350°
20° 340°
30° 330°
350° 10°
340° 20°
330° 30°
40° 320°
50° 310°
60° 300°
70° 290°
80° 280°
90° 270°
100° 260°
110° 250°
120° 240°
130° 230°
140° 220°
150° 210°
160° 200°
170° 190°
180°
190° 170°
200° 160°
210° 150°
320° 40°
310° 50°
300° 60°
290° 70°
280° 80°
270° 90°
260° 100°
250° 110°
240° 120°
230° 130°
220° 140°
15
10
5
0
-5
-10
Attachment 3
13 MHz Azimuth Plane Pattern
---- Given
—— Modaled


PML 142
30° 330°
20° 340°
10° 350°
0
350° 10°
340° 20°
330° 30°
40° 320°
320° 40°
50° 310°
310° 50°
60° 300°
300° 60°
70° 290°
290° 70°
80° 280°
280° 80°
90° 270°
270° 90°
260° 100°
250° 110°
240° 120°
230° 130°
220° 140°
150° 210°
160° 200°
170° 190°
180° 180°
190° 170°
200° 160°
210° 150°
14
9
5
0
-5
-11
Attachment 4
20 MHz Azimuth Plane Pattern
---- Given
—— Modeled



Attachment 5

Model Antenna pattern in Elevation for the Logarithmic Antenna

8 MHz

13 MHz

20 MHz

NAME: RCVRGN, revision 0, function, PML 143
CATEGORY: Antenna gain simulation of a receiving antenna
TITLE: Antenna Gain for a Rhombic Antenna
LANGUAGE: CDC extended Fortran - version 4
PROGRAMMER: B. M. Langworthy, Parke Mathematical Laboratories, Inc.
DATE: June 8, 1976.

DESCRIPTION

Function RCVRGN returns the gain in decibels of a rhombic antenna when given an elevation angle, an azimuth angle, and a frequency of transmission. The antenna gain is modeled on a set of antenna patterns which are attached.

INSTRUCTION SET

To obtain the gain in decibels of the receiver, use function RCVRGN(EL,AZ,F) where

EL is the elevation angle in radians
AZ is the azimuth angle in radians in the dipolar coordinate system with boresight at azimuth angle 64.88° dipolar
F is frequency in MHz.

Values of RCVRGN range from 31.4 to -5 db but may be -100 if a frequency is given which is outside of the values for which the antenna patterns are defined. The function is only valid for frequencies between 8 and 30 MHz. Function RCVRGN has a common area /KYR/ with three variables: PPM, PW, PEl. These variables correspond to P_{P_M}, P_{P_T}, and P_{E_1} in the algorithm which follows.

STORAGE REQUIRED

Function RCVRGN requires 2106 octal words of core storage.

ALGORITHM

The algorithm is broken into two sections, the first for the low band of the rhombic antenna and the second section for the high band.

1. Equations for the low band of the rhombic receiver.

α = azimuth angle from boresight in degrees

β = elevation angle in degrees.

f = transmission frequency in MHz.

Peak power for all lobes is a function of elevation angle.

$$RCVRGN = (P_{P_T} + 5) * \frac{(P_{P_M} + 5)}{(P_{E_1} + 5)} - 5.$$

where P_{P_M} and P_{E_1} are functions of elevation angle and frequency.

Quantities needed to specify each of the seven lobes are:

P_P, α_P, α_H, α_B each a two dimensional table of entries corresponding to 5 frequencies and 7 lobes.

The rear lobe is a direct function of the 6th lobe.
For each lobe tables are interpolated in frequency for the above 4 values. (See Table 1)

The general equation for each lobe is:

$$P_{P_T} = (P_P + 5)\left\{\cos\frac{\pi}{2}\ \frac{(\alpha - \alpha_P)}{(\alpha_B - \alpha_P)}\right\}^C - 5.$$

where $$C = \frac{\log_{10}\left(\frac{P_P + 2}{P_P + 5}\right)}{\log_{10}\left\{\cos\frac{\pi}{2}\ \frac{(\alpha_H - \alpha_P)}{(\alpha_B - \alpha_P)}\right\}}$$

To determine RCVRGN, P_{P_M} and P_{E_1} must be found.

To determine P_{P_M} as a function of frequency and elevation the following quantities are needed.

$$P_{E_1},\ P_{E_2},\ P_{S_2},\ \beta_{B_0},\ \beta_{H_0},\ \beta_{P_1},\ \beta_{H_1},\ \beta_{B_1},\ \beta_{H_2},\ \beta_{P_2},\ \beta_{S_2},\ \beta_{B_2}$$

All are a function of frequency as given in Table 2.

If $\beta_{B_0} \leq \beta < \beta_{P_1}$

$$P_{P_M} = (P_{E_1} + 5)\left\{\cos\frac{\pi}{2}\frac{\beta - \beta_{P_1}}{\beta_{B_0} - \beta_{P_1}}\right\}^C - 5.$$

$$\text{where } C = \frac{\log_{10}\left\{\frac{P_{E_1} + 2}{P_{E_1} + 5}\right\}}{\log_{10}\left\{\cos\frac{\pi}{2}\frac{\beta_{H_0} - \beta_{P_1}}{\beta_{B_0} - \beta_{P_1}}\right\}}$$

If $\beta_{P_1} \leq \beta < \beta_{B_1}$

$$P_{P_M} = (P_{E_1} + 5)\left\{\cos\frac{\pi}{2}\frac{\beta - \beta_{P_1}}{\beta_{B_1} - \beta_{P_1}}\right\}^C - 5.$$

$$\text{where } C = \frac{\log_{10}\left\{\frac{P_{E_1} + 2}{P_{E_1} + 5}\right\}}{\log_{10}\left\{\cos\frac{\pi}{2}\frac{\beta_{H_1} - \beta_{P_1}}{\beta_{B_1} - \beta_{P_1}}\right\}}$$

If $\beta_{B_1} \leq \beta < \beta_{P_2}$

$$P_{P_M} = (P_{E_2} + 5) \left\{ \cos \frac{\pi}{2} \frac{\beta - \beta_{P_2}}{\beta_{B_1} - \beta_{P_2}} \right\}^C - 5.$$

$$\text{where } C = \frac{\log_{10} \left\{ \frac{P_{E_2} + 2}{P_{E_2} + 5} \right\}}{\log_{10} \left\{ \cos \frac{\pi}{2} \frac{\beta_{H_2} - \beta_{P_2}}{\beta_{B_1} - \beta_{P_2}} \right\}}$$

If $\beta_{P_2} \leq \beta < \beta_{S_2}$

$$P_{P_M} = \frac{(P_{E_2} - P_{S_2})}{2} \cos \pi \frac{(\beta - \beta_{P_2})}{(\beta_{S_2} - \beta_{P_2})} + \frac{P_{E_2} + P_{S_2}}{2}$$

If $\beta_{S_2} \leq \beta < \beta_{B_2}$

$$P_{P_M} = (P_{S_2} + 5) \cos \frac{\pi}{2} \frac{(\beta - \beta_{S_2})}{(\beta_{B_2} - \beta_{S_2})} - 5.$$

The rear lobe is generated as follows: α_{P_R} is always π or 180°.

Given P_{P_6}, α_{P_6}, α_{H_6}, α_{B_6}, we determine P_{P_R}, α_{H_R}, and α_{B_R} as follows.

$$P_{P_R} = P_{P_6} + 3 \qquad R = \frac{P_{P_R} + 5}{P_{P_6} + 5}$$

$$\alpha_{H_R} = \alpha_{P_R} + R(\alpha_{H_6} - \alpha_{P_6})$$

$$\alpha_{B_R} = \alpha_{P_R} + R(\alpha_{B_6} - \alpha_{P_6})$$

We then use the general formula for P_{P_M} as a function of P_P, α_P, α_H, and α_B.

Table 1

Tables for azimuth lobes for the low band.

Lobe 1

f	8	10	12	14	16
P_{P_1}	22.2	25.0	29.3	30.0	31.4
α_{P_1}	0.0	0.0	0.0	0.0	0.0
α_{H_1}	5.3	4.5	3.5	3.5	3.2
α_{B_1}	12.5	9.5	8.5	7.1	5.5

Lobe 2

f	8	10	12	14	16
P_{P_2}	-4.4	4.2	11.2	14.1	15.6
α_{P_2}	15.7	14.1	12.2	10.5	9.5
α_{H_2}	16.9	16.0	14.3	12.2	11.5
α_{B_2}	16.9	16.5	16.5	13.0	13.5

Lobe 3

f	8	10	12	14	16
P_{P_3}	-5.0	-5.0	-5.0	2.4	3.3
α_{P_3}	18.0	18.0	18.0	14.5	14.5
α_{H_3}	18.0	18.0	18.0	15.9	16.7
α_{B_3}	18.0	18.0	18.0	16.0	18.1

Lobe 4

f	8	10	12	14	16
P_{P_4}	.2	7.9	7.0	2.2	6.9
α_{P_4}	22.0	26.8	23.0	20.0	23.5
α_{H_4}	24.7	29.2	25.0	21.9	25.6
α_{B_4}	25.2	31.7	26.2	22.3	25.9

Lobe 5

f	8	10	12	14	16
P_{P_5}	-5.0	-5.0	1.9	5.7	3.2
α_{P_5}	26.3	37.0	30.4	26.0	29.7
α_{H_5}	26.3	37.0	32.4	27.8	31.9
α_{B_5}	26.3	37.0	32.5	28.5	32.4

Lobe 6

f	8	10	12	14	16
P_{P_6}	8.3	8.2	11.1	7.9	13.8
α_{P_6}	32.7	46.5	39.7	35.7	39.9
α_{H_6}	36.0	50.0	41.6	37.8	41.5
α_{B_6}	38.9	51.8	42.9	38.9	42.2

Lobe 7

f	8	10	12	14	16
P_{P_7}	-5.0	-5.0	2.7	6.5	12.9
α_{P_7}	42.0	52.0	46.7	43.9	46.0
α_{H_7}	42.0	52.0	48.7	45.7	48.4
α_{B_7}	42.0	52.0	49.6	46.3	49.8

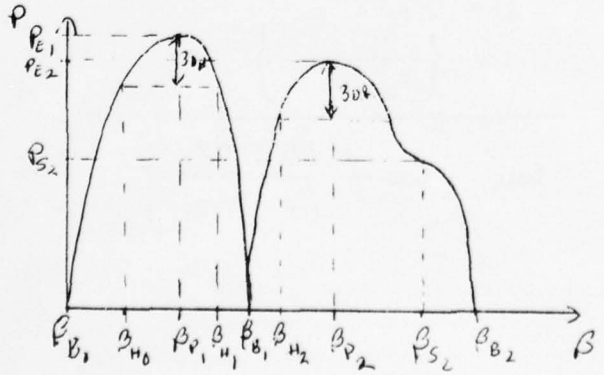
Rear Lobe

f	8	10	12	14	16
P_{P_R}	11.3	11.2	14.1	10.9	16.8
α_{P_R}	180.0	180.0	180.0	180.0	180.0
α_{H_R}	184.0	184.3	182.3	182.6	181.9
α_{B_R}	187.6	186.5	183.8	184.0	182.7

Table 2

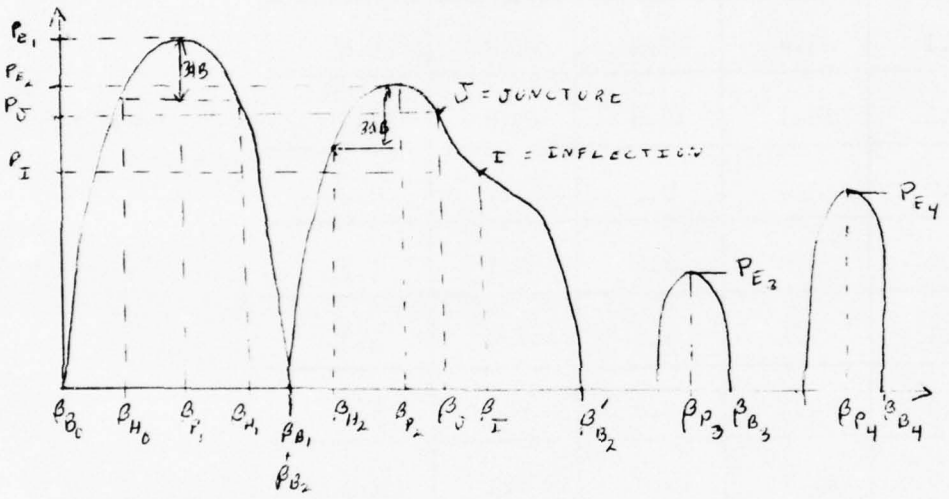
Power vs elevation for low band of rhombic.
Power is in DB, β in degrees, f in MHz.

f	8	10	12	14	16
P_{E_1}	22.2	25.0	29.3	30.0	31.4
P_{E_2}	21.1	23.4	27.0	27.3	28.6
P_{S_2}	12.2	15.1	18.8	19.6	21.4
β_{B_0}	0.0	0.0	0.0	.4	0.0
β_{H_0}	4.0	3.0	2.4	2.1	1.5
β_{P_1}	8.1	5.1	5.0	3.9	3.8
β_{H_1}	11.3	9.3	7.4	6.4	5.5
β_{B_1}	15.0	11.7	10.0	8.3	7.8
β_{H_2}	18.0	14.4	12.2	10.2	8.8
β_{P_2}	21.3	17.5	14.3	12.0	10.0
β_{S_2}	30.3	25.2	21.0	19.2	16.5
β_{B_2}	38.0	34.0	30.8	30.0	27.4



2. Equations for the high band of the rhombic receiver are the same as for the low band in azimuthal variation except that lobe 7 is the predominant side lobe and the rear lobe will be based on it.

In describing the variation in power with elevation for the high band the following quantities are needed. These quantities are obtained from Table 3.



For $\beta_{B_0} \leq \beta \leq \beta_{P_1}$

$$P_{P_M} = (P_{E_1} + 5) \left\{ \cos \frac{\pi}{2} \frac{\beta - \beta_{P_1}}{\beta_{B_0} - \beta_{P_1}} \right\}^C - 5$$

$$\text{where } C = \frac{\log_{10} \left\{ \frac{P_{E_1} + 2}{P_{E_1} + 5} \right\}}{\log_{10} \left\{ \cos \frac{\pi}{2} \frac{\beta_{H_0} - \beta_{P_1}}{\beta_{B_0} - \beta_{P_1}} \right\}}$$

For $\beta_{P_1} \leq \beta \leq \beta_{B_1}$

$$P_{P_M} = (P_{E_1} + 5) \left\{ \cos \frac{\pi}{2} \frac{\beta - \beta_{P_1}}{\beta_{B_1} - \beta_{P_1}} \right\}^C \quad - 5.$$

$$\text{where } C = \frac{\log_{10} \left\{ \frac{P_{E_1} + 2}{P_{E_1} + 5} \right\}}{\log_{10} \left\{ \cos \frac{\pi}{2} \frac{\beta_{H_1} - \beta_{P_1}}{\beta_{B_1} - \beta_{P_1}} \right\}}$$

For $\beta_{B_2} < \beta < \beta_{B_1}$ (May be non existant) take greater of equation above and below.

For $\beta_{B_2} \leq \beta \leq \beta_{P_2}$

$$P_{P_M} = (P_{E_2} + 5) \left\{ \cos \frac{\pi}{2} \frac{\beta - \beta_{P_2}}{\beta_{B_2} - \beta_{P_2}} \right\}^C \quad - 5.$$

$$\text{where } C = \frac{\log_{10} \left\{ \frac{P_{E_2} + 2}{P_{E_2} + 5} \right\}}{\log_{10} \left\{ \cos \frac{\pi}{2} \frac{\beta_{H_2} - \beta_{P_2}}{\beta_{B_2} - \beta_{P_2}} \right\}}$$

For $\beta_{P_2} < \beta \leq \beta_J$

$$P_{P_M} = (P_{E_2} + 5) \cos \quad b(\beta - \beta_{P_2}) \quad - 5.$$

$$\text{where } b = \frac{\cos^{-1} \left\{ \frac{P_J + 5}{P_{E_2} + 5} \right\}}{(\beta_J - \beta_{P_2})}$$

For $\beta_J < \beta \leq \beta_{B_2}$,

$$P_{P_M} = a \tan \frac{\pi}{8} \frac{(\beta - \beta_I)}{(\beta_J - \beta_I)} + P_I$$

$$\text{where } a = \frac{P_J - P_I}{\tan \frac{\pi}{8}}$$

For $2\beta_{P_3} - \beta_{B_3} \leq \beta < \beta_{B_3}$

$$P_{P_M} = (P_{E_3} + 5) \cos^{1/2} \frac{\pi}{2} \left(\frac{\beta - \beta_{P_3}}{\beta_{B_3} - \beta_{P_3}} \right) - 5$$

For $2\beta_{P_4} - \beta_{B_4} \leq \beta < \beta_{B_4}$

$$P_{P_M} = (P_{E_4} + 5) \cos^{1/2} \frac{\pi}{2} \left(\frac{\beta - \beta_{P_4}}{\beta_{B_4} - \beta_{P_4}} \right) - 5 .$$

For all other β's, $P_{P_M} = -5$.

The quantity P_{P_T} is determined in the same manner as for the low band except that the values given in Table 4 are used as a function of frequency. These quantities are interpolated from Table 4 as a function of frequency.

Table 3

Power vs elevation for high band of rhombic.
Power is in DB, β in degrees, f in MHz.

f	16	20	24	28	30
P_{E_1}	25.0	27.7	28.8	29.6	30.3
P_{E_2}	19.0	21.6	22.3	22.9	23.6
P_J	16.8	17.4	18.4	19.4	21.6
P_I	12.9	13.4	14.6	15.6	18.8
P_{E_3}	-5.0	-5.0	-3.2	3.7	8.0
P_{E_4}	-5.0	-5.0	-0.1	13.1	11.4
β_{B_0}	0.0	0.0	0.2	0.2	0.2
β_{H_0}	2.0	2.8	1.9	1.6	1.3
β_{P_1}	6.0	5.0	4.2	3.9	4.1
β_{H_1}	9.3	7.1	6.7	6.8	5.4
β_{B_1}	12.8	10.0	8.5	7.4	7.2
β_{B_2}	12.8	10.0	8.5	7.4	7.2
β_{H_2}	15.0	12.7	9.9	8.4	7.7
β_{P_2}	18.1	14.3	11.7	9.9	9.7
β_J	21.1	18.0	15.0	13.2	11.2
β_I	24.3	20.6	17.2	15.3	13.1
$\beta_{B_2'}$	33.4	28.0	24.0	20.0	18.9
β_{P_3}	37.0	32.0	28.0	24.0	23.0
β_{B_3}	37.0	32.0	29.2	27.1	26.0
β_{P_4}	42.0	40.0	40.7	42.0	39.0
β_{B_4}	42.0	40.0	42.2	48.8	43.5

Table 4

Tables for azimuthal lobes for the high band.

Lobe 1

f	16	20	24	28	30
P_{P_1}	25.0	27.7	28.8	29.6	30.3
α_{P_1}	0.0	0.0	0.0	0.0	0.0
α_{H_1}	4.3	2.9	2.8	2.7	2.5
α_{B_1}	10.0	8.5	6.9	6.0	6.0

Lobe 2

f	16	20	24	28	30
P_{P_2}	4.5	10.8	13.2	14.0	15.5
α_{P_2}	14.1	11.7	10.2	8.0	7.8
α_{H_2}	15.9	13.5	12.0	9.8	9.3
α_{B_2}	16.9	15.0	12.6	10.9	10.8

Lobe 3

f	16	20	24	28	30
P_{P_3}	-5.0	-5.0	2.2	8.9	12.0
α_{P_3}	17.7	18.2	16.0	13.9	12.4
α_{H_3}	17.7	18.2	17.8	15.6	14.3
α_{B_3}	17.7	18.2	18.7	17.1	16.2

Lobe 4

f	16	20	24	28	30
P_{P_4}	-4.0	1.9	3.3	3.8	3.0
α_{P_4}	19.7	23.2	25.7	22.9	20.6
α_{H_4}	20.9	24.9	27.5	24.1	22.1
α_{B_4}	20.9	25.3	28.2	24.5	22.5

Lobe 5

f	16	20	24	28	30
P_{P_5}	5.3	1.0	5.3	-5.0	-4.0
α_{P_5}	27.5	29.2	34.9	28.5	25.5
α_{H_5}	30.7	30.7	36.3	28.5	26.5
α_{B_5}	32.2	31.0	37.0	28.5	26.5

Lobe 6

f	16	20	24	28	30
P_{P_6}	-5.0	-5.0	-5.0	-2.3	1.0
α_{P_6}	36.0	32.9	38.9	33.7	32.2
α_{H_6}	36.0	32.9	38.9	35.5	33.8
α_{B_6}	36.0	32.9	38.9	35.5	34.5

Lobe 7

f	16	20	24	28	30
P_{P_7}	8.7	10.0	7.3	11.8	8.6
α_{P_7}	46.7	39.7	43.3	38.0	36.3
α_{H_7}	49.4	42.0	45.1	40.8	38.0
α_{B_7}	51.4	43.1	46.3	42.0	40.2

Lobe 8

f	16	20	24	28	30
P_{P_8}	-5.0	-3.6	-5.0	7.3	-5.0
α_{P_8}	52.0	46.3	50.0	44.0	43.0
α_{H_8}	52.0	48.1	50.0	45.5	43.0
α_{B_8}	52.0	48.1	50.0	45.7	43.0

Rear Lobe

f	16	20	24	28	30
P_{P_R}	11.7	13.0	10.3	14.8	11.6
α_{P_R}	180.0	180.0	180.0	180.0	180.0
α_{H_R}	183.3	182.6	182.2	183.3	182.1
α_{B_R}	185.7	184.1	183.7	184.7	184.8

SPECIAL CAUTIONS AND FEATURES

Function RCVRGN is only valid for frequencies between 8 and 30 MHz. Outside of this range a value of -100 will be returned.

Values of azimuth and elevation are to be given in radians and the azimuth is assumed to be in the geomagnetic dipolar system.

Computation at 16 MHz is made on the high band antenna.

TIMING

Each use of function RCVRGN takes approximately 2 milliseconds of central processor time.

ERROR MESSAGES

None.

SUBROUTINES

None.

ACCURACY

The model of the azimuthal antenna patterns compares very well with the original. The elevation model is not quite as good. Actual and modeled patterns are given in Attachments 1 to 10.

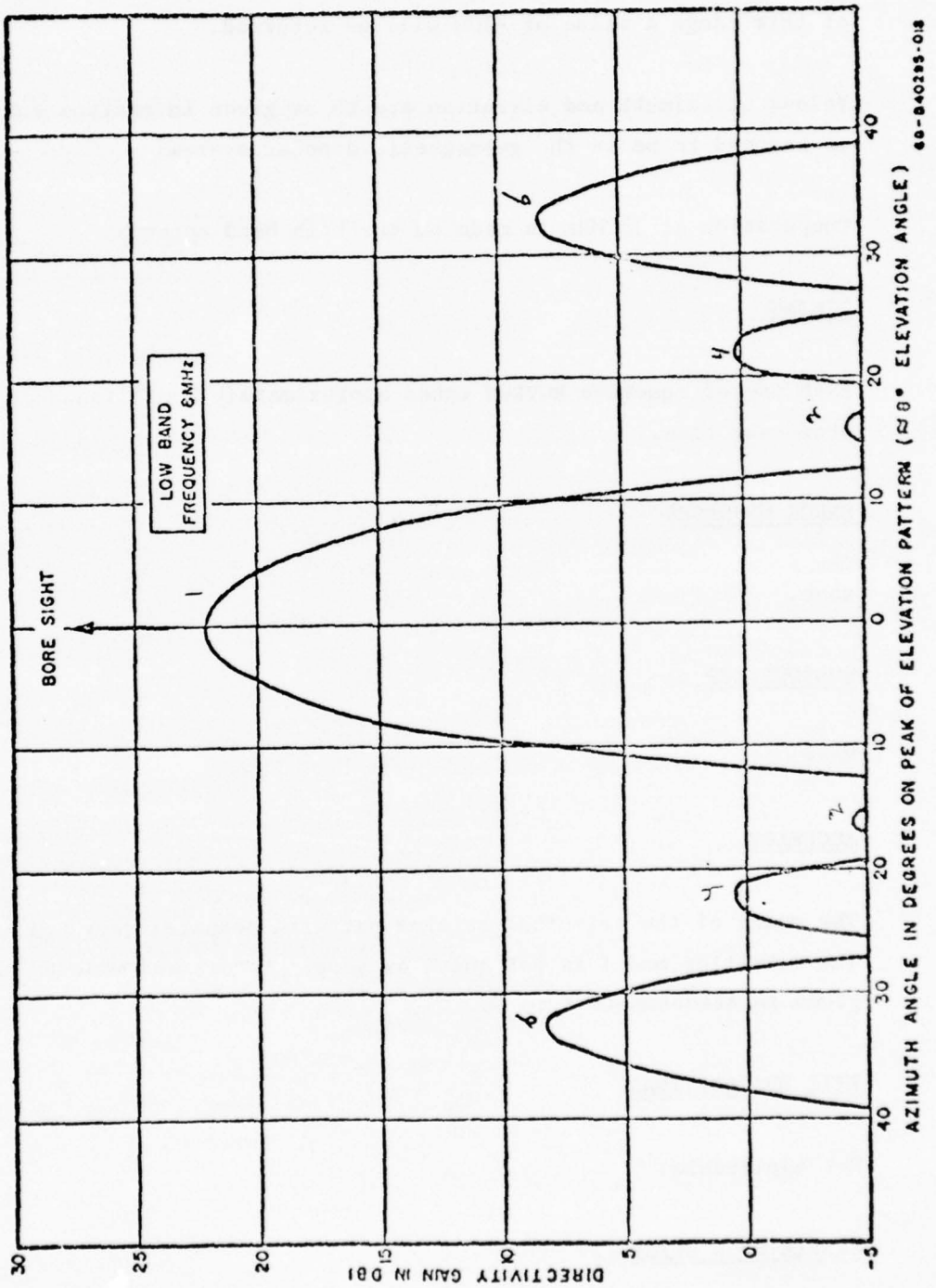
FILE DESCRIPTIONS

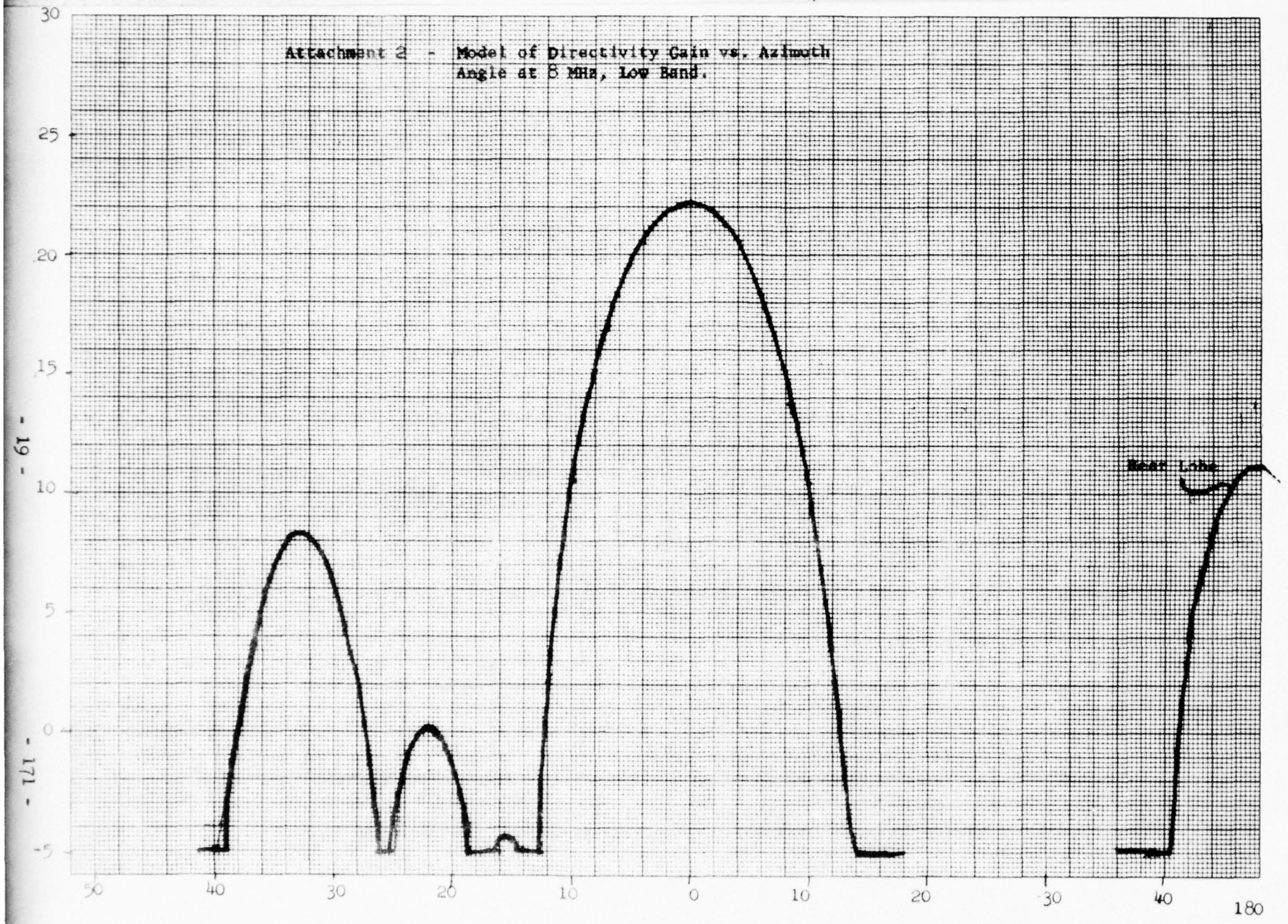
Not applicable.

SAMPLE DECK SETUP

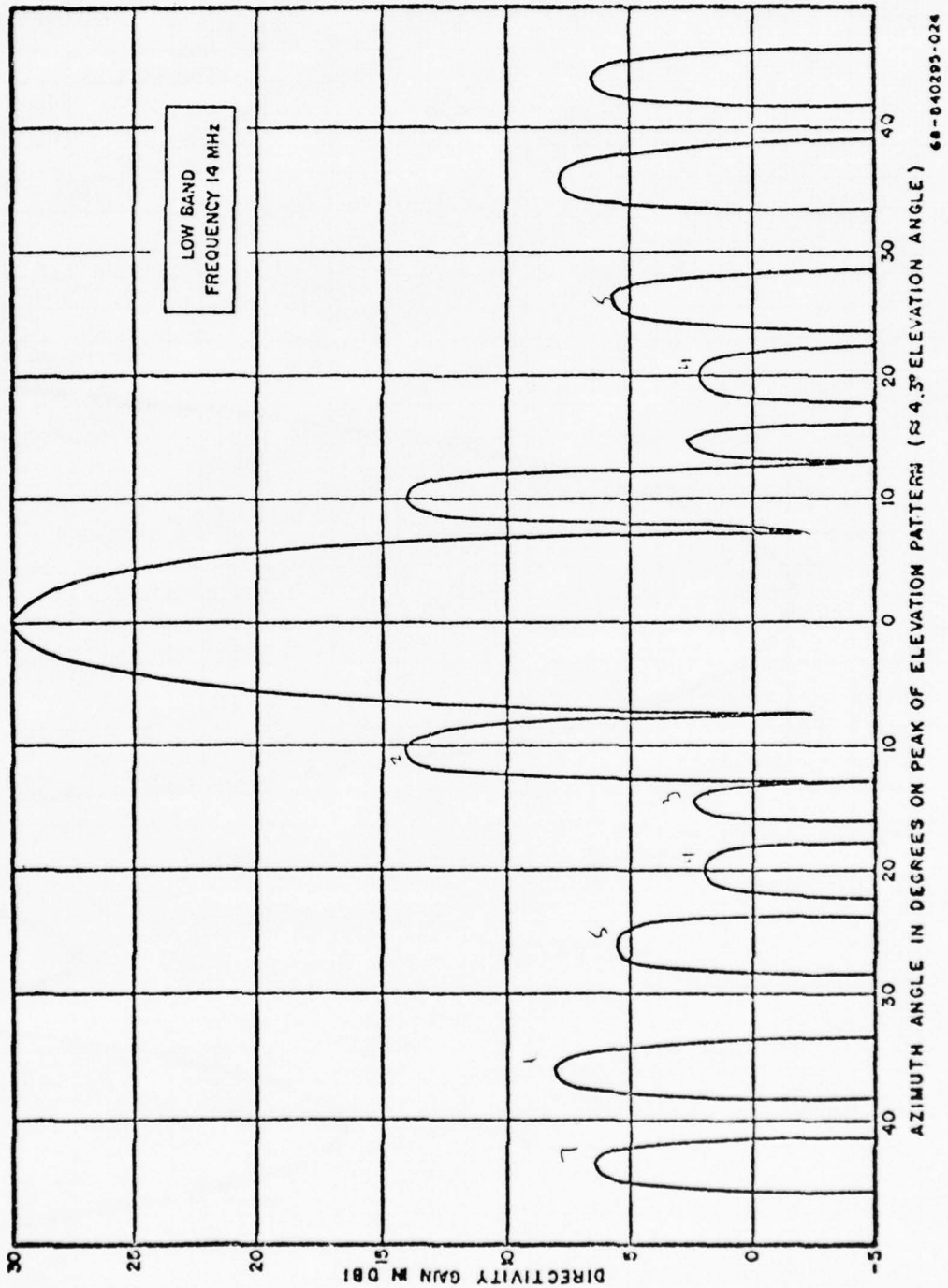
The source deck of RCVRGN is presently on System I permanent file storage under RHOANTX3693818, ID=LANGW.

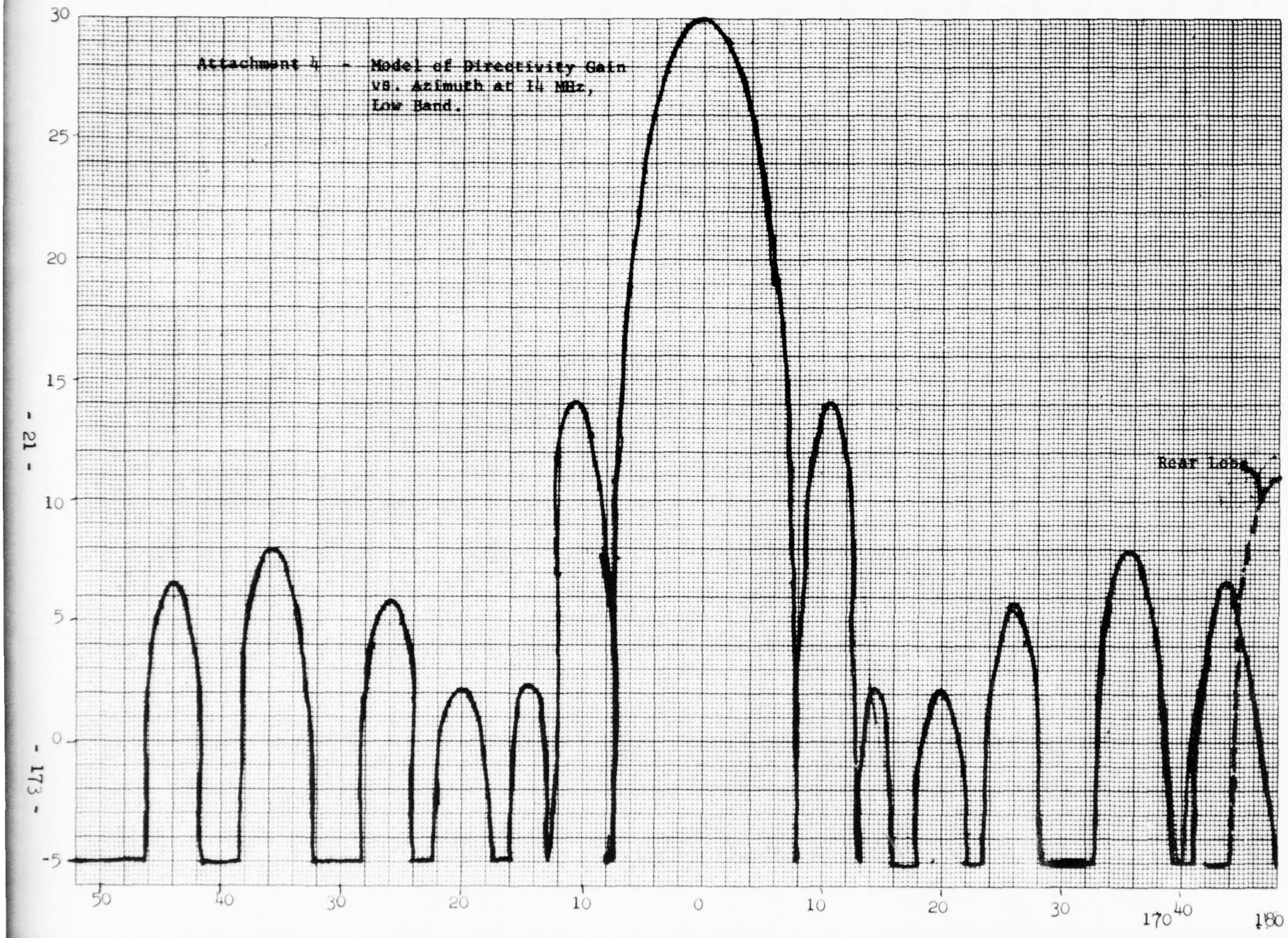
Attachment 1 - Specification of Directivity Gain vs. Azimuth Angle at 8 MHz, Low Band.



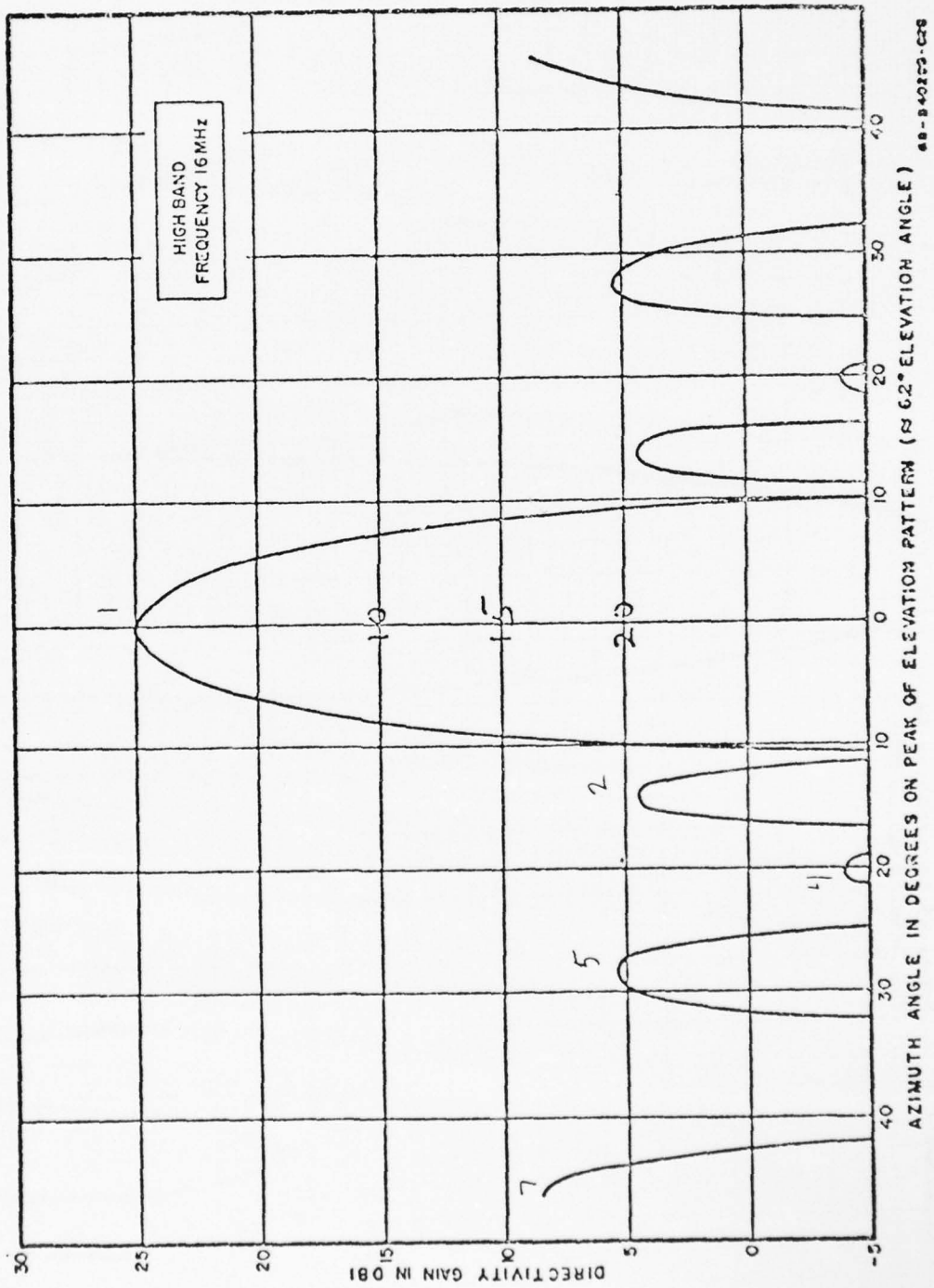

Attachment 2 - Model of Directivity Gain vs. Azimuth Angle at 8 MHz, Low Band.
30
25
20
15
10
5
0
-5
- 19 -
- 171 -
Rear Lobe
50
40
30
20
10
0
10
20
30
40
180

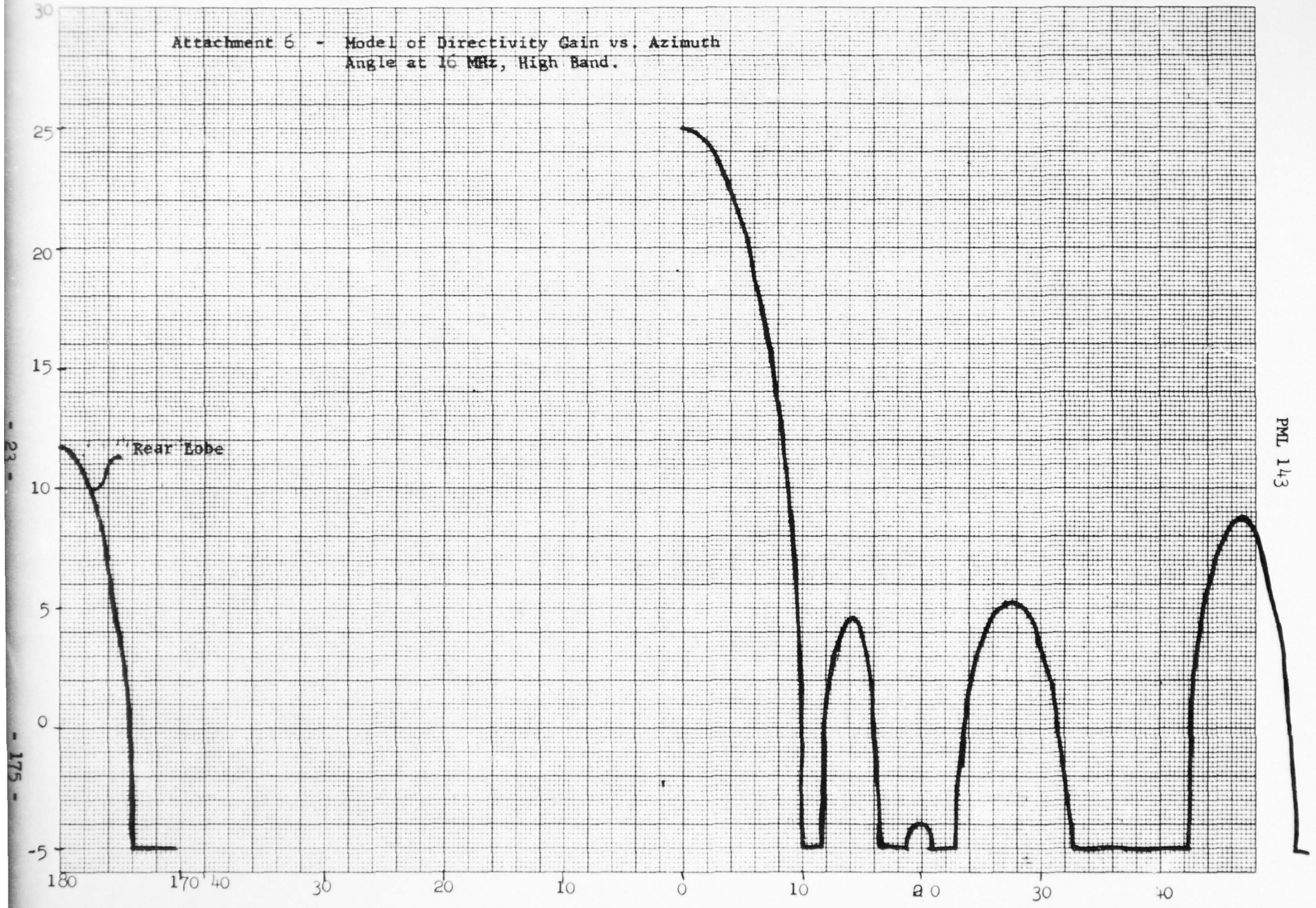
Attachment 3 - Specification of Directivity Gain vs. Azimuth Angle at 14 MHz, Low Band.




Attachment 4 - Model of Directivity Gain vs. Azimuth at 14 MHz, Low Band.
Rear Lobe
30
25
20
15
10
5
0
-5
- 21 -
- 173 -
50
40
30
20
10
0
10
20
30
40
170
180

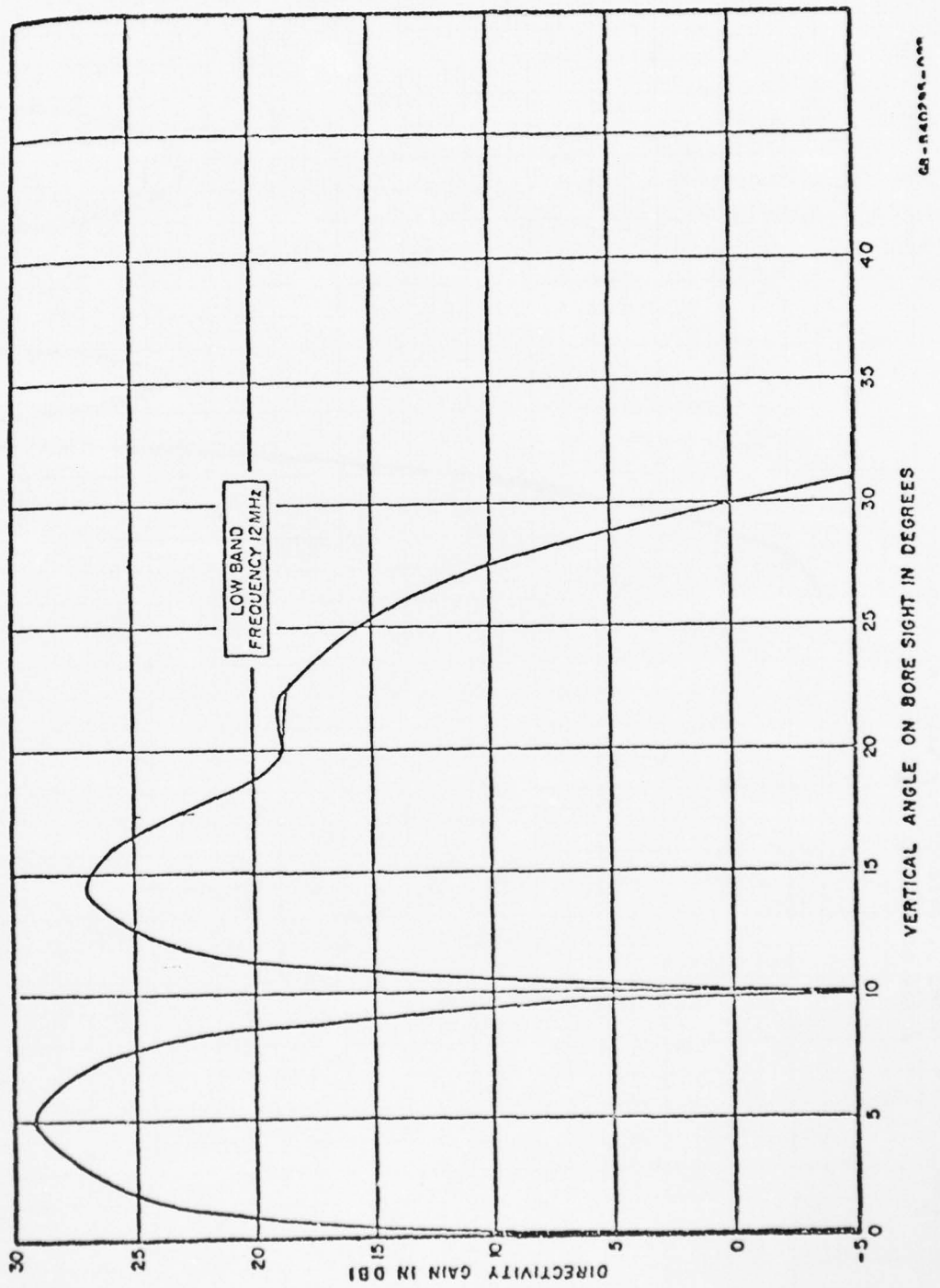
Attachment 5 - Specification of Directivity Gain vs. Azimuth Angle at 16 MHz, High Band.

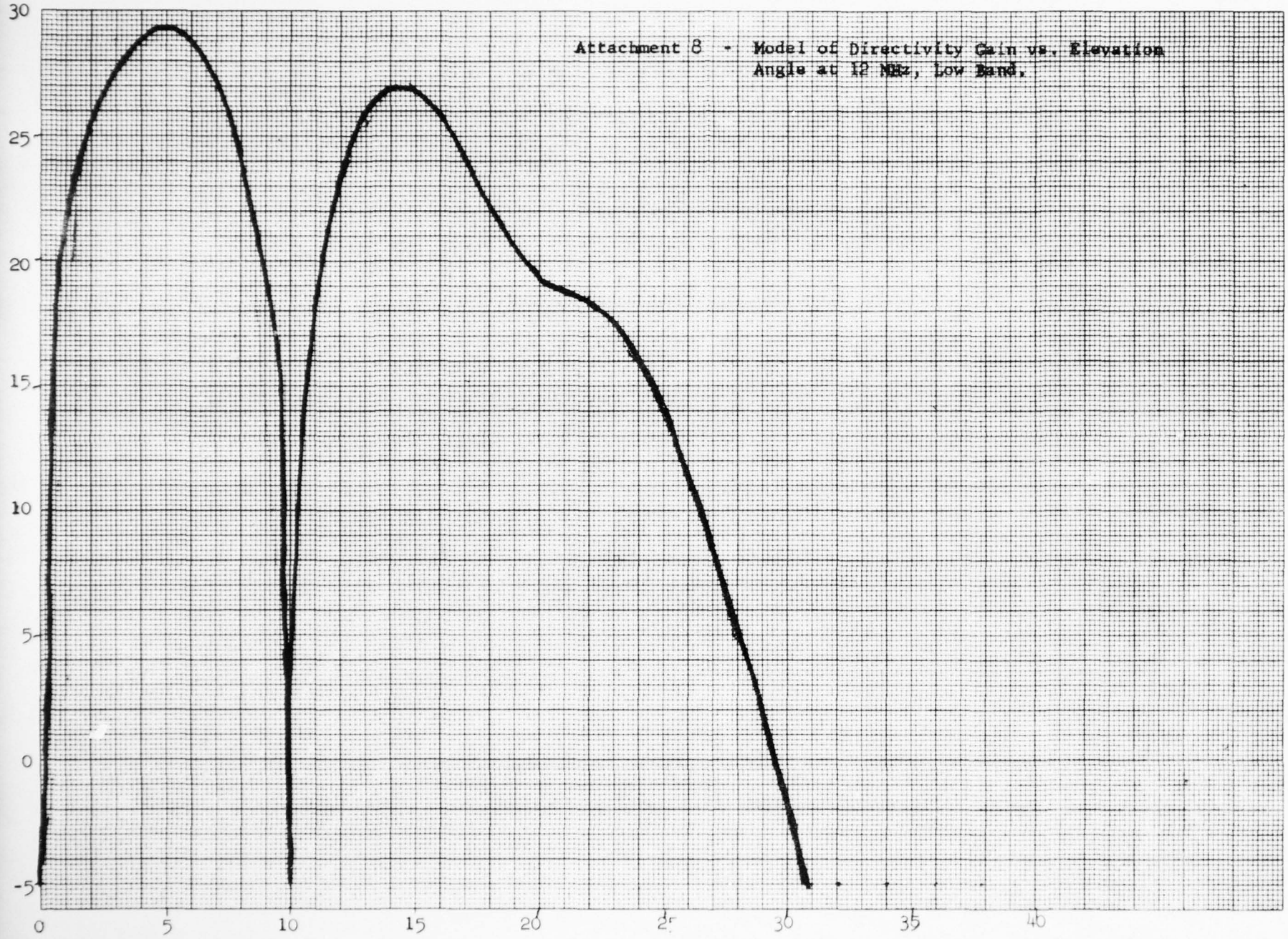



Attachment 6 - Model of Directivity Gain vs. Azimuth Angle at 16 MHz, High Band.
30
25
20
15
10
5
0
-5
Rear Lobe
180
170
40
30
20
10
0
10
20
30
40

PML 143

Attachment 7 - Specification of Directivity Gain vs. Elevation Angle at 12 MHz, Low Band.

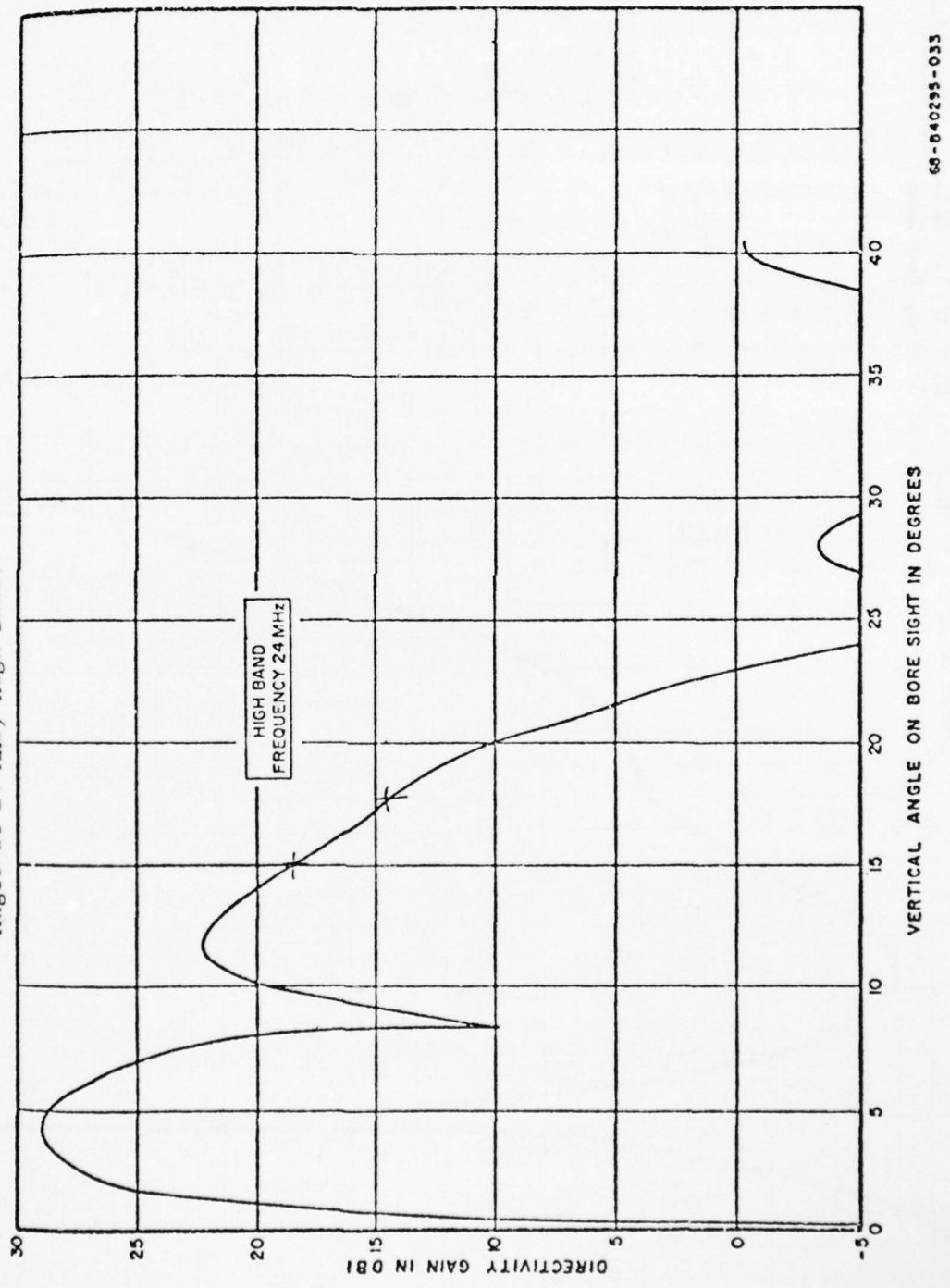




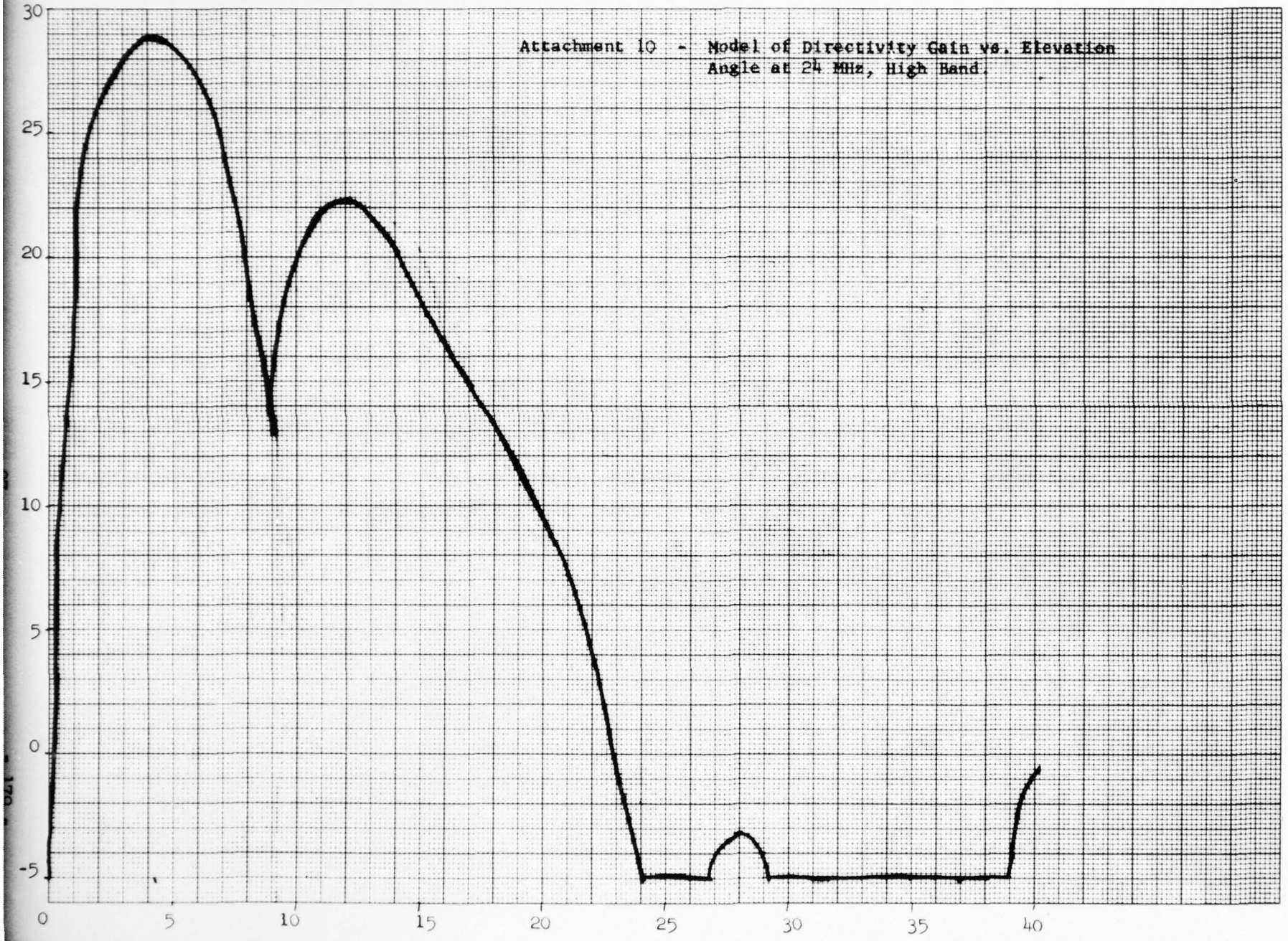
Attachment 8 - Model of Directivity Gain vs. Elevation Angle at 12 MHz, Low Band.

PML 143

Attachment 9 - Specification of Directivity Gain vs. Elevation Angle at 24 MHz, High Band.



Attachment 10 - Model of Directivity Gain vs. Elevation Angle at 24 MHz, High Band.



PML 143

NAME: GPMT, Revision 0, program, PML 144
CATEGORY: Plotting and data sorting program for ray-trace program
TITLE: Minimum Group Path Trace Selection and Plotting.
LANGUAGE: CDC extended Fortran - version 4
PROGRAMMER: B. M. Langworthy, Parke Mathematical Laboratories, Inc.
DATE: June 30, 1976

DESCRIPTION

Program GPMT determines the minimum group path trace for both 1-hop reflections and 90^{o} angles of incidence to enhanced magnetic field lines. The program prints and plots the minimum trace and creates input data for the ray-tracing program to compute the rays necessary for computation of the power loss by program POWER (See PML 145). Group path data of a particular frequency and azimuth angle of transmission is examined as a function of elevation angle. All relative minima in group path length are selected for minimum group path trace. Also, considered as a minimum trace point is the elevation angle which is part of a decreasing function when the next elevation angle shows a penetrated ray. In power computations these rays have been shown to have comparable signal strength to the relative minima rays.

Plots and printouts are grouped so that all frequencies for a given azimuth angle are given one set of axis or one printed table. Points on plots are connected to those points having the closest elevation angle to the points for the previous frequency.

The program is structured in such a way that data is accumulated in tables from previous runs of both GPMT and POWER. If for a given minimum group path ray, the power has not been previously computed, an input case for the ray-tracing program will be set up to compute the absorption for the ray at given frequency, azimuth angle, and elevation angle. It will also prepare input cases to compute the paths for three nearby rays for the spreading computation.

PRECEDING PAGE BLANK-NOT FILMED

INSTRUCTION SET

To use program GPMT, a file of rays refered to as TAPE7 must previously have been generated by the ray-tracing program. This file will contain information on the starting point, points where the ray is perpendicular to magnetic field lines, apogee points, and points where the ray is at the receiver height which for this case is the ground. Since all rays for a particular minimum group path trace may not be run at the same time, there is provision for adding this TAPE7 input to previous runs made on GPMT or POWER. This is through TAPE44, an input tape which contains the previous output tables. The program will add new information from TAPE7 to the existing TAPE44 and will produce a composite plot, printout, and TAPE66 which can then be used as a new TAPE44 or as input to program POWER. These tapes will be explained in more detail under FILE DESCRIPTIONS. TAPE7 and TAPE44, if available, are the only input required by program GPMT.

Output from program GPMT consists of the aforementioned TAPE66, a TAPE88 which becomes INPUT to the ray-tracing program, printed and plotted output. A schematic of the interrelation between program GPMT, POWER, and the ray-tracing program is given in Figure 1.

The printed output consists of a table for each azimuth which is titled by the last four words in the ID title of the original ray-trace run given on TAPE7. Sample printed output is given in Attachment 1. The second word of this title is a five letter code indicating the electron density model used. The third and fourth words of the title give the time and date of the original ray-tracing run from which TAPE7 was created. The azimuth given in the title is in degrees and is in the same coordinate system as the azimuth initially given to the ray-trace program. The system is usually geographic, but it can be dipolar geomagnetic.

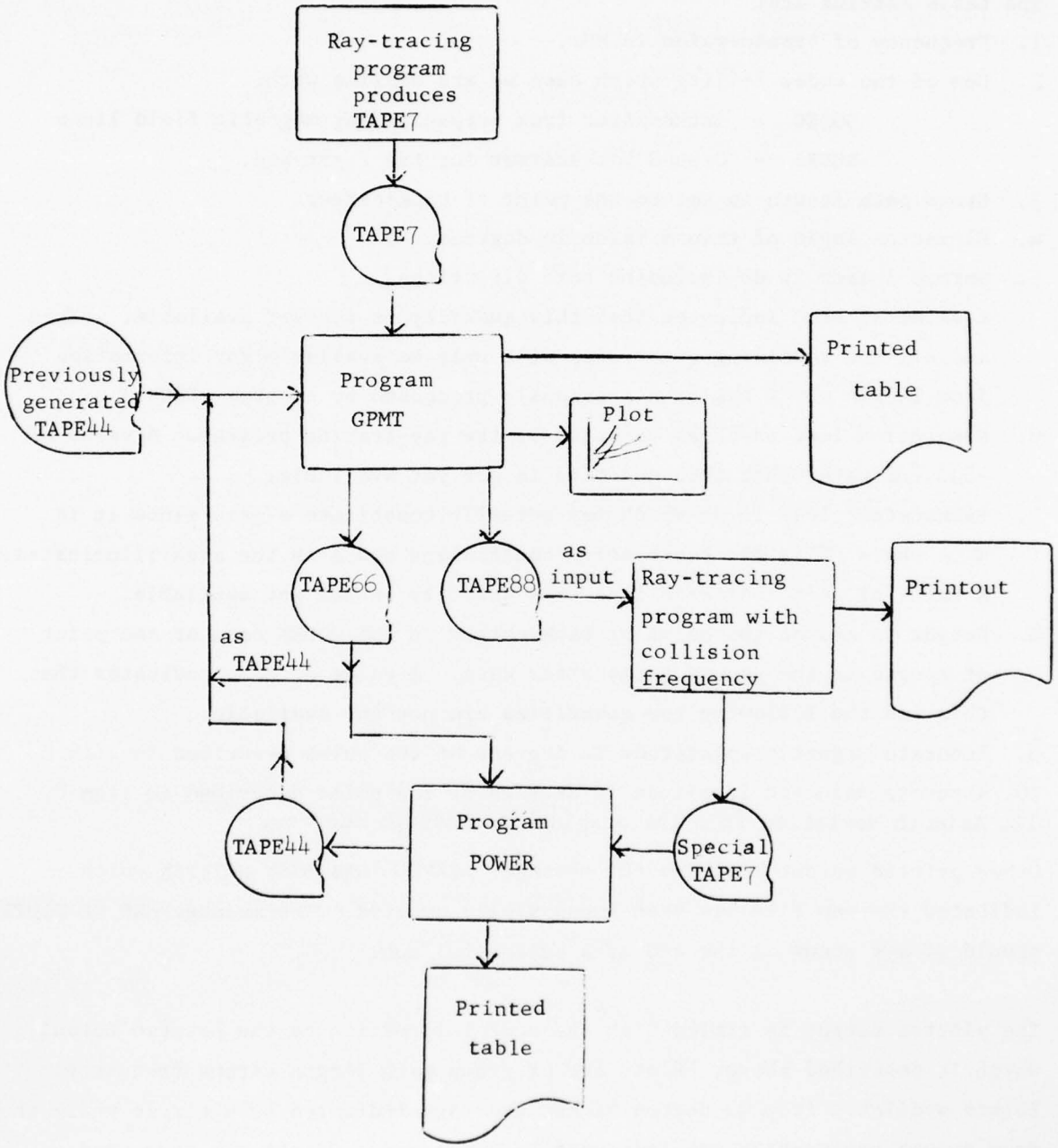


Figure 1: Flow diagram of interaction between the ray-tracing program, program GPMT, and program POWER.

The table entries are:

1. Frequency of transmission in MHz.
2. One of two codes telling which case we are dealing with;
   90DEG - Backscatter from perpendicular magnetic field lines
   RCVR1 - Ground backscatter for the first hop.
3. Group path length in km. to the point of backscatter.
4. Elevation angle of transmission in degrees.
5. Spread losses in db including both directions.
   A value of -200 indicates that this quantity is not yet available. (This and all the remaining quantities will only be available for information from TAPE44 which has been previously processed by program POWER.)
6. Absorption loss in db as computed by the ray-tracing program. A value of -200 indicates that this quantity is not yet available.
7. Backscatter loss in db which may actually constitute a gain since it is σ.A where σ is the backscatter coefficient and A is the area illuminated. A value of zero indicates that this quantity is not yet available.
8. Height in km. of the point of backscatter in the 90DEG case or the point of apogee in the ground backscatter case. A value of zero indicates that this and the following two quantities are not yet available.
9. Accurate magnetic colatitude in degrees of the point described in item 8.
10. Accurate magnetic longitude in degrees of the point described in item 8.
11. Azimuth deviation from the original azimuth in degrees.

Other printed output includes the message: TAPE66 HAS BEEN WRITTEN which indicated the new file has been successfully created. The message END OF PLOTS should always occur at the end of a successful run.

The plotted output is titled with the same information as the printed output which is described above. Plots are of group path length versus frequency. Points available from 90 degree backscatter are indicated by a circle while those from ground backscatter are indicated by a diamond. Points are connected on the same basis of similar elevation angles of transmission. Sample plots are given in Attachment 2.

STORAGE REQUIRED

GPMT requires 160000 octal words of storage.

ALGORITHM

Algorithm for determining the group path minima for a given frequency, f, and a given azimuth, α.

Let G_i correspond to the group path for elevation angle β_i.

Let $i = 1$ and $\Delta_{last} = 1$.

(1) $\Delta_{new} = G_{i+1} - G_i$

If $\Delta_{new} \cdot \Delta_{last} < 0$ and $\Delta_{new} > 0$, we have a minimum and we save it.

We then let $i = i + 1$, $\Delta_{last} = \Delta_{new}$, and return to (1).

Otherwise, let $i = i + 1$.

If $i \geq i_{max}$ and $\Delta_{new} \leq 0$, we save the point as a minimum and terminate the search.

If $i \geq i_{max}$ and $\Delta_{new} > 0$, the search is terminated.

If $i < i_{max}$, set $\Delta_{last} = \Delta_{new}$, and:

If $G_{i+1} \neq 0$, return to (1);

If $G_{i+1} = 0$ and $\Delta_{new} > 0$, the search is terminated;

If $G_{i+1} = 0$ and $\Delta_{new} \leq 0$, we save the point as a minimum point and terminate the search.

SPECIAL CAUTIONS AND FEATURES

At present program GPMT can accept up to 40 frequencies and 10 azimuths. If data for more azimuths and frequencies is given on TAPE7, an error message will be printed out telling which frequencies and azimuths were ignored.

Program GPMT can be run with no TAPE44 or with no TAPE7. If, however, the program is run without a TAPE7, there will be no title on either the plots or the tables and, more seriously, the input cards created on TAPE88 will be incorrect. This is due to the fact that if no TAPE7 is read, no information will be contained in the W-array. TAPE7 may be omitted if a plot and table of an existing TAPE44 is desired.

Program GPMT assumes that no more than 4 minima will be found for each of the two cases of 90^{o} incidence to magnetic field lines and one hop. If there are additional points they will be ignored.

The program also assumes that no more than 10 lines are needed to connect each of the two cases. If this is not the case the points will show up without connecting lines.

Program GPMT will stop looking for minima once a ray has escaped.

The Y plotting axis is of fixed length. If group path lengths greater than 5000 km are encountered, a plotting error will result.

TIMING

Timing varies with the proportion of points which are taken from TAPE7 for plotting against those which are taken from an existing TAPE44. It ranges from 1.7 seconds per hundred points with all points coming from TAPE44 to 3.0 seconds per hundred points with all points coming from TAPE7.

ERROR MESSAGES

INFORMATION NOT SAVED FOR I = XXX FREQ = XX.XX J = XX AZIMUTH = XX.XX

This error message is printed out whenever too many azimuths or frequencies are provided from TAPE7. Program GPMT can handle up to 40 frequencies and up to 10 azimuths. If either of these numbers is exceeded the above message will be printed where:

I is the number of frequencies attempted. If this is 41, the number of frequencies has been exceeded;
FREQ is the frequency in MHz of the neglected information;
J is the number of azimuths attempted. If this is 11, the number of azimuths has been exceeded;
AZIMUTH is the azimuth in radians of the neglected information.

The information for this particular frequency and azimuth combination will be lost but the program will proceed normally.

ATTEMPT TO GAIN MORE DATA WHEN THERE IS NONE.

This error message occurs when subroutine TPRDR has given the signal to program GPMT that it is finished reading TAPE7 and program GPMT has called it again. This should not occur in normal operation. If it does occur the program will stop with a call to exit.

FILE TAPE7 RECORD XXX RECORD MANAGER ERROR - 143
ERROR NUMBER 103 DETECTED BY IOERR AT ADDRESS 000021
CALLED FROM INPB = AT ADDRESS XXXXXX
CALLED FROM TPRDR AT LINE XX
CALLED FROM GPMT AT LINE 30
ROUTINE BEOT ENTERED.

This error message occurs when the last record of the TAPE7 file or tape is incomplete due to a non-standard ending on the ray trace run such as time limit or operator drop. Although this is normally a fatal error, it has been overridden by a call to SYSTEMC and will instead be treated by subroutine TPRDR as an end of file. The program will then proceed normally.

SUBROUTINES

GPMT — This is the main program. It reads any existing TAPE44 and then callsTPRDR and selects the new minima from the information. When all new minima have been found, they are sorted by frequency and azimuth. The information is then given one azimuth at a time to PRNT, GPPLT, and PNCHR. The combined information is then written on TAPE66 to replace TAPE44.

TPRDR — This subroutine reads the desired information off of the ray-trace tape, TAPE7, for one frequency and azimuth at a time.

PRNT(J) — This subroutine prints the output tables described on pages 2 - 4 for the Jth azimuthal entry.

GPPLT(J) — This subroutine creates the plots shown in Attachment 2 for the Jth azimuthal entry.

PNCHR(J) — This subroutine creates the ray-tracing input tape, TAPE88, for the Jth azimuthal entry.

BEOT — This subroutine is the error recovery routine for the Fortran fatal error 103 override, SYSTEMC, explained above. Subroutine BEOT sets an indicator so that subroutine TPRDR will know an end of information exists.

DICOORD — This subroutine converts geographic coordinates to dipolar geomagnetic coordinates when called with a call to entry GTOM and converts dipolar geomagnetic coordinates to geographic coordinates with a call to entry MTOG.

CORRGM2 — This subroutine takes points in the geographic coordinate system and converts them to accurate geomagnetic coordinates. A call to CORRGM2 causes the transforming array to be read in. For points to be converted, a call must be made to CORRGM.

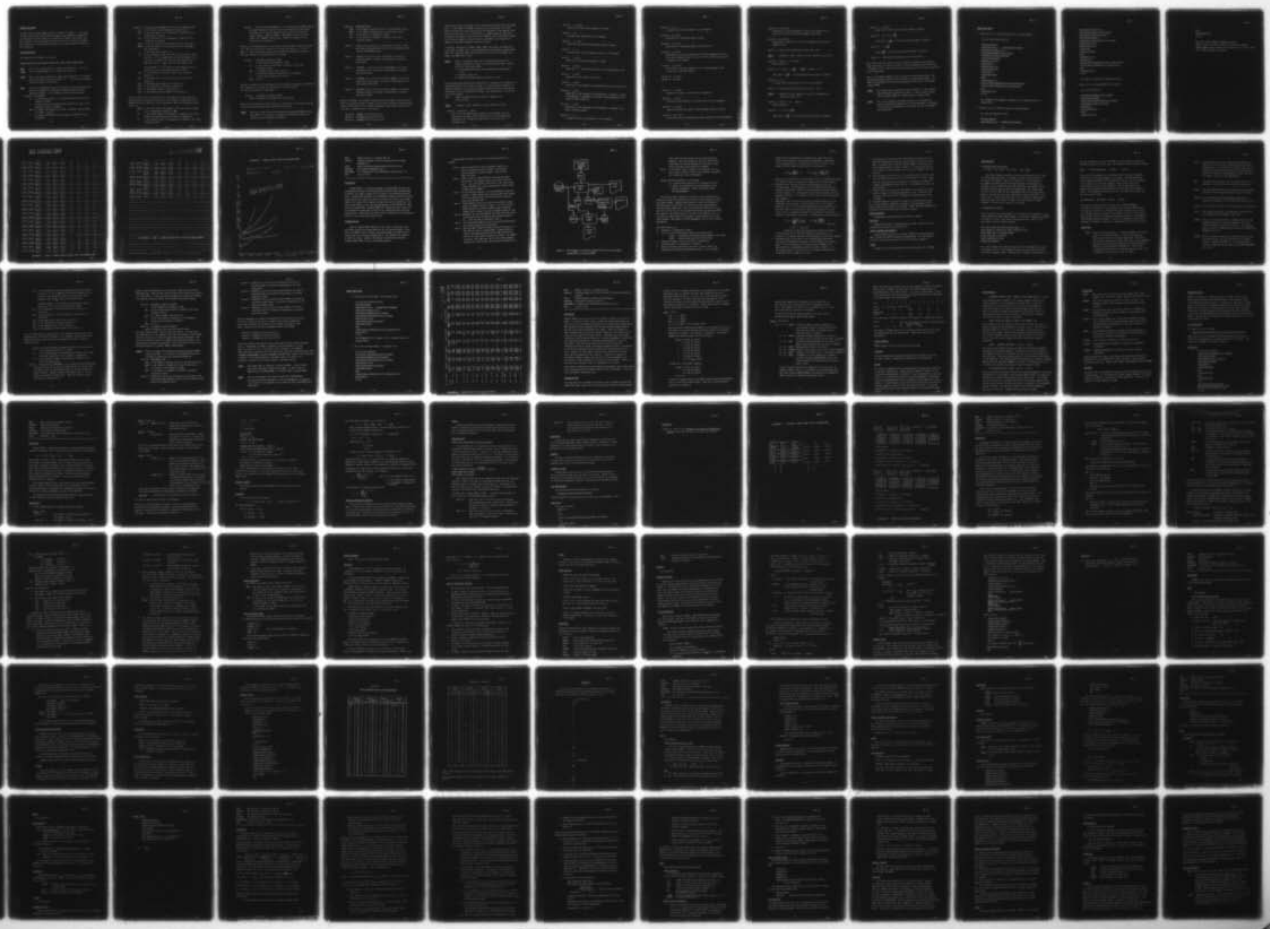
Other subroutines called are AFCRL plotting software. The plotting subroutines used are: PLTID3, AXIS, LINE, NUMBER, PLOT, SYMBOL, and ENDPLT. See Section 14 of the AFCRL User's Guide for descriptions of these subroutines.

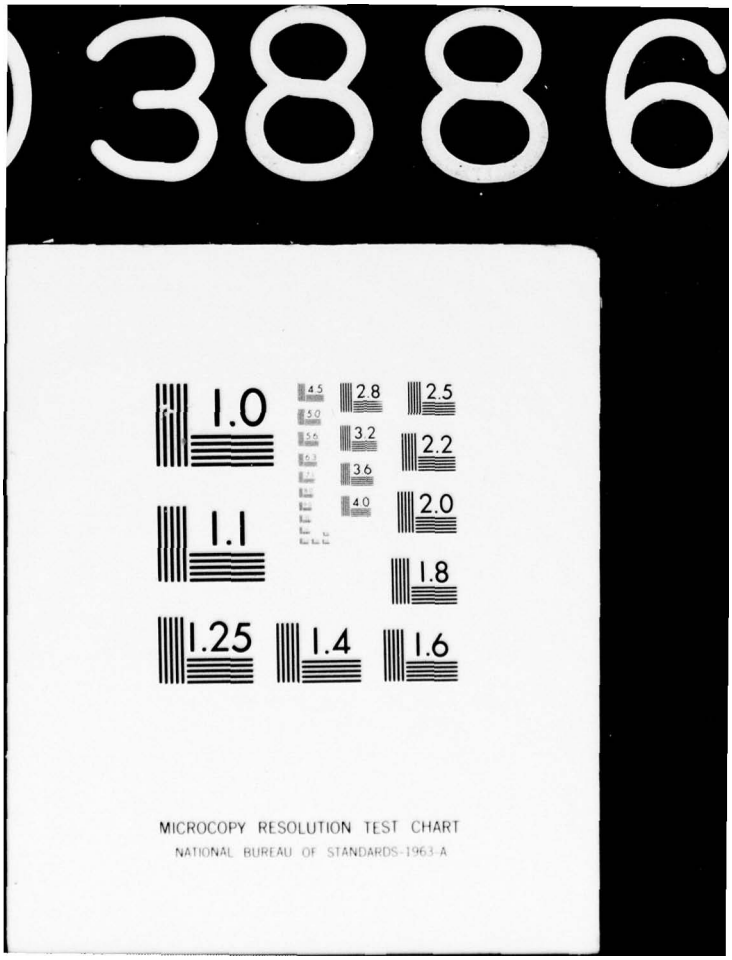
ACCURACY

Not applicable.

AD-A038 869 PARKE MATHEMATICAL LABS INC CARLISLE MASS F/G 20/14
ANALYSIS AND SYNTHESIS OF MODEL IONOGRAMS USING 3D RAY TRACKING--ETC(U)
FEB 77 B LANGWORTHY, T BARRETT, D BANDES F19628-76-C-0029
UNCLASSIFIED RADC-TR-77-60 NL

3 OF 4
AD
A038869




3886
1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

COMMENTS ON USAGE

Possible uses of program GPMT are given in Figure 1 on page 3 . In using the TAPE7 output from the ray-tracing program, the assumption is made that all necessary elevation angles for a particular frequency and azimuth have been computed. It also assumes that once one ray for a particular azimuth and frequency has penetrated the ionosphere, all higher elevation angles will also penetrate.

FILE DESCRIPTIONS

The program card for GPMT is as follows:

PROGRAM GPMT(INPUT,OUTPUT,TAPE4=INPUT,TAPE7,TAPE44,TAPE88,TAPE66,TAPE1)

INPUT
TAPE4 — This is the system input file under normal operation. It is not used by program GPMT but is included for future use.

OUTPUT — This is the system output file under normal operation. It contains the output tables described on pages 2 - 4 plus any error messages from the program or the system.

TAPE7 — This is an unformatted tape put out by the ray-tracing program from which ray information is obtained by GPMT and other ray-trace dependent programs. It is composed of the following records:

Record 1: ID,W,NTYP,N

- ID — an information string of 100 characters which identifies the computer run made,
- W — an array of 400 real numbers which contains all input to the program and other information,
- NTYP — an integer specifying the ray mode - ordinary, extraordinary, or neither (3,1,2)
- N — an integer specifying the total number of equations to be integrated.

Record 2: R(1),R(2),R(3),IHOP,NWHY(1),T,(R(I),I=4,N),SIDEG,F,AZ,EL

R(1) is the radius of the transmitter location in km,

R(2) is the colatitude of the transmitter in spherical polar coordinates in radians,

R(3) is the longitude of the transmitter in spherical polar coordinates in radians,

IHOP is zero to indicate the transmitter data is given here,

NWHY(1) tells the reason for the printout; in this case XMTR or the transmitter,

T is the group path length in km,

R(4) is the component of the wave normal in the positive radial direction. It is of magnitude such that the square root of the sum of its square with those of R(5) and R(6) have a magnitude equal to $\frac{\omega}{c.n}$ when ω is the angular wave frequency($2\pi f$), c is the speed of electromagnetic waves in free space, and n is the phase refractive index.

R(5) is the component of the wave normal in the positive θ direction.

R(6) is the component of the wave normal in the positive ϕ direction.

R(7) is the phase path length in km if W(57) $\neq$ 0.

R(8) is the absorption in decibels if W(58) $\neq$ 0.

R(9) is the doppler shift in Hz if W(59) $\neq$ 0.

R(10) is the geometrical path length in km if W(60) $\neq$ 0.

If any of these last four quantities are not requested then the succeeding quantities will be moved up in position. For instance, if phase path is not computed (W(57)=0), then R(7) will be absorption, R(8) will be doppler shift, and so on.

SIDEG is the angle between the magnetic field and the wave normal.

F is the transmission frequency in MHz.

AZ is the azimuth angle of transmission in radians in the input coordinate system which may be dipolar or geographic.

EL is the elevation angle of transmission in radians in the input coordinate system.

Record 3: R(1),R(2),R(3),IHOP,NWHY(2,3 or 4),T,(R(I),I=4,N),SIDEG,F,AZ,EL

The above record will be given for points along the ray path where the SIDEG is 90^{o}, the ray reaches apogee, or the ray touches the earth's surface. NWHY(2), NWHY(3), and NWHY(4) will be "90 DEG.", "RCVR", and "APOGEE", respectively. IHOP will indicate the hop that the ray is presently on.

Record 3 will be repeated for all such points along the path of this particular ray. Then the next ray will start with a record like Record 2, i.e. with IHOP = 0. When all elevation angles for a particular azimuth and frequency have been listed, the following record will appear:

Record n: F,AZ,DUM,-1,N,(W(I),I=1,NEQ)

- F is the transmission frequency in MHz.
- AZ is the current azimuth angle in radians in the input coordinate system.
- DUM is a dummy variable (most likely 0).
- N is the total number of equations to be integrated.
- W is the W array described in Record 1.
- NEQ is N + 2.

Records 2 through n will be repeated until all azimuth and frequency combinations are completed (or all rays for a particular W-array are given).
The following record will then appear.

Record m: F,AZ,DUM,-2,N,(W(I),I=1,NEQ)

where F,AZ,DUM,N,W, and NEQ are as defined above.

This will be followed by an end-of-file or a new Record 1 will be given and the process will repeat itself until all W arrays are given.

TAPE44 This is an input file containing previously generated TAPE66 output from GPMT. It is written in unformatted records and contains 65 records with the following information:

Record 1: NAZE,NFRE,FT,AZT

NAZE is the number of azimuth entries in the AZT table ($\leq$ 10).

NFRE is the number of frequencies in the FT table ($\leq$ 40).

FT is an array of 40 frequencies in MHz.

AZT is an array of 10 azimuths in radians in the input coordinate system.

Record 2: ((GT(I,J,1),I=1,40),J=1,10) where GT is an array of the group path length minima in km for the Ith frequency entry and the Jth azimuthal entry.

Record 3: ((ELT(I,J,1),I=1,40),J=1,10) where ELT is an array of elevation angles in radians corresponding to the above group paths.

Record 4: ((POWDT(I,. ,1),I=1,40),J=1,10) where POWDT is an array of spreading losses in decibels corresponding to the above group path minima.

Record 5: ((POWBS(I,J,1), I=1,40),J=1,10) where POWBS is an array of backscatter power losses in decibels corresponding to the above group path minima.

Record 6: ((POWAT(I,J,1), I=1,40),J=1,10) where POWAT is an array of absorption losses in decibels corresponding to the above group path minima.

Records 2 through 6 are repeated increasing the third index of each array by 1 until 8 such sets are read. Records 2-21 will give information concerning the minima for the case of 90° incidence to magnetic field lines. Records 22-41 will give information about 1-hop minima.

Record 42: ((HGT(I,J,1),I=1,40),J=1,10)

Record 43: ((COLAT(I,J,1),I=1,40),J=1,10)

Record 44: ((CLONG(I,J,1),I=1,40),J=1,10).

These three records are repeated 7 more times with the third index increasing up to 8. HGT, COLAT, and CLONG correspond to the coordinates of a point which in records 42 to 53 is the point of 90° incidence to the magnetic field line and in records 54 to 65 is the point of apogee along the 1-hop path. Here HGT is the height of this point in km, COLAT is the colatitude in degrees in the dipolar magnetic coordinate system, and CLONG is the longitude in degrees in the dipolar magnetic coordinate system.

In general, information on POWDT, POWBS, POWAT, HGT, COLAT, and CLONG will be available on TAPE44 only if the information was processed by program POWER (PML 145). Otherwise POWDT and POWAT will have value -200 and POWBS, HGT, COLAT, and CLONG will be zero.

TAPE88 This is a formatted output file containing 8-word records to simulate punched card input to the ray-tracing program. Records will be of three basic types:

1) An identification record which signifies the beginning of a case,
2) W-array records, and
3) Blank records which sginify the end of a case.

For each group path minimum point identified by a frequency (f), azimuth (α), and elevation (β) and for which no power has been previously computed (i.e. POWDT and POWAT are -200.) four rays must be traced, one with an absorption calculation. This is accomplished through 3 ray tracing cases. The records will be as follows:

Note: Unless otherwise given, the format is assumed to be (I3,E14.7,63X).

CASE 1. Frequency is f(1), azimuth is $\alpha(1)$, elevation is $\beta(1)$.

Record 1: ID(L),L=1,8 (8A10)

where ID is the identification string of the original ray-trace run except that ID(6) will be RCVR if this is to be a power loss calculation for a 1-hop case and 90 DEG. if it is for a 90° incidence case.

Record 2: 11, $\alpha(1)$

where $\alpha(1)$ is the desired azimuth in radians.

Record 3: 58, 2.

to signal that absorption will be computed.

Record 4: 15, $\beta(1)$

where $\beta(1)$ is the desired elevation angle in radians.

Record 5: 17, 0.

to signal that only one elevation angle is to be computed.

Record 6: 7, f(1)

where f(1) is the desired frequency in MHz.

Record 7: 1, W(1)

where W(1) = 1 for ordinary rays and -1 for extraordinary rays.

Record 8: 2, W(2)

where W(2) is the radius of the earth.

Record 9: 3, W(3)

where W(3) is the height of the transmitter above the earth in km.

Record 10: 4, W(4)

where W(4) is the latitude of the transmitter in radians in either the geographic or geomagnetic coordinate system depending on the values of W(24) and W(25).

Record 11: 5, W(5)

where W(5) is the longitude of the transmitter in radians in the above coordinate system.

Record 12: 9, 0.

to signal that only one frequency is to be computed.

Record 13: 13, 0.

to signal that only one azimuth is to be computed.

Record 14: 22, W(22)

where W(22) is the maximum number of hops.

Record 15: 23, W(23)

where W(23) is the maximum number of steps per hop.

Record 16: 24, W(24)

where W(24) is the north latitude of the north geomagnetic pole in the input coordinate system. If this is $\pi/2$, then the input system is assumed to be dipolar.

Record 17: 25, W(25)

where W(25) is the east longitude of the north geomagnetic pole in the input coordinate system.

Record 18: 41, W(41)

Record 19: 42, W(42)

.

.

.

Record 24: 47, W(47)

Records 18 through 24 control the ray integration.

Record 25: 57, W(57)

where W(57) tells whether or not phase path is to be integrated.

Record 26: 71, W(71)

where W(71) gives the number of steps between periodic printout.

Record 27: 90, W(90)

where W(90) gives the maximum group path length in km to be integrated.

Record 28: 201, W(201)

where W(201) is the gyrofrequency in MHz at the equator on the ground to be used in the dipolar magnetic field model.

Record 29: 249, W(249)

where W(249) is the year to be used in determining the accurate magnetic field.

Record 30: Blank record indicating an end of case. (80x)

CASE 2. Frequency is f(1), azimuth is $\alpha(1)$, elevation is $\beta(1)+\delta$

Record 31: ID(L), L = 1,8 (8A10)

Same as Record 1.

Record 32: 15, $\beta(1) + \frac{\pi}{1800}$ ($\frac{\pi}{1800}$ radians = 0.1^{o})

where $\beta(1) + \frac{\pi}{1800}$ is the desired elevation angle in radians.

Record 33: 58, 0.

to signal that the absorption is not to be calculated.

Record 34: Blank record indicating an end of case. (80x)

CASE 3. Frequency is f(1), azimuth is $\alpha(1)+\delta$, elevations are $\beta(1)$ and $\beta(1)+\delta$.

Record 35: ID(L), L = 1,8 (8A10)

Same as Record 1.

Record 36: 11, $\alpha(1) + \frac{\pi}{1800}$

where $\alpha(1) + \frac{\pi}{1800}$ is the desired elevation angle in radians.

Record 37: 15, $\beta(1)$

where $\beta(1)$ is the initial elevation angle in radians.

Record 38: 16, $\beta(1) + \frac{\pi}{1800}$

where $\beta(1) + \frac{\pi}{1800}$ is the final elevation angle in radians.

Record 39: 17, $\frac{\pi}{1800}$

where $\frac{\pi}{1800}$ is the elevation angle increment in radians.

Record 40: Blank record indicating an end of case (80x).

These three cases are repeated for each frequency, azimuth, and elevation angle combination for which the power is not yet computed except that records 7 through 29 will be omitted from all succeeding case 1's since these quantities do not change value.

Once it is completed TAPE88 can be used with the ray-tracing program. The ray-tracing program, however, must be capable of computing absorption. The TAPE7 thus generated by the ray-tracing program should then be run through program POWER.

TAPE66 This output tape is identical in form to TAPE44. It does include additional points obtained from TAPE7. It may be used as TAPE66 input to program POWER or as TAPE44 input to program GPMT.

TAPE1 This is the permanent file which contains the transformation array for going from geographic to accurate geomagnetic coordinates. It is stored in both Systems I and II under the file name MJDATA with ID=LOGICON.

SAMPLE DECK SETUP

1. Setup for run on a multifile tape with no existing TAPE44.

```
job card with CM140000 and TP1

VSN(MFILE=YOURNO)
REQUEST(MFILE,MF,E)     (YOURNO/NORING)YOURNAME
LABEL(TAPE7,M=MFILE,P=1,R)
ATTACH(BGPMT,BGPMTX3693818,ID=LANGW)
ATTACH(TAPE1,MJDATA,ID=LOGICON)
ATTACH(PEN,ONLINEPEN)
REQUEST(TAPE44,*PF)
REWIND(TAPE44)
REQUEST(TAPE66,*PF)
REQUEST(TAPE88,*PF)
LIBRARY(PEN)
LDSET(PRESET=ZERO)
BGPMT.
DISPOSE(PLOT,*OL)
CATALOG(TAPE66,NEWTAPE66X3693818,ID=LANGW,RP=999)
CATALOG(TAPE88,RAYINX3693818,ID=LANGW,RP=999)
EXIT.
DISPOSE(PLOT,*OL)
6/7/8/9
```

Here BGPMTX3693818,ID=LANGW is assumed to be a compiled version of program GPMT.

2. Setup for run on a multifile tape with existing TAPE44.

```
job card with CM140000 and TP1

VSN(MFILE=YOURNO)
REQUEST(MFILE,MF,E)    (YOURNO/NORING)YOURNAME
```

```
LABEL(TAPE7,M=MFILE,P=2,R)
ATTACH(BGPMT,BGPMTX3693818,ID=LANGW)
ATTACH(TAPE1,MJDATA,ID=LOGICON)
ATTACH(PEN,ONLINEPEN)
ATTACH(TAPE44,OLDTAPE44X3693818,ID=LANGW)
REQUEST(TAPE66,*PF)
REQUEST(TAPE88,*PF)
LIBRARY(PEN)
LDSET(PRESET=ZERO)
BGPMT.
DISPOSE(PLOT,*OL)
PURGE(TAPE44)
CATALOG(TAPE66,NEWTAPE66X3693818,ID=LANGW,RP=999)
CATALOG(TAPE88,RAYINX3693818,ID=LANGW,RP=999)
EXIT.
DISPOSE(PLOT,*OL)
6/7/8/9
```

Here TAPE44 is purged since TAPE66 replaces it.

3. Setup for run on existing TAPE44 with a TAPE7.

```
job card with CM140000

ATTACH(BGPMT,BGPMTX3693818,ID=LANGW)
ATTACH(PEN,ONLINEPEN)
ATTACH(TAPE44,OLDTAPE44X3693818,ID=LANGW)
ATTACH(TAPE1,MJDATA,ID=LOGICON)
LIBRARY(PEN)
LDSET(PRESET=ZERO)
BGPMT.
DISPOSE(PLOT,*OL)
```

EXIT.
*DISPOSE(PLOT,*OL)*
6/7/8/9

Note that neither TAPE66 nor TAPE88 are saved.
TAPE66 is not saved since it will be a duplicate of TAPE44.
TAPE88 cannot be saved because no W-array was read in from a TAPE7 and it would, therefore, be invalid.

BEST AVAILABLE COPY

NEAR INT. DATVD 04/25/76 18.24.38. AZIMUTH = 30.00

FREQUENCY		GROUP PATH LENGTH	ELEVATION ANGLE	SPREAD LOSS	ABSRPTION LOSS	BACKSCAT LOSS	MAX HEIGHT	COLAT	LONG
2.50	90 DEG	935.5	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		494.6	50.00	-200.0	-200.0	0.0	0.0	0.0	0.0
2.50	RCVR1	1206.2	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1338.7	10.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1245.4	34.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1079.1	48.00	-200.0	-200.0	0.0	0.0	0.0	0.0
2.75	90 DEG	939.2	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		539.5	44.00	-200.0	-200.0	0.0	0.0	0.0	0.0
2.75	RCVR1	1222.8	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1362.3	10.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1262.7	34.00	-200.0	-200.0	0.0	0.0	0.0	0.0
3.00	90 DEG	942.7	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		584.6	38.00	-200.0	-200.0	0.0	0.0	0.0	0.0
3.00	RCVR1	1239.7	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1389.2	10.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1289.9	36.00	-200.0	-200.0	0.0	0.0	0.0	0.0
3.25	90 DEG	945.	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		621.0	36.00	-200.0	-200.0	0.0	0.0	0.0	0.0
3.25	RCVR1	1256.7	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1415.7	10.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1336.3	36.00	-200.0	-200.0	0.0	0.0	0.0	0.0
3.50	90 DEG	949.3	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		942.7	12.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		659.4	34.00	-200.0	-200.0	0.0	0.0	0.0	0.0
3.50	RCVR1	1274.5	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1442.8	10.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1418.2	34.00	-200.0	-200.0	0.0	0.0	0.0	0.0
3.75	90 DEG	952.7	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
3.75	RCVR1	1292.9	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1473.8	10.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1537.5	34.00	-200.0	-200.0	0.0	0.0	0.0	0.0
4.00	90 DEG	955.7	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		965.9	12.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		766.8	33.00	-200.0	-200.0	0.0	0.0	0.0	0.0
4.00	RCVR1	1310.4	3.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1508.6	10.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1710.3	33.00	-200.0	-200.0	0.0	0.0	0.0	0.0
4.25	90 DEG	959.0	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		978.3	12.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		852.4	32.00	-200.0	-200.0	0.0	0.0	0.0	0.0
4.25	RCVR1	1331.3	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1544.5	10.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		2021.2	26.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1961.9	32.00	-200.0	-200.0	0.0	0.0	0.0	0.0

Attachment 1. Page 1. Sample printed output table from program GPMT.

BEST AVAILABLE COPY

4.50	90 DEG	962.5	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		990.6	11.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1132.0	30.00	-200.0	-200.0	0.0	0.0	0.0	0.0
4.50	RCVR1	1351.4	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1619.3	9.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1576.7	11.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		2114.6	22.00	-200.0	-200.0	0.0	0.0	0.0	0.0
4.75	90 DEG	965.5	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1003.8	11.00	-200.0	-200.0	0.0	0.0	0.0	0.0
4.75	RCVR1	1372.1	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1638.1	9.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1622.7	11.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		2208.8	21.00	-200.0	-200.0	0.0	0.0	0.0	0.0
5.00	90 DEG	968.8	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1018.2	11.00	-200.0	-200.0	0.0	0.0	0.0	0.0
5.00	RCVR1	1393.3	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1670.2	10.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1671.2	12.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		2318.1	20.00	-200.0	-200.0	0.0	0.0	0.0	0.0
5.25	90 DEG	972.4	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1034.0	11.00	-200.0	-200.0	0.0	0.0	0.0	0.0
5.25	RCVR1	1414.9	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1720.4	11.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		2457.9	19.00	-200.0	-200.0	0.0	0.0	0.0	0.0
5.50	90 DEG	975.6	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1051.4	11.00	-200.0	-200.0	0.0	0.0	0.0	0.0
5.50	RCVR1	1436.9	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1790.1	11.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		2650.0	18.00	-200.0	-200.0	0.0	0.0	0.0	0.0
5.75	90 DEG	979.0	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1070.9	10.00	-200.0	-200.0	0.0	0.0	0.0	0.0
5.75	RCVR1	1459.4	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1857.2	10.00	-200.0	-200.0	0.0	0.0	0.0	0.0
6.00	90 DEG	982.6	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1090.8	10.00	-200.0	-200.0	0.0	0.0	0.0	0.0
6.00	RCVR1	1482.0	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1923.6	10.00	-200.0	-200.0	0.0	0.0	0.0	0.0
6.25	90 DEG	986.1	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1112.3	10.00	-200.0	-200.0	0.0	0.0	0.0	0.0
6.25	RCVR1	1505.3	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1993.5	10.00	-200.0	-200.0	0.0	0.0	0.0	0.0
6.50	90 DEG	989.8	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1135.1	10.00	-200.0	-200.0	0.0	0.0	0.0	0.0
6.50	RCVR1	1528.8	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		2067.5	9.00	-200.0	-200.0	0.0	0.0	0.0	0.0
6.75	90 DEG	993.6	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0

Attachment 1. Page 2. Sample printed output table from program GPMT.

BEST AVAILABLE COPY

		1150.0	1.50	-200.0	-200.0	0.0	0.0	0.0	0.0
6.75	RCVR1	1552.5	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		2139.6	9.00	-200.0	-200.0	0.0	0.0	0.0	0.0
7.00	90 DEG	997.5	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1187.0	9.00	-200.0	-200.0	0.0	0.0	0.0	0.0
7.00	RCVR1	1576.9	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		2216.7	9.00	-200.0	-200.0	0.0	0.0	0.0	0.0
7.25	90 DEG	1001.6	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1213.9	9.00	-200.0	-200.0	0.0	0.0	0.0	0.0
7.25	RCVR1	1601.6	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		2299.7	9.00	-200.0	-200.0	0.0	0.0	0.0	0.0
7.50	90 DEG	1005.8	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1243.6	9.00	-200.0	-200.0	0.0	0.0	0.0	0.0
7.50	RCVR1	1628.3	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		2382.5	8.00	-200.0	-200.0	0.0	0.0	0.0	0.0
7.75	90 DEG	1010.0	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1275.6	8.50	-200.0	-200.0	0.0	0.0	0.0	0.0
7.75	RCVR1	1659.7	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		2469.3	8.00	-200.0	-200.0	0.0	0.0	0.0	0.0
8.00	90 DEG	1014.5	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1312.4	8.00	-200.0	-200.0	0.0	0.0	0.0	0.0
8.00	RCVR1	1691.8	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		2564.5	8.00	-200.0	-200.0	0.0	0.0	0.0	0.0
8.25	90 DEG	1019.1	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1348.8	8.00	-200.0	-200.0	0.0	0.0	0.0	0.0
8.25	RCVR1	1726.0	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		2662.4	7.50	-200.0	-200.0	0.0	0.0	0.0	0.0
8.50	90 DEG	1024.1	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1390.4	8.00	-200.0	-200.0	0.0	0.0	0.0	0.0
8.50	RCVR1	1763.5	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		2767.5	7.00	-200.0	-200.0	0.0	0.0	0.0	0.0
8.75	90 DEG	1030.0	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1437.4	7.50	-200.0	-200.0	0.0	0.0	0.0	0.0
8.75	RCVR1	1806.3	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		2879.9	6.50	-200.0	-200.0	0.0	0.0	0.0	0.0
9.00	90 DEG	1036.5	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1491.1	7.00	-200.0	-200.0	0.0	0.0	0.0	0.0
9.00	RCVR1	1856.8	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		3002.9	6.00	-200.0	-200.0	0.0	0.0	0.0	0.0
9.25	90 DEG	1042.5	1.50	-200.0	-200.0	0.0	0.0	0.0	0.0
		1548.2	7.00	-200.0	-200.0	0.0	0.0	0.0	0.0
9.25	RCVR1	1922.0	2.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		3133.4	6.00	-200.0	-200.0	0.0	0.0	0.0	0.0
9.50	90 DEG	1047.9	1.50	-200.0	-200.0	0.0	0.0	0.0	0.0
		1614.9	6.50	-200.0	-200.0	0.0	0.0	0.0	0.0

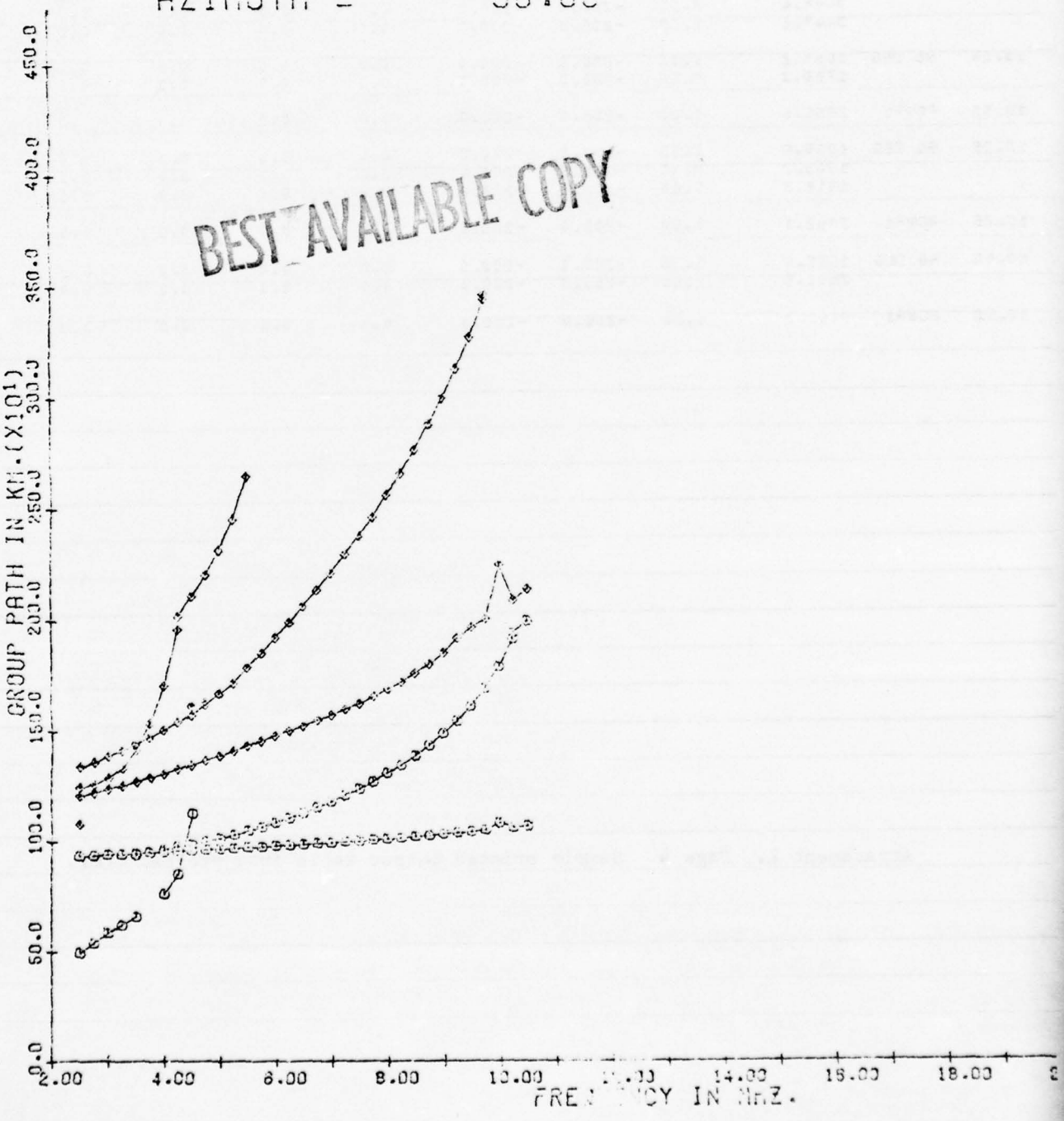
Attachment 1. Page 3. Sample printed output table from program GPMT.

BEST AVAILABLE COPY

		1614.9	6.50	-200.0	-200.0	0.0	0.0	0.0	0.0
9.50	RCVR1	1974.4	1.50	-200.0	-200.0	0.0	0.0	0.0	0.0
		3278.8	5.50	-200.0	-200.0	0.0	0.0	0.0	0.0
9.75	90 DEG	1053.6	1.50	-200.0	-200.0	0.0	0.0	0.0	0.0
		1694.8	6.00	-200.0	-200.0	0.0	0.0	0.0	0.0
9.75	RCVR1	2009.6	1.50	-200.0	-200.0	0.0	0.0	0.0	0.0
		3468.4	3.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		3447.1	5.00	-200.0	-200.0	0.0	0.0	0.0	0.0
10.00	90 DEG	1088.2	1.00	-200.0	-200.0	0.0	0.0	0.0	0.0
		1795.3	6.00	-200.0	-200.0	0.0	0.0	0.0	0.0
10.00	RCVR1	2251.6	1.00	-200.0	-200.0	0.0	0.0	0.0	0.0
10.25	90 DEG	1065.9	1.50	-200.0	-200.0	0.0	0.0	0.0	0.0
		1948.2	2.50	-200.0	-200.0	0.0	0.0	0.0	0.0
		1918.3	5.00	-200.0	-200.0	0.0	0.0	0.0	0.0
10.25	RCVR1	2092.1	1.50	-200.0	-200.0	0.0	0.0	0.0	0.0
10.50	90 DEG	1072.4	1.50	-200.0	-200.0	0.0	0.0	0.0	0.0
		2001.5	2.50	-200.0	-200.0	0.0	0.0	0.0	0.0
10.50	RCVR1	2142.2	1.50	-200.0	-200.0	0.0	0.0	0.0	0.0

Attachment 1. Page 4. Sample printed output table from program GPMT.

Attachment 2. Sample plotted output from program GPMT.



BEST AVAILABLE COPY

NAME: POWER, Revision 0, program, PML 145
CATEGORY: Companion program to ray-tracing program and program GPMT(PML 144)
TITLE: Power Calculations Along Minimum Group Path Trace
LANGUAGE: CDC Extended FORTRAN-version 4
PROGRAMMER: B. M. Langworthy, Parke Mathematical Laboratories, Inc.
DATE: July 28, 1976

DESCRIPTION

Program POWER is a companion program to program GPMT (See PML 144). Program GPMT determines the minimum group path trace while program POWER determines the signal strength for the points of the minimum group path trace. Program GPMT produces a file (TAPE88) of input for the ray-tracing program which will cause four rays to be traced for each of the minimum points, one of which will compute the power loss due to absorption. The output tape (TAPE7) of the ray-tracing program will then become the input tape for program POWER. Program POWER will compute the power losses due to spreading and backscatter as well as the transmitting and receiving antenna gains. These gains and losses will be combined with the absorption loss for a total power loss figure in db.

INSTRUCTION SET

Input to program POWER consists of two files or tapes which were previously produced by program GPMT and the ray-tracing program. These files are TAPE66, which is output from program GPMT, and TAPE7 which is output from the ray-tracing program when it uses TAPE88 (also produced by GPMT) for input. Figure 1 illustrates the relationship between programs GPMT, POWER, and the ray-tracing program.

Program POWER is meant to be part of an iterative procedure as follows:

Step 1. Rays are traced for a given ionospheric model for a variety of frequencies and azimuths at sufficiently fine increments in elevation angle. This produces TAPE7 as output.

Step 2. TAPE7 is used by program GPMT in selecting the group path minima. The program produces an output table and plots - but also produces input tape for the ray-tracing program (TAPE88) and a tape containing the tabular output of GPMT which is TAPE66.

Step 3. The ray-tracing program is run using TAPE88 as input and produces a TAPE7 output which contains sufficient information for computing power loss for all points along the group path minimum trace.

Step 4. Program POWER is then run using this new TAPE7 as input along with TAPE66 produced in Step 2. It will produce a TAPE44 identical in form to TAPE66 except containing complete data on power loss and the apogee height of rays.

Step 5. Meanwhile the user has examined the plots from Step 2 and decided which other frequencies and azimuths should be run. (This step may be automated later.)

Step 6. The ray-tracing program is run for additional frequencies and azimuths producing another TAPE7.

Step 7. This new TAPE7 along with TAPE44 produced from Step 4 is then used as input to program GPMT. Program GPMT will extract new minimum points from the TAPE7 data and add them to the table given on TAPE44 to produce a new TAPE66 and plots. The input tape for the ray-tracing program (TAPE88) will only contain runs for power information for

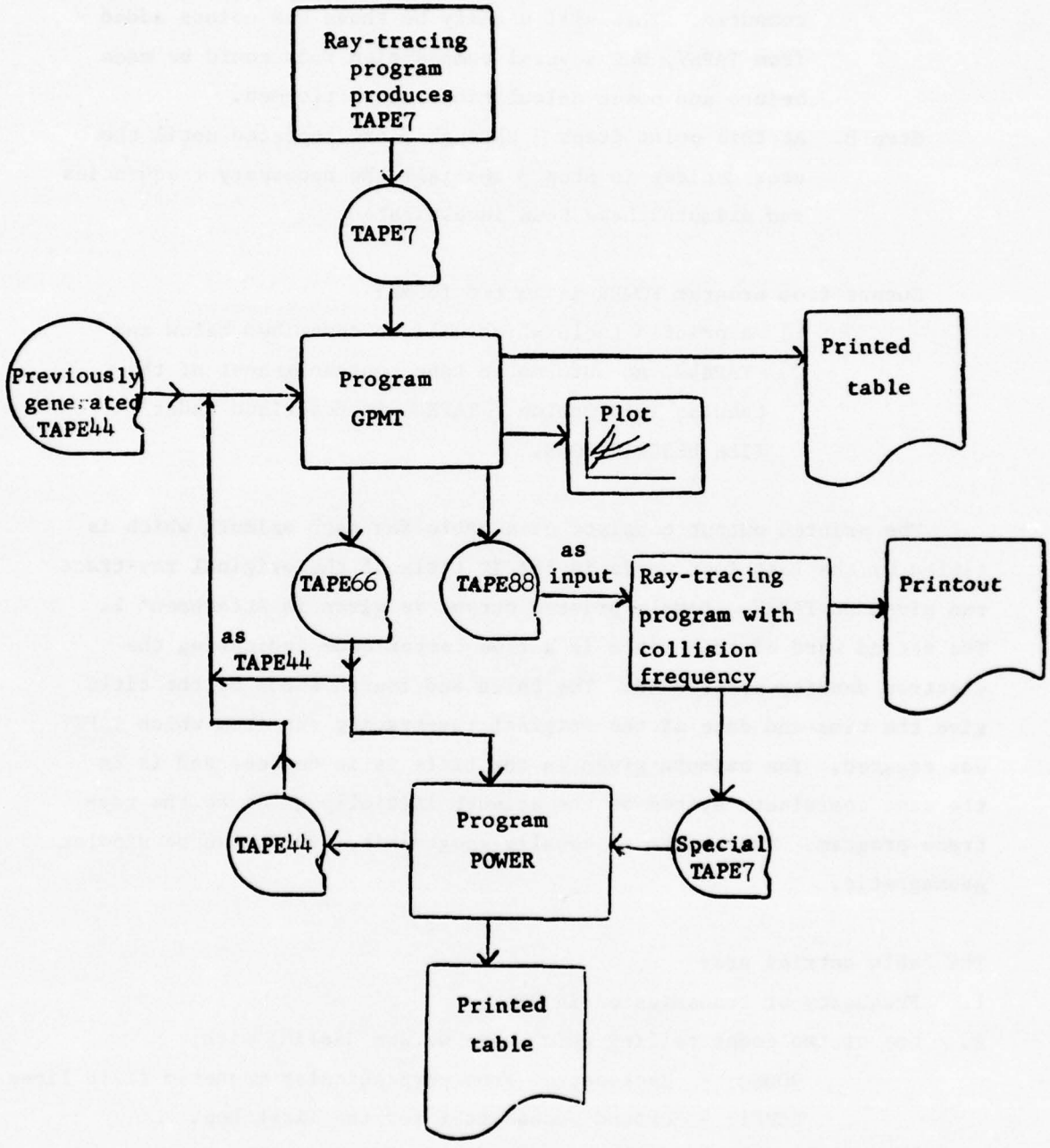


Figure 1: Flow diagram of interaction between the ray-tracing program, program GPMT, and program POWER.

those points for which power has not previously been computed. This will usually be those new points added from TAPE7, but several composition runs could be made before and power calculations are performed.

Step 8. At this point Steps 3 through 7 are repeated until the user decides in Step 5 that all the necessary frequencies and azimuths have been investigated.

Output from program POWER is in two forms:

1) a printed table which will be described below and
2) TAPE44, an unformated tape containing most of the tabular information. TAPE44 is described under FILE DESCRIPTIONS.

The printed output consists of a table for each azimuth which is titled by the last four words in the ID title of the original ray-trace run given on TAPE7. Sample printed output is given in Attachment 1. The second word of this title is a five letter code indicating the electron density model used. The third and fourth words of the title give the time and date of the original ray-tracing run from which TAPE7 was created. The azimuth given in the title is in degrees and is in the same coordinate system as the azimuth initially given to the ray-trace program. The system is usually geographic, but it can be dipolar geomagnetic.

The table entries are:

1. Frequency of transmission in MHz.
2. One of two codes telling which case we are dealing with;
   - 90DEG - Backscatter from perpendicular magnetic field lines
   - RCVR1 - Ground backscatter for the first hop.
3. Group path length in km. to the point of backscatter.
4. Elevation angle of transmission in degrees.
5. Spread loss gives the two way loss due to inverse square law spreading and focusing and defocusing effects. It is computed

using the four ray approach and assuming that backscatter will follow a path similar to the original transmission. Computation of the spreading loss is covered in more detail in TM-23. In the notation of TM-23 what we have listed as spread loss is

$$-20\cdot\log_{10}\left(\frac{\pi A_p}{\Delta S}\right) \quad \text{or} \quad 10\cdot\log_{10}\left\{\left(\frac{\Delta S}{4\pi A_p}\right)^2\right\}$$

where A_p is the flux tube cross-sectional area at the backscatter point. If a value of -200. is listed, this quantity has not been computed.

6. Absorption loss is taken directly from the output of the ray-tracing program. The absorption in db is integrated along the ray path and then doubled. Details of this calculation are also found in TM-23, pages 2 - 5. The quantity listed here corresponds to twice quantity A in TM-23. If a value of -200. is listed this value has not been computed.

7. Backscatter loss in most of the cases is actually again due to the backscatter area involved. The backscatter loss is $10\cdot\log_{10}\sigma$ where $\sigma = \sigma_o \cdot A$. σ is the total backscatter cross-section, A is the total area of the backscatter region intercepted by the flux tube, and σ_o is dependent on whether the backscatter is from the ground or field aligned ionization. If backscatter is from the ground,

$$\sigma_o = \frac{\sin\psi}{\frac{0.001}{\psi^2} + \sin^2\psi} + 10\cdot\exp\left(\frac{-3.35}{\tan^2\psi}\right) \quad \text{if } \psi > 10^{-5}$$

$$= 10^{-50} \quad \text{if } \psi \leq 10^{-5}$$

where ψ is mean flux tube angle of arrival in radians. If the backscatter is from field aligned ionization, $\sigma_o = 3.5$. A value of -50 usually indicates that backscatter has not been computed.

8. Transmitter gain is the gain of the transmitting antenna which may vary with transmission frequency, azimuth angle, and elevation angle. This is obtained from a function routine TRNSGN. One such routine for a logarithmic antenna is given in PML 142. If the antenna pattern is not known or if the antenna effects are to be ignored, this is simply set to zero.

9. Receiver gain is the gain of the receiving antenna which may vary with the transmission frequency, azimuth angle, and elevation angle. This is obtained from a function routine RCVRGN. One such routine for a rhombic antenna is given in PML 143. If the antenna pattern is not known or if the antenna effects are to be ignored, this will be zero.
10. The total gain is the sum of all the previous gains and losses with one additional quantity added in. This quantity is $10 \cdot \log_{10} \frac{\lambda^2}{4\pi}$. Thus the total gain should correspond to $10 \cdot \log_{10} P_r$ as given by Georges and Stephenson except that $P_o = 1$ and absorption losses are also included.
11. Height in km. of the point of backscatter in the 90DEG case or the point of apogee in the ground backscatter case. A value of zero indicates that this and the following two quantities are not yet available.
12. Accurate magnetic colatitude in degrees of the backscatter point.
13. Accurate magnetic longitude in degrees of the backscatter point.

Other printed output includes the message: TAPE44 HAS BEEN WRITTEN which indicated the new file has been successfully created.

STORAGE REQUIRED

Program POWER requires 200000 octal words of storage.

ALGORITHM

Algorithms for power calculations may be found in TM-23, entitled, "Absorption, Flux Tubes, and Backscatter."

SPECIAL CAUTIONS AND FEATURES

Program POWER can only be run using a TAPE7 from a ray-tracing run made on input from TAPE88 produced by program GPMT. All other TAPE7's will give errors since rays are not run in the proper sequence.

TIMING

Timing for a run producing power loss for 100 points is 4 seconds.

ERROR MESSAGES

NO MATCHING TABLE ENTRY FOUND.
F = XX.XX AZ = XX.XX EL = XX.XX WHY = AAAAA

This error message is printed out when ray information off of TAPE7 has no matching table entry in frequency, azimuth, and elevation in the tables from TAPE66. There is a tolerance of .000001 radians in the angles to account for any round off problems. The message will generally occur if TAPE66 and the TAPE7 generated using TAPE88 did not originate from the same execution of program GPMT. The program will proceed to search TAPE7 for more meaningful data, but the user should check to see that the power loss information is being attributed to the correct rays. F,AZ,EL, and WHY refer to the frequency, azimuth angle, elevation angle, and "90DEG" or "RCVR" code from the rays on TAPE7. Frequency is in MHz and the angle are in degrees.

NO INFORMATION ON TAPE66

Since the purpose of program POWER is to add power loss information to the tables produced by GPMT, namely TAPE66, a run with no information on TAPE66 is invalid. If this is the case, execution will stop. Check to see that TAPE66 has been properly attached.

FILE TAPE7 RECORD XXX RECORD MANAGER ERROR - 143
ERROR NUMBER 103 DETECTED BY IOERR AT ADDRESS 000021
CALLED FROM INPB = AT ADDRESS XXXXXX
CALLED FROM GET4R AT LINE XX
CALLED FROM POWER AT LINE 41
ROUTINE BEOT ENTERED.

This error message occurs when the last record of the TAPE7 file or tape is incomplete due to a non-standard ending on the ray trace run such as time limit or operator drop. Although this is normally a fatal error, it

has been overridden by a call to SYSTEMC and will instead be treated by subroutine GET4R as an end of file. The program will then proceed normally.

ERROR LATITUDE,LONGITUDE xxx.xxx xxx.xxx

This is an error message from the subroutine which converts geographic coordinates to accurate geomagnetic coordinates. The message appears if the inputted latitude is not between $+90^{\circ}$ and -90° or if the longitude is less than zero. The first quantity listed is the latitude and second quantity listed is the longitude. This message should not appear since subroutine DICOORD should check for this before CORRGM is called. The subroutine will try to correct the longitude by adding 360° but will proceed anyway.

NO TRIANGLE FOR XLAT,XLONG = xxx.xxx xxx.xxx

This may occur when the point for which the accurate geomagnetic coordinates are desired is in the region of the accurate magnetic pole. The region is divided into triangles which have the pole as one vertex. If a triangle cannot be located, it is assumed that this is the accurate magnetic pole and the above message will appear giving the coordinates in the geographic system in degrees. No value is assigned to the accurate geomagnetic longitude in this case. Computation then proceeds as normal.

SUBROUTINES

POWER This the the main program. It reads the TAPE66 to initialize tables to the current values. It then calls subroutine GET4R to get four rays for power calculation. With these four rays it locates their proper table entry, calls subroutine DEN for the flux density calculations, and calls subroutine BACSCAT for the backscatter calculations. This process is repeated until all sets of four rays have been read. The information is then printed and put on TAPE44.

GET4R This subroutine reads the information from TAPE7 for the required sets of four rays. GET4R requires that TAPE7 information be in a particular order which means that the TAPE7 must be ray-tracing results which used input data generated on TAPE88 of program GPMT. For more details on the format of TAPE7 see FILE DESCRIPTIONS.

DEN This subroutine calculates the ray density given four ray endpoints. See TM-23 for a more detailed description.

AREA1 This subroutine which is used by subroutine DEN calculates the area of the rectangle described by the four points.

BACSCAT This subroutine calls the appropriate subroutine to compute the backscatter for the "RCVR" and "90DEG" cases.

SIGGND This subroutine computes the ground backscatter coefficient.

SIG90 This subroutine computes the backscatter coefficient for rays perpendicular to magnetic field lines.

TRNSGN This is a function subroutine to compute the gain of the transmitting antenna. One such subroutine for a log periodic antenna is described in PML 142. If no antenna gain is desired, a dummy subroutine should be included with TRNSGN set to zero.

RCVRGN This is a function subroutine to compute the gain of the receiving antenna. One such subroutine for a rhombic antenna is described in PML 143. If no antenna gain is desired, a dummy subroutine should be included with RCVRGN set to zero.

PRNT(J) This subroutine computes total power and prints out tables of power values for the Jth azimuthal entry.

DICOORD This subroutine through entry point MTOG converts a point in colatitude and longitude from dipolar geomagnetic to geographic coordinates.

CORRGM2 This subroutine takes points in the geographic coordinate system and converts them to accurate geomagnetic coordinates. A call to CORRGM2 causes the transforming array to be read in. For points to be converted, a call must be made to CORRGM.

BEOT This subroutine is the error recovery routine for the Fortran fatal error 103 override, SYSTEMC, explained above. Subroutine BEOT sets an indicator so that subroutine GET4R will know an end of information exists.

ACCURACY

For a discussion on the accuracy of subroutine DEN on the absorption calculations, see TM-23. Accuracy of antenna patterns is covered in the individual PML describing them.

COMMENTS ON USAGE

Program POWER is used to give an indication of the POWER associated with points along the minimum trace. For some points this may not be possible since nearby rays needed for the computation of spread losses may penetrate.

FILE DESCRIPTIONS

The program card for POWER is as follows:

PROGRAM POWER(INPUT,OUTPUT,TAPE4=INPUT,TAPE7,TAPE66,TAPE44,TAPE1)

INPUT
TAPE4 — There is no system input under normal operation. This is included for future modifications.

OUTPUT — This is the system output file under normal operation. It contains the output tables described on pages 3-5 plus any error messages from the program or the system.

TAPE7 — This is an unformatted tape which is output from the ray-tracing program. This particular TAPE7 must have been created as a result of a ray-tracing run using TAPE88 of program GPMT as ray-trace input. It is composed of the following records:

Record 1: ID,W,NTYP,N

- ID — an information string of 100 characters which identifies the computer run made. The sixth word of this string identifies which characteristic of this ray we are interested in, RCVR or 90DEG.
- W — an array of 400 real numbers which contains all input to the program and other information,
- NTYP — an integer specifying the ray mode - ordinary, extraordinary, or neither (1,-1,0),
- N — an integer specifying the total number of equations to be integrated.

Record 2: R(1),R(2),R(3),IHOP,NWHY(1),T,(R(I),I=4,N),SIDEG,F,AZ,EL

- R(1) is the radius of the transmitter location in km,
- R(2) is the colatitude of the transmitter in spherical polar coordinates in radians,
- R(3) is the longitude of the transmitter in spherical polar coordinates in radians,
- IHOP is zero to indicate the transmitter data is given here,
- NWHY(1) tells the reason for the printout in this case XMTR or the transmitter,
- T is the group path length in km,

R(4) is the component of the wave normal in the positive radial direction. It is of magnitude such that the square root of the sum of its square with those of R(5) and R(6) have a magnitude equal to $\frac{\omega}{c.n}$ when ω is the angular wave frequency ($2\pi f$), c is the speed of elctromagnetic waves in free space, and n is the phase refractive index.

R(5) is the component of the wave normal in the positive θ direction.

R(6) is the component of the wave normal in the positive $\emptyset$ direction.

R(7) is the phase path length in km if W(57) $\neq$ 0.

R(8) is the absorption in decibels if W(58) $\neq$ 0.

R(9) is the doppler shift in Hz if W(59) $\neq$ 0.

R(10) is the geometrical path length in km if W(60) $\neq$ 0.

If any of these last four quantities are not requested then the succeeding quantities will be moved up in position. For instance, if phase path is not computed (W(57)=0), then R(7) will be absorption, R(8) will be doppler shift, and so on.

SIDEG is the angle between the magnetic field and the wave normal.

F is the transmission frequency in MHz.

AZ is the azimuth angle of transmission in radians in the input coordinate system which may be dipolar or geographic.

EL is the elevation angle of transmission in radians in the input coordinate system.

Record 3: R(1),R(2),R(3),IHOP,NWHY(2,3 or 4),T,(R(I),I=4,N),SIDEG,F,AZ,EL

The above record will be given for points along the ray path where the SIDEG is 90^{o}, the ray reaches apogee, or the ray touches the earth's surface. NWHY(2), NWHY(3), and NWHY(4) will be "90 DEG.", "RCVR", and "APOGEE", respectively. IHOP will indicate the hop that the ray is presently on.

Record 3 will be repeated for all such points along the path of this particular ray. Then the next ray will start with a record like Record 2, i.e. with IHOP = 0. When all elevation angles for a particular azimuth and frequency have been listed, the following records will appear:

Record n: F,AZ,DUM,-1,N,(W(I),I=1,NEQ)

- F is the transmission frequency in MHz.
- AZ is the current azimuth angle in radians in the input coordinate system.
- DUM is a dummy variable (most likely 0).
- N is the total number of equations to be integrated.
- W is the W array described in Record 1.
- NEQ is N + 2.

Record m: F,AZ,DUM,-2,N,(W(I),I=1,NEQ)

where F,AZ,DUM,N,W, and NEQ are as defined above.

The next W-array record, record 1, will then appear. For each point for which power is to be computed, three sets of records 1-m will appear. The first set gives one elevation angle and computes absorption. The second set gives the ray for a slightly higher elevation angle at the same azimuth, but gives no absorption. The third set gives rays for two elevation angles at a slightly higher azimuth angle.

TAPE66 This is an input file containing previously generated TAPE66 output from GPMT. It is written in unformatted records and contains 65 records with the following information:

Record 1: NAZE,NFRE,FT,AZT

- NAZE is the number of azimuth entries in the AZT table (≤ 10).
- NFRE is the number of frequencies in the FT table (≤ 40).
- FT is an array of 40 frequencies in MHz.
- AZT is an array of 10 azimuths in radians in the input coordinate system.

Record 2: ((GT(I,J,1),I=1,40),J=1,10) where GT is an array of the group path length minima in km for the Ith frequency entry and the Jth azimuthal entry.

Record 3: ((ELT(I,J,1),I=1,40),J=1,10) where ELT is an array of elevation angles in radians corresponding to the above group paths.

Record 4: (POWDT(I,J,1,),I=1,40),J=1,10) where POWDT is an array of spreading losses in decibels corresponding to the above group path minima.

Record 5: ((POWBS(I,J,1),I=1,40),J=1,10) where POWBS is an array of backscatter power losses in decibels corresponding to the above group path minima.

Record 6: (POWAT(I,J,1),I=1,40),J=1,10) where POWAT is an array of absorption losses in decibels corresponding to the above group path minima.

Records 2 through 6 are repeated increasing the third index of each array by 1 until 8 such sets are read. Records 2-21 will give information concerning the minima for the case of 90^{o} incidence to magnetic field lines. Records 22-41 will give information about 1-hop minima.

Record 42: ((HGT(I,J,1),I=1,40),J=1,10)
Record 43: ((COLAT(I,J,1),I=1,40),J=1,10)
Record 44: ((CLONG(I,J,1),I=1,40),J=1,10)

These three records are repeated 7 more times with the third index increasing up to 8. HGT,COLAT, and CLONG correspond to the coordinates of a point which in records 42 to 53 is the point of 90^{o} incidence to the magnetic field line and in records 54 to 65 is the receiver along the 1-hop path except that the height is the apogee height. Here HGT is the height of this point in km, COLAT is the colatitude in degrees in the dipolar magnetic coordinate system, and CLONG is the longitude in degrees in the dipolar magnetic coordinate system.

TAPE44 This output tape is identical in form to TAPE66. It does include additional information obtained from TAPE7. It may be used as TAPE66 input to program POWER or as TAPE44 input to program GPMT.

TAPE1 This is the permanent file which contains the transformation array for going from geographic to accurate geomagnetic coordinates. It is stored in both systems I and II under the file name MJDATA with ID=LOGICON.

SAMPLE DECK SETUP

1. Set up for run where TAPE7 is a multifile tape.

```
Job card with CM200000 and TP1
VSN(MFILE=YOURNO)
REQUEST(MFILE,MF,E)    (YOURNO/NORING)YOURNAME
LABEL(TAPE7,M=MFILE,P=1,R)
ATTACH(BPOW,BPOWERX3693818,ID=LANGW)
ATTACH(TAPE66,OLDTAPE66X3693818,ID=LANGW)
REQUEST(TAPE44,*PF)
ATTACH(TAPE1,MJDATA,ID=LOGICON)
LDSET(PRESET=ZERO)
BPOW.
CATALOG(TAPE44,NEWTAPE44X3693818,ID=LANGW,RP=999)
PURGE(TAPE66)
6/7/8/9
```

Here BPOWERX3693818 is assumed to be a compiled version of program POWER.

2. Set up for run where TAPE7 is a permanent file.

```
Job card with CM200000.
ATTACH(TAPE7,SPECIALTP7X3693818,ID=LANGW)
ATTACH(BPOW,BPOWERX3693818,ID=LANGW)
ATTACH(TAPE66,OLDTAPE66X3693818,ID=LANGW)
REQUEST(TAPE44,*PF)
ATTACH(TAPE1,MJDATA,ID=LOGICON)
LDSET(PRESET=ZERO)
BPOW.
CATALOG(TAPE44,NEWTAPE44X3693818,ID=LANGW,RP=999)
PURGE(TAPE66)
6/7/8/9
```

BEST AVAILABLE COPY

MEAS INT. DATA0 08/18/76 21.59.06. AZIMUTH = 30.00 TRNS LAT = 58.00 TRNS LONG = 1.80

FREQUENCY		GROUP PATH LENGTH	ELEVATION ANGLE	SPREAD LOSS	ABSRPTION LOSS	BACKSCAT LOSS	TRANSMITR GAIN	RECEIVER GAIN	TOTAL GAIN	MAX HEIGHT	AC MAG COLAT	AC MAG LONG
2.50	90 DEG	935.5	2.00	-200.0	-200.0	0.0	0.0	0.0	-469.4	0.	0.00	0.00
		494.6	50.00	-200.0	-200.0	0.0	0.0	0.0	-469.4	0.	0.00	0.00
2.50	RCVR1	1206.2	2.00	-133.0	-18.1	4.4	0.0	0.0	-216.1	102.	24.96	19.44
		1338.7	10.00	-200.0	-.1	0.0	0.0	0.0	-269.5	0.	0.00	0.00
		1245.4	34.00	-500.0	-2.3	0.0	0.0	0.0	-571.7	300.	27.70	12.83
		1079.1	48.00	-106.7	-.1	-5.5	0.0	0.0	-181.8	325.	28.45	11.37
2.75	90 DEG	939.2	2.00	-110.2	-2.1	-50.9	0.0	0.0	-233.4	102.	25.04	19.18
		539.5	44.00	-85.9	-.0	-63.1	0.0	0.0	-219.2	326.	28.35	11.55
2.75	RCVR1	1222.8	2.00	-134.1	-15.6	3.8	0.0	0.0	-216.2	102.	24.91	19.59
		1362.3	10.00	-121.0	-.1	-.3	0.0	0.0	-191.7	215.	25.34	18.82
		1262.7	34.00	-106.5	-.7	-5.3	0.0	0.0	-182.7	308.	27.53	13.13
3.00	90 DEG	942.7	2.00	-110.0	-1.9	-51.0	0.0	0.0	-233.9	102.	25.02	19.23
		584.6	38.00	-89.2	-.0	-61.4	0.0	0.0	-221.6	324.	27.90	12.44
3.00	RCVR1	1239.7	2.00	-133.9	-13.9	3.3	0.0	0.0	-215.6	103.	24.91	19.59
		1389.2	10.00	-119.6	-.2	-1.2	0.0	0.0	-191.9	217.	25.30	18.96
		1289.9	36.00	-110.3	-.4	-3.3	0.0	0.0	-185.0	323.	27.45	13.26
3.25	90 DEG	945.9	2.00	-110.6	-1.8	-50.7	0.0	0.0	-234.8	103.	25.00	19.28
		621.7	36.00	-92.7	-.0	-59.6	0.0	0.0	-224.0	330.	27.63	12.95
3.25	RCVR1	1256.7	2.00	-134.2	-12.7	3.1	0.0	0.0	-215.5	104.	24.90	19.59
		1415.7	10.00	-117.5	-.2	-2.4	0.0	0.0	-191.8	219.	25.25	19.10
		1336.3	36.00	-112.2	-.1	-2.1	0.0	0.0	-186.0	335.	27.17	13.71
3.50	90 DEG	949.3	2.00	-110.4	-1.7	4.7	0.0	0.0	-179.8	103.	24.99	19.33
		942.7	12.00	-97.1	-.0	-57.4	0.0	0.0	-226.9	245.	25.40	18.65
		659.4	34.00	-92.2	-.0	-59.9	0.0	0.0	-224.4	335.	27.34	13.47
3.50	RCVR1	1274.5	2.00	-134.2	-11.7	2.7	0.0	0.0	-215.5	104.	24.85	19.74
		1442.8	10.00	-118.5	-.2	-2.0	0.0	0.0	-193.0	221.	25.21	19.24
		1418.2	34.00	-116.4	-.1	-.4	0.0	0.0	-189.2	344.	26.68	14.53
3.75	90 DEG	952.7	2.00	-110.2	-1.6	4.6	0.0	0.0	-180.2	104.	24.97	19.38
3.75	RCVR1	1292.3	2.00	-133.8	-10.8	2.1	0.0	0.0	-215.4	105.	24.85	19.74
		1473.8	10.00	-115.6	-.2	-3.6	0.0	0.0	-192.4	223.	25.17	19.39
		1537.5	34.00	-106.8	-.1	-5.9	0.0	0.0	-185.8	361.	26.32	15.20
4.00	90 DEG	955.7	2.00	-110.7	-1.6	4.8	0.0	0.0	-181.0	104.	24.95	19.43
		965.9	12.00	-96.7	-.1	-57.7	0.0	0.0	-227.9	250.	25.31	18.97
		766.8	33.00	-91.2	-.0	-60.4	0.0	0.0	-225.1	365.	26.71	14.56
4.00	RCVR1	1310.4	3.00	-500.0	-7.3	0.0	0.0	0.0	-580.8	129.	24.50	22.59
		1508.6	10.00	-120.6	-.3	-1.3	0.0	0.0	-195.7	225.	25.11	19.68
		1710.3	33.00	-102.8	-.2	-9.3	0.0	0.0	-185.8	376.	25.98	16.15
4.25	90 DEG	959.0	2.00	-110.7	-1.5	4.8	0.0	0.0	-181.4	105.	24.93	19.48

Attachment 1 Sample Output Table of Program POWER

NAME: S2POBL, revision 0, program, PML 146
CATEGORY: Plotting program for use with electron density preprocessing program.
TITLE: Plasma frequency contour plotting program
LANGUAGE: CDC extended Fortran, version 4
PROGRAMMER: D. Bandes, Parke Mathematical Laboratories, Inc.
DATE: July 15, 1976

DESCRIPTION

Program S2POBL may be used to plot the contours of models to be given to S2PPR. It will plot models with or without the layer modifications and will also plot the perturbation models with or without layer modifications. This program has been separated from the preprocessing program S2PPR to keep the latter within reasonable core limits. Plots are composed of layer characteristics such as the plasma frequency of the various layers versus ground range for any great circle. The great circle over which contours are plotted must be specified in dipolar geomagnetic coordinates; the program does appropriate coordinate transformations to find the input data. Plots may contain more than one contour within the following limitations: height and plasma frequencies cannot be intermixed on one plot and the model cannot be mixed with the perturbation on one plot. However, all four types of plots may be specified on one data card, and the program will sort them out. A plot can be made up of the plasma frequency contours for all five sine square segments, any one contour, or any combination of several of these contours. Plots can deal with only one great circle on a particular set of axes.

Since S2PLT is separated from the preprocessing program, layer modifications can be assessed before the plasma frequency array is generated. This can save considerable computation time.

INSTRUCTION SET

The input data set to S2POBL is the same as that for S2PPR (see PML 136) except that input cards 1 and 2 are omitted since these two cards specify

the grid size for an output array which is not generated by this program. Additional input, to describe the plot, must follow the model and perturbation specifications. For each set of plots desired two input cards are required: the first specifies which contours are to be included on the plot, and the second specifies up to eleven great circles along which contours are to be plotted. These cards follow the complete contour specification as would be given to S2PPR.

Card 1 (24I3)

Cols 1 - 3 IPL(1)
Cols 4 - 6 IPL(2)
Cols 7 - 9 IPL(3)
etc. up to a maximum of 24 values of IPL

IPL is an array containing the numbers of the contours to be plotted. The number of a contour depends on the total number of sine square segments in the model. For instance, if there are 3 sine square segments the contour numbers will be as follows:

Contour 1 is the f_1 contour
" 2 is the f_2 contour
" 3 is the f_3 contour
" 4 is the H_0 contour
" 5 is the H_1 contour
" 6 is the H_2 contour
" 7 is the H_3 contour
" 8 is the H_{max} contour

If there are 5 sine square segments:

Contour 4 is the f_4 contour
" 5 is the f_5 contour
" 6 is the H_0 contour
and so on.

To plot the plasma frequency and height contours for the perturbation, add 20 to the above contour number. Thus the f_3 contour would be specified as contour 23.

Contour code numbers may be specified in any order; that is, the contours of the same type need not be contiguous on the card. Code numbers 0 or less, 13 to 19 inclusive, and 33 or more are invalid and are ignored by the program.

All IPL entries must be right-justified in the three spaces allowed and must not contain decimal points.

Card 2 (I10,7F10.0)

Cols 1 - 10	NPATHS	If this is between 1 and 11 inclusive, it is the number of great circles for which the plots are to be made. The number must be right-justified to column 10.
11 - 20	THETA0	Colatitude (dipolar geomagnetic) of start of all great circles specified on this card, in degrees.
21 - 30	PHI0	Longitude (dipolar geomagnetic) of start of all great circles specified on this card, in degrees.
31 - 40	RANGE	Maximum ground range to be plotted.
41 - 50	STARTAZ	Azimuth of first great circle to be plotted (degrees)
51 - 60	ENDAZ	Azimuth of last great circle to be plotted (degrees)
61 - 70	DELAZ	If NPATHS is zero or negative and if DELAZ is at least one-tenth ENDAZ-STARTAZ, this is the azimuth increment in degrees between successive great circles.

At most 11 great circles may be specified on any one card. If NPATHS is greater than 11 or if NPATHS is non-positive and DELAZ less than one-tenth ENDAZ-STARTAZ, the program resets NPATHS and DELAZ to specify 11 equally spaced great circles.

These two cards may be repeated as often as necessary to obtain all desired sets of plots. Suppose the model has 5 sine square segments and the perturbation has 3; we wish to plot f_3, f_4, and f_5 of the model, f_2 and f_3 of the perturbation, and H_0, H_3 and H_{max} of both the model and the perturbation (sample input case listed below):

		1	1	2	2	3	3	4	4	5	5	6	6	7
Cols	5	0	5	0	5	0	5	0	5	0	5	0	5	0
Card 1	3 4 5 22 23 6 9 12 24 27 28													
Card 2		3	32.		0.		5000.		30.		40.		0.	
Alternative Card 2		0	32.		0.		5000.		30.		40.		5.	

If we wish only to plot the H_4 layer of the model for a single great circle:

Card 1 10 (note that the number need only be right-justified in any I3 field)

Card 2	1	34.7	88.5	5000.	46.6

The output of S2POBL consists of a listing of input tables for the model and perturbation and the desired plots. Sample plots are given in attachment 1.

STORAGE REQURED

S2POBL requires 140000 octal words of core storage.

ALGORITHM

The layer modification computations performed by S2POBL are the same as those in S2PPR. Please refer to PML 136 for this algorithm.

TIMING

The timing of S2POBLwill vary, becoming greater with increased complexity of the layer modification. Timing will also depend on whether or not coordinate transformations are required. Since coordinates for the great circle to be traversed must be given in dipolar geomagnetic coordinates, coordinate transformations will be eliminated if the input data is also dipolar. It takes longer to plot curves over several great circles than the same number of curves all over the same great circle. A typical timing with plots for 5 layer characteristics is two seconds per great circle.

ERROR MESSAGES

PARAMETERS OUTSIDE LIMITS NUMBER OF SINE SQUARE CURVES = nnn nnn

This indicates that the number of sine square segments specified for either the model or the perturbation is greater than 5. The first number given is for the model and second is for the perturbation. If the error occurred in the model, the second number given will be 0. Check to see that the numbers are punched in the correct columns and that the input deck is set up correctly. This will cause a program stop before any plotter output is generated.

LAYER nnn HAS nnn ENTRIES. MODEL = nnn PERTURB = nnn

This indicates that the number of table entries from the Ith layer of either the model or the perturbation is greater than 25. The additional printout allows one to tell if it is the model or the perturbation which is out of range. Check to see that these numbers are punched in the correct columns and that the input deck is set up in the correct order. This will cause a program stop before any plotter output is generated.

ERROR LATITUDE, LONGITUDE nnn.nnn nnn.nnn

This is an error message from the subroutine which converts geographic coordinates to accurate geomagnetic coordinates. The message appears if the inputted latitude is not between $+90^{\circ}$ and -90° or if the longitude is less than zero. The first quantity listed is the latitude and the second is the longitude. This message whould never appear since subroutine DICOORD should check for this before CORRGM is called. The subroutine will try to correct the longitude by adding 360° and will proceed anyway.

NO TRIANGLE FOR XLAT, XLONG = nnn.nnn nnn.nnn

This may occur when the point for which the accurate geomagnetic coordinates are desired is in the region of the accurate magnetic pole. The region is divided into triangles which have the pole as one vertex. If a triangle cannot be located, it is assumed that this is the accurate magnetic pole, and the above message will appear, giving the coordinates in the geographic system in degrees. No value is assigned to the accurate geomagnetic longitude in this case. Computation then proceeds as normal.

SUBROUTINES

SETLAB This routine sets the entries of array LAB(12,3) in common /LABELS/ to be the correct labels (Hollerith data) for the layers.

ONEPLOT This routine plots the contours on one set of axes. Its parameters specify which contours to plot and how to label the plot.

S2POBLA This subroutine reads input data, writes it to OUTPUT with headings, and checks that the number of colatitudes read in is within bounds.

SOLVTRI This routine computes the colatitude and longitude reached after traveling a given distance along a given great circle. The algorithm is an application of the laws of cosines in spherical triangles.

ADDER This subroutine computes the additive and multiplicative variations to be made in the model and the perturbation as a function of colatitude and longitude.

S2POBLB This subroutine computes the values to be plotted.

DICOORD Transforms dipolar geomagnetic coordinates to geographic coordinates.

CORRGM2 Transforms geographic coordinates to accurate magnetic coordinates.

Program S2POBL also requires some AFCRL plotting software subroutines. See Section 14 of the AFCRL User's Guide for details. The plotting subroutines called are PLTID3, AXIS, ENDPLT, LINE, NUMBER, PLOT and SYMBOL.

ACCURACY

The curves put out by S2POBL are made up of points calculated every 10 km. of ground range. If a particular sine square segment happens over a small interval of distance, it will tend to look squared off. This in no way affects the values computed by S2PPR.

COMMENTS ON USAGE

S2POBL is intended as a companion program to S2PPR. It may be used when constructing models and modifying layers, and should be helpful in evaluating layer modifications. It may also be used before tracing rays to give an idea of ionoshpere characteristics along the ray paths. However, it only depicts characteristics of the sine square segments. If profiles of plasma frequency versus height are desired, S2PPR must be run first and then PROPLT must be used with electron density subroutine E14994. See PML 129 for a description of PROPLT and PML 127, rev. 1, for a description of E14994.

FILE DESCRIPTION

The program card is as follows:

```
PROGRAM S2POBL(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1)
```

TAPE1 should be the file MJDATA,ID=LOGICON, which has the data for conversion from geographic to accurate geomagnetic coordinates. The input deck to S2POBL is described on page 2.

DECK SET UP

```
Job Card with CM140000.
ATTACH(SOURCE,S2POBLX3693818,ID=BANDES)
FTN(I=SOURCE,B=BS2P)
ATTACH(TAPE1,MJDATA,ID=LOGICON)
ATTACH(PEN,ONLINEPEN)
LIBRARY(PEN)
LDSET(PRESET=ZERO)
BS2P.
DISPOSE(PLOT,*OL)
EXIT.
DISPOSE(PLOT,*OL)
7/8/9
model description data cards
perturbation description data cards
sets of 2 plot description cards
6/7/8/9
```

NAME:	AURLAT, Revision 0, function, PML 150
CATEGORY:	Coordinate transformation function
TITLE:	Northern Hemisphere Auroral Zone Accurate Geomagnetic Latitude Function
LANGUAGE:	CDC extended FORTRAN, Version 4
PROGRAMMER:	D. Bandes, Parke Mathematical Laboratories, Inc.
DATE:	August 12, 1976

DESCRIPTION

AURLAT returns the approximate accurate geomagnetic latitude of a point given in dipolar geomagnetic coordinates. The point should be in or near the northern-hemisphere auroral zone. The program uses a nine term Fourier series to compute the accurate geomagnetic latitude; coefficients of the Fourier series are computed from parabolas which fit the data at 60°, 68°, and 76° north latitude (dipolar). Only 27 data items need be stored. This program runs about twice as fast as the combination of DICOORD and CORRGM2 which may otherwise be used to do the coordinate transformations and requires about 13100 (=31500 octal) fewer words of memory.

CALLING SEQUENCE

A call to AURLAT is of the form
GMLAT=AURLAT(DLAT,DLON). DLAT and DLON are the dipolar latitude and east longitude of a point; the value returned by AURLAT is the accurate geomagnetic latitude of that point. DLAT is a latitude (not a colatitude), must be in degrees, and should be between 45 and 85. DLON must be in degrees, and may be positive or negative. Values of DLON greater than 360° are allowed.
No common storage is required.

STORAGE REQUIRED

AURLAT requires about 140 octal (96 decimal) words of memory.

PRECEDING PAGE BLANK-NOT FILMED

ALGORITHM

AURLAT approximates GMLAT(DLAT, φ) by

$$A_o(DLAT) + A_1(DLAT)\cos\varphi + B_1(DLAT)\sin\varphi + \ldots + A_4(DLAT)\cos 4\varphi + B_4(DLAT)\sin 4\varphi \ .$$

Each coefficient A_o ... B_4 is given by a quadratic polynomial which was chosen to fit the correct Fourier coefficients at latitudes of 60, 68, and 76 degrees. The 3 coefficients of each of these 9 quadratics are stored in the array X local to AURLAT. For instance,

$$A_o(DLAT) = X_{1,1}*(DLAT-68.)^2 + X_{2,1}*(DLAT-68.) + X_{3,1}.$$

ACCURACY

For dipolar latitudes between 48 and 84 degrees, AURLAT returns the geomagnetic latitude accurate to within .43 degrees. For latitudes between 56 and 80 degrees, AURLAT returns the geomagnetic latitude accurate to within .12 degrees, with standard deviation under .05 degrees.

TIMING

A test program calling AURLAT at each of 50 latitudes and 400 longitudes ran in 7.35 seconds on the CDC 6600, giving an average of under .4 microseconds per call.

NAME: MLRA, revision 0, program, PML 155
CATEGORY: Data fitting program
TITLE: Multiple Linear Regression Analysis
LANGUAGE: CDC Extended Fortran version 4
PROGRAMMER: B.M. Langworthy, Parke Mathematical Laboratories, Inc.
DATE: November 3, 1976

DESCRIPTION

Program MLRA is a generalized program to perform multiple regression analysis on arbitrary sets of data producing coefficients of the equation

$$Y_1 = a + b_1X_1 + b_2X_2 + b_3X_3 + \cdots + b_NX_N$$

The program is linear in that it uses each X-table entry in a first order manner. One may, however, enter a table of X_2's say which is in effect X_1^2 or exp (X_1) or whatever other non-linear function is desired. Input data is flexible allowing up to 12 variables with up to 20 entries each. Any entry table may be used as a dependent variable for up to 8 other tables as independent variables. If data is not available for certain observations of a variable, this can be indicated and that observation will be excluded.

The program computes means, coefficients, and the standard estimate of error. Since it has the flexibility to make multiple runs on various combinations of dependent and independent variables, the effective contribution of each independent variable can be measured.

The technique used for determining the multiple linear regression is least squares as given by Ezekiel + Fox, 1959.

INSTRUCTIONS

To use program MLRA the following data must be entered:

Card 1 (215)

Cols 1-5 M - The number of observations to be given in each table ($\leq$ 20)

Cols 6-10 N - The number of tables to be entered. ($\leq$ 12)

Card 2 (3A10)

Cols 1-30 TITLE(J,1),J=1,3 - A three word title for the data table which will appear in 3 lines of 10-characters each.

Card 3 + (8F10.0)

X(J,1),J=1,M - First data table of M entries. Entries are given 8 to a card. If data is not available for a particular observation, a-500. should be entered.

Cards 2 on are repeated giving a three word title followed by the data entries for each table. Once all tables are read in, the following cards appear.

Card n (14I5)

NDEP(I),I=1,6 - Numbers of the tables which are to be used as the dependent variables. There should be at least one entry but not more than 6 entries. The program will stop reading with the first blank, zero or nonpositive entry.

NIND(I),I=1,8 - Numbers of tables which are to be used as independent variables. No more than 8 such table numbers should be listed. The program will stop reading with the first blank, zero or nonpositive entry.

Card n may be repeated as many times as necessary to accomodate all the desired combinations of independent and dependent variables.

Last Card - This should be a card which is blank in columns 1-5.

A listing of sample input is given in Attachment 1.

The output from program MLRA consists of a table for each dependent variable with its independent variables. The column headings for the dependent variable and the independent variables are those given in the input data. In equation notation the columns are:

$Y\ X_1\ X_2\ \cdots\ X_N\ Y'\ Z$

\- - - - - - - -
\- - - - - - - -
.
. data entries
.
\- - - - - - - -

COLUMN MEANS

$\bar{Y}\ \bar{X}_1\ \bar{X}_2\ \cdots\ \bar{X}_N$

CONSTANT AND COEFFICIENTS

$a\ b_1\ b_2\ \cdots\ b_N$

STANDARD ERROR OF ESTIMATE = .XXXXE XX

COEFFICIENT OF MULTIPLE CORRELATION = .XXX EXX

where Y is the dependent variable

Y' is the linear regression value of Y

and Z is Y - Y'

The standard error of estimate and the coefficient of multiple correlation are explained on page 4.

Sample output is given in Attachment 2.

If the data entries for one observation contains a -500., that observation is not used. Y' for that observation will be -500. and Z will be zero. The standard error of estimate will be based only on the remaining observations.

STORAGE REQUIRED

The storage required for program MLRA is 40000 octal words of core storage.

ALGORITHM

The coefficients for the equation

$$Y = a + b_1X_1 + b_2X_2 + b_3X_3 + \cdots + b_NX_N$$ are solved for in the following manner:

Let $Y = Y_1, Y_2, \cdots, Y_M$

$X_1 = X_{11}, X_{12}, \cdots, X_{1M}$

$X_N = X_{N1}, X_{N2}, \cdots, X_{NM}$

The least squares constant, a, is defined as

$$a = Y - b_1\bar{X}_1 - b_2\bar{X}_2 - b_3\bar{X}_3 - \cdots - b_N\bar{X}_N$$

where the bars represent means. Substituting this definition for a in the initial equation will give us

$$Y_i - \bar{Y} = b_1(X_{1i}-\bar{X}_1) + b_2(X_{2i}-\bar{X}_2) + \cdots + b_N(X_{Ni}-\bar{X}_N)$$

for i = 1,2, $\cdots$ M

Substituting $y_i = Y_i - \bar{Y}$

$$x_{1i} = X_{1i} - \bar{X}_1$$

etc.

we have a set of M linear equations in N unknowns of the

$$y_i = b_1x_{1i} + b_2x_{2i} + b_3x_{3i} + \cdots + b_Nx_{Ni}.$$

The coefficients b_1, b_2, $\cdots$,b_N are solved for using IMSL subroutine LLSQAR give a least squares solution of an overdetermined systems of linear equations. Matrix solution is by Householder's *reduction to bidiagonal form.*

To evaluate the effectiveness of the regression coefficients, the adjusted standard error of the estimate, S, is computed.

$$S = \left[\frac{\sum_{i=1}^{M}(Y_i - Y^1)^2}{M - N}\right]^{1/2}$$

where Y^1 is the computed value of Y_i

M is the number of observations

N is the number of independent variables

The coefficient of multiple correlation is:

$$R = S \cdot \left[\frac{\sum_{i=1}^{M} Y_i^2}{M}\right]^{-1/2}$$

SPECIAL CAUTIONS AND FEATURES

No more than 20 observations may be entered without changing program MLRA. If some of the entries are not available for a particular observation, a -500. may be entered which will indicate that if this table is used that observation must be ignored in determining the coefficients. If -500. is a valid data entry, then -500.00001 should be used instead.

TIMING

Execution time will vary with the number of observations and the number of independent variables but should remain under one second per set of dependent variables for the current limits of 20 observations and 8 independent variables.

ERROR MESSAGES

THERE ARE NO INDEPENDENT VARIABLES REQUESTED.

This message occurs if a regression analysis is requested but no independent variable table numbers are given. This is input of the form of card n described on page 2. Check to see that the independent table numbers are given starting in Cols 41 to 45. If values are placed before this area, they will be regarded as dependent variables. If a value does not occur in columns 41 to 45, the program assumes that no data follows. If this message occurs, the program will proceed to the next "card n" and continue execution.

IMSL(UERTST) TERMINAL {LPSDOR / LLSQAR} 1 (IER=129)
ERROR RETURN FROM LLSQAR.
INPUT MATRIX IS SINGULAR.

This indicates that two of the independent tables are not orthogonal functions. Check to see that an independent variable table was not requested twice on a single "card n" (See page 2). Also check to see that one table is not a constant times another table. The program will proceed to the next "card n" and continue execution.

A normal stop of program MLRA is EXIT; any other stop indicates an error condition. The following stops may occur:

STOP 1000 — This will occur if an end of file is encountered while attempting to read a table title. Check that N, the number of table, is punched right justified in columns 6-10 of card 1.

STOP 1001 — This will occur if an end of file is encountered while attempting to read a data table. Check to see that card 1 is punched properly.

STOP 1002 This indicates that an end of file was encountered while attempting to read the first card. Check to see that data was included or that an extra 7/8/9 card was not in the deck.

SUBROUTINES

Program MLRA uses IMSL (International Mathematical and Statistical Library) subroutines LLSQAR, LPSDOR, LSVALR, UERTST, and VSORTM. Detailed documentation on these subroutines can be found in the Programmer of the Day room at the AFGL Computer Center.

ACCURACY

Accuracy of the matrix inversion will be controlled to the number of significant figures to which the data is specified to be accurate. Presently this is set at six significant figures.

COMMENTS ON USAGE

Program MLRA can also be used on curvilinear functions provided data is entered in table form, See page 1 for details. Results of the coefficient of multiple correlation can be used to measure the importance of individual independent variables in determining the dependent variable.

FILE DESCRIPTIONS

The program card for this program is as follows:

PROGRAM MLRA(INPUT,OUTPUT,TAPE4=INPUT)

INPUT and OUTPUT are described on pp. 2-3 and given in Attachments 1 and 2.

DECK SET UP

Job card w/ CM40000.
FTN(SL,R=3)
LGO.
7/8/9
Decks of MLRA,LLSQAR,LPSDOR,LSVALR,UERTST, and VSORTM.
7/8/9
Input data cards
6/7/8/9

REFERENCES

Ezekiel, M., and K.A. Fox, Methods of Correlation and Regression Analysis, John Wiley and Sons, N.Y., 1959, pp. 170-203.

Attachment 1 - Listing of sample input cards to program MLRA.

```
    6     6
STARTING  RANGE FS  FREQ=6.25
55.         58.         62.5        71.         82.         98.
SLOPE OF  FS WALL   FREQ=6.25
.08888      .08         .06956      .06154      .04624      .02667
GPL AT 90 UPPER ELEVFREQ=6.25
970.4       1031.9      1112.3      1221.5      1826.7      2024.7
GPL AT RCVUPPER ELEVFREQ=6.25
1913.6      1946.4      1993.5      2079.8      -500.       3315.7
SLOPE 90  UPPER ELEVFREQ=6.25
39.15       42.         44.3        48.2        183.6       154.15
SLOPE RCVRUPPER ELEVFREQ=6.25
142.5       141.2       143.9       126.95      -500.       297.7
     1     2                           3     4
     1     2                           3     4     5     6
     1     2                           5     6
     1     2                           3     5
     1     2                           4     6
```

STARTING RANGE F5 FREQ=6.25	GPL AT 90 UPPER ELEV FREQ=6.25	GPL AT RCV UPPER ELEV FREQ=6.25	COMPUTED Y	DIFFERENCE
5.5000E+01	9.7040E+02	1.9136E+03	5.4270E+01	7.3008E-01
5.8000E+01	1.0319E+03	1.9464E+03	5.8407E+01	-4.0683E-01
6.2500E+01	1.1123E+03	1.9935E+03	6.3679E+01	-1.1786E+00
7.1000E+01	1.2215E+03	2.0798E+03	7.0117E+01	8.8339E-01
8.2000E+01	1.8267E+03	-5.0000E+02	-5.0000E+02	0.
9.8000E+01	2.0247E+03	3.3157E+03	9.8028E+01	-2.8033E-02

COLUMN MEANS

6.8900E+01 1.2722E+03 2.2498E+03

CONSTANT AND COEFFICIENTS

3.4146E+01 8.4516E-02 -3.2342E-02

STANDARD ERROR OF ESTIMATE = 9.7789E-01

COEFFICIENT OF MULTIPLE CORRELATION = .9862E+00

SLOPE OF F5 WALL FREQ=6.25	GPL AT 90 UPPER ELEV FREQ=6.25	GPL AT RCV UPPER ELEV FREQ=6.25	COMPUTED Y	DIFFERENCE
8.8880E-02	9.7040E+02	1.9136E+03	8.7889E-02	9.9076E-04
8.0000E-02	1.0319E+03	1.9464E+03	8.0522E-02	-5.2196E-04
6.9560E-02	1.1123E+03	1.9935E+03	7.1218E-02	-1.6578E-03
6.1540E-02	1.2215E+03	2.0798E+03	6.0312E-02	1.2279E-03
4.6240E-02	1.8267E+03	-5.0000E+02	-5.0000E+02	0.
2.6670E-02	2.0247E+03	3.3157E+03	2.6709E-02	-3.8866E-05

COLUMN MEANS

6.5330E-02 1.2722E+03 2.2498E+03

CONSTANT AND COEFFICIENTS

9.5898E-02 -1.6115E-04 7.7539E-05

STANDARD ERROR OF ESTIMATE = 1.3554E-03

COEFFICIENT OF MULTIPLE CORRELATION = .9803E+00

Attachment 2 - Sample output from program MLRA.

NAME: CRRSET, revision 0, subroutine, PML 157
CATEGORY: Special purpose file processor
TITLE: RAYSET (elements) to RAY (bundles)
LANGUAGE: CDC Extended Fortran
PROGRAMMER: T.B. Barrett, Parke Mathematical Laboratories, Inc.
DATE: October 5, 1976

DECRIPTION

The purpose of CRRSET is to select various "ray elements" as produced by the ray-trace program and to combine these elements into "rays" which are then suitable for use with various information extraction subroutines. Thus CRRSET itself extracts information, restructures it but does not display it.

Most of the "useful" information from ray-tracing is contained in knowledge of each ray at its "landing points". These landing points (if any) include points where the ray strikes the earth, the earth's magnetic field perpendicularly, and the point where the ray leaves the transmitter. (The latter point is a "landing point" for backscattered rays but will not be referred to as a "landing point" in the sequel.) Such knowledge, which is also referred to as a "ray element", includes ray position, ray direction (wave normal and ray direction are coincident at the points mentioned) and quantities which are integrated along the ray such as group path length, phase path length, ray path length and ray path attenuation.

Thus a file is created during a ray trace run which includes a summary of each ray in the form of knowledge of the ray at various "interesting" points including the landing points mentioned above. Such interesting points include points where the ray enters and exits the ionosphere, "apogee" and"perigee" points, etc. This unformatted file which is called RAYSET in the sequel is the TAPE7 referred to in NITIAL. It has the following form:

ID - record
ray records (or elements)
ID - record
ray records (or elements)

PRECEDING PAGE BLANK-NOT FILMED

Each ID-record corresponds to a new W-array in the ray trace program and is of the form:

ID(10), W(400), NTYPE, N

where ID(10) - is a W-array identification record, the last 2 words of which include date and time of start of a new W-array respectively.

W(400) - is the W-array (see ref. 1)

NTYPE - is an integer specifying ray mode - ordinary, extra-ordinary, or neither (1, -1, 0)

N - is an integer specifying the total number of equations integrated (i.e. r(1) - r(N) contain useful quantities - see ref. 1)

Each ray record or element is of the form

R(1),R(2),R(3),HOP,WHY,T,R(4)-R(N),SIDEG,FREQ,AZ,EL

where HOP is an integer which is a key to the significance of the data in each ray record as follows:

HOP = -1

R(1)= FREQ (see below)

R(2)= AZ (see below)

R(3)= (no significance)

WHY = N

The rest of the record consists of the first 9 values of the W-array.

This record occurs at the end of each "elevation loop" of the ray trace program.

HOP = -2

same as for HOP = -1 except this record occurs at the end of the "frequency loop" i.e. just before another W-array (if any) is "executed".

For all other values of HOP the ray record contains information about the ray at the point described by HOP and WHY where:

R(1) - R(3)	= ray dipolar coordinates (r,θ,ϕ) (km, radians, radians)
T	= group path length (km)
R(4) - R(6)	= ray wave-normal vector (k_r,k_θ,k_ϕ)
R(7) - R(N)	= are integrated quantities such as phase path length or absorption. The actual significance of a particular R(i) depends on some W-array parameters (W(57) - W(60) see ref. 1)
SIDEG	= is the angle between the wave normal and accurate magnetic field (if any) (degrees)
FREQ	= transmitter frequency (Mhz.)
AZ EL	= ray take off azimuth and elevation angle (radians) where azimuth is with respect to geographic north-clockwise positive
WHY	= is a descriptive character string like XMTR for transmitter, 90 DEG. for wave normal perpendicular to mag. field, RCVR for earth intersection. (Hollerith left justified)
HOP	= is an integer which is the HOP number unless HOP = 0 in which case it signifies the ray starting point (as does XMTR).

For each ray there are usually several ray records starting with HOP = 0 at the transmitter and in the order of increasing group path. The terminal record for each ray is dependant on the fate of the ray.

As noted above each ray in RAYSET is usually represented by several records. For ray density calculations and many of the displays it is convenient to produce a secondary file which will be referred to simply as RAY. This is done by CRRSET which extracts data from a set of RAYSET records pertaining to a single ray to produce a single ray record. Each such record has the form:

frequency, azimuth, elevation, $P_1,P_2, \cdots P_n$

where	frequency	= transmitter frequency (Mhz)
	(azimuth, elevation)	= initial ray direction (radians) relative to a dipolar coordinate system
	P_1 to P_n	are ray data at selected "landing" points

Each P_i contains the data

HOP,WHY,R,Θ,$\emptyset$,K_r,K_Θ,$K_\emptyset$,T,SIDEG,Q(1)-Q(m)

where:

HOP	= hop (1,2,3,...)
WHY	= landing point description, e.g. 90 DEG. or RCVR
R,Θ,$\emptyset$	= point dipolar coordinates (km, radians, radians)
K_r,K_Θ,$K_\emptyset$	= wave normal vector at point
T	= group path length at point (km)
SIDEG	= angle between wave normal and earth magnetic field (degrees)
Q(1)-Q(m)	= are the "integrated quantities" (if any) phase path length, etc. where m is the number of such quantities (possibly 0).

The n points appear in increasing group path length order in each record.

The output of CRRSET is a "bundle" of "rays" of the above form which may or may not be in any particular order (by frequency and initial direction). As it now stands, however, the ray trace program puts out raysets in ascending order by elevation, azimuth and frequency in that sequence so that the output of CRRSET will have this order. Thus diagramatically the structure of RAYSET is:

1st W-array header
- frequency (1 to n-$freq_1$)
  - azimuth (1 to n -az_1)
    - elevation (1 to n -el_1)

2nd W-array header
- frequency (1 to n-$freq_2$)
  - azimuth (1 to n-az_2)
    - elevation (1 to n-el_2)

.
.
.

nth W-array header
- frequency (1 to n-$freq_n$)
  - azimuth (1 to n-az_n)
    - elevation (1 to n-el_n)

By using CRRSET in the proper mode and possibly using a sorting subroutine (see subroutine BIBSUB1,PML158) it is possible to assemble "rays" so that they will appear in the proper order for flux tube calculations. This order must be such that all rays of the same frequency are together and rays with neighboring azimuths must appear in neighboring records (see subroutine DEN,PML161).

In addition to ray records, the first record in RAY is an identification record with the following structure:

-999.,-999.,-999.,N,M,NINT,ID(4),R,Θ,Ø, (HOP,WHY,i=1,N),(title,i=1,NINT)

where:

N	= number of landing points in each ray (see below)
M	= length of each ray record
NINT	= number of integrated quantities
ID(4)	= last four words of the W-array record identifier. Note that for all modes this identifier is obtained from the first W-array record encountered on a call to CRRSET.
R,Θ,Ø	= transmitter location (assumed to be the same for all rays) (km, radians)
HOP,WHY	= landing point identifiers
TITLE	= title of integrated quantities, e.g. "PHASE", "ABSORP", "DOPPLER", "PATH"

USAGE

CALL CRRSET (IN1,OUT1,INC,OUTC,NET,STOP1,ARRAY,NREC,INDIC,M,IPO)

Input parameters

All input arguments are left justified Hollerith unless specified otherwise. Blank values cause defaults to be chosen as shown in parentheses. For explanation of permissible values and role of parameters see the notes indicated by asterisks.

IN1	input file name ("RAYSET")	*1
OUT1	output file name ("RAY")	*1
INC	input rewind control ("RR")	*2
OUTC	output rewind control ("RR")	*2

NET 6 element array for range control *3
as follows:

1,2	frequency	(0.,100.)
3,4	azimuth	(-180.,360.)
5,6	elevation	(0.,90.)

STOP1 subroutine stop condition (no default) *4

ARRAY an array of landing point name pairs in *5
the form of hop number (integer) and code
name for the point. (no default but first
element 0 causes all points to be chosen.)

IPO (integer array) position numbers on the *6
multifile tape if STOP1 = "MFILE"

Notes on input parameters:

*1 File names are the "logical file names" which may be up to 7 alphanumeric characters in length starting with a letter.

*2 The possible input values are as follows:
"RR" - rewind on file open and close
"NR" - rewind file on close only
"RN" - rewind file on open only
"NN" - do not rewind the file at all

Normally the input and output files are rewound when they are opened and closed. There are times however when it makes sense to not rewind. An example of this is given in the sample program shown at the end of this guide. In this case CRRSET was used to extract information from several input files and put it on one output file. The values of INC and OUTC interact somewhat with the values of STOP1. (refer to *4)

*3 The action of NET is such that only if frequency, initial azimuth and initial elevation angles fall within the ranges indicated is an output "RAY" produced. The possible input values are blank or Hollerith string which can be interpreted as a floating point number. Net values 1,2 refer to frequency; 3,4 to azimuth and 5,6 to elevation. The following example illustrates the use of NET:

(1) blank (2) freq 2	all frequencies greater than or equal to freq 2
(1) freq 1 (2) blank	all frequencies less than or equal to freq 1
(1) freq 1 (2) freq 2	all frequencies within the interval (freq 1, freq 2) inclusive

*4 The "recognized" values of STOP1 are "F", "W" and "MFILE". Any other value of STOP1 will cause a stop at end of input file; thus one can use a value of "EOF" as a reminder. The significance of these values is as follows:

"F" - stop whenever the frequency of the input ray changes value from the first value observed. If "F" is used, the output file should not be rewound on open the first time CRRSET is called since an automatic skip over the output header record takes place after an initial rewind.

"W" - stop whenever a new W-array occurs

"MFILE" - stop only when the last file of a multifile tape has been read. This stop is useful if ray elements from several files are to be combined into a single output file. If MFILE is used the input file must be on tape (multifile) and the control card, REQUEST(MFILE, MF,E) used.

*5 The values of array (in pairs) are used to match array names associated with the various ray elements in RAYSET. Whenever a match occurs, the corresponding element is added to RAY. Normally the first value in the pair to be matched is an integer hop number (not Hollerith) while the second is a descriptive Hollerith code e.g. 1, "RCVR". For a given set of ray elements (which make up a complete ray in RAYSET) an element is extracted and put into RAY only if an unmatched name pair of ARRAY remains. For example if the name pair 1,"90 DEG." appears once in ARRAY and several ray elements have the corresponding name pair, only the first such element will be selected. If the first value of ARRAY is set to Ø, all ray elements are taken. It should be noted that if this is done the records in RAY will

generally be of different lengths and at present unsuitable for the various display programs. A value of Ø should be inserted in ARRAY immediately following the last name pair since Ø is used as a stopping value for "reading" ARRAY into CRRSET.

*6 If the first element of IPO is Ø, all files of the "requested" multifile tape are used for input. Otherwise only those indicated in IPO are read. A value of Ø should immediately follow the last desired value.

output parameters

All values are integer unless otherwise specified.

NREC The number of rays in the output "bundle" of rays. Note that the header record is not included in this count. If the output file is not rewound on open, NREC should be set to zero in the calling program.

INDIC A value of 1 is returned by CRRSET when an "end-of-file" is encountered for the current input file. It is set to 0 whenever CRRSET is called.

M is the length of each output record if landing points have been specified in ARRAY.

Use of common storage

The following labeled common storage areas should be declared in the calling program or in some subroutine which must be loaded before CRRSET:

/FITAOUT/ (35)
/OUTBUF/ (514)
/DATA/ (512) (or M if M>412 which is unlikely)
/IBUF/ (19)

The following labeled common areas are used by CRRSET for communications with other subroutines:

/POSIT/ (1)
/SUB/ (2)
/FITACR/ (35)

STORAGE REQUIRED

2520_8 (1360_{10}) not including labeled common.

ALGORITHM

The parameters are first "translated" by subroutine TRANS. If a parameter can not be properly interpreted an exit is made from the subroutine.

All file manipulation (with the exception of OUTPUT) is done using Record Manager call statements. If STOP1 = "MFILE", CRRSET uses subroutine POSI to search for the required files (as specified in IPO).

CRRSET "keys" on values of HOP in the input file. The first record in RAYSET is assumed to be a "W-array, header" record. If it is not (as determined by length), CRRSET terminates. After the first record a check is made on HOP. If HOP = -2, the next record (if any) is a new "header" record; if HOP = 0, the start of a new ray is indicated; HOP = -1 is ignored (it indicates a new frequency); positive hop numbers are actual hop numbers to be associated with the ray.

The subroutine is essentially composed of various "phrases" which are so commented in the source listing. Thus there is the:

"translate parameters phrase"
"landing point check phrase"
"end of W-array phrase"
"output a record phrase"
"start a new ray phrase"
" 'net check' phrase"
"end of file phrase"
"possible multifile mode phrase"
"end of program phrase"

The phrases which are composed of one or more FORTRAN statements appear "at random" within the subroutine and are usually "addressed" by a statement number and left by a GO TO statement.

The only computation required is the conversion of ray initial azimuth from geographic to dipolar (or compute) coordinates. Rather than

converting a → a' directly, a' is computed from the initial values of k_θ and k_ϕ as:

$$a' = \pm \cos^{-1} \left(\frac{-k_\theta}{\sqrt{k_\theta^2 + k_\phi^2}} \right)$$

and a' has same sign as k_ϕ.

Since a' is a computed quantity, it is truncated to the 40 most significant bits (including exponent).

SPECIAL CAUTION AND FEATURES

Most precautions have been mentioned. They are summarized here:

1) If STOP1 = "MFILE" the input file must be on multifile tape and REQUEST(MFILE,MF,E) used along with the appropriate VSN statement immediately following the job card.
2) The input file must have the structure indicated under DESCRIPTION and summarized under FILE DESCRIPTION.
3) If STOP1 = "F" then the output file should not be rewound the first time CRRSET is called. In addition, NREC should be set to 0 in the calling subroutine.
4) Labeled common areas /FITAOUT/, /OUTBUF/, /IBUF/ and /DATA/ should be declared with the minimum sizes indicated by subroutines which appear before CRRSET in the load sequence.

In addition to various error messages which might be printed via OUTPUT, CRRSET also prints the following data:

1) input parameter values and their "translations" in octal. This translation is the actual value used with the subroutine and includes default substitution when appropriate.
2) header information including ID (10 words) and the first 17 values of the W-array.
3) the number of input records read, the number of output records, record length and the number of integrated quantities and their titles.
4) If STOP1 = "MFILE", the file position sought for and file label.

TIMING

Compilation time is approximately 2 seconds on the CDC6600. CP time is on the order of a second and probably always less than 10 seconds even for very large input files.

ERROR MESSAGES

"INPUT FILE DOES NOT START WITH W-ARRAY"

Occurs if the first record is not 412 words long or if any record after a record with HOP = -2 is not 412 words long.

"OUTPUT FILE DOES NOT HAVE A HEADER"

Occurs only if STOP1 = "F" and an output file was not previously created.

"CRRSET, FATAL ERROR, IHOP = n"

Occurs if the record following a header record does not have a HOP value of 0 indicating the start of a new ray.

"CRRSET, FATAL ERROR, PARAMETER n HAS BAD VALUE"

Occurs if an input parameter has a value which can not be properly interpreted. The value of n runs from 1 for IN1 to 11 for STOP1.

SUBROUTINES

(subroutines marked with * are considered to be special purpose and considered to be part of CRRSET. Others are normally obtained from a user library.)

EOFILE - end of file subroutine
ERROR2 - error subroutine associated with output file
ERROR3* - error subroutine for input file
IEOF - end of file function
INTCHK* - used for determining the "integrated quantities"
IPOS - converts numbers for POSI
NETCHK* - checks for "within range"

POSI - positions multifile tape to desired file

TRANS - translates character string parameters and substitutes defaults if required

ACCURACY

not applicable

COMMENTS ON USAGE

CRRSET is most often called from a main program which also includes one or more display programs which require records in a form produced by CRRSET. It is possible to extract information piecemeal from a single input file, for example the data corresponding to a single W-array can be processed through other subroutines then a loop made back to CRRSET for another batch of data which is associated with the next W-array. At the other extreme it is possible to extract data from several files, which may or may not exist as multifiles on several separate physical tapes. Once RAY has been produced it can be sorted (by BIBSUB1, for example) for further processing if necessary.

FILE DESCRIPTIONS

Three files are used by CRRSET. Only OUTPUT should be declared by a main program. The input file RAYSET and output file RAY were described under DESCRIPTION. Their structure can be summarized as follows:

RAYSET (w-type records as produced by Fortran unformatted write)

records are variable length with the first record length 412 words (this record is called a "header" record - there may be several "header" records). Header records are assumed to have the structure:

ID(10), W(400), NTYPE, N

ID is assumed to be Hollerith

W(1-17) is assumed to be floating point

N is an integer which specifies the <u>total</u> no. of integrated quantities.

All other records are assumed to be "ray element" records provided HOP $\geq$ 0 (see below). If HOP = -2 the next record, if any, is assumed to be a "header" record. If HOP = -1 the record is not used. HOP = 0 signifies the start of a new set of "ray elements". The structure of a ray element record is:

$$r_r, r_\theta, r_\phi, \text{HOP}, \text{WHY}, t, k_r, k_\theta, k_\phi, (\ -\ ),\ \psi_g, f, a, e$$

where:

- (r_r, r_θ, r_ϕ) is the dipolar coordinates of the element (km, radians) [floating point]
- HOP,WHY is the landing point name [integer, Hollerith]
- t is the group path length of this element (km) [floating point]
- (k_r, k_θ, k_ϕ) is the wave normal direction of the element [floating point]
- (-) is empty if N (total number of integrated quantities is 6. It consists of N-6 floating point numbers representing other integrated quantities [floating point]
- ψ_g is the angle between the wave normal and earth's magnetic field. (radians) [floating point]
- f,a,e initial frequency (MHz), azimuth, elevation (radians) [floating point]

RAY (w-type record, output file, usually fixed length records)

This file consists of 1 header record and the rest ray records. Each record, including the header record has the same length provided the parameter ARRAY does not have a first element of $\emptyset$. If ARRAY (1) = $\emptyset$ the length of each ray record is determined by the total number of elements for a given ray from RAYSET. The structure of each record is:

header record
(npts $\neq$ 0)
-999., -999., -999., npts, m, nint, ID(4), r, θ, $\emptyset$,
(HOP,WHY, i = 1, npts), (title (i), i = 1, nint)

where:

- npts = number of ray elements [integer]

- m = total record length [integer]
- nint = number of "integrated quantities" [integer]
- ID(4) = last four words from ID(10) in header record of RAYSET [Hollerith]
- r,Θ,Ø = transmitter dipolar coordinates (radians) [floating point]
- HOP,WHY = element hop and name [integer, Hollerith]
- TITLE = titles of "integrated quantities" [Hollerith]

If npts = Ø then m is set to zero and the (HOP,WHY) pairs are not included.

ray record

$f,a,e,P_1,P_2, \cdots, P_{npts}$ (npts ≠ 0)

or

$f,a,e,P_1,P_2, \cdots, P_{last}$ where P_{last} corresponds to the last element of a set of ray elements in RAYSET

and

P_i = HOP,WHY,$r_r,r_\Theta,r_\phi,k_r,k_\Theta,k_\phi$,t, Ψ_g,q(1), ··q(nint)

where:

- f,a,e = initial ray frequency, azimuth, elevation (MHz, radians) [floating point]
  note: azimuth is relative to compute coordinates
  Azimuth in RAYSET is relative to geographic coordinates.
- HOP,WHY = hop number, name [integer, Hollerith]
- (r_r,r_Θ,r_ϕ) = element coordinates (km, radians) [floating point]
- (k_r,k_Θ,k_ϕ) = wave normal direction of element [floating point]
- t = group path length (km) [floating point]
- Ψ_g = angle between wave normal and earth's magnetic field (degrees) [floating point]
- q(1)-q(nint) = integrated quantities [floating point]

EXAMPLE OF USE

In this example, data from three multifile tapes are combined into one RAY file. It is assumed that the transmitter location remains fixed, the ionospheric model remains the same and the number of integrated quantities is fixed. In order to accomplish this, three main programs are used since three different tapes must be mounted. These three main programs

are identical except that parameter OUTC is respectively "RR", "NN", "NR". Note that NREC is set to zero in MAIN since the output file is not rewound on open for the second and third MAIN programs. These 3 programs are called MAIN but stored (as object modules) in cycles 1, 2, 3 respectively of permanent file MAINX3683818. All required subroutines which are not part of CRRSET are stored in user library BARLIB. First the control deck, then the source listing of the 3 MAIN programs:

```
BARMA,CM77777,T100,TP1.
VSN(MFILE=RAYSE1/RAYSE2/RAYSE3)
ATTACH (BARLIB,····)
ATTACH (MAB,MAINX3683818,CY=1,···)
LIBRARY (BARLIB)
REQUEST (MFILE,MF,E)   (RAYSE1/NORING)
MAB.
RETURN(MAB)
UNLOAD (MFILE)
REQUEST (MFILE,MF,E)    (RAYSE2/NORING)
ATTACH(MAB,MAINX3693818,CY=2,···)
MAB.
RETURN(MAB)
UNLOAD (MFILE)
REQUEST (MFILE,MF,E)    (RAYSE3/NORING)
ATTACH (MAB,MAINX3693818,CY=3,···)
MAB.
```

Source listing:

```
      PROGRAM MAIN (OUTPUT)
      COMMON/FITAOUT/FITAOUT(35)
      COMMON/OUTBUF/OUTBUF(35)
      COMMON/IBUF/IBUF(100)
      COMMON/DATA/DATA(500)
      DIMENSION ARRAY (5), NEW(6), IPO(1)
      DATA (ARRAY=1, "90 DEG.", 1, "RCVR", 0)
      DATA (NET=6*0), (IPO=Ø)
      NREC=0
                                          "RR"
      CALL CRRSET ("RAYSET","RAY","RR", "NN" ,NET,"MFILE",
                                          "NR"
     $ARRAY,NREC,INDIC,M,IPO)
      END
```

References

1) Jones, R.M., Stephenson, J.J. : (1975) A Versatile Three-Dimensional Ray Tracing Computer Program for Radio Waves in the Ionosphere. U.S. Dept. of Commerce OT Report 75-76

NAME: BIBSUB1, revision 0, subroutine, PML 158
CATEGORY: General purpose
TITLE: File sort
LANGUAGE: CDC Extended Fortran - Version 4 or later
PROGRAMMER: T.B. Barrett, Parke Mathematical Laboratories, Inc.
DATE: July 16, 1976

DESCRIPTION

BIBSUB1 does a file sort using Fortran statements which can access the CDC Sort/Merge package.

USAGE

CALL BIBSUB1

Use of common storage for input

All of the controlling data for BIBSUB1 is passed through labeled common /PARAM/. This data is in the form of character strings (Hollerith), one word per datum. A blank word is replaced with default values as shown in brackets below: Data is listed in the order in which it should appear in /PARAM/. Refer to the notes marked by asterisks for more details and restrictions.

1) Input file name ["FIL"] *1
2) Output file name [input file name with "S" added at the end of the name]
3) Number of sort keys, NKEY [1] *2
4) Key origin (character position in record) [1]
5) Key length (characters) [10]
6) Sort code ["DISPLAY"] *3
7) Sort order (ascending, "A", or descending, "D") ["A"]
8) Colating sequence [depends on sort code] *4

9) repeat 4 through 8 NKEY times

3 + NKEY · 6) maximum input record length(characters) [5120]

notes on input

*1 File names are the "logical file names" which may be up to 7 alphanumeric characters in length starting with a letter. The input file must be essentially of the same type as produced by a Fortran unformatted write statement. Specifically it must be a sequential file with W-type records. Records need not be of constant length however. The output file is the same as the input file with a possible re-ordering of records. Both input and output files are re-wound before and after sorting. BIBSUB1 checks to see if the input file is empty and if so returns with a "fatal" error message.

*2 Sorting is done by key 1, then by key 2, etc. where key 1 starts at the character position given by /PARAM/(4) and key 2, if it exists, starts a /PARAM/(9) etc. The start of a record is at character position 1.

*3 The permissable values of sort code are:

"DISPLAY"	(display code)
"FLOAT"	(floating point number)
"INTEGER"	(integer number)
"LOGICAL"	(binary integer (unsigned))

*4 Collating sequence has meaning only if the sort code (see *3) is "DISPLAY". The permissable values are:

"ASCII6"	see Appendix A
"COBOL6"	"
"DISPLAY"	"
"INTBCD"	"
"MINE"	see Appendix B

It should be pointed out that data which is to be interpreted as a character string should be left justified. For example /PARAM/(6) might be "DISPLAYØØØØ". On the other hand, numerical data can appear anywhere within the word. For example /PARAM/(5) might be "ØØ1ØØØØØØØ" and would be correctly interpreted as integer 10.

Use of other common storage

In addition to /PARAM/ the following common storage areas must be declared in the calling program or in some subroutine which is loaded before BIBSUB1:

/FITAIN/(35)
/FITAOUT/(35)
/INBUF/(514)
/OUTBUF/(514)
/MESSA/(21)
/IBUF/ (length of first record of input file)

Also BIBSUB1 uses common storage /SUB/(2) to communicate with the file error processing subroutines.

STORAGE REQUIRED

BIBSUB1 requires 655_8 (429_{10}) not including labeled common. The "average" sort job (see EXAMPLE OF USE) should run with 60 to 65 K of central memory. Larger amounts of CM may speed up the sort process. It is imperative that an RFL directive be included in the control deck since the sorting subroutines depend upon the use of "scratch" storage over and above that required to store the load module. If RFL is not used, storage is automatically decreased to that required to store the load module.

ALGORITHM

Parameters (in /PARAM/) are translated and defaults substituted where required. If a proper translation can not be made an error message is produced and a return is made from the subroutine with IFATAL set to one.

The input file is opened and a record read. If parameter "FP" in the file information table is 100_8, an end of file has been encountered (no information) and the subroutine returns with an error condition (IFATAL = 1).

The sort is done using the following calls to Sort/Merge:

```
        CALL SMSORT     (max. record length)
        CALL SMOPT ("RETAIN")
        CALL SMFILE ("SORT",···)
        CALL SMFILE ("OUTPUT",···)
NKEY  ( CALL SMKEY (····)
times { CALL SMSEQ (···)   (if collating sequence is "MINE")
        CALL SMEND
```

Record Manager calls are used for all file manipulation except output messages are handled using standard Fortran write statements.

SPECIAL CAUTION AND FEATURES

BIBSUB1 can only be used with w-type records, thus it can not sort "formatted" files. Be sure to use LIBRARY(COBOL) (see EXAMPLE OF USE) and RFL. Also be sure to declare the labeled common areas indicated previously. IFATAL (in /MESSA/) is set to 1 if a fatal error occurs in that BIBSUB1 can not do the required sort because of an inappropriate parameter or non-existant input file. The calling program can check this error flag and take appropriate action.

The input parameters are printed out in the form they are received in /PARAM/ along with their "translations" in octal form.

BIBSUB1 also prints out the CPU seconds required for the sort.

TIMING

The time required for a sort depends upon the number of records in the input file, the length of these records, the amount of central memory required, the number of sort keys and the sort code ("integer" or "logical" sorts are the fastest). Experience has shown that 20

seconds is adequate for files with around 10000 records, 2 or 3 sort keys, a record length of 32 words and "scratch" space of 20 to 30 kilowords.

ERROR MESSAGES

"INPUT FILE input file name DOES NOT EXIST"

Occurs if the input file is empty.

"BIBSUB1", FATAL ERROR, PARAMETER n HAS BAD VALUE"

Occurs if an input parameter has a value which can not be properly interpreted. The value of n is the parameter number as given in USAGE.

Sort/Merge also provides a large repertoire of error messages. Refer to CDC Sort/Merge manual page E-1.

SUBROUTINES

The following "user" subroutines are required. They are normally obtained from a user library.

ERROR1	error subroutine associated with input file
ERROR2	error subroutine associated with output file
MESSAGE	prints out information and error messages
TRANS	translates character string parameters and substitutes defaults if required

FILE DESCRIPTIONS

The only files used explicitly by BIBSUB1 are the input file to be sorted and the sorted output file. Neither of these files need be declared in a program statement although it has been found in practice they may be declared there if other subroutines have to use conventional Fortran statements to manipulate these files. These files are essentially those which would be produced by an unformatted write statement in Fortran.

Error messages are printed out via subroutine MESSAGE which usually uses OUTPUT for its output file. In any case OUTPUT should be declared in the main program statement.

EXAMPLE OF USE

In this example all default values are used except a floating point sort is done. The output file is stored on a permanent file for later use.

First the control deck, then the source listing of the MAIN program: All subroutines except MAIN itself are assumed to be stored in user library BARLIB.

```
BARSR,CM65000,T30
ATTACH(BARLIB)
ATTACH(FIL,------)
REQUEST(FILS,*PF)
FTN(SL,R=3)
LIBRARY(COBOL,BARLIB)
RFL(65000)
LGO.
CATALOG(FILS,-----)
7/8/9
  [program MAIN]
6/7/8/9
Source listing of MAIN
PROGRAM MAIN(OUTPUT)
COMMON/FITAIN/FITAIN(35)
COMMON/FITAOUT/FITAOUT(35)
COMMON/INBUF/INBUF(514)
COMMON/OUTBUF/OUTBUF(514)
COMMON/MESSA/IFATAL,MFSSA(20)
COMMON/IBUF/BUF(100)
COMMON/PARAM/PARAM(10)
DATA (PARAM = 5* " ","FLOAT", 4* " ")
CALL BIBSUB1
END
```

APPENDIX A

6-BIT CHARACTER CODE COLLATING SEQUENCES

COBOL6†		DISPLAY†		INTBCD		ASCII6††	
Graphics	Display Code	Graphics	Display Code	Graphics	CDC INTBCD	Graphics	Sequence
blank	55	:†	00†	0	00	blank	00
≤	74†	A	01	1	01	!	01
%†	63	B	02	2	02	"	02
[	61	C	03	3	03	#	03
→	65	D	04	4	04	$	04
≡	60	E	05	5	05	%††	05
∧	67	F	06	6	06	&	06
↑	70	G	07	7	07	'	07
↓	71	H	10	8	10	(	10
>	73	I	11	9	11	)	11
≥	75	J	12	:	12	*	12
¬	76	K	13	=	13	+	13
.	57	L	14	≠	14	,	14
)	52	M	15	≤	15	-	15
;	77	N	16	%	16	.	16
+	45	O	17	[	17	/	17
$	53	P	20	+	20	0	20
*	47	Q	21	A	21	1	21
-	46	R	22	B	22	2	22
/	50	S	23	C	23	3	23
,	56	T	24	D	24	4	24
(	51	U	25	E	25	5	25
=	54	V	26	F	26	6	26
≠	64	W	27	G	27	7	27
<	72	X	30	H	30	8	30
A	01	Y	31	I	31	9	31
B	02	Z	32	<	32	:	32
C	03	0	33	.	33	;	33
D	04	1	34	)	34	<	34
E	05	2	35	≥	35	=	35
F	06	3	36	¬	36	>	36
G	07	4	37	;	37	?	37
H	10						

APPENDIX A (continued)

COBOL6 †		DISPLAY †		INTBCD		ASCII6 ††	
Graphics	Display Code	Graphics	Display Code	Graphics	CDC INTBCD Code	Graphics	Sequence
I	11	5	40	-	40	@	40
∨	66	6	41	J	41	A	41
J	12	7	42	K	42	B	42
K	13	8	43	L	43	C	43
L	14	9	44	M	44	D	44
M	15	+	45	N	45	E	45
N	16	-	46	O	46	F	46
O	17	*	47	P	47	G	47
P	20	/	50	Q	50	H	50
Q	21	(	51	R	51	I	51
R	22	)	52	∨	52	J	52
]	62	$	53	$	53	K	53
S	23	=	54	*	54	L	54
T	24	blank	55	↑	55	M	55
U	25	,	56	↓	56	N	56
V	26	.	57	>	57	O	57
W	27	≡	60	blank	60	P	60
X	30	[	61	/	61	Q	61
Y	31	]	62	S	62	R	62
Z	32	%†	63†	T	63	S	63
:	00 †	≠	64	U	64	T	64
0	33	→	65	V	65	U	65
1	34	∨	66	W	66	V	66
2	35	∧	67	X	67	W	67
3	36	↑	70	Y	70	X	70
4	37	↓	71	Z	71	Y	71
5	40	<	72	]	72	Z	72
6	41	>	73	,	73	[	73
7	42	≤	74	(	74	\	74
8	43	≥	75	→	75	]	75
9	44	¬	76	≡	76	^	76
		;	77	∧	77	_	77

†Under the CDC 63-character set, there is no percent graphic; the colon is display code 63. Display Code 00 is not used.

††Under the ASCII 63-character set, there is no percent graphic; the colon collates in position 05, not position 32.

APPENDIX B

If the value of parameter 6 is "DISPLAY" and parameter 8 is "MINE" (and similarly for corresponding parameters for additional sort keys), the following collating sequence is used:

: (actually 00_8)
blank
A
|
Z
0
|
9
+
–
*
/
(
)
$
=
,
.
#
[
]
%
"
_ (underline)
!
&
'
?
<
>
@
\
^
;

NAME: BIBSUB2, revision 0, subroutine, PML 159
CATEGORY: General purpose file processor
TITLE: Sequential to word addressable transformer
LANGUAGE: CDC Extended Fortran
PROGRAMMER: T.B. Barrett, Parke Mathematical Laboratories, Inc.
DATE: July 22, 1976

DESCRIPTION

BIBSUB2 creates a "random access" (acutally word addressable,WA) file from a sequential file with fixed length records. This WA file is similar to that produced by a CDC Fortran WRITMS. However no record index is produced as is done using the standard "mass storage file" routines which are available within CDC Extneded Fortran. For this reason BIBSUB2 is most useful for creating files with fixed length records wherein the desired record location may be easily computed. (see next section) Because no index array is required, the use of BIBSUB2 can save considerable storage in programs which require the use of random access capabilities.

USAGE

CALL BIBSUB2

Use of common storage for input

All of the controlling data for BIBSUB2 is passed through labeled common /PARAM/. This data is in the form of character strings (Hollerith), one word per datum. A blank word is replaced with default values as shown in brackets below. Data is listed in the order in which it should appear in /PARAM/. Refer to the notes marked by asterisks for more details and restrictions:

1) Input file name ["FIL"] *1
2) Output file name (input file name with "R" at end)

note

*1 The input file name is the logical file name which consists of up to 7 alpha-numeric characters starting with a letter. If

PRECEDING PAGE BLANK-NOT FILMED

the default output file name is to be used, the input file name can consist of at most 6 characters. File names should be left justified. The input file should be sequential and consist of w-type record (e.g. as produced by a standard Fortran unformatted write statement). The maximum record length is 512 words. Input and output files are rewound before and after use.

Use of common storage

The following labeled common storage areas should be declared in the calling program or in another subroutine which is loaded before BIBSUB2:

/FITAIN/ (35)
/FITAOUT/ (35)
/INBUF/ (514)
/OUTBUF/ (514)
/IBUF/ (maximum record length)
/MESSA/ IFATAL, (20)

The following labeled common storage areas are used by BIBSUB2 to communicate with other subroutines:

/SUB/ (2)

STORAGE REQUIRED

BIBSUB2 requires 322_B (210_{10}) not including common storage areas and other anciliary subroutines. It will usually run with a good deal less than 60 K (octal) words.

ALGORITHM

The parameters are first "translated" by subroutine TRANS. If a parameter can not be properly interpreted an exit is made from the subroutine.

All file manipulation is accomplished using Record Manager call statements.

A check for a non empty input file is made before further processing is done. If input is empty BIBSUB2 returns with IFATAL=1 in /MESSA/.

The length of records in the output file, is obtained by a call to IFETCH(···,2LRL) on the first input record. Thus, even though the input file does not necessarily consist of fixed length records, the output file does.

After the last record has been copied from input to output, a zero length "record" is appended to the output file. A test for this record may be made (see EXAMPLE OF USE) if, for example, a sequential read is done and a normal stop at end of file is desired.

SPECIAL CAUTION AND FEATURES

The input file need not consist of fixed length records but in most instances they should be, since the output file does contain fixed length records, the length of which is equal to the length of the first input record.

IFATAL is set to 1 if the input file is empty.

TIMING

Very little CP time is required since this is basically a file manipulation subroutine. In most instances less than 10 CP seconds are required.

ERROR MESSAGES

"BIBSUB2, INPUT FILE file name EMPTY"

Occurs if the input file does not exist i.e. contains no records.

"BIBSUB2, FATAL ERROR, PARAMETER n HAD BAD VALUE"

Since the only input parameters are file names the message will occur only if the file name is more than 7 characters in length.

SUBROUTINES

These subroutines are normally obtained from a user library.

EOFILE, IEOF } (end of file for input)
ERROR1 (error processing for input)
ERROR2 (error processing for output)
TRANS (translates input parameters)
MESSAGE (writes informative and error messages)

ACCURACY

Not applicable.

COMMENTS ON USAGE

BIBSUB2 is useful for creating a random access file within a system of subroutines, in particular when it is desirable to eliminate the need for an index array. It is necessary, however, to access this file using Record Manager calls (see EXAMPLE OF USE).

FILE DESCRIPTIONS

Files need not (and probably should not) be declared on a program statement.

input sequential, fixed length (maximum 512 words), w-type records (e.g. Fortran unformatted file)
output word addressable, fixed length (maximum 512 words), w-type records

EXAMPLE OF USE

The following program creates a random access file with logical file name RAYSR from an input file with logical file name RAYS.

```
PROGRAM MAIN (OUTPUT)
COMMON/FITAIN/FITAIN(35)
COMMON/FITAOUT/FITAOUT(35)
COMMON/INBUF/INBUF(514)
COMMON/OUTBUF/OUTBUF(514)
COMMON/MESSA/IFATAL,MESSA(20)
```

```
COMMON/PARAM/PARAM(2)
DATA (PARAM = "RAYS", "  ")
CALL BIBSUB2
END
```

The following example gives the elements for accessing the file produced by the above example. Common storage areas /FITAIN/, /INBUF/ and /IBUF/ are used here but need not be in common in general. The error subroutine ERROR1 should be declared external.

```
COMMON/FITAIN/FITAIN(35)
COMMON/INBUF/INBUF(35)
COMMON/IBUF/BUF(100)
EXTERNAL ERROR1
CALL FILEWA(FITAIN,3LLFN,5LRAYSR,2LRT,1LW,
$3LMRL,5120,2LEX,ERROR1,3LBFS,514,3LFWB,INBUF)
CALL OPENM(FITAIN,5LINPUT)
:
IWA = (LREC + 1) * (IDREC - 1) + 1
```

IWA is the address of the desired record in RAYSR. It is found by adding 1 to the nominal record length, LREC (since w-type records include a control word which is not normally included in computing record length), multiplying this by the desired record number IDREC less one and adding 1.

```
CALL GET(FITAIN,BUF,IWA)
:
CALL CLOSEM(FITAIN)
```

The "end of file" subroutine IEOF may be used in case the number of records in RAYSR is not known. For example:

```
IF (IEOF(X) .EQ. Ø) 100,110
(IEOF is zero if no error condition exists)
```

If the file is being read sequentially a check can be made for zero record length as follows:

```
IF (IFETCH(FITAIN,2LRL) .EQ. 0) GOTO 100
```

NAME: TRANS, revision 0, subroutine, PML 160
CATEGORY: General purpose
TITLE: Translation of character string data
LANGUAGE: CDC Extended Fortran
PROGRAMMER: T.B. Barrett, Parke Mathematical Laboratories, Inc.
DATE:

DESCRIPTION

TRANS "decodes" character string data into the form it is normally used in a standard Fortran program. Specifically it will produce the following data forms:

integer
floating point
character string left justified, blank fill
character string left justified, zero fill
character string left justified, ; fill (special)

In addition it substitutes default values when required.

USAGE

CALL TRANS(PARAM,VAR,ITYPE,DEFAULT), RETURNS(M)

parameters

PARAM Character string parameter to be translated or default " " or "?".

VAR variable into which the translated parameter is placed (output of TRANS). If ITYPE = 4, VAR may also be used for input. See DEFAULT

ITYPE controls decoding as follows (integer)

1 integer
2 floating point
3 character string left justified, blank fill
4 " " " " , zero fill
5 " " " " , semi-colon fill

DEFAULT default value to be put in VAR if PARAM is "blank" or "?". DEFAULT should contain data in a form which

PRECEDING PAGE BLANK-NOT FILMED

agrees with ITYPE (with one exception (see below)). For example if ITYPE = 2 (floating point), DEFAULT should contain a floating point number (not in character string representation). The exception is for ITYPE = 4 which is used mainly to produce file names which must be left justified, zero filled. In this case DEFAULT can be in the form

"/string"

where string is a sequence of characters which are to be concatenated with those in VAR to produce the output of TRANS.

For example:

```
F1 = "RAY"
F2 = " "
CALL TRANS(F2,F1,4,"/S"), RETURNS(1000)
```

will produce an output value of F1 ="RAYS". Note that in this case VAR is used both for input and output. If ITYPE = 5, the results are similar except that the word is filled with semi-colons rather than zeroes.

use of common storage

The following labeled common storage area should be declared in the calling program or in a subroutine loaded before TRANS:

```
/MESSA/IFATAL, (20)
```

The following labeled common storage area is normally used for passing the calling subroutine name to TRANS:

```
COMMON/SUB/SUB,ICODE
```

If SUB is set to the character string name of the calling subroutine, this name will be printed along with the value of PARAM and its "translation" (octal) in VAR.

abnormal return

If a "translation" can not be made, TRANS will return to the calling subroutine at the statement number assigned to M. In addition IFATAL is set to one. This condition occurs when ITYPE=1 or 2 if PARAM can not be properly decoded. For example, if ITYPE=1 and PARAM="1.5" an error occurs since "1.5" is not an integer. If ITYPE=4 and the number of letters in VAR is greater than 7, an error has also occurred. If ITYPE=4 or 5 and VAR is blank, an error has also occurred.

STORAGE REQUIRED

The amount of storage required for TRANS and its associated special subroutines (those marked with * under SUBROUTINES) is 442_8 (290_{10}).

ALGORITHM

If ITYPE=1 or 2, PARAM is right justified and a DECODE with the appropriate format used. If "illegal data in field" is encountered, the count in ERRSET is incremented. A check for this condition is made and if it occurs, TRANS returns to M. If ITYPE=3 (character string data) TRANS merely equates VAR with PARAM if PARAM is neither " " or "?"; otherwise VAR is equated with DEFAULT. If ITYPE=4 and DEFAULT is not of the form "/string", the shifted blanks in PARAM or DEFAULT are searched for from the right by shifting to the right and masking. Zero fill is then done by "and-ing" PARAM or DEFAULT with the appropriate length mask. A similar scheme is used if ITYPE=5 except by "or-ing" with the complement of the mask; fill with semi-colons is obtained since semi-colon is octal 77. If ITYPE=4 and DEFAULT= "/string", the contents of VAR is zero filled as PARAM or DEFAULT was then "or-ed" with the appropriately shifted and masked contents of DEFAULT.

SPECIAL CAUTION AND FEATURES

The calling subroutine should have an appropriate return location if an error is detected in TRANS. Be sure to store the name of the calling subroutine in /SUB/.

The input values of PARAM and its translation in octal are printed out using a call to MESSAGE.

TIMING

Not applicable

ERROR MESSAGES

"calling subroutine, PARAMETER param VALUE = value BAD"

where: calling subroutine is the name of the calling subroutine provided it has been put into /SUB/.
param is the character string value of PARAM
value is the octal value of VAR

This message is printed if a translation can not be made as previously explained.

"calling subroutine, PARAMETER param VARIABLE var + DEFAULT default TOO LONG"

where: var is the character string value of VAR (used as input)
default is the character string value of DEFAULT

This message is printed if the result of concatenating VAR with DEFAULT (when ITYPE=4 and DEFAULT starts with "/") is longer than 7 characters.

SUBROUTINES

(Subroutines marked with * are considered to be special purpose and considered to be part of TRANS. Others are normally obtained from a user library.)

FILEN*	(replaces right hand blanks in a word with zeroes or semi-colons)
INT*	(decodes integer or floating point strings)
MESSAGE	(writes informative error messages)

ACCURACY

Not applicable

COMMENTS ON USAGE

TRANS is used primarily by subroutines which receive their controlling parameters as character strings.

EXAMPLE OF USE

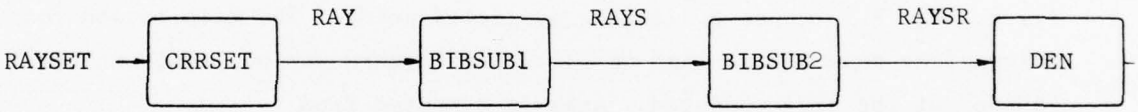
```
      SUBROUTINE BIBSUB1
      COMMON/SUB/SUB,IFATAL
      COMMON/PARAM/SFN1,SFN2,NKEY,PARAM(1)
      SUB="BIBSUB1"
      CALL TRANS(SFN1,SFN11,4,3LFIL),RETURNS(1000)
      SFN22=SFN1
      CALL TRANS(SFN2,SFN22,4,"/S"),RETURNS(1000)
      CALL TRANS(NKEY,NK,1,1),RETURNS(1000)
           .
           .
           .
           .
1000    RETURN
           .
           .
           .
```

NAME: DEN, revision 0, subroutine, PML 161
CATEGORY: Special purpose (ray-trace analysis)
TITLE: Ray density calculations and flux tube generation
LANGUAGE: CDC Extended Fortran
PROGRAMMER: T.B. Barrett, Parke Mathematical Laboratories, Inc.
DATE: depends on version (1976)

DESCRIPTION

Subroutine DEN is used to output "flux tubes" and to calculate "ray density" in the form of areas subtended by the landing points of neighboring rays. Briefly it works as follows:

The input to DEN is the bundle of rays (file RAY from CRRSET, see PML 157) properly sorted by frequency, azimuth and elevation. In addition this bundle must have been made word addressable. Thus typically we have the sequence:



For each frequency DEN performs flux tube calculations on 4 neighboring rays where neighboring rays are defined as having neighboring take off directions. More specifically consider two adjacent azimuth "planes" labeled by α_1 and $\alpha_2 = \alpha_1 + \Delta\alpha$ respectively; within plane 1 are rays at elevation angles $\beta_{11}, \beta_{12}, \cdots \beta_{1M_1}$ and in plane 2 $\beta_{21}, \beta_{22}, \cdots, \beta_{2M_2}$. The first tube is defined by the rays $R_1 = (\alpha_2, \beta_{22})$, $R_2 = (\alpha_2, \beta_{21})$, $R_3 = (\alpha_1, \beta_{11})$, $R_4 = (\alpha_1, \beta_{12})$; the next tube by $R_1 = (\alpha_2, \beta_{23})$, $R_2 = (\alpha_2, \beta_{22})$, $R_3 = (\alpha_1, \beta_{12})$, $R_4 = (\alpha_1, \beta_{13})$; etc. until there are no more rays for this frequency.

Several things should be noted about this scheme of selecting rays to define tubes:

1) There should be two or more rays in adjacent azimuth planes.

PRECEDING PAGE BLANK-NOT FILMED

2) The increment in elevation angles for adjacent azimuth planes need not be the same nor do they need to start at the same elevation angle.

3) Tubes are always started at the bottom (lowest elevation angles) and work up. If the number of rays in adjacent azimuth planes are unequal, the highest excess rays will not take part in flux tube calculations.

Having found four neighboring rays, DEN then checks to see if these rays all land at the first landing point or landing region e.g. "90 deg." or "rcvr" (specified in ARRAY -see CRRSET); if so it proceeds to calculate landing point area, etc. If one or more rays do not land, then the number of these rays (numbered as shown above) are stored. Landing region area is calculated using plane geometry (even though as a rule 4 points do not define a plane). One of the points, P_4 (that associated with ray 4 (R_4)), is projected into the plane defined by points P_1, P_2, P_3 associated with R_1, R_2, R_3 so that the area computed is that defined by P_1, P_2, P_3, P_4' where P_4' is the projected point. The main reason for doing plane geometry is that it is easier to make various tests on the "shape" of the area computed. Area is computed from

$$A = \frac{1}{2} |\sin\gamma| \, d_{13} \cdot d_{24}$$

where γ = angle between diagonals d_{13}, d_{24} where d_{13} is the line P_1P_3 etc.

In the current version of DEN an area is computed only if the area P_1, P_2, P_3, P_4' satisfies the following "good area" criteria:

1) The points P_2, P_3 must be in the same half-plane bordered by line P_1P_4'.

2) The intersection of the diagonals d_{13}, d_{24} must be in the same half-plane (see 1) as P_2 and P_3.

3) Let P be the point of intersection of the two diagonals. Then $< P_4P_1P_2$ must be greater than $< P_4P_1P$.

4) The length of d_{13} must be greater than d_{1P} (distance between point P_1 and point of intersection of the diagonals.

Typically P_1P_4 means the line through points P_1 and P_4; $< P_4P_1P_2$ means the angle between the lines P_1P_4 and P_1P_2 going counter-clockwise from P_1P_4 to P_1P_2.

These criteria essentially force "good" areas to be convex quadrilaterals with the points in the same or reverse order of the outgoing rays.

If a good area can not be found, a value of -1 is returned from the subroutine AREA1 which calculates the landing point area. In addition a "skewness" code is returned indicating where the points failed to meet one of the above criteria. In an attempt to summarize the shape of the areas defined by the flux tubes some other quantities are computed and returned by AREA1. These are:

1) area "polarity" (+1 or -1). As noted above P_1P_4 (here we are thinking of a directed line in the direction P_1 to P_4) divides the plane into "positive" and "negative" half planes where the positive half plane is to the left of line P_1P_4 looking in the direction P_1 to P_4. If P_2 and P_3 fall in the positive region then the area is positive, etc.

2) $\cos\psi$ where ψ is the angle between the current diagonal d_{13} and a reference diagonal. The purpose of this parameter is to indicate how rapidly the area changes its orientation as defined by the reference diagonal. The reference diagonal is chosen as follows:

   a) if the azimuth of R_1 of the current flux tube is the same as that of the preceeding flux tube, then the reference diagonal is d_{13} of the preceeding flux tube.

   b) if the azimuth of R_1 of the current flux tube is different from that of the preceeding flux tube, then the reference diagonal is d_{13} of the flux tube with lowest elevation angle and preceeding azimuth.

   c) at the start where there is no preceeding flux tube, use the current diagonal d_{13}.
   (the azimuth and elevation angle of R_1 of the reference diagonal is included in the information printed by DEN.)

3) the sine of the angle between the two diagonals of the area.

4) length of the two diagonals (this and 3 essentially define the shape of the area).

5) the distance which P_4 is moved into the plane P_1, P_2, P_3 to obtain P_4'.

The printed output of DEN includes the following information for each landing region and each flux tube:

1) the ray number as it pertains to the flux tube (1, 2, 3 or 4 for R_1, R_2, R_3, R_4 - see above) and azimuth and elevation of each ray in the flux tube.

2) the mean azimuth and elevation of the flux tube (essentially the direction of the flux tube).

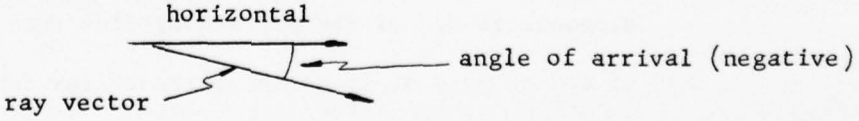
3) if an area can not be defined because of missing rays, the numbers of such rays (e.g. 1, 2 if R_1 and R_2 do not land).

4) if the points P_1, P_2, P_3, P_4 do not satisfy the "good area" criteria then a "skewness code" corresponding to one of the above conditions e.g. SKEW1 means that condition 1 was not satisfied, etc. (The points can fail to meet more than one criteria. Only the first one not met in the order given above is given.)

5) for good areas, the following data:

mean group path length (km)

mean angle of arrival (degrees from horizontal)



projected area (km^2) (projected perpendicular to arrival vector)

relative projected area = projected area divided by the 1 km area (area at 1 km from transmitter)

area polarity - (see above)

cosine of the angle between the diagonal d_{13} and the reference diagonal (see above)

azimuth and elevation angle (degrees) of P_1 of the reference diagonal

relative sine of the angle between area diagonals = sine of angle between d_{13}, d_{24} divided by the sine of the angle between the diagonals of the 1 km area.

relative ratio of the diagonals, i.e. d_{13}/d_{24} divided by d'_{13}/d'_{24} where prime refers to the 1 km area.

In addition to the printed output which describes the landing regions, DEN produces a "flux-tube" file referred to as FLUXT. This file is similar to file RAY produced by CRRSET except that instead of each record representing a ray at its landing point, each record represents a flux tube at its landing region.

USAGE

CALL DEN(IN,OUT,NREC,M,NEL,STORE1,STORE2)

Input parameters

For explanation of permissable values and role of parameters see the notes indicated by asterisks. There are no default values.

IN	input file name (left justified, zero fill)	*1
OUT	output file name (left justified, zero fill)	*2
NREC	number of input data records (integer)	*3
M	input record length (words, integer)	*4
NEL	maximum number of elevation angles per azimuth (integer)	*5
STORE1 STORE2	scratch storage arrays which must be declared in the calling subroutine	*6

Notes on input parameters

*1 This is the logical file name of the input file, 7 characters or less. It is typically a "random access" version of "RAY" produced by CRRSET or of a sorted "RAY" - see diagram in DESCRIPTION. For a detailed description of the record structure of the file see PML 157.

*2 This is the sequential output file containing flux tubes. See FILE DESCRIPTIONS for details of the record structure, etc.

*3 The input file is assumed to contain a "header" (first) record with the structure shown in PML 157. NREC should not include this header.

*4 Record length is included in the header record of the input file but is also required here in order to define the structure of the arrays STORE1, STORE2 which are declared to have dimensions (M,NEL) in DEN.

*5 See note *4

*6 STORE1 and STORE2 must be declared in the calling subroutine to have a size at least as large as M x NEL.

use of common storage

The following labeled common storage areas should be declared in the calling program or in some subroutine which is loaded before DEN:

/FITAIN/(35)
/FITAOUT/(35)
/INBUF/(514)
/OUTBUF/(514)
/IBUF/(maximum of input/output record length, usually 100 is sufficient)

The following labeled common areas are used by DEN for communications with other subroutines:

/SUB/SUB,ICODE
/DEN/(26) - communicates with AREA1 (see appendix A)

miscellaneous

1) Record Manager calls are used for manipulating the main input and output files, IN and OUT, thus normally they should not be declared in the main program. Both files are rewound at subroutine

start and end. The input file must not be empty since DEN is not "protected" for empty input files. If NREC is less than 3, however, DEN will make a normal return having produced no output file.

2) DEN tries to compute "ray densities" and flux tubes for the entire input file. It processes the input file in equi-frequency batches. For a given frequency a flux tube is defined as long as there exists one or more "landing points" on 4 adjacent rays where adjacent rays are defined by 2 rays each in two adjacent azimuthal "planes" e.g.

(az,el) = (25.5,1.), (25.5,2.), (26,1.5), (26,2.5)

Note that elevation angles do not have to correspond. In the present version of DEN azimuthal separation is irrelevent but it is an easy modification to define flux tubes only if $/\Delta az/ \leq C$ where C is a number which can be passed to DEN.

STORAGE REQUIRED

Storage required for DEN and its associated "special"subroutines is 3170_8 (1656_{10}). This figure does not include space required by the arrays STORE1 and STORE2.

ALGORITHM

The random access input file is first searched for a change in ray frequency while storing the record numbers of azimuth changes. Using the stored information on azimuths, records for $azimuth_i$ are read into STORE1 then records for $azimuth_{i+1}$ are read into STORE2. A search is then made through STORE1 and STORE2 for flux tubes using the scheme outlined in DESCRIPTION. Landing point areas are computed for all landing points along a ray for which they can be "well defined" as outlined in DESCRIPTION. The area computation is done within subroutine AREA1 which is briefly described in Appendix A. AREA1 is also used to calculate the flux tube cross-sectional area at 1 km from the transmitter. In addition to computing flux tube landing point areas, DEN

is used to compute various other quantities which are associated with each flux tube. These quantities are stored on the output "flux tube" file which is described later under FILE DESCRIPTIONS. The number of such quantities is somewhat variable so their computation is not described here but is briefly mentioned in the later section. As mentioned previously, an area may be "undefined" either because the 4 rays do not all land or if they do land because the points do not have a particular pattern. The former case is defined by setting R=Ø (see PML 157, page 4) in the various P_i which describe each potential landing point. This is reflected in the search through STORE1 and STORE2 for rays which have landed.

SPECIAL CAUTION AND FEATURES

1) DEN, of course, depends upon having a particular input record structure which is at present defined by CRRSET, in particular for locating data which is to be used in computing various quantities for the output file of "flux tubes". From time to time it may be necessary to change the structure of the input file, thus DEN should be modified accordingly.

2) The number of values, M2, used to describe each "point" in the output flux tube file may be changed by suitably modifying DEN. This variable M2 is assigned a value in a data statement in DEN and should be changed if necessary.

3) DEN is limited to a maximum of 9 landing points per ray by some dimension statements.

4) If the number of landing points exceeds 4 the output file "PRINT" will only contain information about the first 4 because of page size limitations.

5) In addition to printing information about landing point areas, DEN prints (on OUTPUT) the min. and max. frequencies, min. and max. starting elevation and azimuthal angles (mean values) over all flux tubes stored on file OUT.

TIMING

No accurate timing figures are available. However it es estimated

that for two landing points approximately .015 seconds per flux tube is required.

ERROR MESSAGES

"NO FLUX TUBES CAN BE DEFINED"

If for any frequency, less than two initial azimuths are found, this message is printed. Note that this message may occur several times for a given input file which consists of several "frequency sets".

"DEN - NO. RECS TOO SMALL, n"

Occurs if there are less than 4 input records, in which case it is impossible to generate flux tubes. The value of n is the number of input records (NREC).

SUBROUTINES

(Subroutines marked with * are considered to be special purpose and part of DEN. Others are normally obtained from a user "universal" library.)

AREA1*	(computes landing point areas - see Appendix A)
COORD*	(converts from spherical to cartesian coordinates)
DEV*	(computes azimuthal deviation)
ERROR1	(error handling for input file)
ERROR2	(error handling for output file)

ACCURACY

There are several classes of errors to consider when DEN is used. The first is strictly computational errors due to finite length computer words, etc. This source of error can be neglected compared to the others. The second source of errors to consider is that due to the use of plane geometry in place of spherical geometry at earth landing points. Plane geometry was chosen mainly because it simplifies considerably the classification of the landing point geometry as discussed under DESCRIPTIONS. Typically this simplification introduces an error of less than 1% which is not considered excessive. The third source of error lies

outside of the domain of DEN in that it has to do with the fact that the flux tube areas are used for estimating back scatter power calculations and/or as a measure of ray focusing or defocusing. A discussion of this properly belongs elsewhere.

COMMENTS ON USAGE

DEN is a special purpose program which is designed to work with ray records as they are typically produced by CRRSET. It can be used with any input file which preserves the input file structure as given in PML 157. The main requirement for the definition of a "flux tube" as seen by DEN, is that neighboring rays must lie on vertical azimuthal planes i.e. 4 rays must be related by (α, β_1), (α, β_2), $(\alpha + \Delta\alpha, \beta_3)$, $(\alpha + \Delta\alpha, \beta_4)$ where (α, β) means (initial take off azimuth, initial take off elevation). Note that $\beta_1, \beta_2, \beta_3, \beta_4$ can be all different; but $\beta_1 \neq \beta_2$ and $\beta_3 \neq \beta_4$. The angular differences $|\Delta\alpha|$ and $|\Delta\beta|$ are immaterial to DEN but common sense and experience puts limits on their magnitudes for sensible results.

FILE DESCRIPTIONS

There are four files used by DEN. They are briefly described below:

IN — fixed length, word-addressable, w-type records. The record structure of this file is given in PML 157. Note however that CRRSET produces a sequential file; thus typically IN is produced by running the output of CRRSET through BIBSUB1. The file is rewound at the start and end of DEN. Record Manager calls are used in manipulating this file, thus it is not necessary to declare it in a main program.

PRINT — formatted output file. It must be declared in a main program. This file contains most of the printed (or printable) output of DEN. As noted previously if NPTS is greater than 4, only information on the first 4 points is printed.

AD-A038 869 PARKE MATHEMATICAL LABS INC CARLISLE MASS F/G 20/14
ANALYSIS AND SYNTHESIS OF MODEL IONOGRAMS USING 3D RAY TRACKING--ETC(U)
FEB 77 B LANGWORTHY, T BARRETT, D BANDES F19628-76-C-0029
UNCLASSIFIED RADC-TR-77-60 NL

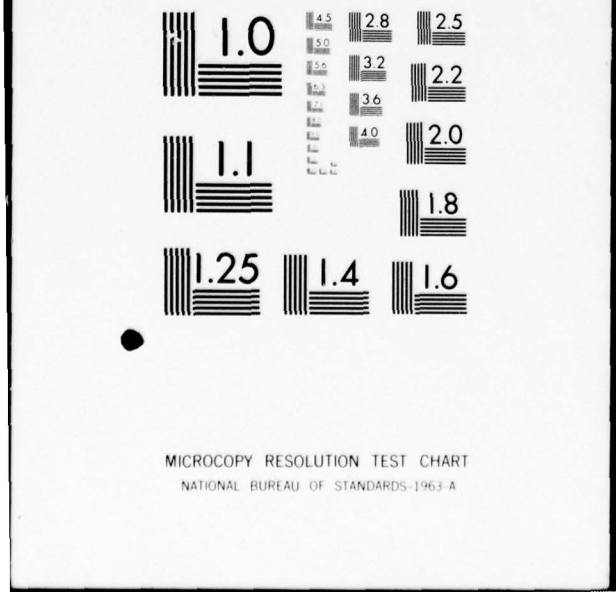
4 OF 4
AD
AD38869



END
DATE FILMED
5-77

OF 4

038869



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

OUTPUT usual output file which should be declared in the main program. It is used for error and informative messages only.

OUT fixed length, sequential, w-type records. It may be read using Fortran standard unformatted READ statements. File manipulation is done within DEN using Record Manager calls. The file is opened and closed within DEN with rewind. The record structure is as follows:

header record

-999. -999., -999., npts, m, nint, ID(4), r, θ, ϕ, (HOP,WHY,i = 1, npts), (title(i), i = 1, nint)

where:

- npts = number of landing points [integer]
- m = total record length [integer]
- nint = number of integrated quantities [integer]
- ID(4) = last 4 words from ID(10) in header record of RAYSET [Hollerith]
- HOP,WHY = landing point name (hop number and description code) [integer, Hollerith]
- title = titles of "integrated quantites" [Hollerith]

flux tube records (all quantities are floating point unless specified otherwise. Parentheses around a symbol indicates that it is a mean quantity, i.e. the average value of 4 rays defining a flux tube.)

f, (a), (e), S, P_1, P_2, $\cdots$ P_{npts}

P_i = HOP,WHY, (r_r), (r_θ), (r_ϕ), A, (ψ), (t), σ_t, (Δa), (α), (l)

where:

- f = transmitter frequency (Mhz)
- a = initial azimuth (degree)
- e = initial elevation (degree)
- S = cross sectional area at 1 km (km^2)

- HOP = hop number [integer]
- WHY = landing point name code [Hollerith]
- A = landing point area (km^2). If the landing point is not "defined" for one of the reasons mentioned in DESCRIPTION, A is set to -1.
- ψ = angle of arrival (rad) (see DESCRIPTION)
- t = group path length (km)
- σ_t = mean square deviation of group path length (see note)
- Δa = azimuthal deviation (deg.) (see note)
- α = absorption (db) or zero if absorption is not calculated during ray tracing
- l = geometric path length (km) or zero if it is not calculated during ray tracing

Note

σ_t is calculated using the expression:

$$\sigma_t = \left(\frac{\sum_{i=1}^{4} t_i^2 - 4\bar{t}^2}{3} \right)^{1/2}$$

where $\bar{t} = 1/4 \sum_{i=1}^{4} t_i$

and t_i is the group path length for ray -i.

Δa is calculated using the expression:

$$A = \cos^{-1} (\cos\theta_t \cos\theta + \sin\theta_t \sin\theta \cdot \cos(\phi - \phi_t))$$

if $A = 0$, then $\Delta a = 0$.

if $\sin\theta_t = 0$, then $\Delta a = \phi - a$

otherwise

$$D = \text{sign}\left[\frac{\cos^{-1}(\cos\theta - \cos\theta_t \cdot \cos A)}{\sin\theta_t \sin A}, \sin(\phi - \phi_t) \right]$$

and $\Delta a = D - a$; where:

θ_t, ϕ_t are transmitter spherical coords.

θ, ϕ are landing point spherical coords.

a is the initial azimuth.

trailer record

999., 999., 999., f_{min}, f_{max}, $(a)_{min}$, $(a)_{max}$, $(e)_{min}$, $(e)_{max}$, $(t)_{min}$, $(t)_{max}$

where min. and max. are over all the flux tubes produced by DEN.

EXAMPLE OF USE

Normally DEN is called after CRRSET has been called to produce a sequential file which can be transformed to word-addressable through BIBSUB2 to produce an input suitable for DEN. The following example shows the additional statements which must be added to the example shown in PML 157 in order to run DEN:

```
COMMON/FITAIN/FITAIN(35)
COMMON/INBUF/INBUF(35)
COMMON/PARAM/PARAM(2)
DIMENSION STORE1(1600), STORE2(1600)
DATA (PARAM="RAYS","RAYSR")
CALL CRRSET("RAYSET","RAYS", ····
CALL BIBSUB2
CALL DEN(5LRAYSR,5LFLUXT,NREC,M,100,STORE1,STORE2)
    .
    .
    .
```

Note that values of NREC and M (the number of input data records and their length) is obtained as output of CRRSET. If CRRSET is not run in the same main program as DEN these values must be given. NREC can be given a value less than the total number of data records output by CRRSET if less flux tubes are desired.

APPENDIX A

Subroutine AREA1

DESCRIPTION

Given four points (x_i, y_i, z_i) i = 1,2,3,4 in a cartesian system, this subroutine projects (x_4, y_4, z_4) perpendicularly on to the plane 1 2 3 to obtain (x'_4, y'_4). The plane quadrilateral "defined" by 1 2 3 4' (assuming it is defined - see below) is computed and returned. The following criteria must be met by the points 1,2, 3,4' in order to consider that the plane area is well defined. (Assume for convenience that point 1 lies at the origin of a coordinate system of the plane and that 4' lies on the x- axis - from here on 4' is referred to as 4):

1) points 2 and 3 must lie in the same half plane defined by the x - axis
2) "diagonals" $\overline{1\ 3}$ and $\overline{2\ 4}$ must intersect in the same half plane as points 2 and 3 lie
3) $\angle$ 214 must be greater in magnitude than $\angle$ i14 where i is the intersection of the diagonals
4) length $(\overline{13})$ must be greater than length $(\overline{1i})$

If these conditions are not met the value of area returned is -1 and a "skewness" code corresponding to the condition 1), 2), 3) or 4) above which was not satisfied is returned. Note that it may happen that more than one of these conditions may fail to be satisfied. Only the first condition which fails to be met is returned. In addition to the area, other quantities as listed below are returned.

USAGE

CALL SUBROUTINE AREA1

All input and output is through labeled common area /DEN/. Additional useful information is contained in /COORD1/. (all values are floating point except MESSAGE which is Hollerith)

COMMON /DEN/ x(3,6), AREA, GAMMA, RL1, RL2, COSREF, P, POLE, MESSAGE

X(3,1-4) contains the Cartesian coordinates of the four points (x_i, y_i, z_i), i = 1,4 (here only 4 refers to point 4 before projection)

X(3,5-6) define a reference line $\overline{5\ 6}$ (see below)

AREA = quadrilateral area 1 2 3 4 or -1 if the points fail to meet conditions 1), 2), 3), 4)

GAMMA = angle ($\leq \frac{\pi}{2}$) between the diagonals $\overline{1\ 3}$ and $\overline{2\ 4}$

RL1 = length ($\overline{1\ 3}$)

RL2 = length ($\overline{1\ 4}$)

COSREF = cosine of the angle between diagonal $\overline{1\ 3}$ and reference line $\overline{5\ 6}$. (Note that $\overline{5\ 6}$ does not necessarily lie in the plane 1 2 3 4)

P = distance (positive number) which original point 4 must be moved to lie in the plane 1 2 3

POLE = $\pm$ 1 depending on whether the points 2 3 are above (+) or below (-) the x - axis defined by directed line 1 4. Note that during calculations, 2 and 3 may be shifted so that they always lie in the positive half plane.

MESSAGE = a code word indicating which condition was <u>first</u> not met. For example if condition 2) is the first one a value of "SKEW1" is returned.

COMMON/COORD1/ XP1, YP1, ZP1, ····, XP4, YP4, ZP4, XI, YI

This common storage area contains the coordinates of points 1, 2, 3, 4 such that point 1 lies at the origin (i.e. XP1, YP1, ZP1 should all be zero), point 4 lies on the positive x axis and points 2 and 3 are in the positive half plane. XI, YI is the point of intersection of the diagonals $\overline{1\ 3}$, $\overline{2\ 4}$ in this reference frame. Note that ZP1, ZP2, ZP3 and ZP4 should all be zero. They are stored here merely as a check on AREA1.

APPENDIX B

Subroutine DEN1

DESCRIPTION

Subroutine DEN1 is a mutation of DEN such that

1) The flux tube quantities are not averaged e.g. group path length is not averaged over the 4 rays which define a given flux tube but rather is associated with the lower-left (least azimuth, least elevation) ray of the quartet.

2) The angle between the wave normal and earth's magnetic field is stored in the flux tube output. This value is stored after 1, the geometric path length i.e. it is the last item in each P_i as described in FILE DESCRIPTIONS for DEN.

3) Landing point "PAST 90." is recognized as special in that it always occurs immediately after "90 DEG." or after another "PAST 90.". Values of area for the preceeding "90 DEG." point are associated with the current "PAST 90." point.

4) A print out of "ray density" is not given.

USAGE

Same as for DEN except CALL DEN1(····).

NAME: DEN1, revision 1, subroutine, PML 161

CATEGORY: Special purpose (ray-trace analysis)

TITLE: Ray density calculations and flux tube generation

LANGUAGE: CDC Extended Fortran

PROGRAMMER: T.B. Barrett, Parke Mathematical Laboratories, Inc.

DATE: December 14, 1976

DESCRIPTION (of the revision)

DEN1 is a modification of DEN (DEN remains the way it is described in revision 0) which does not compute mean quantities for flux tubes. It uses the ray at the lower left corner (lowest elevation angle, left azimuth) of each flux tube to define quantities such as group path length, landing point deviation, etc. In addition DEN1 treats the landing points "PAST 90." and "APOGEE" in a special way (see ALGORITHM). The output file OUT has been modified to include the following quantities (see FILE DESCRIPTIONS):

ψ_g = angle between wave normal and earth's magnetic field (degrees)

r_a = radial coordinate (km) at apogee (stored for "RCVR" only)

code = polarity, ratio of ratio of diagonals and γ, the change in the angle between the diagonals from take-off to landing point. (Hollerith)

g_r = ground range (km) from transmitter to landing point (or landing point projected onto the earth).

The print out of ray "density" information has been eliminated.

ALGORITHM (changes)

Basically DEN1 operates in the same manner as DEN. As mentioned, mean quantities are not computed. If the landing point "APOGEE" is encountered, the value of r_a is stored and put into the appropriate location for association with the ground backscatter point "RCVR". Thus no flux tube quantities are computed for "APOGEE". The landing point "PAST 90." is treated in a special manner in that landing point area is not computed.

However group path length and other quantities are computed for this point (only if the preceeding 90 DEG. point had a flux tube associated with it).

FILE DESCRIPTIONS (changes)

PRINT is used only to print information on rays which do not form a flux tube

OUT

header record

remains the same

flux tube records (all quantities are floating point unless specified otherwise)

f a e S P_1, ... P_{npts} (P_i = HOP, ..., g_r)

- f = transmitter frequency (Mhz)
- a = initial azimuth (degree)
- e = initial elevation (degree)
- S = cross sectional area at 1 km (km^2)
- HOP = hop number [integer]
- WHY = landing point name code [Hollerith]
- A = landing point area (km^2). If the landing point is not"defined" for one of the reasons mentioned in DESCRIPTION, A is set to -1.
- ψ = angle of arrival (rad) (see DESCRIPTION)
- t = group path length (km)
- σ_t = mean square deviation of group path length (see note)
- Δa = azimuthal deviation (deg.) (see note)
- α = absorption (db) or zero if absorption is not calculated during ray tracing
- l = geometric path length (km) or zero if it is not calculated during ray tracing
- ψ_g = angle of arrival between wave normal and earth's magnetic field (degrees)
- r_a = radial coordinate of "apogee" - stored with P_i with WHY = "RCVR" only. For other landing points r_a is meaningless.

- code = code describing the landing point area [Hollerith]
  This code is in the form $\pm$ aaaa bbbbb where $\pm$ is the area polarity, aaaa is the ratio of ratio of area diagonals and bbbbb (a signed quantity) is the change in γ from take off to landing point where γ is the angle between the diagonals.
- g_r = is the ground range (km) between the transmitter and landing point (projected onto the earth).

NAME: FLUXPR(or FLUXPR1), revision 0, subroutine, PML 162
CATEGORY: Special purpose display
TITLE: Flux tube report generator
LANGUAGE: CDC Extended Fortran
PROGRAMMER: T. B. Barrett, Parke Mathematical Laboratories, Inc.
DATE: September 1976

DESCRIPTION

Subroutine FLUXPR (or FLUXPR1) is basically a device for inputting data to subroutine BACSCA1 (see PML 163) and for outputting information generated by this subroutine. The two versions FLUXPR and FLUXPR1 differ essentially in the amount of information to be printed. FLUXPR also serves as a good example of a subroutine which calls the general purpose print program PRINT3.

USAGE

SUBROUTINE FLUXPR (IN, IPRNT)
or FLUXPR1

Input parameters

IN - file name (left justified, zero fill) of the file which contains data for use by BACSCA1.

IPRNT - file name (left justified, zero fill) of the file to receive the printer output.

note: See PML 163 for the assumed structure of the input file. The output file contains formatted records which are suitable for printing.

Labeled common input

/SYSTEM/ DUM(20), PW, PØ

PW is the transmitted pulse "width" in kilometers

PØ is the total transmitted power. Although this value is passed to BACSCA1 it is not used in any caluclations.

/QUANT1/ (20) in FLUXPR
(30) in FLUXPR1

is the storage location for returning information from

BACSCAl which is to be printed through FLUXPR. These quantities are given the following titles (listed below in the order in which they are stored; first for FLUXPR then for FLUXPR1):

FLUXPR

"frequency (mhz)"
"azimuth mean compute (degrees)"
"elevation mean (degrees)"
"flux tube x-section (km..2)"
"wave length gain (db km..2)"
"transmit antenna gain (db)"
"receive antenna gain (db)"
"distance (R-4) gain (db km-4)"
"radar x-section (db km..2)"
"pulse spread gain (db)"
"absorption (2-way) gain (db)"
"reflection gain multi-hop (db)"
"total path gain (db)"
"ray-focus gain (db)"
"arrival angle (degrees)"
"group path length (mean) (km)"
"group path length spread (km)"
"azimuthal deviation (degrees)"

FLUXPR1

"backscatter point"
"frequency (mhz)"
"azimuth compute (degrees)"
"elevation (degrees)"
"flux tube x-section (sterrad)"
"total path gain (db)" (note 1)
"wave length gain (db km..2)"
"transmit antenna gain (db)"
"receive antenna gain (db)"
"distance (R-4) gain (db km-4)"
"radar x-section (db km2)"

"pulse spread gain (db)"
"absorption (2-way) gain (db)"
"reflection gain multi-hop (db)"
"total path gain (db)" (note 2)
"ray focus gain (db)"
"group path length (km)"
"group path length spread (km)"
"ground range (km)"
"azimuthal deviation (degrees)"
"apogee or height (km)" (note 3)
"arrival angle (degrees)"
"accurate geomag. colatitude (degrees)"
"polarity ratio of diag.-rat. del-gamma" (note 4)

note 1: Although it is not indicated explicitly on the report, data for the "past 90." landing point is included in the report. Most entries in the report for this point are empty (indicated by R). The value for "total path gain" is actually the incremental loss due to the fact that the ray path and earth's field are slightly off perpendicularity.

note 2: Total path gain is repeated so that it will appear on both pages of the flux tube report.

note 3: For the landing point "90 DEG.", the height of the point is given. For landing point "RCVR", the apogee of the ray is given.

note 4: The "polarity" of the landing point area is discussed in PML 161. The "ratio of the ratio of the diagonals" is a measure of the change in shape of the flux tube. The diagonals referred to are at 1 km from the transmitter and at the landing point. "Del-gamma" is the change in angle between the cross section diagonals.

Use of other common storage

The following common storage areas must be declared with the minimum sizes as indicated in another subroutine (usually the main program) loaded prior to FLUXPR.

/FITAIN/(35) input file FIT area
/INBUF/(514) input file buffer area
/IBUF/ (input file record size)
/FLUXPR/(1) used by FLUXPR1 only to communicate with BACSCA1.

STORAGE REQUIRED

The storage sizes required for FLUXPR and FLUXPR1, not including common storage and "system" subroutines, are 546_8 (358_{10}) and 631_8 (409_{10}) respectively.

ALGORITHM

Both FLUXPR and FLUXPR1 read one record at a time from IN into /IBUF/, pass the location of information (pertaining to the desired landing point) to BACSCA1 and then call PRINT2 or PRINT3 (basically the same - see PRINT3, PML 165) to print out information. There is a fundamental difference between FLUXPR and FLUXPR1. FLUXPR is designed to work with any file of the type "FLUXT" as described in PML 161. It will display the entire contents of the file. FLUXPR1 on the other hand is designed to display data ("explicitly") for the (1 hop) "90 DEG." and "RCVR" landing points only. When this FLUXT file is processed through FLUXPR1 and BACSCA1, the landing points "APOGEE" and "PAST 90." are treated in a special way by BACSCA1 so that height information and backscatter losses due to slightly off perpendicularity conditions can be added to the appropriate section of the flux tube report.

SPECIAL CAUTION AND FEATURES

FLUXPR prints out data on the number of quantities which have been integrated. (FLUXPR1 does not.) Both FLUXPR and FLUXPR1 print out all the data for a given landing point before going to the next landing point. Special note should be made of the requirement that height information in the form of apogee or landing point height must be available for FLUXPR1.

TIMING

Timing depends on the size of the input file. Typically only a few seconds are required to process even large files.

ERROR MESSAGES

"FLUXPR, NO FLUX TUBES AVAILABLE" if the input file IN is empty.

SUBROUTINES

Aside from the file error processing subroutines EOFILE and ERROR1, FLUXPR requires BACSCA1 (PML 163) and PRINT2 or PRINT3 (PML 165).

ACCURACY

Not applicable

COMMENTS ON USAGE

None

FILE DESCRIPTIONS

IN - see the description of file OUT in PML 161 revision 1.

PRINT - formatted output file for printing. This file is not rewound on entry or exit to FLUXPR.

EXAMPLE OF USE

```
PROGRAM MAIN (OUTPUT, PRINT)
COMMON/FITAIN/FITAIN(35)
COMMON/INBUF/INBUF(514)
COMMON/IBUF/IBUF(100)
COMMON/SYSTEM/ANTENNA(10), RECEIVE(10), TRANS(10)
DATA(TRANS=50., 100000.)
CALL FLUXPR(5LFLUXT,5LPRINT)
END
```

This program assumes that the appropriate flux tube file has been created and attached as local file FLUXT. The print file, PRINT, should be rewound and copied to OUTPUT after MAIN has run.

NAME: BACSCA1, revision 0, subroutine, PML 163
CATEGORY: Special purpose
TITLE: Backscattered power calculations
LANGUAGE: CDC Extended Fortran
PROGRAMMER: T.B. Barrett, Parke Mathematical Laboratories, Inc.
DATE: October 1, 1976

DESCRIPTION

BACSCA1 computes the incremental power from a single flux tube backscattered at a single "landing point" which is capable of scattering power back to the transmit-receive location. Two such "points" are recognized by BACSCA1: "90 DEG." and "RCVR". The first is the code name for that point along a ray (or flux tube) where the earth's magnetic field and wave normal (also ray direction) are perpendicular. The second is the code name for the point where a ray intersects and is reflected from the earth's surface. Both of these points are assumed to scatter power back along the same paths; the "90 DEG." point because of field aligned electron density enhancements and the "RCVR" point because of irregularities in the earth's surface which is a partially reflecting surface.

There are several factors in the expression for backscatter power (see ALGORITHM). These factors are computed separately and also returned by BACSCA1 so they can be used to analyze flux tube backscatter power losses. At the same time a few other quantities (see below) are returned for use in a "flux tube report" - see FLUXPR, PML 162.

USAGE

FUNCTION BACSCA1(IS,PO,PW,R1,THETA1,PHI1)

input parameters

All input parameters are floating point except IS which is integer. Units, if they matter, are shown in parentheses. Refer to the notes indicated by asterisks for further details if necessary.

IS pointer to the location in the flux tube record at which landing point data should be obtained *1

PO transmitter power

PW transmitted pulse "width" (km)

(R1, THETA1, PHI1) coordinates of the transmitter in the compute coordinate system (km, rad., rad.).

In the current version of BACSCA1, these quantities are not used for anything and need not be given values.

Notes

*1 Refer to PML 161 for the structure of the flux tube file as output by subroutine DEN. Each record (except for header and trailer records) is of the form:

$$f,a,e,S,P_1,P_2,\cdots P_n$$

where f,a,e is the flux tube label giving frequency and ray mean starting direction (azimuth and elevation), S is the initial flux tube solid angle, and each P_i consists of landing point data starting with the landing point identifier which consists of hop number, HOP, and code name, WHY. The value of IS points to the location of HOP for a given landing point.

output

There are two "outputs" of BACSCA1; the first is the total incremental backscatter power given by BACSCA1 (a function). The second output is stored in labeled common area /QUANT1/ as follows: (all values are floating point, units - if any, are shown in parentheses, notes given explanations are indicated by asterisks. Each quantity is given a sequence number according to its relative location in /QUANT1/.

1) Transmitter frequency (Mhz)
2) Initial flux tube azimuth (degrees-relative to the compute coordinate system where 0 degrees is north and positive azimuth is clockwise)
3) Initial flux tube elevation (degrees-horizon is zero, positive elevation above horizon)
4) Initial flux tube solid angle (steradians)
5) wave length gain (db km^2) *1

6) transmit antenna gain (db) *2
7) receive antenna gain (db) *2
8) distance (R^{-4}) gain (db km^{-4}) *3
9) total radar cross section (db km^2) *4
10) pulse spread gain (db) *5
11) 2-way absorption gain (db) *6
12) multi-hop ground reflection gain (db) *7
13) total gain (sum of preceeding gains) (db)
14) ray-focus gain (db) *8
15) arrival angle at landing point (degrees) *9
16) group path length (km)
17) group path spread (km)
($=\sigma_t$ see PML 161)
18) azimuthal deviation (degrees - positive deviation clockwise)

In addition, two dummy locations, 19) and 20), are declared for future use. *Subroutines loaded prior to BACSCA1 must declare* /QUANT1/ to be at least 20, if they declare it at all.

notes

*1 see G_1 in ALGORITHM. If the frequency is less than or equal to 0., BACSCA1 returns with no computations.

*2 see G_2, G_3 in ALGORITHM. These may be set to zero if antenna patterns are not available.

*3 see G_4 in ALGORITHM. Also note that if the flux tube landing point area is not defined (signified by A = -1 - see DEN), a value of $G_4 = 0$ is returned and BACSCAT1 returns to the calling subroutine.

*4 see G_5 in ALGORITHM and Appendix A

*5 see G_6 in ALGORITHM

*6 see G_7 in ALGORITHM

*7 see G_8 in ALGORITHM

*8 see the discussion on ray focus gain in the ALGORITHM. This quantity is defined only if the geometric path length is computed in the ray trace program.

*9 The absolute value of ψ as given by DEN is used by converting to degrees

use of other common storage

Label common /IBUF/ is used as input to BACSCA1. It contains the current flux tube record for which IS is the pointer.

STORAGE REQUIRED

The amount of storage required for BACSCA1 including the current versions of SIGGND and SIG90, but not including antennapattern subroutines if required, is 323_8 (211_{10}). Additional space is required for subroutine AURLAT which is required by SIG90 (see PML 150). This figure does not include thatrequired for common storage areas /QUANT1/ and/IBUF/.

ALGORITHM

The algorithm for computing the incremental backscattered power follows from the "radar equation":

$$P_r = \frac{P_t G_t A_r \sigma}{(4\pi R^2)^2} \qquad \text{(see ref. 1)}$$

where: P_r = received power
P_t = transmitted power
G_t = transmit antenna gain
A_r = effective aperature of the receiving antenna
σ = total radar cross section (units of area)
R = "distance" to the target

The effective aperature of the receiving antenna is given by $A_r \approx \frac{\lambda^2 G_r}{4\pi}$ where G_r is the receiving antenna gain (relative to isotropic antenna) and λ is the free space wavelength of the radio wave.

The radar equation implicitly assumes that the "radar rays" travel in straight lines from the transmitter, thus the power density at the target is given by $\frac{P_t G_t}{4\pi R^2}$ where $4\pi R^2$ is the area of a sphere of radius R.

Here R is not known but can be estimated if the flux tube cross section area at the landing point is known. The relationship which follows from flux conservation arguments is that $R = \sqrt{A_p / S}$ where A_p is the projected area and S is the initial solid angle size of the flux tube. Another quantity of interest is a measure of ray focusing or defocusing. If focusing occurs there is an apparent gain in received signal strength over that which would be observed if the rays traveled in straight lines. A good measure of this gain is $40 \log_{10} (R_1/R)$ where R_1 is the geometric path length. (The factor 40 occurs because distance R appears to the inverse 4^{th} power in the power equation.) R_1 can be computed in the ray trace program but often is not. If so this "ray density gain" is returned as zero, though it is actually unknown.

In addition to the factors given by the "pure" radar equation, the effects of absorption, pulse spreading (see ref. 2), and multi-hop ground reflection can be included as multiplicative factors. In BACSCA1 the total path loss is computed as a sum of components as follows:

$$G_T = G_1 + G_2 + G_3 + G_4 + G_5 + G_6 + G_7 + G_8$$

where:

G_1 = "wave length gain" $10 \log_{10} (\lambda^2/4\pi)$ (db km^2)
where $\lambda = .3/f(Mhz)$ and f is given as part of the flux tube label.

$G_2 = G_t$ = transmit antenna gain (db). Antenna gain is computed in general by various function subroutines which are attached as necessary. In general G_t is a function of flux tube initial direction.

$G_3 = G_r$ = receive antenna gain (db). See above remarks for G_t. On occasion transmit and receive gains are lumped into one gain figure with the other set to zero.

G_4 = distance gain (db-km^{-4})
$= 10 \log_{10} \left(\frac{1}{4\pi R^2}\right)^2$ where $R^2 = \frac{A/\sin \Psi/}{S}$; S = initial flux tube cross section = $\Delta a \, \Delta e \cos e$ (steradian) nominally if the flux tube cross section is rectangular. S is actually computed in DEN for any flux tube geometry.

G_5 = total radar cross section "gain" (db-km^2)

$10 \log_{10} \sigma^o \cdot A$

where A = landing point area computed by DEN and σ^o is the cross section per unit area (no dimensions). The value of σ^o is returned by one of two function subroutines called SIG90 and SIGGND for field aligned and earth backscatter respectively. These subroutines are discussed briefly in Appendix A.

G_6 = pulse spread gain (db)

$= 10 \log_{10} \frac{pw}{2\sigma_t}$

where pw is the pulse "width" in km and σ_t is the flux tube group path length "standard deviation". This expression is essentially that used in ref. 2 except their expression for "σ_t" is somewhat different. If $pw/2\sigma_t > 1$ or if $\sigma_t = 0$, $G_6 = 0$.

G_7 = absorption (db)

This is obtained directly from the ray trace program if it has been calculated. However total path absorption is twice the value given and is multiplied by -1 since the given absorption is positive. If absorption is not calculated G_7 is set to zero.

G_8 = earth reflection loss for multi hop (db)

= 1.(2(HOP-1)) i.e. a loss of 1 db in each direction is assumed.

The value returned in BACSCAl is given by $P_r = P_0 10^{GT/10}$

The distance gain factor, G_4 is set to zero if a flux tube landing point is not defined (signified by a value of -1 when computed in DEN) so G_4 should be checked in total power calculations.

SPECIAL CAUTION AND FEATURES

BACSCAl is dependant on the data record structure of the flux tube as output by DEN. Any change in this structure should be reflected

in an appropriate change in BACSCAl.

Special note should be taken of the convention that G_4 is set to zero if and only if backscatter power can not be calculated at a particular landing point.

TIMING

Unknown. The computations in BACSCAl are relatively short and simple; little CPU time should be required.

ERROR MESSAGES

None

SUBROUTINES

BACSCAl requires 4 function subroutines called GT, GR, SIG90 and SIGGND to provide antenna gains (db) and radar cross section per unit area. These functions will vary according to various assumptions about the radar system and backscatter models. Listed below are the functions and their parameters (arguments). SIG90 and SIGGND (as they are currently structured) are briefly described in Appendix A.

GT(F,AZ,EL)
GR(F,AZ,EL)
SIG90(R,THETA,PHI)
SIGGND(PSI)

where:

F = transmit frequency (Mhz)
AZ= initial azimuthal angle (radians) w.r.t. compute coord. system (0^o = north, clockwise positive)
EL= initial elevation angle (radians)
(R,THETA,PHI)=landing point coords. (compute system) (km, radians, radians)
PSI= absolute value of the angle of arrival w.r.t. horizontal (radians)

ACCURACY

Not applicable

COMMENTS ON USAGE

None

FILE DESCRIPTIONS

A file is used only implicitly by BACSCA1 in the fact that information from a flux tube file, typically produced by DEN, is obtained through common area /IBUF/. A complete flux tube record is usually stored at this location.

EXAMPLE OF USE

FLUXPR(see PML 162) shows a typical way BACSCA1 is used.

```
      COMMON/QUANT1/Q(20)
        .
        .
        .
160   CALL GET(FITAIN,BUF)
      IF (IEOF(X).NE 0) GOTO 150
      IF (BUF(1).EQ. 999.) GOTO 150
      F = BACSCA1(JSTART,PO,PW,R1,THETA1,PHI1)
      IF (Q(8) .EQ.0) GOTO 160
      CALL PRINT2(···
      GOTO 160
        .
        .
        .
```

APPENDIX A

This appendix provides a brief description of the algorithms used for computing radar backscatter cross sections (per unit area) at field aligned irregularities and at the ground.

I. Backscatter cross section due to field aligned electron density irregularities at the point where the ray direction is perpendicular to the earth's magnetic field. (commonly known as auroral backscatter)

It is extremely difficult to obtain a reliable figure for this cross section, first because it apparently moves over a wide range and second because it is difficult if not impossible to know the geometry of the enhanced region with respect to the illuminating radar. Subroutine SIG90 uses a "mean value" of 3.5 for σ^{o}_{90} which is independant of frequency, height, etc. However this value is returned only if the 90 degree landing point falls within a specified "accurate geomagnetic" band. Furthermore there is an exponential decrease from 3.5 at the edges of this band. The value of 3.5 is obtained by converting some radar cross sections per unit volume in ref. 3 to non-dimensional radar cross section. Referring to Table 4.2 on page 108 of this reference, the average cross section per unit volume (m^{-1}) is .0035 for the "DF mode" auroral backscatter. The DF mode is the direct backscatter from field aligned irregularities in the F region. No data is available for similar backscatter from E region irregularities but there is no height dependancy in SIG90. Using an assumed range resolution of 1 km, the resulting non-dimensional backscatter cross section value of 3.5 is obtained.

As mentioned above, this value of 3.5 is returned only if the latitude of the backscatter point ($\frac{\pi}{2} - \theta$) falls within a particular accurate geomagnetic band. This latitude is converted to accurate geomagnetic latitude θ_g using subroutine AURLAT (see PML 150). If $65 \leq \theta_g \leq 72$, the value of 3.5 is returned. Outside of this region an exponential fall off is used

$$\sigma^{o}_{90} = 3.5\ e^{-/\text{edge}-\theta_g/}$$ where edge is either 65 or 72 and

/edge-θ/ ≤ 10. Beyond this SIG90 is given the value 10^{-50} meaning *essentially no backscatter.*

II. Backscatter cross section due to backscatter from the earth's surface.

Subroutine SIGGND combines some results from references 4 and 5 to obtain a frequency independant non-dimensional radar cross section per unit area. This cross section σ_g^o depends only upon the elevation angle of arrival ψ. It has a Gaussian fall off from large values of ψ (see ref. 5, page 465) then σ_g^o increases to a "knee" followed by a rapid fall off at very small values of ψ (see ref. 4, page 147). The result is the following expression:

$$q = \min\left(\frac{B}{\tan^2\psi}\ ,\ 100\right)$$

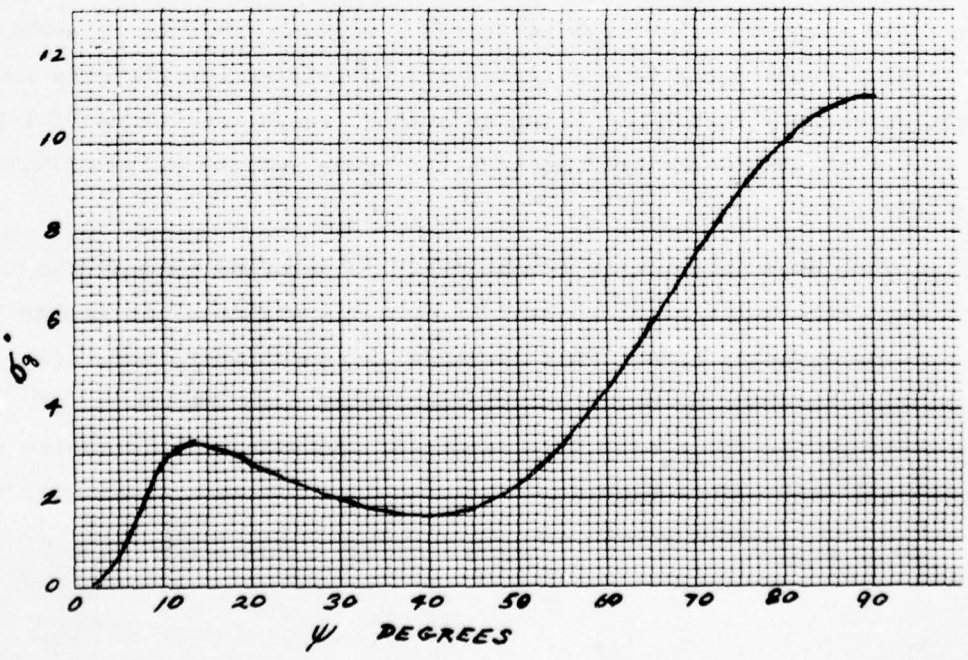
$$\sigma_g^o = \frac{\sin\psi}{\frac{A}{\psi^2} + \sin^2\psi} + C\,e^{-q} \qquad \psi > 10^{-5}$$

$$= 10^{-50} \qquad \psi \leq 10^{-5}$$

where $A = 10^{-3}$

$B = 3.35$

$C = 10$



REFERENCES

1) Skolnik, M.I., "Introduction to Radar Systems"
McGraw Hill (1962)

2) Georges, T.M., Stephenson, J.J., "HF Radar signatures of traveling ionospheric irregularities, 3D ray-tracing simulation"
Radio Science 4 , 679 (1969)

3) Katz, A.H. "HF Auroral Backscatter Study"
(Phase II) (AVCO Corporation)
RADC-TR-71-297 Final Report (1971)

4) Steele, J.G., "Backscatter of Radio Wave from the Ground"
(Stanford Electronics Lab.)
SU-SEL-65-064 (1965)

5) Beckmann, P., Spizzichino, A., "The Scattering of Electromagnetic Waves from Rough Surfaces"
Pergamon (1963)

NAME: PLTDEV; revision 0, subroutine, PML 164
CATEGORY: Special purpose display
TITLE: Azimuth deviation plot/print
LANGUAGE: CDC Extended Fortran
PROGRAMMER: T.B. Barrett, Parke Mathematical Laboratories, Inc.
DATE: October 1976

DESCRIPTION

The primary purpose of PLTDEV is to provide equi-take off azimuth curves of group path length vs. azimuth deviation for various specified landing points. More specifically, given a "bundle" of rays (from CRRSET; PML 157, for example) which have been generated or sorted by frequency, then initial-azimuth and then initial-elevation, PLTDEV produces these curves on separate coordinate frames for each frequency. For each initial azimuth, the azimuth at the given landing point is computed for all elevation angles. These azimuths and corresponding group path lengths (as computed during ray tracing) are plotted in ascending elevation angle order. After all "equi-az" curves have been plotted, "equi-el" (elevation angle) curves are plotted in dotted line i.e. points with the same elevation angle are connected in ascending initial-azimuth order.

Another option of PLTDEV is to plot only a group path length curve for each initial-azimuth, i.e. group path length is plotted in order of ascending initial elevation angle. In order to spread the points (since they otherwise fall on a line through the initial azimuth) each point is given an incremental x-value.

In addition to plots, PLTDEV produces a print-out of group path length, azimuthal deviation and accurate geomagnetic latitude of the landing point.

There are various options for PLTDEV which are described in the next section.

PRECEDING PAGE BLANK-NOT FILMED

USAGE

SUBROUTINE PLTDEV(IN,NREC,MODE,ISUP)

Input parameters

Parameter type is shown in parentheses; refer to notes indicated by asterisks for details. There are no default values.

IN - input file name (left justified-zero fill) *1

NREC - number of records to be plotted. NREC should be less than or equal to the number of records in the input file. (integer)

MODE - an integer between 0 and 3 inclusive which controls the mode of operation. (integer) *2

ISUP - if ISUP = 0, a separate frame is used for each landing point. Otherwise all landing point curves appear in the same frame. (integer)

notes

*1 The input file is the "bundle of rays" produced by CRRSET and made random access through BIBSUB2 for example. See FILE DESCRIPTION for more details on this file.

*2 MODE = Ø means that information in the first NREC records of IN are plotted (and printed) using coordinate frame scales determined by PLTDEV (see ALGORITHM) i.e. there is no user control over what gets plotted.

MODE = 2 means that no plotting is done; only a print-out is provided of information which would be plotted if MODE were set equal to zero. As noted previously, landing point accurate geomagnetic latitude is also printed.

MODE = 3 means that group path length (along initial azimuth lines) are plotted. (Modes 0 - 2 refer to azimuth deviation.) Note, however, that the print out still includes azimuth deviation information when MODE = 3.

Labeled common input

If MODE = 1, the user must provide data for selecting what information is to be plotted and how it is to be plotted. This data

is obtained from labeled common area PDEV as follows (numbers in parentheses refer to locations in PDEV; all values are floating point unless indicated otherwise)

(10) frequency (mHz) of the rays to be plotted

(11) name of the landing point (Hollerith). This name should be in the form nxxxxxxxxx where n is the hop number (1-4) and xxx... is the landing point code name. For example "1RCVR" is the name for the first hop ground landing point.

(12) frame minimum x-value (the x-axis is azimuth)

(13) frame Δx (1 inch on the plot is equivalent to a Δx range in x values.

(14) frame maximum x-value

(15) frame minimum y-value

(16) frame Δy

(17) frame maximum y-value. (Since plots are limited in the y-direction, if y-axis length as computed from $(y_{max}-y_{min})/dy$ is greater than 8, it is set equal to 8 and some points may not be plotted.

(18) az_{min} - lower bound of initial azimuths

(19) az_{max} - upper bound of initial azimuths

(20) el_{min} - lower bound of initial elevations

(21) el_{max} - upper bound of initial elevations

The above sequence of values can be repeated as often as desired. A value of frequency (item (10) or (22) ... (10 + (n-1)12)) of zero terminates the subroutine.

PLTDEV produces plots for all frequency "matches" with the input file up to NREC. When NREC records have been plotted (or input) the file is rewound and the next set of parameters from PDEV are read in.

Use of other common storage

The following common storage areas must be declared with the sizes as indicated in another subroutine loaded prior to PLTDEV:

/PDEV/ (minimum of 7) If mode = 1 then the size of /PDEV/ is controlled by the location of the last frequency value which must be Ø.

/DATA/ ((3 · NPTS + 2) · NREC) where NPTS is the no. of landing points

/FITAIN/ (35) input file FIT area

/INBUF/ (514) input file buffer area

/IBUF/ (input file record size)

In addition, the following common areas are used by PDEV to communicate with other subroutines:

/SUB/ (2)

/CURVE/ (7) communicates with CURVE (PML 166)

/XARRAY/ (100)

/YARRAY/ (100)

If plotting is to be done (mode ≠ 2) then calls to PLTID3 and ENDPLT must be made sometime before and after PLTDEV respectively.

STORAGE REQUIRED

The amount of storage required by PLTDEV and its special subroutines (see SUBROUTINES), but not including common storage (except /XARRAY/ and /YARRAY/), is 3361_8 (1777_{10}).

ALGORITHM

The word addressable input file is opened and the first record read. This record provides some labeling information, transmitter location for azimuth deviation computations and information on the form of the "ray" records.

In the following description of the algorithm it is assumed the mode = Ø; modifications required for the other modes are summarized later.

The first frequency is found from the first "ray" record. As long as the frequency remains the same, records are read and data is obtained and/or computed and stored in common area /DATA/. The information stored in /DATA/ is later used to produce print outs and plots. This information is stored as follows:

$a_1,e_1,t_1,d_1,s_1,t_2,d_2,s_2,\ldots.t_n,d_n,s_n,a_2,e_2,t_1,d_1,\ldots$

where a_1,e_1 are the initial-azimuth and elevation angles of the first "ray" record; the first sequence of t_i,d_i,s_i are the group path lengths, landing point azimuths and acc.-geom. latitudes associated with the NPT landing points on the first ray record.

Similarly a_2,e_2, etc. are associated with the second ray record. If a ray does not actually land at one of the specified landing points, as indicated by r = ∅ (see CRRSET, PML 157), a value of 37777 77777 77777 77777B is given to the corresponding t_i,d_i,s_i. This has the effect on print out of having the letter "R" printed. It is used in plots to indicate a break in the curve of group path length vs. azimuth which may be bridged by a dotted line if it is a truebreak.

When the next frequency is found (or NREC records have been read), PLTDEV leaves the /DATA/ - fill loop and prints out the contents of /DATA/. Next the curve plotting loop is entered. X and y axis scales, etc. are determined. The y-axis origin and scale can be obtained from gpl_{min} and gpl_{max} values which are determined as data is transferred from the input file to /DATA/. However, it has been found expedient to override the logic which calculates y_{min} and Δy by setting y_{min} = 500. and Δy = 500. Similarly x_{min} and Δx are found using max. and min. values of landing point azimuths. If ISUP ≠ 0 (indicating superimpose landing point curves), the scaling is determined by the max and min over all landing points.

New sets of axes are drawn for each landing point if ISUB = 0, otherwise only one set of axes is drawn for each frequency.

For each landing point, i, the t_i,d_i information in /DATA/ is plotted in increasing elevation angle order as one curve as long as the initial azimuth ($a_1,a_2,\ldots$) remains the same. If a "break" occurs as indicated by t_i = 377777...., the curve sections (if there are more than 1) are joined by dotted straight lines.

A curve section is drawn by subroutine CURVE and initial-azimuth labels are plotted using subroutine LBL (part of the PLTDEV "family")

if the curve section is a first or last one. As each curve (as labeled by initial-azimuth) is completed, the location of the $a_1, a_2, \ldots$ are stored in array NEWAZ.

After all curves have been drawn, a call is made to PLTEL (also part of the PLTDEV "family") to draw dashed curves through equi-elevation points (gpl and az-dev. are functions of initial elevation angle for fixed initial azimuths) in increasing initial-azimuth order. If there is a break in elevation angle, the break is not connected as azimuth breaks are.

The above procedures are repeated for the next new frequency after the plot origin is moved by XLEN + 3 inches.

If MODE = 1, the algorithm is changed as follows:

(1) plot scales, etc. are determined from information in /PDEV/
(2) t,d,s data is put into /DATA/ only if there is a frequency match (with /PDEV/(10) for example)
(3) curves are plotted for the landing point specified by the current value of landing point name (PDEV(11) for example)
(4) points are plotted only if initial azimuth and elevation are within the ranges specified by the current /PDEV/ values, (for example by /PDEV/(18),(19),(20),(21)).

If MODE = 2, the algorithm is the same as for MODE = 0, except no plotting is done.

If MODE = 3, the algorithm is the same as for MODE = Ø except that for plots, instead of plotting t_i, d_i (group path length, landing point azimuth), the point t_i, $a + \Delta$ is plotted where a is the current initial azimuth value and Δ is a sequenced increment to prevent points from falling on the same line.

SPECIAL CAUTION AND FEATURES

PLTDEV is fairly well self protected but the following points should be observed:

(1) The input file must be word-addressable and structured as shown in FILE DESCRIPTION.

(2) The common storage areas /PDEV/, /FITAIN/, /IBUF/, /INBUF/ and /DATA/ must be declared as explained previously. Particular care must be taken to see that /DATA/ is at least as large as (3 · NPTS + 2) · NREC.

(3) If MODE = 1 is chosen /PDEV/ must be filled in as explained under USAGE. Be sure to terminate the data in /PDEV/ with a zero at a "frequency" location.

(4) If NREC is very large (and MODE ≠ 1) it is best to run PLTDEV only on weekends with a note to the operator (use a control card "PAUSE!" for example) that a long plot is to be produced.

TIMING

Timing depends on several factors. As a rule, even long plots should take less than 100 CPU seconds.

ERROR MESSAGES

None

SUBROUTINES

Subroutines marked with * are considered to be part of PLTDEV; others are obtained typically from User Library, BARLIB.

AURLAT (computes acc. geom. latitude)
CURVE (draws curves w/solid or dashed lines)
DEC* (decodes landing point name in /PDEV/)
ERROR1 (error subroutine for input file)
IEOF (error function for input file)
LBL* (labels curve end-points)
PLTEL* (draws equi-elevation curves)

ACCURACY

Not applicable

COMMENTS ON USAGE

For an overall look at "deviation data" PLTDEV should be called with MODE = 0. For later and perhaps less confused plots, MODE = 1 should be used on the same input file.

FILE DESCRIPTIONS

PLTDEV uses two files. The printed output file is created using Fortran formatted write statements. The file name is PRINT and should be so declared in the main program statement.

The input file (called IN) is a word-addressable file with fixed length records. This file is typically created by CRRSET (PML, 157) in sequential form followed by a conversion to word-addressable form through BIBSUB2.

This file consists of 1 header record and the rest ray records. Each record, including the header record has the same length. The structure of each record is:

header record
(npts ≠ 0)

-999., -999., -999., npts, m, nint, ID(4), r, Θ, Ø,
(HOP,WHY, i = 1, npts), (title (i), i = 1, nint)

where:

- npts = number of ray elements [integer]
- m = total record length [integer]
- nint = number of "integrated quantities" [integer]
- ID(4) = last four words for ID(10) in header record of RAYSET [Hollerith] from ray tracing
- r,Θ,Ø = transmitter dipolar coordinates (radians) [floating point]
- HOP,WHY = element hop and name [integer, Hollerith]
- TITLE = titles of "integrated quantities" [Hollerith]

ray record
$f,a,e,P_1,P_2, \cdots, P_{npts}$ (npts ≠ 0)

or

$f,a,e,P_1P_2, \cdots, P_{last}$ where P_{last} corresponds to the last element of a set of ray elements in RAYSET

and

P_i = HOP,WHY,$r_r,r_\theta,r_\phi,k_r,k_\theta,k_\phi,t,\Psi_g,q(1), \cdots q(nint)$

where:

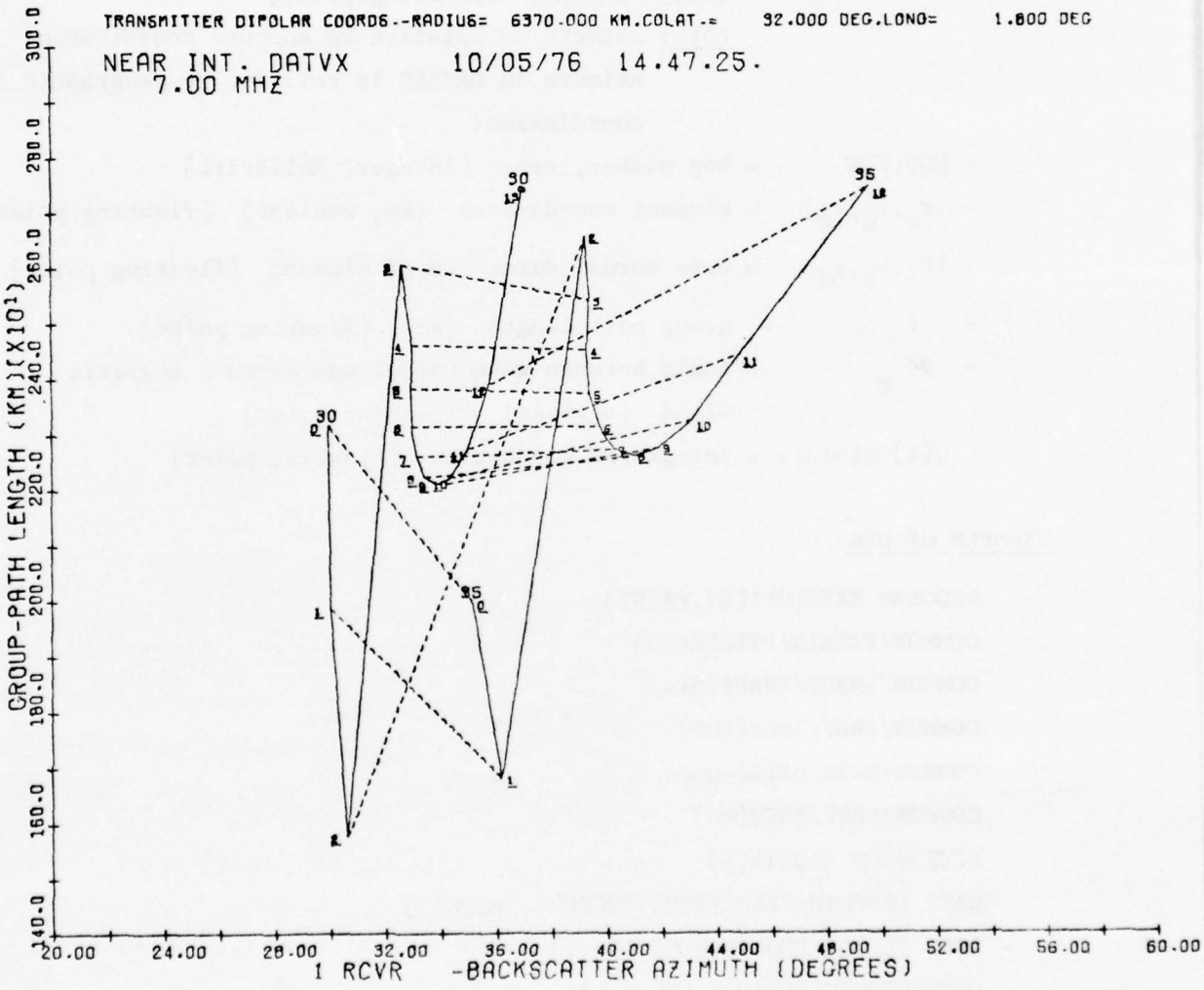
- f,a,e = initial ray frequency, azimuth, elevation (MHz, radians) [floating point]
  note: azimuth is relative to compute coordinates Azimuth in RAYSET is relative to geographic coordinates.
- HOP,WHY = hop number, name [integer, Hollerith]
- (r_r,r_θ,r_ϕ) = element coordinates (km, radians) [floating point]
- (k_r,k_θ,k_ϕ) = wave normal direction of element [floating point]
- t = group path length (km) [floating point]
- Ψ_g = angle between wave normal and earth's magnetic field (degrees) [floating point]
- q(1)-q(nint) = integrated quantities [floating point]

EXAMPLE OF USE

```
PROGRAM MAIN(OUTPUT,PRINT)
COMMON/FITAIN/FITAIN(35)
COMMON/INBUF/INBUF(514)
COMMON/IBUF/IBUF(100)
COMMON/DATA/DATA(5000)
COMMON/PDEV/PDEV(20)
DIMENSION PROGID(3)
DATA (PROGID="BARRETT", "2628", "AZDEV")
CALL PLTID3(PROGID,200.,11.,1)
CALL PLTDEV(5LRAYSR,456,3,1)
CALL ENDPLT
END
```

File RAYSR is assumed to have been created by CRRSET followed by BIBSUB2 for example.

An example of PLTDEV output is shown below.



NAME: PRINT3 (formerly PRINT2), revision 0, subroutine, PML 165
CATEGORY: General purpose
TITLE: Automatic formatted output
LANGUAGE: CDC Extended Fortran
PROGRAMMER: T.B. Barrett, Parke Mathematical Laboratories, Inc.
DATE: July 19, 1976

DESCRIPTION

PRINT3 may be used to output an array of data under variable format control. It is most useful in those situations where the number of columns of data to be output exceeds the capacity of a standard print page. The user may select data to be included as "header" information on each page as well as a column of data to be repeated on each page if more than one page is required to include all columns. In addition to the data, the user must supply an array of formatting data to control output. Spacing of data columns is taken care of by the subroutine.

USAGE

CALL PRINT3(M,M1,M2, TITLE, B, FORM, INDIC, NCHARMX, NROWS, IPRINT)

parameters

M - number of values to be printed. For example, an array of numbers of M columns and N rows are to be output. Some "columns" may be considered header data (see M1) [integer > 0]

M1 - number of header values. More explicitly, the first M1 "columns" in the data array are to be used in headers on each page. [integer $\geq$ 0]

M2 - column number within the array of M columns which is to be repeated on each page. [integer $\geq$ 0]

TITLE - an array of dimension (4,M) which contains Hollerith titling information for each of the M data columns. The four words in each title are printed in column fashion. (See the example at the end of the subroutine write-up.)

B - an array of up to 12 words (Hollerith) which are to be printed on each page as a title. This array should be filled with blanks if no title is to appear.

FORM - an array of M words (Hollerith) which contain valid format field specifiers. These specifiers must be of the form

lf#n

where:

l is one of the format conversion letters A, E, F, G, I, L, O, R

f is the integer field length

may be missing or a period

n may be missing or an integer

For example - A10, E16.4, I5, etc.

INDIC - an indicator which controls print out. PRINT3 is designed to be used in place of a line by line WRITE statement i.e. it accumulates one row of data to be output each time it is called. The first time PRINT3 is called INDIC must be set to zero in order to force PRINT3 to establish format arrays, etc. PRINT3 then changes INDIC to 1. When PRINT3 is called the last time i.e. the N^{th} row has been generated and is to be printed, INDIC should be set to 999 to force print out of accumulated data.

NCHARMX - maximum number of characters/line. [integer > 0]

NROWS - maximum number of rows per page, not including header and titling information. 15 lines are used if a 4 row header (i.e. the header can be contained in 4 rows) is used. (See example) [integer > 0]

IPRINT - logical file name (left justified zero fill) of the print file to receive the printed output. This file must be declared in a program statement.

use of common storage

The following labeled common storage areas should be declared in the calling subroutine or in a subroutine loaded before PRINT3:

/DATA/ (M x NROWS)

This common storage area is a scratch storage area used to accumulate data prior to print out of a page of data.

/QUANT1/ (M)

On each call to PRINT3 /QUANT1/ should contain one row of data to be printed.

STORAGE REQUIRED

The storage required for PRINT3, not including the above mentioned common storage areas is 1407_8 (824_{10}).

ALGORITHM

When PRINT3 is called with INDIC = 0, the format arrays PHEADER, QHEADER, HEADER and NUMBER are constructed so they can be used in an "execution time" format statement e.g. WRITE (IPRINT, HEADER(1,K)) (data list). These format arrays are established in such a fashion that the last column of data will fall within the page width boundary specified by NCHARMX. Two spaces between columns are used. As mentioned previously, INDIC must be zero on first call to PRINT3. After the various format arrays have been established, the subroutine continues into storage phase wherein one row of data from /QUANT1/ is stored in /DATA/.

One or more pages of data are printed whenever one of the following occurs:

(1) a header datum changes value
(2) NROWS have been accumulated
(3) INDIC has been set to 999.

If the number of columns of data cannot fit on a single page, the excess columns are printed on pages immediately following the starting page which is printed whenever one of the above conditions occur. Header information is repeated as is the "repeat" column specified by M2.

SPECIAL CAUTION AND FEATURES

Header information, if any, must appear only in the first M1 locations in /QUANT1/. The repeat column specified by M2 may appear anywhere. It will always appear on the printed page as the first column.

PRINT3 is limited to a maximum of 7 continuation pages (a values which should seldom be exceeded in practice).

TIMING

Not applicable.

ERROR MESSAGES

None

SUBROUTINES

None

ACCURACY

Not applicable.

COMMENTS ON USAGE

PRINT3 is useful in any situation for printing out data arrays in a neat fashion. It is particularly useful when the number of columns of data will not fit on a single page.

EXAMPLE OF USE

The following listing of subroutine FLUXPR1 (see PML 162) is a good example of the use of PRINT3. A sample of the printed output is also included.

```
      SUBROUTINE FLUXPR1(IN,IPRNT)
C REVISION---JANUARY 11,1977
C PURPOSE---PRINT OUT FLUX TUBE DESCRIPTION AT BACKSCATTER POINTS
C ARGUMENTS--- IN  FLUX TUBE FILE NAME (LEFT JUSTIFIED,ZERO FILL)
C              IPRNT  PRINT FILE FOR OUTPUT (LEFT JUSTIFY,ZERO FILL)
      COMMON/SYSTEM/DELA,DELE,DUM(8),DELG,DELF,DUN(8),PW,PI
      COMMON/IBUF/BUF(1)
      COMMON/FITAIN/FITAIN(1)
      COMMON/INBUF/INBUF(1)
      COMMON/FLUXPR/C
      COMMON/QUANT1/Q(30)
      DIMENSION FORM(30),TITLE(4,30),DID(12),ARRAY(11),RINT(4)
      DIMENSION NAME(2)
      EXTERNAL EOFILE,ERROR1
      DATA (FORM="A10","F4.1","F7.2","F7.2","E10.3",14*"F7.2","F7.3",
     3"F7.2","F7.3","F7.2","A10")
      DATA (((TITLE(I,J),I=1,4),J=1,15)=
     2"BACK-","SCATTER","POINT"," ",
     3"FREQUENCY"," "," ","(MHZ)",
     4"AZIMUTH"," ","COMPUTE","(DEGREES)",
     5"ELEVATION"," ",1H ,"(DEGREES)",
     6"FLUX TUBE","X-SECTION",1H ,"(STERRAD.)",
     7"TOTAL","PATH","GAIN","(DB/DEG)",
     8"WAVELENGTH","GAIN",1H ,"(DB KM**2)",
     9"TRANSMIT","ANTENNA","GAIN","(DB)",
     *"RECEIVE","ANTENNA","GAIN","(DB)",
     1"DISTANCE","(R-4) GAIN",1H ,"(DB KM-4)",
     2"RADAR","X-SECTION",1H ,"(DB KM2)",
     3"PULSE","SPREAD","GAIN","(DB)",
     4"ABSORPTION","(2-WAY)","GAIN","(DB)",
     5"REFLECTION","GAIN","MULTI-HOP","(DB)",
     6"TOTAL","PATH","GAIN","(DB/DEG)")
      DATA (((TITLE(I,J),I=1,4),J=16,24)=
     4"RAY-FOCUS","GAIN"," ","(DB)",
     5"GROUP PATH","LENGTH"," ","(KM)",
     6"GROUP PATH","LENGTH","SPREAD","(KM)",
     7"GROUND","RANGE"," ","(KM)",
     8"AZIMUTHAL","DEVIATION",1H ,"(DEGREES)",
     9"APOGEE","OR HEIGHT"," ","(KM)",
     1"ARRIVAL","ANGLE"," ","(DEGREES)",
     2"ACCURATE","GEOMAG.","COLATITUDE","(DEGREES)",
     3"POLARITY","RATIO OF","DIAG.-RAT.","DEL-GAMMA")
      DATA (NAME="90 DEG.","RCVR")
      CALL FILESO(FITAIN,3LLFN,IN,3LMRL,3120,2LDX,EOFILE,2LEX,ERROR1,
     33L8FS,314,3LFWB,INBUF)
      CALL OPENM(FITAIN,5LINPUT)
      CALL GET(FITAIN,BUF)
      IF (IEOF(X) .NE. 0) GOTO 1000
      R1=BUF(11)
      THETA1=BUF(12)
      THETAD=THETA1*57.2957795
      PHI1=BUF(13)
      PHID=PHI1*57.2957795
      NPTS=BUF(4) .0. 0
      N2=2*NPTS
      M=BUF(5) .0. 0
      IF (NPTS .EQ. 0) GOTO 1000
```

```
      M1=(M-4)/NPTS
      JSTART=5
      DO 100 I=1,4
100   DID(I)=BUF(I+6)
      ENCODE(80,8,DID(5)) THETAD,PHID,C
      DO 120 I=1,4
120   RINT(I)=BUF(13+N2+I)
      DO 130 II=1,2
      INDIC=0
160   CALL GET(FITAIN,BUF)
      IF (IEOF(X) .NE. 0) GOTO 150
      IF (BUF(1) .EQ. 999.) GOTO 150
      F=BACSCA1(JSTART,F0,PW,R1,THETA1,PHI1)
      IWHY=O(1) .O. 0
      IF (IWHY .EQ. NAME(II))155,160
155   IF (Q(25) .EQ. 0.) GOTO 161
      CALL PRINT3(24,3,4,TITLE,DID,FORM,INDIC,132,80,IPRNT)
      GOTO 160
150   CALL REWND(FITAIN)
      CALL GET(FITAIN,BUF)
      IF (INDIC .NE. 0) GOTO 170
      WRITE 7  $ GOTO 1000
170   INDIC=999
      CALL PRINT3(24,3,4,TITLE,DID,FORM,INDIC,132,80,IPRNT)
130   CONTINUE
1000  CALL CLOSEM(FITAIN)
      RETURN
1     FORMAT(1H1," FLUX TUBE REPORT"/4X4A10)
2     FORMAT(" HOP NUMBER "I2", LANDING POINT "A10)
3     FORMAT(" PHASE PATH CALCULATED")
4     FORMAT(" ABSORPTION CALCULATED")
5     FORMAT(" DOPPLER CALCULATED")
6     FORMAT(" GEOMETRIC PATH CALCULATED")
7     FORMAT(" FLUXPR,NO FLUX TUBES AVAILABLE")
8     FORMAT(" TRANSMITTER AT "F7.2" (DEG. COLAT.) "F7.2
     $" (DEG LONG.), DPSI LOSS="F3.1" DB/DEG")
      END
```

NEAR INT. DATVX 11/16/76 15.21.03. TRANSMITTER AT 32.00 (DEG. COLAT.) 1.80 (DEG LONG.), DPSI -DSS=3.0 DB/DEG

BACK-SCATTER POINT RCVR FREQUENCY 5.0 AZIMUTH 40.00

(MHZ) COMPUTE (DEGREES)

ELEVATION (DEGREES)	FLUX TUBE X-SECTION (STERADS.)	TOTAL PATH GAIN (DB/DEG)	WAVELENGTH GAIN (DB (4PI^2)	TRANSMIT ANTENNA GAIN (DB)	RECEIVE ANTENNA GAIN (DB)	DISTANCE (R-4) GAIN (DB (4-4)	RADAR X-SECTION (DB KM2)	PULSE SPREAD GAIN (DB)	ABSORPTION (2-WAY) GAIN (DB)	REFLECTION GAIN MULTI-HOP (DB)
2.00	.3048E-04	-172.45	-35.+3	0.00	0.00	-151.25	19.86	-15.03	0.00	0.00
4.00	.3145E-04	-151.37	-35.+3	0.00	0.00	-144.34	29.31	-3.91	0.00	0.00
5.00	.3138E-04	-146.60	-35.+3	0.00	0.00	-132.52	31.53	-.13	0.00	0.00
6.00	.3135E-04	-150.37	-35.+3	0.00	0.00	-138.77	34.39	-5.55	0.00	0.00
7.00	.3125E-04	-153.37	-35.+3	0.00	0.00	-151.15	35.79	-3.35	0.00	0.00
8.00	.3115E-04	-157.59	-35.+3	0.00	0.00	-150.12	35.87	-7.81	0.00	0.00
9.00	.3107E-04	-143.37	-35.+3	0.00	0.00	-134.44	32.97	-3.03	0.00	0.00
10.00	.3087E-04	-145.59	-35.+3	0.00	0.00	-143.42	27.44	-+.11	0.00	0.00
11.00	.3085E-04	-147.41	-35.+3	0.00	0.00	-141.52	26.25	-5.71	0.00	0.00
12.00	.3078E-04	-151.49	-35.+3	0.00	0.00	-147.38	28.85	-7.02	0.00	0.00
13.00	.3066E-04	-153.49	-35.+3	0.00	0.00	-149.17	25.53	-5.61	0.00	0.00
14.00	.3056E-04	-152.52	-35.+3	0.00	0.00	-148.30	27.64	-5.83	0.00	0.00
15.00	.3045E-04	-152.58	-35.+3	0.00	0.00	-149.45	28.58	-5.33	0.00	0.00
16.00	.3037E-04	-152.58	-35.+3	0.00	0.00	-147.38	25.75	-+.81	0.00	0.00
17.00	.3019E-04	-152.84	-35.+3	0.00	0.00	-147.37	25.11	-+.67	0.00	0.00
18.00	.2990E-04	-157.63	-35.+3	0.00	0.00	-143.37	24.87	-+.73	0.00	0.00
19.00	.2977E-04	-154.83	-35.+3	0.00	0.00	-149.77	24.53	-+.83	0.00	0.00
20.00	.2956E-04	-154.45	-35.+3	0.00	0.00	-148.57	24.15	-+.61	0.00	0.00
21.00	.2935E-04	-154.19	-35.+3	0.00	0.00	-149.11	23.52	-+.11	0.00	0.00
22.00	.2917E-04	-153.91	-35.+3	0.00	0.00	-145.78	22.59	-3.21	0.00	0.00
23.00	.2895E-04	-151.51	-35.+3	0.00	0.00	-144.63	21.23	-1.53	0.00	0.00
24.00	.2875E-04	-147.19	-35.+3	0.00	0.00	-140.47	18.96	-.01	0.00	0.00
25.00	.2855E-04	-133.19	-35.+3	0.00	0.00	-124.19	10.52	-.01	0.00	0.00
26.00	.2837E-04	-151.57	-35.+3	0.00	0.00	-141.26	18.86	-2.84	0.00	0.00
27.00	.2795E-04	-153.57	-35.+3	0.00	0.00	-151.11	23.59	-6.81	0.00	0.00
28.00	.2787E-04	-150.31	-35.+3	0.00	0.00	-153.71	27.30	-11.07	0.00	0.00

NEAR CNT, CATVX 11/16/76 15.31.03. TRANSMITTER AT 32.00 (DEG. COLAT.) 4.80 (DEG .ONG.), DPSI -OSS=3.0 DB/DEG

BACK-SCATTER POINT — RCVR — FREQUENCY 5.0 (MHZ) — AZIMUTH 40.00 COMPUTE (DEGREES)

ELEVATION (DEGREES)	TOTAL PATH GAIN (DB/DEG)	RAY-FOCUS GAIN (DB)	GROUP PATH LENGTH (KM)	GROUP PATH -LENGTH SPREAD (KM)	GROUND RANGE (KM)	AZIMUTHAL DEVIATION (DEGREES)	APOGEE OR HEIGHT (KM)	ARRIVAL ANGLE (DEGREES)	ACCURATE GEOMAG. COLATITUDE (DEGREES)	POLARITY RATIO OF MAG.-RAT. DEL-GAMMA
3.00	-172.88	0.00	2831.80	330.67	2537.41	21.311	189.31	1.887	31.47	+1.18 167.
4.00	-151.87	0.00	2287.67	71.75	2086.64	15.658	202.78	9.804	28.39	- .88 170.
5.00	-146.65	0.00	2160.38	7.41	1959.74	14.481	216.54	13.365	27.85	+ .91 -34.
6.00	-156.33	0.00	2159.32	33.89	1941.23	13.256	231.12	14.363	28.08	-1.50 149.
7.00	-159.17	0.00	2213.74	51.24	1959.78	17.120	245.44	13.295	28.69	-1.48 143.
8.00	-157.68	0.00	2337.58	45.22	2023.28	13.471	259.45	11.353	29.52	-1.52-150.
9.00	-149.07	0.00	2374.78	15.23	2070.93	21.617	273.38	11.001	30.32	-1.25 37.
10.00	-145.52	0.00	2397.45	13.72	2072.63	22.748	286.24	11.121	30.69	+ .59 158.
11.00	-147.41	0.00	2368.33	35.18	2030.63	22.954	298.51	12.297	30.65	- .80-166.
12.00	-151.48	0.00	2310.39	37.73	1964.42	22.439	310.28	14.181	30.35	- .89 174.
13.00	-152.48	0.00	2246.10	34.34	1895.07	21.648	321.10	16.384	29.96	- .92-163.
14.00	-152.52	0.00	2187.07	29.78	1831.75	20.830	331.05	18.508	29.59	- .93 175.
15.00	-152.58	0.00	2136.41	25.88	1776.71	19.939	339.99	20.715	29.25	- .94-162.
16.00	-152.66	0.00	2091.75	23.18	1728.22	19.070	347.89	22.553	28.95	- .94 176.
17.00	-152.86	0.00	2051.73	21.36	1685.07	18.239	354.80	24.423	28.67	- .95 176.
18.00	-153.59	0.00	2013.31	22.41	1645.31	17.332	360.57	25.957	28.41	- .97-162.
19.00	-154.39	0.00	1975.32	22.79	1606.63	16.391	365.23	27.374	28.14	- .98 176.
20.00	-154.46	0.00	1935.39	21.56	1568.95	15.356	368.80	28.573	27.87	- .99-162.
21.00	-154.12	0.00	1898.52	19.83	1533.61	14.320	371.30	29.886	27.61	- .99-162.
22.00	-152.81	0.00	1864.39	15.55	1501.57	13.328	372.81	31.033	27.38	- .98 178.
23.00	-150.51	0.00	1837.71	11.37	1473.85	12.456	373.36	32.093	27.19	- .94-160.
24.00	-147.00	0.00	1818.34	5.31	1450.78	11.775	373.01	33.173	27.05	- .86-158.
25.00	-139.10	0.00	1808.31	2.64	1433.12	11.362	371.82	33.375	26.97	- .80 177.
26.00	-150.67	0.00	1812.36	14.43	1422.43	11.329	374.09	34.743	26.87	-1.16-144.
27.00	-150.87	0.00	1837.10	36.14	1422.29	11.477	382.65	35.293	27.09	-1.17 177.
28.00	-168.71	0.00	1898.15	95.85	1441.52	13.438	396.21	35.313	27.44	-1.15 172.

NAME: CRT1: revision 0, subroutine, PML 168
CATEGORY: General purpose display
TITLE: CRT (intensity) display
LANGUAGE: CDC Extended Fortran
PROGRAMMER: T. B. Barrett, Parke Mathematical Laboratories, Inc.
DATE: September 1976

DESCRIPTION

The primary purpose of CRT1 is to display a function of two variables as an apparent intensity variation as a function of x-y coordinates. Actually the intensity variation is obtained through the use of a simulated "half tone dots". The dots are generated by a subroutine, PLOTI, which is essentially the same as that used by Stephenson and Georges in "Computer Routines for Synthesizing Ground Backscatter From Three-Dimensional Raysets", (1969) ESS/Tech. Rpt. ERL 120-ITS84.

There are various modes of operation possible which allow the user to "normalize" the function to be displayed in various ways (see below).

USAGE

SUBROUTINE CRT1(DB,GAIN,NLEV,INDIC,IX,IY,MODE)

Input parameters

Parameter type is shown in parentheses; refer to notes indicated by asterisks for details. Default values, if any, are indicated by brackets. (A default value is substituted for a given value of 0.)

DB (floating point) [20.] intensity display "dynamic range". *1
It functions in various ways according to MODE.

GAIN (floating point) [1.] intensity display "gain". *1
DB and GAIN interact as indicated in the notes to produce different display effects according to MODE of operation.

NLEV (integer) [11] is the number of intensity levels to be used in the display. The maximum at present is 11. *2

INDIC (integer) is a parameter which can be used to superimpose functions on the same coordinate frame. *3

IX, IY } (integer) are the inches per bin (in "cth's") in the x and y directions. *4

MODE (integer) (mode of operation) *5

= 0 no special normalization, DB and GAIN are used in the "standard" manner.

= 1 GAIN is interpreted as the maximum intensity to be plotted; the minimum intensity to be plotted is DB db below this value.

= 2 GAIN has no function. Each column to be plotted is normalized (divided) by the maximum function value for that column. Intensity levels are adjusted in equal steps to DB db below the maximum.

= 3 same as for MODE = 2 except substitute "row" for "column ".

= 4 is used to "smooth" the display using a 2 dimensional weighting function and convolution. At the time of writing it should be considered experimental.

notes

*1 In the "standard" mode (MODE = 0) of operation the functional value to be displayed is represented by a half-tone dot pattern which is the same for all values within a given "contour" interval. These intervals, I_i, are defined by DB, GAIN and NLEV by

$$I_i = I_{min} + (i-1) \cdot \Delta I \; ; \quad i = 1 \text{ to NLEV}$$

where

$$I_{min} = \frac{fg}{K \cdot GAIN} \quad ; \quad \Delta I = \frac{I_{max} - I_{min}}{NLEV-1}$$

$$I_{max} = \frac{fg \cdot K}{GAIN}$$

fg = geometric mean of the function f (assumed positive)

$$K = \text{"contrast"} = 10^{\frac{DB}{20}}$$

For MODE = 1 operation I_{max} = GAIN and

$$I_{min} = I_{max} / K$$

For MODE = 2,3 operation, the levels are determined according to:

$$I_i = I_{max}\ 10^{(DB/20)(i-NLEV)} \quad ; \quad i = 1,NLEV$$

where

I_{max} = f(c or r max) (column or row max of the function f)

*2 There are at present 11 distinctive "grey levels" available. If a function value falls below I_{min}, it is not plotted (i.e. is represented by "white"). If a function value falls at or above I_{max} and NLEV is 11, it is represented by "black". Note that if NLEV is less than 11, function values which fall on or above I_{max} are represented by a "grey".

*3 If INDIC = 0, the axis and calibration strips, etc. are drawn as well as the function values poltted. If INDIC >0 only the function values are plotted. This means that it is possible to superimpose functions (or different parts of the function generated at different times) on the same coordinate frame.

*4 A single "half tone dot" can be imagined to occupy a square which is C inches on a side. (To be more exact, the elemental points which make up a "dot" fall within this square.) It is frequently desirable to represent the function value at the spatial "point" x,y by more than one dot, e.g. in order to fill up more of the available plotting area. Thus we can imagine function values being represented by "super-dots" or rectangles which are C·IX by C·IY on a side. The values of IX and IY which the user chooses should be considered to be maximum values. Subroutine CRT1 will decrease these quantities if necessary to contain the plot on the frame. (see "labeled common input")

*5 If MODE = 2 or 3, a small "strip chart is plotted on the display to indicate the relative values of the maximum function values (row or column maxima) which were used to set the intensity levels for the display.

Labeled common input

Labeled common area /DATA/ contains the function values (z-value) to be plotted as well as values which indicate the corresponding values of x,y. Thus the function, f, should be stored in /DATA/ in the sequence:

/DATA/$x_0, \Delta x, n_x, y_0, \Delta y, n_y$, $f(x_0,y_0)$, $f(x_0+\Delta x,y_0)$,

... $f(x_0+(n_x-1)\Delta x,y_0)$, $f(x_0,y_0+\Delta y)$, $f(x_0+(n_x-1)\Delta x,y_0+\Delta y)$,

... $f(x_0+(n_x-1)\Delta x,y_0+(n_y-1)\Delta y)$

Note that the "dot" or superdot" which represents f(x,y) has local origin at relative location x,y in the plot frame.

Title (in Hollerith) for the plot are contained in common area /TITLES/ as follows:

ID(4)	4 words of plot indentification which are plotted at upper left in frame.
TITLE(6)	additional plot identification plotted beneath ID.
XTITLE(6)	x-axis title.
YTITLE(6)	y-axis title.
PTITLE(6)	"parameter" information plotted below TITLE.

Following this Hollerith information the integer length (number of characters) of the titles should be included in the order NTC, NXC, NYC, NPC.

Use of other common storage

/DDXY/ communicates with subroutine DXY.

STORAGE REQUIRED

The amount of storage required by CRT1 and its special subroutines (see SUBROUTINES), but not including common storage except /DDXY/ is 2517_8 (1359_{10}).

ALGORITHM

Most of the subroutine is a straight forward implementation of the various modes of operation as defined above. The available total

plotting area is 11" by 17" (equivalent lengths in the y,x directions respectively if this were the usual "pen-plot"). Some of this space is used for titles, etc. so that the effective plot space is considered to be 9.5" by 15.5". As mentioned above this space can be filled with rectangular dots which are up to IX·C by IY·C inches on a side. The value of C is given in a data statement which at the time of writing is .05. Each dot contains one or more points which are multiples of C/7 (= .007143 with the present value of C). Since the minimum CRT plotter increment is .005, C should not be less than .035 inches. The function to be plotted is essentially an n_x by n_y matrix of real numbers. This matrix can be plotted provided $n_x \leq 15.5/C$ and $n_y \leq 9.5$. If this condition is not fulfilled, the program halts execution with a fatal error message.

The coordinate frame axes are drawn and partially labeled by system subroutine AXISL. Among other arguments required are NUMDIV, the number of major division; DIVLEN, length in inches of a major division; NUMSUB, the number of minor divisions; BEGNUM, the number used to label the beginning of the axis; DELNUM, incremental change in the number used for axis labeling; NUMDEC, the number of decimal places to be used in the number labels. These quantities are determined as follows:

let a_{max} = maximum total axis length

a = DIVLEN = length of major subdivision (each major subdivision is labeled with a scale number)

n = NUMDIV = number of major subdivisions

q = length of superdot along axis

m = NUMSUB = number of minor subdivisions / major division (possible values used here are 10,5,2,1 where 1 is equivalent to minor subdivision = major subdivision)

k = total number of elements to be plotted in the axis direction (n_x or n_y)

Δ = change in function argument from one superdot to the next (Δ_x or Δ_y)

Δ_s = DELNUM = change in scale value per major division

then:

(1) adjust q in increments of C such that $q \cdot k \leq a_{max}$

(2) adjust m such that $m \cdot q \leq a_{max}$

(3) $a = m \cdot q$

(4) $n = [\frac{q \cdot k}{a} + 1]$; [ ] means ineger part of

(if $n \cdot a > a_{max}$, reduce n by 1)

(5) $\frac{\Delta_s}{\Delta} = \frac{a}{q}$

For printing purposes the scale numbers (BEGNUM, DELNUM) are adjusted (by factors of 10) such that Δ_s can be expressed in the form $\pm$ xx.xx $\cdot 10^i$ where i is a positive or negative integer or zero and the first x is not zero.

SPECIAL CAUTION AND FEATURES

The user should pay attention to values assigned to XMIN, DELX, YMIN, DELY since these values eventually determine the scale numbers which appear on the display (see ALGORITHM). The function to be plotted should be positive (or normalized in some fashion to be made positive). Function values of zero are permitted but will not be plotted. (A zero value can be considered to correspond to a point where the function is not defined.)

The following "debugging" information is printed on the OUTPUT file:

"CRT1,COMMON INPUT XMIN, DELX, NX, YMIN, DELY, NY

x_0 , Δx , n_x, y_0, Δy , n_y

n_d , k , i

n , a , m , s , Δ_s for x-axis then for y-axis

where n_d = number of dots/superdot

s = BEGNUM = scale factor labeling the axis origin

See ALGORITHM for the meaning of the other symbols. The values of s and Δ_s have been normalized such that $\Delta_s = \pm$ xx.xx $\cdot 10^i$.

The subroutine is "protected" for an all zero function (axes and labels will be drawn but nothing plotted).

Note that calls to CRTPLT and ENDPLT must be made prior to and after CRT1 is called. (see EXAMPLE OF USE)

TIMING

Timing depends mainly on the number of function elements which fall above the minimum "contour" level, I_{min}, and the number of dots in each superdot used to represent a function value. At present no timing information is available.

ERROR MESSAGES

"CRT1 - PLOT OUT OF BOUNDS IN X-DIRECTION"

or

"CRT1 - PLOT OUT OF BOUNDS IN Y-DIRECTION"

if $q \cdot k > a_{max}$ (see ALGORITHM) for the minimum value of q

"DATA IS ALL ZEROES"

if the function to be plotted is identically zero.

SUBROUTINES (not including "standard" plot software)

Subroutines marked with * are considered to be part of CRT1.

AXISL - "systems linear axis plotting subroutine
FCONTR - determines contour levels
INTENS4 - used only if MODE = 4 for smoothing
PLOTI* - "half-tone" dot generator
SCALF - used for finding factors of 10, etc. for scaling purposes

ACCURACY

Not applicable

COMMENTS ON USAGE

CRT1 can be used to display any positive function of two variables (usually in the "standard" mode (MODE = 0)). The other modes are useful for simulating special devices such as ionosondes where the particular method of normalization implied by a particular mode is actually used.

FILE DESCRIPTIONS

The only files used are OUTPUT for formatted output messages and (implicitly) a plot file which should be disposed properly at the end of a job (see EXAMPLE OF USE).

EXAMPLE OF USE

The following example assumes that subroutine BIN (see PML 169) is used to generate the function to be displayed thus some of the common declarations are made for this subroutine.

```
      PROGRAM MAIN (OUTPUT,PRINT)
      COMMON/FITAIN/FITAIN(35)
      COMMON/INBUF/INBUF(514)
      COMMON/MESSA/IFATAL,MESSA(20)
      COMMON/IBUF/IBUF(100)
      COMMON/DATA/DATA(6000)
      COMMON/SYSTEM/ANTENNA(10), RECEIVE(10), TRANS(10)
      DIMENSION PROGID(3)
      DATA (PROGID="BARRETT", "2628", "CRT-PLOT")
      DATA (ANTENNA=1.,1.), (RECEIVE=200.,.5), (TRANS=50.,10000.)
      CALL BIN(5.00,2,1,5LFLUXT,6000)
      IF (IFATAL.EQ.1) GOTO 1000
      CALL CRTPLT(PROGID,1.,17.)
      CALL CRT1(200.,1.,11,-1,6,6,2)
      CALL ENDPLT
1000  STOP
      END
```

Two examples of CRT1 output are shown in Figures 1 and 2. Figure 2 includes a "normalization" curve and a calibration strip.

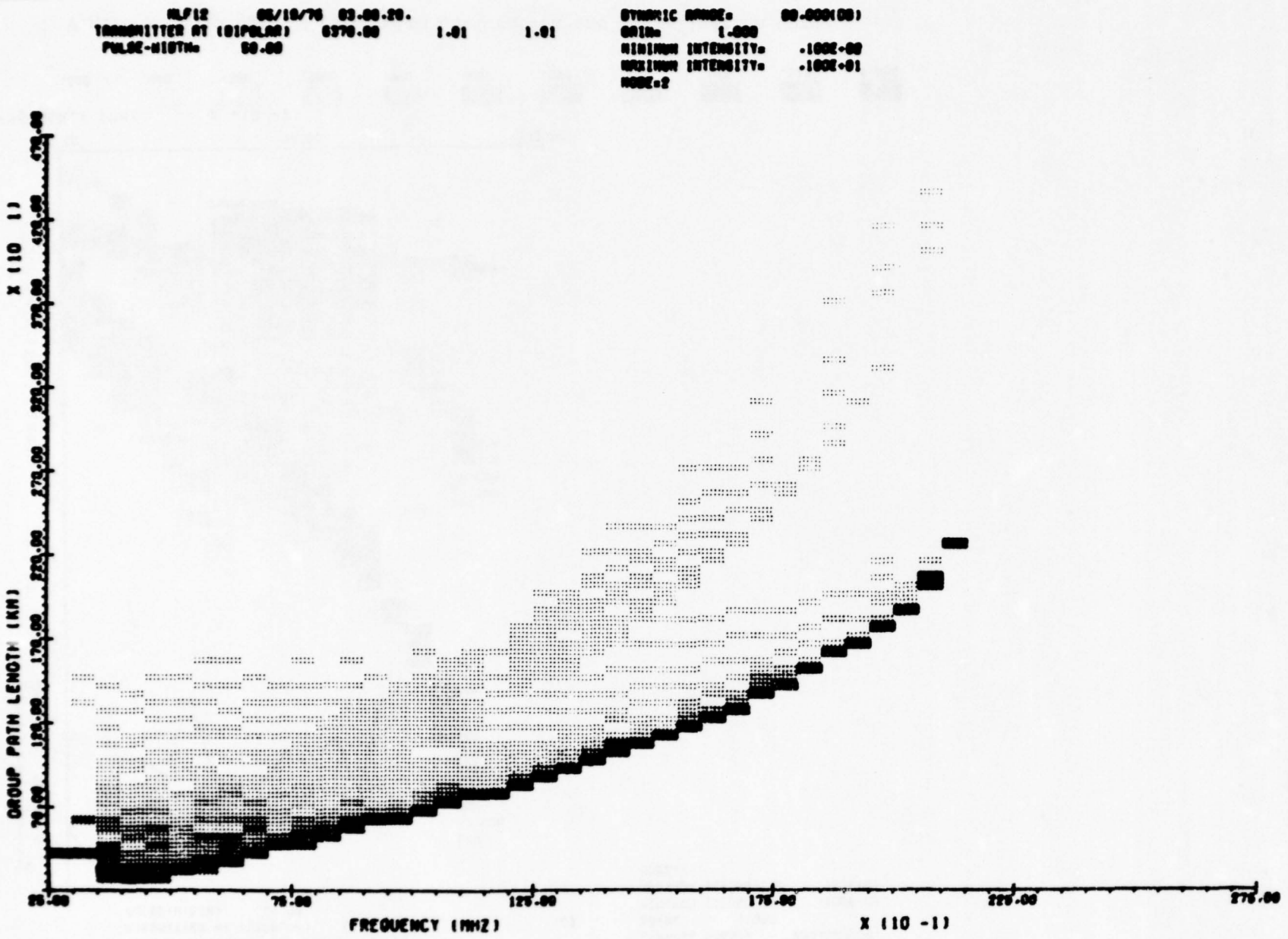


Figure 1 Ionogram Sample

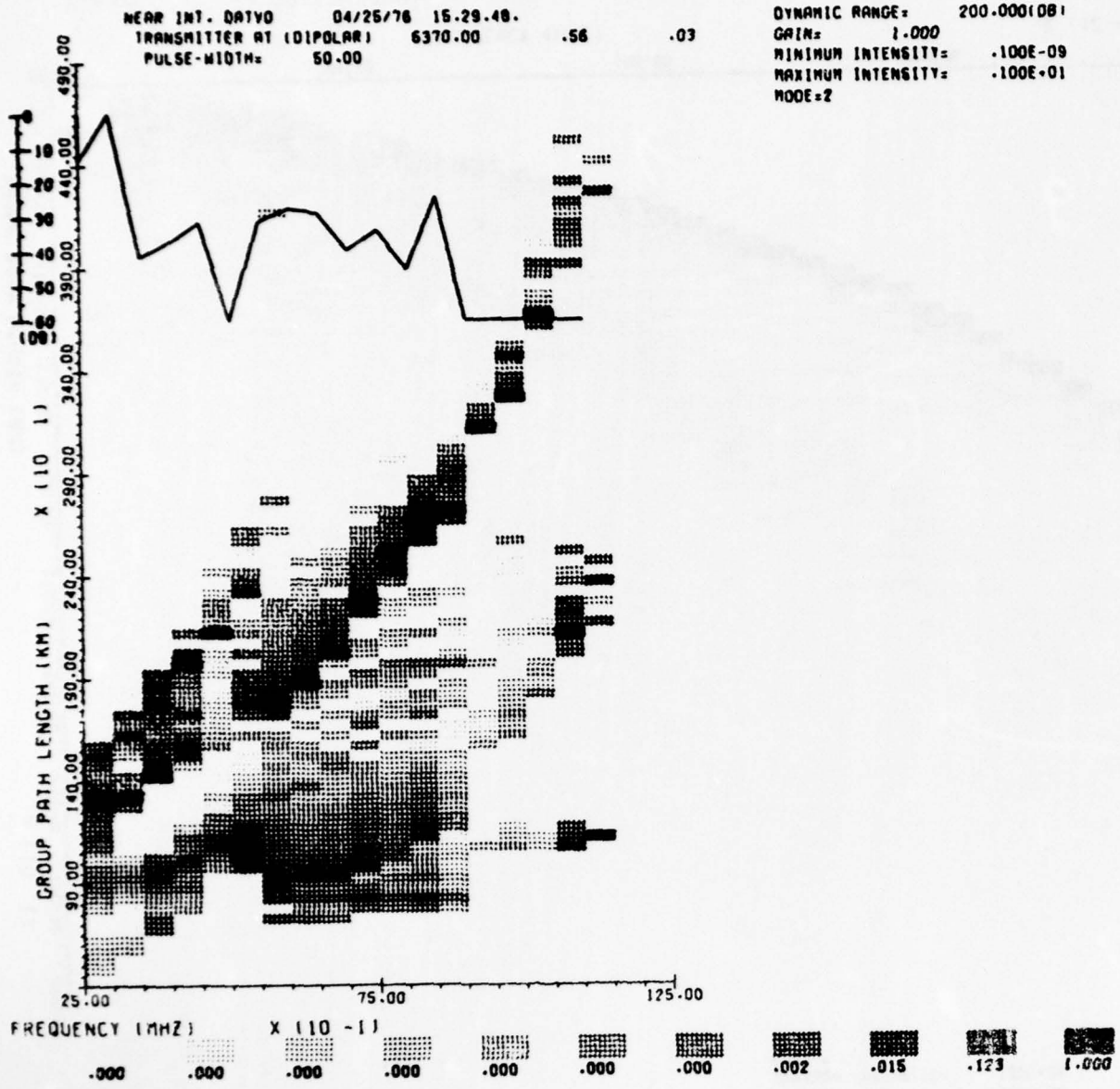


Figure 2 Ionogram Sample with Calibration and Normalization Curve

NAME: BIN, revision 0, subroutine, PML 169
CATEGORY: Special purpose
TITLE: Binning of power for ionogram displays
LANGUAGE: CDC Extended Fortran
PROGRAMMER: T.B. Barrett, Parke Mathematical Laboratories, Inc.
DATE: June 1976

DESCRIPTION

This subroutine is used to sort (and sum) "incremented backscattered power" into various "bins" according to various criteria. These criteria fall into 3 categories which are coded as CAA, CIA and AII; they are described below. It first should be pointed out that although this subroutine was written for a very specific purpose, i.e. ionogram synthesis, it can be used in more general situations where a similar "binning" is required. Thus BIN is described in fairly general terminology although in practice its operation is governed by the specific structure of the file from which input data to be binned is obtained (see FILE DESCRIPTION).

The easiest way to understand BIN is to imagine a table of values of P, G, f, a, e where, for example:

P = incremental power
G = group path length
f = frequency
a = azimuth
e = elevation angle

In general there will be several different values of P and G for the same set of values of f, a, e.

In CAA binning the table is first searched for a given value (constant) of f, a or e to produce a sub-table of values P, G, f, a, e (for example $f = f_o$ in this sub-table). A set of "bins" is then established for the remaining f, a, e quantities (for example a_1, $a_1 + \Delta a$, $a_1 + 2\Delta a$, $\cdots a_2$; e_1, $e_1 + \Delta e$, $\cdots e_2$) and values of P from the sub-table are summed into these bins regardless of the value of G.

In CIA binning the table is searched as before to produce a sub-table with a constant value of f, a or e. A set of bins is then established for one of the remaining f, a, e quantities (the A quantity) and G. Values of P from the sub-table are summed into these bins regardless of the value of the third (the I quantity).

In AII binning a set of bins is established for one of the quantities of f, a or e and G. Values of P from the table are summed into these bins regardless of the values of the other two quantities.

USAGE

SUBROUTINE BIN(F1,A1,E1,IN,ISIZE)

Input parameters

Parameter type (integer, etc.) is shown in parentheses; refer to notes indicated by asterisks for details. There are no default values.

F1	frequency "indicator" (floating point, integer)	*1
A1	azimuth " " " "	*1
E1	elevation " " " "	*1
IN	input file name (left justified, zero fill)	*2
ISIZE	amount of storage available (integer) in /DATA/	*3

notes:

*1 These parameters indicate whether the quantities f, a, e are C, A or I-variables. In addition, if they are A-variables, whether the binning should be done such that they are x or y-variables i.e. whether they normally supply the x-values or y-values for intensity plots. See output for further details on the "x-y" characteristics of the quantities. If, for example, F1 is a floating point number, F1 is treated as a constant (a C-variable) and the sub-table produced from the table (in this case the input file) will contain records with values of frequency equal to F1. The other acceptable values are integer 1, 2 or 3:

1 means this is an I-variable
2 means this is an A-variable (x-variable)
3 means this is an A-variable (y-variable)

Any other value (e.g. 4) will be treated as floating point and will probably produce erroneous results.

*2 This is the logical file name of the input file which contains the "table" of P, G, f, a, e values. See FILE DESCRIPTION for the assumed structure of this file.

*3 The main output of BIN (see output) is stored in labeled common area /DATA/. ISIZE should set to the dimension value which has been declared for /DATA/ to prevent a possible subroutine malfunction.

labeled common input

Additional input is required by BIN in order to establish bin widths. This is done through labeled common area /SYSTEM/. The following locations in system provide the indicated bin widths:

(1) a (azimuth resolution, degrees)
(2) e (elevation resolution, degrees)
(12) f (frequency resolution, MHz)
(21) G (group path length resolution, km)

In addition, location (22) in /SYSTEM/ contains the total transmitted power which is passed to BACSCAl (see ALGORITHM and PML 163) for incremental power calculations.

labeled common output

The main output of BIN is backscatter intensity vs x and y in labeled common /DATA/ as follows:

$x_0, \Delta x, M_x, Y_0, \Delta y, M_y, b(x_1,y_1), b(x_2,y_1),\ldots,b(x_{M_x},y_1), b(x_1,y_2), \ldots,$

$b(x_{M_x},y_2), \ldots, b(x_{M_x},y_{M_y})$.

The x - bin boundaries are defined as x_o, $x_0 + \Delta x$, $x_o + 2\Delta x$, ..., $x_o + M_x \cdot \Delta x$, and similarly for the y - bin boundaries.

The value of parameter ISIZE (see above) should be at least as large as $M_x \cdot M_y + 6$. If it is not, BIN will terminate with an error message.

In addition BIN provides title information for use with various display subroutines in labeled common /TITLES/ as follows:

ID(4) This is the 4 words of ID information obtained from RAYSET by subroutine CRRSET (see PML 157)

TITLE(6) At present TITLE gives the location of the transmitter in dipolar coordinates.

XTITLE(6) x-axis title

YTITLE(6) y-axis title

PTITLE(6) Parameter title. The contents of PTITLE depends on the type binning done as determined by the variables F1, A1, E1.

NTC,NXC, NYC,NPC number of characters in TITLE, XTITLE, YTITLE, PTITLE respectively.

use of other common storage areas

Storage areas /TITLES/ and /SYSTEM/ are declared in BIN to have lengths 32 and 22 respectively. The following storage areas must be declared in a subroutine which is loaded prior to BIN. (Usually this is done in the main program - see EXAMPLE OF USE.) The sizes of these areas are indicated in parentheses.

/MESSA/ (21)
/IBUF/ (length of input record - see FILE DESCRIPTION)
/FITAIN/ (35)
/FITAOUT/ (35)
/INBUF/ (35)
/OUTBUF/ (35)

STORAGE REQUIRED

The amount of storage required by BIN and its associated subroutine ADDS, but not including common storage areas is 1335_8 (734_{10}).

ALGORITHM

First, the number of constant variables (there can be only 1 or 0) and their "location" e.g. F1, A1 or E1 are determined and the sub-table created. If there is no C-variable, a sub-table of sorts is created anyway, in that a record from the input file is copied to the scratch "sub-table" only if a value of P can be calculated from this record. (Thus the input "table" does not really contain P-values but only quantities from which P can be calculated. P can be calculated only if a "flux tube" has one or more landing points and a well defined cross-section there. See subroutine DEN, PML 161.) It may happen that there are two "matches" between the C-value and the corresponding value of f, a or e - whichever is to be selected. In this case, BIN rewinds the input table, selects the first value of f, a or e found and uses this as the C-value.

The bin widths are obtained from labeled common /SYSTEM/ as mentioned above. The bin boundaries, however, are obtained from the appropriate maximum and minimum values of the quantities f, a or e. If there are no C-variables, these max. and min. are obtained from the last record in the input file. Otherwise they are obtained from the sub-table after it has been created.

The sub-table is actually a file which is created by BIN with the name "FILE". Both "FILE" and the input file are manipulated using Record Manager calls.

After FILE has been created, the mode of operation CAA, CIA, or AII is determined and titles appropriately filled in. (These titles can be used by various display subroutines.) In addition the bin boundaries, number of bins and bin widths are determined.

For all binning operations, the "sub-table" is read into IBUF and then a call is made to BACSCAl for calculation of the incremental powers which are contributed by this "line" in the table. Actually, because of the "multi-valuedness" of the f, a, e → P, G relationship because of multiple landing points, each record in IBUF may correspond to several

lines in the "sub-table".

As noted in FILE DESCRIPTION, each record is of the form f, a, e, P_1, ... P_n where the P_i contain the information required to compute incremental power at possibly n landing points in addition to group path lengths to these landing points. For CAA binning group path length is irrelevant so incremental powers are summed over all landing points and put into the appropriate f, a or e bins. For CIA or AII binning, group path length is one of the bins so the summation over landing points is not done prior to binning - rather the binning is done with each call to BACSCAl.

SPECIAL CAUTION AND FEATURES

1) The values of Fl, Al, El should be set to integer 1, 2, 3 or a floating point number.
2) The value of ISIZE should be set to the size of labeled common area /DATA/.

TIMING

Unknown, but CPU time should be small since calculations are elementary. Timing is probably heavily dependant on the time required to execute BACSCAl.

ERROR MESSAGES

The following fatal error messages are printed through subroutine MESSAGE. A value of IFATAL = 1 in /MESSA/ is also returned and BIN returns with no further action:

"BIN, TOO MANY C-VARIABLES (n)"
(if the number of C-quantities is greater than 1. The value of n gives the number of C-variables.)

"BIN MODE CII NOT PERMITTED"
(if the illegal CII mode is used)

"TOO MANY X-VARIABLES"
(if 2 or more of the parameters Fl, Al, El are set to 2)

"TOO MANY Y-VARIABLES"
(if 2 or more of the parameters F1, A1, E1 are set to 3)

"BIN - NO CONSTANT MATCH"
(if no match between the constant quantity and input table is found. This is not a fatal error since BIN continues as described in USAGE.)

"NO TERMINAL RECORD IN INPUT"
(A terminal record is recognized by the value of 999. for the first word. If this record can not be found, appropriate max., min-values are missing.)

"BIN, FATAL ERROR, NX·NY TOO LARGE, NX = nx, NY = ny"
(where nx and ny are the number of columns and rows in the output matrix in /DATA/. This message occurs if the size of /DATA/, as indicated by ISIZE, is not large enough to contain the output of BIN.)

SUBROUTINES

The major subroutine required by BIN is BACSCA1 (see PML 163). In addition the following subroutines are required (those marked with * are special purpose and are considered to be part of BIN).

ADDS *
EOFILE
ERROR1
ERROR2
IEOF

ACCURACY

Not applicable

COMMENTS ON USAGE

Although BIN has been designed for an express purpose, it can be used in more general situations where this type of binning is to be done. Normally BIN is followed immediately by a "3-D" display program such as PRINT1 (PML 170) or CRTI (PML 168).

FILE DESCRIPTIONS

BIN uses two files. One is the main input file or "table" as it is refered to in the text. The other is the "sub-table" produced by BIN which selects records from "table". Both of these files are manipulated using Record Manager calls and need not be declared in the main program.

The files are sequential with fixed length w-type records (Fortran unformatted file). For a complete description of the main input file structure refer to subroutine DEN (PML 161)

EXAMPLE OF USE

```
PROGRAM MAIN(OUTPUT,PRINT)
COMMON/FITAIN/FITAIN (35)
COMMON/FITAOUT/FITAOUT (35)
COMMON/INBUF/INBUF (514)
COMMON/OUTBUF/OUTBUF (514)
COMMON/MESSA/IFATAL,MESSA (20)
COMMON/IBUF/IBUF (100)
COMMON/DATA/DATA (6000)
COMMON/SYSTEM/ANTENNA (10), RECEIVE (10), TRANS (10)
DATA (ANTENNA = 1., 1.), (RECEIVE = 200., .5), (TRANS = 50., 1.)
CALL BIN(5.75, 2, 1, 5LFLUXT, 6000)
CALL PRINT1(100., 1., 41, 2)
END
```

In this example FLUXT is assumed to have been created (by DEN, for example). It is an example of CIA operation with frequency the C-quantity, azimuth the A-quantity and elevation the I quantity.

NAME: PRINT1: revision 0, subroutine, PML 170
CATEGORY: General purpose display
TITLE: Printer display of a function of two variables
LANGUAGE: CDC Extended Fortran
PROGRAMMER: T.B. Barrett, Parke Mathematical Laboratories, Inc.
DATE: May 1976

DESCRIPTION

PRINT1 is basically a simulator (on the line printer) for CRT intensity plots as produced by subroutine CRT1 (see PML 168). It may be used to display a general function of two variables which has been stored in suitable "matrix" form. As for CRT1, there are several modes of operation which permit various function normalizations. Two displays are produced; the first is an "intensity" display wherein function values are represented by symbols; the second displays the function as numbers in floating point notation.

USAGE

Subroutine PRINT1(DB,GAIN,NLEV,MODE)

Input parameters

Parameter type is shown in parentheses; refer to notes indicated by asterisks for details. Default values, if any, are indicated by brackets. (A default value is substituted for a given value of 0.)

DB — (floating point) [20.] intensity display "dynamic range". It functions in various ways according to MODE. *1

GAIN — (floating point) [1.] intensity display "gain". DB and GAIN interact as indicated in the notes to produce different display effects according to MODE of operation. *1

NLEV — (integer) [41] is the number of "intensity" levels to be used in the display. The maximum at present is 41. *2

MODE (integer) (mode of operation)

= 0 no special normalization, DB and GAIN are used in the "standard" manner.

= 1 GAIN is interpreted as the maximum intensity to be plotted; the minimum intensity to be plotted is DB db below this value.

= 2 GAIN has no function. Each column to be plotted is normalized (divided) by the maximum function value for that column. Intensity levels are adjusted in equal steps to DB db below the maximum.

= 3 same as for MODE = 2 except substitute "row" for "column".

notes

*1 In the "standard" mode (MODE = 0) of operation the functional value to be displayed is represented by a printed symbol which is the same for all values within a given "contour" interval. These intervals, I_i, are defined by DB, GAIN and NLEV by

$$I_i = I_{min} + (I-1) \cdot \Delta I \; ; \quad i = 1 \text{ to NLEV}$$

where

$$I_{min} = \frac{fg}{K \cdot GAIN} \quad ; \quad \Delta I = \frac{I_{max} - I_{min}}{NLEV-1}$$

$$I_{max} = \frac{fg \cdot K}{GAIN}$$

fg = geometric mean of the function f (assumed positive)

K = "contrast" = $10^{DB/20}$

For MODE = 1 operation $I_{max} \approx GAIN$ and

$$I_{min} \approx I_{max}/K$$

For MODE = 2,3 operation, the levels are determined according to:

$$I_i = I_{max}\, 10^{(DB/20)(i-NLEV)} \; ; \quad i = 1,NLEV$$

where

I_{max} = f(c or r max) (column or row max of the function f)

*2 There are at present 41 distinctive symbols available. If a function value falls below I_{min}, it is not plotted (i.e. is represented by "blank"). If a function value falls at or above I_{max} and NLEV is 41, it is represented by "#". Note that if NLEV is less than 41, function values which fall on or above I_{max} are represented by a symbol other than #.

The intensity symbols in ascending order are:

blank · - + 0 → 9 A → Z #

Labeled common input

Labeled common area /DATA/ contains the function values (z-value) to be plotted as well as values which indicate the corresponding values of x, y. Thus the function, f, should be stored in /DATA/ in the sequence:

/DATA/$x_0, \Delta x, n_x, y_0, \Delta y, n_y$, $f(x_0,y_0)$, $f(x_0+\Delta x,y_0)$,

... $f(x_0+(n_x-1)\Delta x,y_0)$, $f(x_0,y_0+\Delta y)$, $f(x_0+(n_x-1)\Delta x,y_0+\Delta y)$,

....$f(x_0+(n_x-1)\Delta x,y_0+(n_y-1)\Delta y)$

Note that the "symbol" which represents f(x,y) has local origin at relative location x,y in the print frame.

Title (in Hollerith) for the print are contained in common area /TITLES/ as follows:

ID(4)	4 words of print identification which are printed at upper left in frame.
TITLE(6)	additional identification printed beneath ID.
XTITLE(6)	x-axis title.
YTITLE(6)	y-axis title.
PTITLE(6)	"parameter" information printed below TITLE.

Following this Hollerith information the integer length (number of characters) of the titles should be included in the order NTC,NXC,NYC,NPC.

STORAGE REQUIRED

The amount of storage required by PRINT1, but not including common storage areas is 1463_8 (819_{10}).

ALGORITHM

Two arrays are used for each print line. Array TARRAY is used to hold a letter of the y-axis title (if necessary), possibly a letter of scaling information and a numerical scale value. Array PARRAY is used to contain the various intensity symbols.

The x and y axis numerical scale values are scaled so that the arithmetic mean of these values is in the form $\pm$ xx.xx using subroutine SCALF.

PRINT1 makes two passes through the print out loop. In the first pass intensity values are represented by symbols and on the second pass by actual numerical values.

SPECIAL CAUTION AND FEATURES

Note that XMIN, DELX, YMIN and DELY are used to determine scale values. These numbers should be chosen so that the scale numbers can be adequately represented in the form shown above. (In other words don't use too many digits.) (XMIN = x_0, DELX = Δ x, etc.)

TIMING

None available. In any case should be quite small.

ERROR MESSAGES

None

SUBROUTINES

FCONTR
SCALF

ACCURACY

Not applicable

COMMENTS ON USAGE

PRINT1 can profitably be used as a simulator for CRT1 (see PML 168). In addition it has the value of being able to print out actual numerical function values which CRT1 can not do.

FILE DESCRIPTIONS

Formatted file PRINT contains the output of PRINT1. The file should be rewound and copied to OUTPUT after PRINT1 has executed.

EXAMPLE OF USE

The following example assumes that subroutine BIN (see PML 169) is used to generate the function to be displayed thus some of the common declarations are made for this subroutine.

```
      PROGRAM MAIN (OUTPUT,PRINT)
      COMMON/FITAIN/FITAIN(35)
      COMMON/INBUF/INBUF(514)
      COMMON/MESSA/IFATAL,MESSA(20)
      COMMON/IBUF/IBUF(100)
      COMMON/DATA/DATA(6000)
      COMMON/SYSTEM/ANTENNA(10), RECEIVE(10), TRANS(10)
      DATA (ANTENNA=1.,1.), (RECEIVE=200.,.5), (TRANS=50.,10000.)
      CALL BIN(5.00,2,1,5LFLUXT,6000)
      IF (IFATAL.EQ.1) GOTO 1000
      CALL PRINT1(200.,1.,41,0)
1000  STOP
      END
```

An example of the "intensity" display output by PRINT1 is shown in Figure 1.

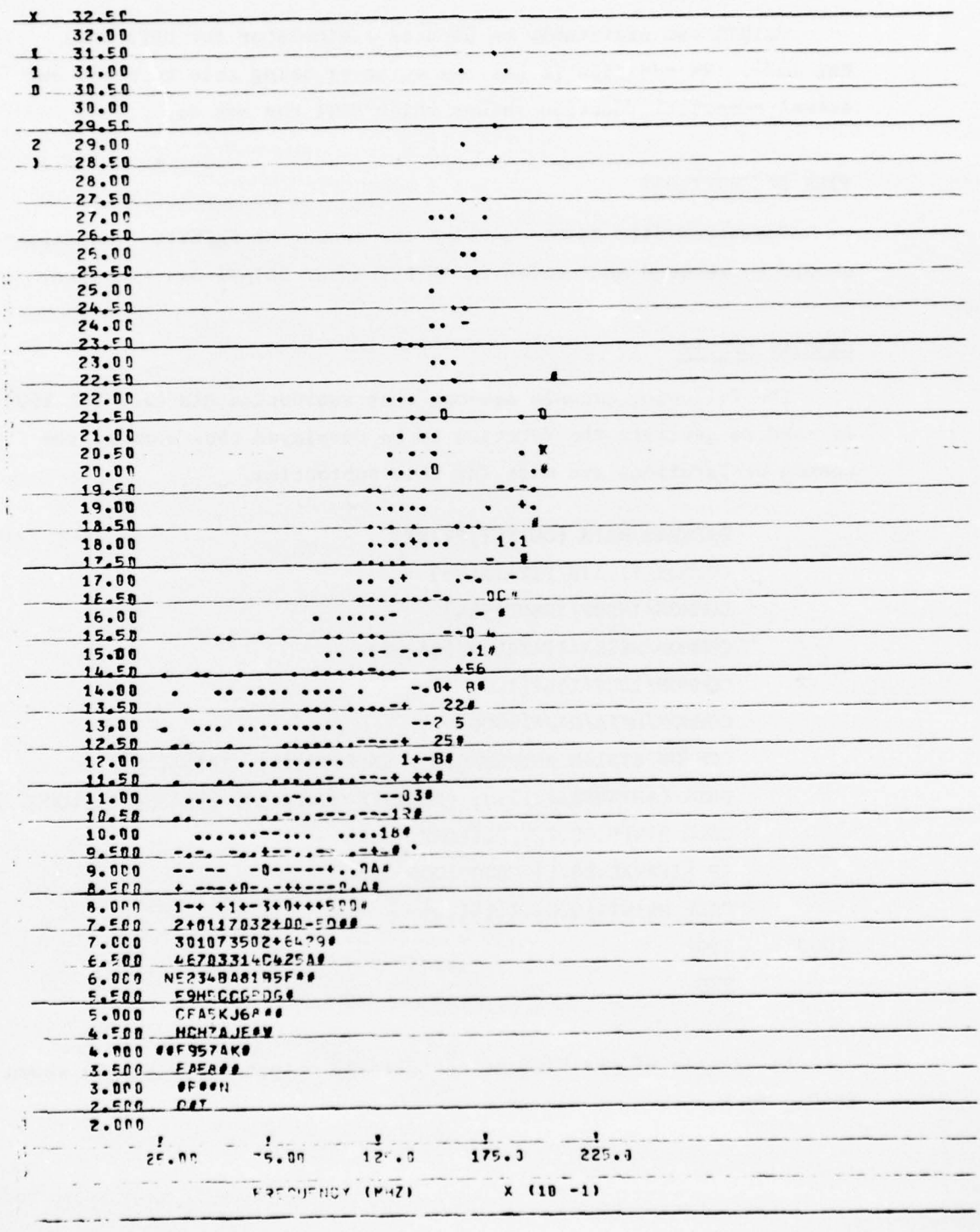


Figure 1 Coded Print Ionogram

U. S. GOVERNMENT PRINTING OFFICE: 1977--700-002--71