DOE/ID/12079-31
ESL 55

MT2D: An Interactive 2 Dimensional Magnetotelluric and Line Source Modeling Program (User's Guide and Documentation for Rev. 3)

by
Carleen Nutter

EARTH SCIENCE LABORATORY
UNIVERSITY OF UTAH RESEARCH INSTITUTE
420 Chipeta Way
Suite 120
Salt Lake City, Utah 84108

April 1981

Prepared for the
Department of Energy
Division of Geothermal Energy
Under Contract DE-AC07-80ID12079

DISCLAIMER

NOTICE

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

NOTICE

Reference to a company or product name does not imply approval or recommendation of the product by the University of Utah Research Institute or the U.S. Department of Energy to the exclusion of others that may be suitable.

Table of Contents

ABSTRACT

MT2D.REV3 is the latest revision of a 2-dimensional, finite-element, interactive MT-line source modeling program. The original program was a batch-mode program developed by John Stodt at the University of Utah, Department of Geology and Geophysics. An interactive program was developed based on Stodt's program for a UNIVAC 1108 at the University of Utah Computer Center.

The program uses linear interpolation of the unknown field over triangular sub-domains of the region where a solution is sought in conjunction with the Galerkin technique to derive a system of linear equations which approximate the governing partial differential equation. The solution of this linear system of equations gives the approximate field values at the nodes of the discretized domain.

MT2D has an interactive data management system for data manipulation and display built around the finite-element program.

This revision of the interactive program was developed for a PRIME 400 at the University of Utah Research Institute, Earth Science Laboratory. The revisions are centered around the I/O handling routines to make them more comfortable and convenient to use. A pseudosection for multiple frequencies has been added.

The specific changes and additions are detailed in the text of the user's guide.

This report is a user's guide and documentation for MT2D.REV3.

INTRODUCTION

MT2D.REV3 is the newest revision of the 2D interactive, finite-element, MT-Line Source modeling program. The program uses linear interpolation of the unknown field over triangular sub-domains of the region where a solution is sought in conjunction with the Galerkin technique to derive a system of linear equations which approximate the governing partial differential equation. The solution of this linear system of equations gives the approximate field values at the nodes of the discretized domain. Stodt (1978) contains a description of the theory and mathematics of this finite-element program.

This program is operational on the University of Utah Research Institute's (UURI) PRIME 400 computer operating under PRIMOS IV, Rev 17.

MT2D is a modification of a program originally written by Luis Rijo and later modified by John Stodt while they were students at the University of Utah, Department of Geology and Geophysics.

This user's guide documents the interactive interface written around the finite-element modeling program. It describes the input parameters and the textual and graphic output. It also describes the data manipulation functions such as an internal editor and an internal file utility for handling the work file and the catalog of work files, called the merge file. The user is referred to Stodt (1978) for descriptions of the finite-element method as applied to magnetotellurics and line source.

USER'S GUIDE

MASTER

MT2D is a menu-driven program. The user has a variety of available options that allow him to perform necessary I/O, data manipulation and computation to produce a model.

By selecting the appropriate options and responding to the prompts correctly, the user can work through the modeling process.

The program makes use of two direct-access files. One, the "work" file, contains the current model. Most of the major functions interact with the work file for storing and retrieving data. Two of the major functions work from the

second direct-access file, called the "merge" file. The merge file is a collection of work files. The FILE MAINTENANCE function and the PHANTOM EXECUTE function use the merge file. The FILE MAINTENANCE also uses the work file for some of its options.

The user enters the program at the operating system level by typing

MT2D

The program responds by typing the name of the program and the revision # at the user terminal. The user then encounters the first option query as follows:

TERMINL- OPTION:

The user enters the appropriate option number to continue with the program. If the user doesn't know or can't remember the available options and their corresponding option numbers, he can enter a zero or carriage return, the program will display the options list at the terminal, and again request an option number.

The options list for TERMINL- is

1 HARD COPY 80 COLUMN
2 TEKTRONIX 4014
3 CRT 80 COLUMN

The user enters the option number corresponding to the type of terminal being used. If option #1 is chosen, the program does no paging during the LIST function. Option #2 generates Tektronix TCS commands for erasing the terminal screen.

The next option request is for the controlling option list called MASTER.

MASTER-- OPTION:

The available options under MASTER are

1 EXIT FROM SYSTEM
2 INPUT
3 EDIT
4 LIST
5 EXECUTE
6 FILE MAINTENANCE
7 PLOT
8 PHANTOM EXECUTE

The MASTER options are discussed separately in the following sections.

Options are nested, with TERMINL and MASTER at level 1. Lower-level options are indented from the next higher one. Option #1 will always return the user to the next higher options list unless the current options list is for device selection. Option #1 in MASTER causes the program to terminate.

A zero input to the options query will always result in a listing of the options available.

Data are read from the work file (called MT2D-WORK.3) upon entering one of the major functions. Data are rewritten to the work file upon exiting one of the major functions. If editing is done, the work file itself is not actually updated until the editing function is terminated by choosing EDIT option #1. The exception to this is the FILE MAINTENANCE function. This function immediately acts on the work file and/or merge file (called MT2D-MERGE.3).

INPUT function

The following is a description of the needed input parameters for the finite-element modeling subroutines. The program limits for the parameters are listed also. Input is free format unless stated otherwise.

```
Maximum # of X nodes = 90
        "     Z nodes = 30
        "     element groups = 80
        "     frequencies = 10
        "     boundary layers = 10
```

PN	- 60 char. title (owner-project)
AN	- 60 char. title of model
IDX	- mode= 1 for line source 2 for MT - TE 3 for MT - TM
NFREQ	- no. of frequencies
FREQ(i)	- NFREQ frequencies (in HZ)
NODEX	- no. of X nodes in the mesh
NXX	- no. of X-element groups
NX(i)	- NXX X-element group #'s
DELX(i)	- NXX X-element group spacings (in meters)
NODEZ	- no. of Z nodes in the mesh
NZZ	- no. of Z-element groups
NZ(i)	- NZZ Z-element group #'s
DELZ(i)	- NZZ Z-element group spacings (in meters)
LINE1, LINE2	- 1st and 2nd line source positions (in meters)

Enter this only if IDX=1.
If no 2nd line source, enter LINE2 as zero.

IS THIS A FLAT EARTH? - "YES" or "NO"

If "YES" - enter MM(1) = no. of air layers on the left edge; this value is extended to MM(2) and IAEZ(i).

If "NO" - enter IAEZ(i), i=1,NODEX where IAEZ(i) is the Z-node of the air-earth interface for the ith X-node. MM(1) is set to the value of IAEZ(1), and MM(2) is set to IAEZ(NODEX). (Note that IAEZ is 1 greater than the number of air layers at that point.)

NPSUDO(i) - select the X-nodes to be used in the pseudosection (for MT only). Enter "N" or "Y".

NRES - no. of resistivities (Maximum number of resistivities is 36.)

Y(i) - resistivities (note that the resistivity of air is not infinity but 10**37, in ohm-meters).

If IDX=1, skip the next five inputs

NLR(1), NLR(2) - no. of layers for boundary conditions on the left and right. There must be at least 2 layers on each side. The resistivities of adjoining layers may be the same. The number of layers on the left and right sides do not have to be equal.

H(i,1) - depth to layer interfaces on the left boundary (in meters)

P(i,1) - associated resistivities (zero is an invalid resistivity), in ohm-meters.

H(i,2) - depth to layer interfaces on the right boundary (in meters)

P(i,2) - associated resistivites (in ohm-meters)

NPT's - alphanumeric codes for the finite-element mesh. Code "1" is for the first resistivity, code "2" for the second, ...code "0" for the tenth, code "A" for the 11th, ... code "Z" for the 36th (in 79A1 format).

A sub-portion of the total mesh can be entered by specifying the X- and Z- nodes of the corners of the area. The mesh is then extended to the top and bottom and left and right using

the values of the outer perimeter of the sub-mesh specified. The individual triangular elements can be entered separately or the larger square elements can be entered (giving the same value to the four triangular elements making up the square element). The program checks for input values within the specified range of resistivities. For example, if NRES=3, the only valid mesh codes are 1,2 and 3. If an invalid code is given, the program requests the information again.

EDIT function

MT2D has an internal editor to provide limited editing capability under program control. Any input parameter can be edited. The new parameters are not copied to the work file until EDIT option # 1 (return to MASTER) is executed.

All editing is done via the CHANGE subroutine except EDIT option # 4 (edit entire mesh) where a new mesh is generated by specifiying the start and end X- and Z- nodes of the sub-mesh (as in the INPUT function). Editing is done on groups of related parameters. Input values of edited parameters are checked for validity.

Below is a list of EDIT options. Selecting EDIT option #7 results in another level of option request.

```
EDIT---- OPTION:
    1  RETURN TO MASTER OPTIONS
    2  EDIT MESH CODE ELEMENTS
    3  EDIT MESH CODE ROWS
    4  EDIT ENTIRE MESH CODE
    5  EDIT # OF MEDIA CODES
    6  EDIT MEDIA RESISTIVITY
    7  EDIT MESH GEOMETRY
    8  EDIT TITLES
    9  EDIT MODE,L.S. POSITIONS
    0  EDIT FREQUENCIES
    +  EDIT EDGE DEPTH,RES.
    *  EDIT PSUDO-SEC.NODES(MT)

EMGEO--- OPTION:
    0  RETURN TO EDITOR
    1  EDIT # OF X NODES
    2  EDIT # OF X ELEMENT GROUPS
    3  EDIT X ELEMENT GROUP #S
    4  EDIT X GROUP SPACINGS
    5  EDIT # OF Z NODES
    6  EDIT # OF Z ELEMENT GROUPS
    7  EDIT Z ELEMENT GROUP #S
    8  EDIT Z GROUP SPACINGS
    9  EDIT AIR-EARTH INTERFACES
```

The editor works by allowing the user to enter a new value (in free format) from the keyboard. If changing arrays, the user must specify the index of the array element to be changed and then enter the new element. To change all values of the array, the user enters a "-1" and the program will request a new value for each element without the user specifying all the indices.

After an element of an array has been changed, the program requests another index (for the same array). In this way, the user can change multiple elements of an array with a minimum amount of typing. When the array is corrected, a "0" input for the index will return the program to the options list.

LIST function

Listings of the input parameters and results of execution can be obtained on the terminal or line printer. Terminal listings are formatted to 80 columns and paging is done so that information does not scroll off the screen before the user can read it. A carriage return will allow printing to continue. The user is prompted for this carriage return when it is necessary.

The LIST options are given below.

```
LIST---- OPTION:
    1 RETURN TO MASTER OPTIONS
    2 LIST TITLES,MODE,FREQ
    3 LIST GEOMETRY PARAMETERS
    4 LIST MEDIA CHARACTRISTCS
    5 LIST MESH CODE TRI. ELE.
    6 LIST MESH CODE SQR. ELE.
    7 LIST RESULTS
    8 LIST PSEUDO-SEC(MT-ONLY)
    9 SEND WORK FILE TO PRNTR
```

Options 2 thru 8 print at the terminal. Option 9 prints at the on-site printer.

Option 7 contains the following information for each frequency:

for MT

1) distance from the center of mesh to each node,
2-5) vertical field components (real, imaginary, magnitude and phase),
6-9) field components perpendicular to strike

(real, imaginary, magnitude and phase),
10-13) field components parallel to strike (real, imaginary, magnitude and phase),
14) apparent resistivity,
15) negative phase of the impedance.

for LINE SOURCE

1) horizontal distance from the + line source to the nodes at z=0,
2-5) transverse magnetic field components (real, imaginary, magnitude and phase),
6-9) vertical magnetic field components (real, imaginary, magnitude and phase),
10) ratio of magnitude of transverse field to primary field,
11) ratio of magnitude of vertical field to primary field,
12) primary field.

Option 8 generates two pseudosections for MT and none for Line Source. The two MT pseudosections are for apparent resistivity and negative phase. The user can determine which of the two (or both) he wants.

Line printer listings can be in a long or short format. The user is queried as to which he wants. The long format prints the information from LIST option #'s 2,3,4,6,7 and 8 (both pseudosections). The short format prints the information from LIST option #'s 2,4,8 (both pseudosections) and partial information from #3. The information in the short format is organized in a more concise manner and gives a better visual representation between the mesh and the pseudosection output. The mesh code numbers for only the columns of the mesh at the x-nodes chosen for the pseudosection are printed. The x-node number and distance from the center of the mesh are printed at the top of the partial mesh. The heights above the air-earth interface and the depths below are printed to the side of the mesh. The pseudosection information is aligned with the mesh. The format for mesh and pseudosections allows only 15 values across a page. If more than 15 nodes are requested for the pseudosections, the mesh and pseudosections are broken up into segments 15 values wide. The sample terminal session has a sample of this output format.

EXECUTE function

This function executes the finite-element program using the parameters in the work file. The results are copied into the work file. The only terminal output generated during

execution of this function is a message about which frequency is being processed. The user must execute MASTER option #4 (LIST) to see the results.

FILE MAINTENANCE function

This option allows the user to save his work file on a merge file (a catalog of work files). This option contains utilities for manipulating the merge file. The merge file should be initialized (with option #7) before attempting to copy the first work file.

Below is a list of the FILE MAINTENANCE options.

```
FMAIN--- OPTION:
   1 RETURN TO MASTER PROGRAM
   2 LIST MERGED FILE DIRECTORY
   3 SAVE WORK FILE ON MERGED FILE
   4 RESTORE WORK FILE FROM MERGED FILE
   5 DELETE AND PACK MERGED FILE
   6 SYSTEM PACK MERGE FILE
   7 INITIALIZE MERGE FILE
   8 SEND MERGE FILE CONTENTS TO PRINTER
```

Option #5 deletes and packs the merge file but does not reduce the physical size of the file. Option #6 does reduce the physical size of the file, thereby reducing the amount of disc space needed to store it. Option #8 requests the user to select the subfiles of the merge file to be printed. These are selected by typing "Y" (do print) or "N" (don't print) under the subfile #. The user can select long or short format as described in LIST for option #9.

PLOT function

The plot options are listed below.

```
PLOT--- OPTION:
   1 RETURN TO MASTER OPTIONS
   2 PLOT CURRENT MODEL
   3 PLOT MULTIPLE SAVED MODELS
```

Option #2 plots the model in the work file. Option #3 produces plots from selected subfiles in the merge file.

The PLOT option produces a device-selection option request to determine which plotting device is to be used. The current available devices are

DEVICE-- OPTION:
1 TEKTRONIX GRAPHICS PLOT
2 STATOS PLOT
3 PRINTER (DEFAULTS TO #2)
4 ZETA PEN PLOTTER
5 UUCC CALCOMP PLOT

Statos, Zeta and Calcomp plots are generated but not plotted until the user terminates the program and types the proper PRIMOS-level command.

Two plots can be generated. 1) A plot of the cross section of the model (or some sub-portion) with or without an overlaying plot of the mesh. The user specifies the page size of this plot in inches. For MT models, the cross section is centered on the page. For Line Source models, the cross section is plotted on the page in relation to the position of the first line source (plotted at the center of the plot). 2) A plot of two graphs of data is generated. For MT these graphs are apparent resistivity versus distance and negative phase versus distance. The apparent resistivity is plotted on a log(10) scale. For Line Source the graphs are transverse magnetic field components divided by the primary field versus distance and vertical magnetic field components divided by the primary field versus distance. The user enters the page size for this plot in inches and also selects the frequencies for which he wants graphs. All frequencies are plotted on the same graph.

If the user wants to compare the cross section/mesh with the graphs (and preserve relative distances), he should enter the same X-dimension on the page size for both plots and plot the cross section from nodes 2 to (NODEX-1).

Multiple plots are all done on the same page size and plotting device specified by the user. Different sub-portions of cross sections and different frequencies may be chosen for each model.

Both plots are labeled with the two titles from the input parameters, the mode and the units of distance. The units of distance are M(meters), KM(kilometers), and XX(unknown). (Remember that all distances should be entered in meters.)

The sample terminal session contains examples of the plots produced by the PLOT function.

PHANTOM EXECUTE function

Execution of single or multiple models can be done in a batch (background) environment.

Set up the models to be executed in the merge file. (Enter model into work file, edit if necessary, and copy to merge

file.)

When MASTER option #8 is selected, the user selects the desired merge sub-files to be executed. This option then causes the program to terminate. The user cannot access the merge files while the phantom is executing. The user can delay the start of the phantom by using the WAIT command and entering the date and time for the phantom to begin. A query in MT2D will request this information. The user can then access the work/merge files until the phantom starts.

The selected merge subfiles will be executed and the results copied into the appropriate subfile on the merge file for future reference. A listing will be sent to the printer showing the model and the results of each subfile executed. A COMO file will be generated, residing in the current UFD, called MT2D.COMO. This file can be examined for error messages if the phantom job terminates abnormally. Another file will be generated, residing in the current UFD, called PH_MT2D. This is the phantom command file. This command file is created and executed but not deleted by the PHANTOM EXECUTE function.

NOTES ON PROPER MESH DESIGN

Proper mesh design is important in obtaining meaningful results.

Mesh design is based on the unit of distance called "skin depth", defined as the distance in which the amplitude of a plane wave is attenuated by 1/e as it propogates through a homogeneous conducting medium.

$$\text{skin depth} = \delta = 500\sqrt{\frac{\rho}{f}} \quad \text{meters}$$

where

ρ = resistivity of the medium
f = frequency (HZ)

The following rules of thumb will serve as a guideline in designing meshes. With experience, the user will learn how to manipulate and bend these rules to create smaller but still accurate meshes.

1. Anything deeper than several skin depths cannot be resolved.
2. Anything further away laterally than about four skin depths has no influence.
3. Element dimensions should not change from one

element to the next by more than a factor of 3 to 5.

4. In the vicinity of a change in resistivity of the medium, the element dimensions should be approximately 1/6 of a skin depth in the medium where the element resides.

5. Two to three skin depths away from any variation in resistivity, the element dimension may be increased to the order of a skin depth of the medium.

6. Vertical element dimensions may be increased approximately logarithmically from the air-earth interface because of the exponential decay of the fields. Ideally however, the maximum vertical element should still be held to 1 to 2 skin depths.

7. The air layer for the TE-MT and line source problems should consist of 7 or 8 elements logarithmically increasing in vertical dimension from the air-earth interface, starting with about 10-100 m for frequencies < 1 HZ and 1-10 m for frequencies > 1 HZ.

8. A 1 or 2 node air layer is required computationally in this program (not theoretically) for the TM-MT case. A 3 m and 10 m layer gives good results.

9. Vertical mesh boundaries should ideally extend 3 to 6 skin depths away from the nearest 2-D structure.

10. The bottom mesh boundary should ideally be four to six skin depths of the background resistivity from the air-earth interface.

11. In solving the line source problem, the mesh should be fine in the region of the source. The mesh boundaries should extend to where the fields due to the source are approximately zero.

12. At low frequencies, a very high aspect ratio of the x and z dimensions of a mesh element can give incorrect results at the boundary edges. Making the mesh elements more square can improve the results.

13. At low frequencies, when the fields are not changing too fast, a coarser mesh can give better results.

14. In processing multiple frequencies for the same mesh design, the mesh design should conform to the rules of thumb for skin depths calculated from all frequencies being processed. Meshes designed for high frequencies (above 1 HZ) are not properly designed for low frequencies (below 1 HZ). Generally, the mesh is designed for the highest frequency of the group of frequencies to be executed and the rules of thumb checked for the lowest frequency. If the mesh is okay for the highest and lowest frequencies, it should be okay for the intervening ones.

15. A group should not span the air-earth interface.

DIRECT-ACCESS FILE STRUCTURE

This program uses two direct-access files for data storage. These direct-access files have fixed-length records. The record length is defined in the $INSERT file as PARAMETER IWIDTH.

DEFILE is the subroutine that defines the direct-access file.

MT2D-WORK.3 is the name of the direct-access file that contains the current model parameters and the calculated values. Most options work on data from this file and store results into this file. The exceptions are MASTER Option #6 (File Maintenance) which provides a number of utilities that access the second direct-access file called MT2D-MERGE.3, the PLOT Option #3 which plots models and curves from selected files stored on the merge file, and the PHANTOM EXECUTE.

MT2D-MERGE.3 is essentially a catalog of different work files. MT2D-WORK.3 can be stored on the merge file. The work files on the merge file can be moved back into MT2D-WORK.3 for manipulation with the program options. Rev. 3 is of similiar design and structure as Rev.2 but with some changes in input parameters and output formats. The work and merge files are not compatible between the two revisions. Rev.3 files have a ".3" appended to the names of the files so users will not overwrite Rev.2 files.

The direct-access work file has 90 words per record and is of variable length. The variables that are written to the work file are stored in COMMON blocks defined in the $INSERT file. An entire record is written to the work file with a single write statement by specifying the beginning variable in the COMMON block associated with that record irregardless of mixed type within the COMMON block. The COMMON blocks are filled out with dummy arrays to insure correct record length.

See Appendix I for details on PRIME's direct-access I/O.

SYSTEM IMPLEMENTATION

The PRIME 400 is a virtual memory machine. Some attempt was made to optimize paging for this program.

MT2D uses 640K words of virtual memory (16 bits/word). Most floating point and integer numbers are 32 bits. Most system subroutines require short integer arguments (INTEGER*2). PRIME FORTRAN has two functions, INTL and INTS, that convert short integers to long (INTEGER*4) and long integers to short. Some complex arithmetic is done in double precision to increase the accuracy of repeated calculations. (The PRIME does not have a DOUBLE PRECISION COMPLEX type.) A machine with a larger word size than 32 may get by with

single precision.

The large size of the program is primarily a function of the size of the global matrix, although only part of this matrix is actually stored (because the matrix is symmetrical and banded).

The amount of storage required by this program can be reduced considerably by decreasing the maximum allowable number of x- and z-nodes. The values need only be changed in the $INSERT file. (See $INSERT and PARAMETER statement descriptions in Appendix I.) Changing the width of the direct-access file records will make previously created direct-access files unusable. The $INSERT file contains some notes on changing these PARAMETER variables.

The entire code in the $INSERT file could be substituted for the $INSERT MT2D.COM statements throughout the program if no equivalent mechanism exists at the site installing the program.

The program makes use of direct-access files. See the previous section and Appendix I for more details.

The ENCODE/DECODE statements are used in the program to read and write data between memory locations. Appendix I describes ENCODE/DECODE statements.

All subroutines are compiled using PRIME FORTRAN's -BIG option to allow large arrays to span segment boundaries (1 segment=64K words). Long integer is the default on ESL's FOR compiler.

MT2D makes use of a device-independent plotting software package resident on the system for plotting on various graphic devices. This software library, UUPLT, is available from UURI/ESL. It uses Tektronix PLOT10 software to interface to the resident Tektronix 4014 graphics terminal. PLOT10 is available from Tektronix. It uses Varian Graphic's DATAPLOT III software to interface to the resident Varian STATOS electrostatic plotter. Nicolet ZETA software is interfaced to UUPLT to provide access to the pen plotter. MT2D generates plot files that can be read by the University of Utah Computer Center's (UU/CC) CalComp plotting system. The CalComp plot files are written to magnetic tape and transferred to UU/CC for actual plotting.

The plotting subroutines in MT2D are very modular and could easily be replaced to conform to the plotting capabilities of another installation or be removed entirely.

The program uses four logical units for I/O:

LU 1 - terminal read/write

LU 8 - assigned to the work file
LU 9 - assigned to the merge file
LU 11 - assigned to temporary print files.

Logical unit numbers are converted to PRIME unit numbers for use by system subroutines by subtracting 4.

MT2D makes use of some system subroutines. See Appendix II for a list and description. Integer arguments for the system subroutines are short integer unless otherwise stated. An equivalent system routine is probably available at any site. PRIME manual SUBROUTINES, P/N MAN3251-001 contains more detail if needed.

REFERENCES

Stodt, J. A., 1978, Documentation of a finite element program for solution of geophysical problems governed by the inhomogeneous 2-D scalar Helmholtz equation: University of Utah technical report, NSF grant AER 76-11155.

Nutter, Carleen, 1979, An interactive modeling system for 2-D magneto-telluric and line source resistivity data (MT2D user's guide and documentation): University of Utah, DOE topical report.

APPENDIX I

PRIME FORTRAN EXTENSIONS

The following extensions are a part of PRIME's FORTRAN IV library. Additional information on the use of these extensions can be found in the PRIME system manual titled "The FORTRAN Programmers Guide", PDR3057, dated November 1977. PRIME Computer Inc. has given permission to reproduce the descriptions that follow.

CODING statements

For the 3 coding statements, the following variable descriptions apply.

c	Number of ASCII characters to be transferred.
f	Format statement label.
a	Array name.
sn	The statement number to which control is transferred if a FORMAT/DATA mismatch occurs.
list	I/O list of elements (same as in READ or WRITE statement)

Formatted DECODE statement

```
DECODE(c,f,a[,ERR=sn]) list
```

converts the first c characters in the array, a, from ASCII data into the I/O list elements according to the specified format, f. If the optional error branch is inserted, a FORMAT/DATA mismatch will cause a transfer to the statement labelled sn.

List-directed DECODE statement

```
DECODE(c,*,a[,ERR=sn]) list
```

allows the user to input/decode data from free-format input devices such as the user terminal. The requirements on input

and delimiters are the same as for the list-directed READ statement.

ENCODE statement

ENCODE(c,f,a) list

converts the elements of the I/O list into ASCII data according to format, f, and stores the first c characters of the resultant string into array a.

List-directed READ statement

READ(u,*[,END=a][,ERR=b]) list

where u is the FORTRAN read unit and * is the symbol indicating that this is a free-format read. The a and b are the statement numbers to which control is passed if an end of file, or a read error is encountered.

List-directed READ frees the programmer from including format statements for READs from free-format devices such as the user terminal. The input data is converted according to the data type of items in the I/O list. Additionally, this feature provides a method to indicate in the input data that an item in the I/O list is to remain unchanged by the READ statement.

Delimiters: Values in list-directed input are separated by a blank, comma, or slash. A slash or comma may be preceded and followed by any number of blanks. An end of record is treated as a blank. A slash terminates a READ and leaves the remaining items in the I/O list unchanged. Two adjacent commas with no intervening characters except blanks will leave the corresponding item in the I/O list unchanged. A list-directed READ will read any number of records until a slash is encountered or until all items in the I/O list have been satisfied.

Numerical Input: If an item in the I/O list is a long or short integer variable or array element, the corresponding input field must contain a string of decimal digits optionally preceded by a + or - sign, as in

```
-357      100514      +12387
```

If a real or double precision item is in the I/O list, the corresponding input field must contain a string of decimal digits with an optionally embedded decimal point. An exponent field may follow either an E or D format, as in:

```
51      -27.68      7.65E-14      863D2
     .503      +265.
```

The input field corresponding to a complex item must contain two real numbers (as described above) separated by a comma and enclosed in parentheses, as in

```
(1E2,-2.)      (5.67E-6,8.09)
```

Character String Input: A variable or array of any type can be set equal to a character string using list-directed READ. A character string must be enclosed in single quotation marks in the input data. Within a character string, a quotation mark is represented by two consecutive single quotation marks. A character string, regardless of length, matches a single item in the I/O list whether it is a variable, array element, or whole array (represented by including the unsubscripted array name in the I/O list). If the character string is shorter than the list item, the rightmost characters of the list item are blank-filled. If the character string is longer than the list item, the rightmost characters of the character string are ignored.

Direct-access READ and WRITE statements

The FORTRAN compiler and run-time library support direct-access READ and WRITE statements. READ and WRITE statements may contain a record number to randomly access file records. With sequential access, record n-1 must be read or written before record n. The syntax implemented is compatible with both IBM FORTRAN and the new ANSI standard FORTRAN.

Usage: Special action is required by the user when creating and opening files to be used for direct-access I/O. Files used for direct-access I/O should be DAM files (direct-access I/O statements may be used with SAM files but execution time will be longer). If the file is formatted, the ATTDEV

subroutine must be called so that fixed-length records are written (the ATTDEV subroutine is also used to set the record length). DAM files are created by opening a new file using the K$NDAM subkey in either a SRCH$$ or TSRC$$ call (see Reference Guide, PRIMOS Subroutines for details).

The ATTDEV subroutine may be used to alter the mapping of FORTRAN units to file system units. The records of a direct-access formatted file must be fixed length. This is done by setting the second argument of ATTDEV to 8. The records of an unformatted file are fixed length by default.

A program that creates a direct-access file cannot write record n before record n-1 has been written. A separate program should be used. Once the file has been created, it can be read or written in random order.

After a direct-access I/O statement, the file is positioned at the record following the one just transferred. If the direct-access file is then accessed sequentially, using other forms of the READ or WRITE statement, it is not necessary to include the record number. This enhances performance by eliminating the positioning call.

IBM compatibility: The READ and WRITE statements are identical to IBM FORTRAN. The DEFINE FILE and FIND statements of IBM FORTRAN are not supported. The record size in the DEFINE FILE statement must appear in the ATTDEV call. The record size in the DEFINE FILE statement is measured in bytes or 32-bit words rather than 16-bit words required by ATTDEV. If the U specified is used in the DEFINE FILE statement, the record size of the DEFINE FILE statement should be doubled for the ATTDEV call; otherwise the record size should be halved.

The ATTDEV call requires INTEGER*2 arguments. If the INTL option is used during compilation, constants used as arguments in the ATTDEV calls must be converted to INTEGER*2 by the INTS function (e.g., INTS(8)).

There is no equivalent of the DEFINE FILE associated variable in Prime's implementation of direct-access files. In IBM FORTRAN, the value of the associated variable is the number of the record that follows the record just transferred.

Direct-access READ statements

```
READ(u'r[,f][,ERR=b]) list          IBM format
READ(u[,f],REC=r[,ERR=b]) list      ANSI format
```

u — A long or short integer constant or variable whose value is the FORTRAN unit number.

r — The long or short integer expression whose value is the record number to be accessed.

f — The statement number of the format specified (optional).

b — The statement number to which control is transferred if a device or format error is encountered during transfer (optional).

The apostrophe (') is required in the IBM form of direct-access READ and WRITE statements.

The END=specifier is not allowed in the direct-access READ statement. This restriction is consistent with both IBM FORTRAN and the new ANSI standard FORTRAN.

Direct-access WRITE statements

```
WRITE(u'r[,f][,ERR=b]) list        IBM format
WRITE(u[,f],REC=r[,ERR=b]) list    ANSI format
```

Parameters are the same as those described for the READ statement.

NOBIG / BIG

-BIG (a parameter on the compilation statement)

BIG treats all dummy arrays as arrays that span segment boundaries. BIG forces the 64V mode and thus cannot be used in the 32R or 64R modes. If a dummy argument array may become associated with an array spanning a segment boundary

(through a subroutine call or function reference), the compiler must be made aware of this by including BIG in the parameter list. The code generated will work whether or not the array actually spans segment boundaries.

PARAMETER statement

```
PARAMETER(V1=C1,...,Vn=Cn)
```

The Vs are variables (arrays are not allowed) and the Cs are constants or constant expressions of the same mode as the corresponding variables. The operands in the constant expressions may be constants or previously defined parameters. Allowed operations include +, -, *, and / on INTEGER*2, REAL*8, and REAL*4 operands. INTEGER*2 XOR, OR, AND, MOD, shift, and truncate function references are also allowed. An error message, ILL. CONSTANT EXPR., is generated if these restrictions are violated. The variable names must be typed explicitly prior to the PARAMETER statement or default-typed implicitly. All other uses of the PARAMETER names must follow the PARAMETER statement. PARAMETER names may be used wherever a constant would be used (including DATA and DIMENSION statements) except in FORMAT statements. Since the parameters are named constants, PARAMETER names may not be used in COMMON or EQUIVALENCE statements.

Enclosing the parameter list in parentheses is required by the FORTRAN 77 standard. Prime's FORTRAN will accept a PARAMETER statement with or without the parentheses.

$INSERT statement

```
$INSERT insert-file
```

Insert into the program, at compilation time, the file whose pathname is insert-file. The $INSERT command should not be nested; do not include a $INSERT command in a file which will be inserted into a program by a $INSERT command.

$INSERT is used for:

- Insertion of COMMON specification into programs,
- Commonly used one-line functions,

- Data initialization statements,
- Parameter definitions, especially for the file management system, applications library, MIDAS, etc.

APPENDIX II

PRIMOS SUBROUTINES

The following subroutines are a part of the PRIMOS subroutine library. Additional information on the use of PRIMOS subroutines can be found in the PRIME system manual titled "PRIMOS Subroutines Reference Guide", PDR3621, dated March 1979. PRIME Computer Inc. has given permission to reproduce the descriptions that follow.

ACCPR

ACCPR is a local accounting routine for the printer.

CNAM$$

CNAM$$ changes the name of a file in the current UFD.

CALL CNAM$$(oldnam, oldlen, newnam, newlen, code)

oldnam	The name of the file to be changes.
oldlen	The length in characters of oldnam.
newnam	The new name of the file.
newlen	The length in characters of newnam. code An integer variable set to the return code.

The user must be the owner of the UFD of the file to change the name. CNAM$$ does not change the last modified date-time of the file or any of the other attributes of the file. However, the last modified date-time of the UFD in which the file resides is changed. CNAM$$ may cause the position of the file in the UFD to change with respect to the other files if the new name is longer than the old name. It is illegal to change the name of the MFD, BOOT, BADSPT. A NO RIGHT error message is generated if this is attempted.

ERRPR$

ERRPR$ interprets a return code and, if it is non-zero, prints a standard message associated with the error return code, followed by optional user text.

CALL ERRPR$(key,code,text,txtlen,filnam,namlen)

key — An integer specifying the action to take subsequent to printing the message. Possible values are:

- K$NRTN — Exit to the system, never return to the calling program.
- K$SRTN — Exit to the system, return to the calling program following an 'S' command.
- K$IRTN — Return immediately to the calling program.

code — An integer variable containing the return code from the routine that generated the error. If code is 0, ERRPR$ always returns immediately to the calling program and prints nothing.

text — A message to be printed following the standard error message. Text is omitted by specifying both text and txtlen as 0.

txtlen — The length in characters of text.

filnam — The name of the program or subsystem detecting or reporting the error. Filnam is omitted by specifying both filnam and namlen as 0.

namlen	The length in characters of filnam.

EXIT

The EXIT subroutine provides a way to return from a user program to PRIMOS; it prints 'OK,' (or 'OK:') at the terminal, and PRIMOS awaits a user command.

CALL EXIT

The user may open or close files or switch directories and restart a FORTRAN program at the next statement by typing 'S' (i.e., START).

EXST$A

EXST$A is a logical function which returns .TRUE. if the file exists and .FALSE. if the file does not exist or if an error was encountered.

LOG = EXST$A(name,namlen)

name	File name (may be a treename).
namlen	Length of name in characters. (Mode is INTEGER*2, but the type of name doesn't matter.)

SPOOL$

A user program can insert a file into the spool file directory by calling the SPOOL$ subroutine from the

applications program. This subroutine SPOOL$ is in the SPOOL$ library (R-mode) and VSPOO$ library (V-mode).

CALL SPOOL$(key,name,namlen,info,buffer,buflen,code)

key	1 - copy named file into queue. 2 - open file on unit info(2) for writing.
name	File to be copied (key=1). Name to appear on banner (key=2).
namlen	Length of name, in characters (1-32).
info	Information array, 12 elements, as follows:

1	temp file unit 1 (may range from 1-62, Rev. 16 and above).
2	temp file unit 2 (may range from 1-62, Rev. 16 and above).
3	print option word (see below).
4-6	form type (6 ASCII characters).
7	plot raster scan size (plot only). This represents #words/raster scan.
8-10	spool filename (returned).
11	deferred print time (valid only if defer bit specified in option word).
12	file size, returned if key 1.

buffer	Scratch buffer - this is used to set up control info and to copy the file to the spool queue (key=1) - it must be at least 40 words long. Copy time is inversely proportional to buffer size. Nominal size is between 300-2000 words.
buflen	Length of buffer.

code — Return code (non-zero is file system error).

SRCH$$

SRCH$$ is used to open a file, close a file, delete a file, or check on the existence of a file.

CALL SRCH$$(action+ref+newfil,filnam,namlen,funit,type,code)

action — A subkey indicating the action to be performed. Possible values are:

- K$READ — Open filnam for reading on funit.
- K$WRIT — Open filnam for writing on funit.
- K$RDWT — Open filnam for reading and writing on funit.
- K$CLOS — Close file by filnam or by funit.
- K$DELE — Delete file filnam.
- K$EXST — Check on existence of filnam.

ref — A subkey modifying the action subkey as follows:

- K$IUFD — Search for filnam in the current UFD (this is the default).
- K$ISEG — Perform the action specified by action on the file that is a segment directory entry in the directory open

on file unit filnam.

K$CACC — Change the access mode of the file already open on funit to action. (K$READ, K$WRIT, K$RDWR only).

K$GETU — Open filnam on an unused file-unit selected by PRIMOS. The unit number is returned in funit. When this key is used, SRCH$$ supplies a unit number not currently in use.

newfil — A subkey indicating the type of file to create if filnam does not exist. Possible values are:

K$NSAM — New threaded (SAM) file (this is the default).

K$NDAM — New directed (DAM) file.

K$NSGS — New threaded (SAM) segment directory.

K$NSGD — New directed (DAM) segment directory.

It is not possible to generate a new UFD with SRCH$$; use CREA$$ instead.

filnam — Name of the file to be opened (2 characters per word). K$CURR can be used to open the current UFD (ACTION keys K$READ, K$WRIT, or K$RDWR only). If ref is K$ISEG, filnam is a file unit from 1 to 62 (1 to 15 under PRIMOS II) on which a segment directory is already open.

namlen	The length in characters (1-32) of filnam.
funit	The number (1-15 under PRIMOS II) of the file unit to be opened or closed, or returned argument with K$GETU key.
type	An integer variable that is set to the type of the file opened. Type is set only on calls that open a file -- it is unmodified for other calls. Possible values of type are: 0 SAM file. 1 DAM file. 2 SAM segment directory. 3 DAM segment directory. 4 UFD.
code	An integer variable set to the return code.

SRCH$$ is a complex subroutine that has multiple uses. The most common use is to open and close files.

TIMDAT

TIMDAT returns the date, time, CPU time, and disk I/O time used since LOGIN, the user's unique number on the system, and the user's login UFD name in an array as follows:

CALL TIMDAT(array,num)

array

(1)	Two ASCII characters representing month. Example: 11
(2)	Two ASCII characters representing day. Example: 30
(3)	Two ASCII characters representing year. Example: 75

	(4)	Integer time in minutes since midnight.
	(5)	Integer time in seconds.
	(6)	Integer time in ticks.
	(7)	Integer CPU time used in seconds.
	(8)	Integer CPU time used in ticks (standard is 330 ticks/second).
	(9)	Integer disk I/O time used in seconds. (see note)
	(10)	Integer disk I/O time used in ticks. (see note)
	(11)	Integer number of ticks per second.
	(12)	User number.
	(13-15)	Six-character login name, left-justified. Example: MSMITH
num		num words of array are set. This routine does not return any useful information under PRIMOS II.

Disk I/O time is from start of seek to end of transfer, including both explicit file I/O and paging operations. CPU time used in controlling the transfer is counted under CPU time, array(7) and array(8).

TNOU

TNOU outputs count characters to the user terminal followed by the LINE FEED, CARRIAGE RETURN. Buffer is expected to contain 2 characters per word.

CALL TNOU(buffer,count)

TNOUA

TNOUA outputs count characters to the user terminal, but is not followed by the LINE FEED, CARRIAGE RETURN.

CALL TNOUA(buffer,count)

TSRC$$

TSRC$$ is used to open a file, close a file, delete a file, or check on the existence of a file, anywhere in the PRIMOS file structure.

CALL TSRC$$(action+newfil,treename,funit,chrpos,type,code)

action		A subkey indicating the action to be performed. Possible values are:
	K$READ	Open treename for reading on funit.
	K$WRIT	Open treename for writing on funit.
	K$RDWT	Open treename for reading and writing on funit.
	K$CLOS	Close treename.
	K$DELE	Delete file treename.
	K$EXST	Check on existence of treename.
newfil		A subkey indicating the type of file to create if treename does not exist. Possible values are:

K$NSAM New threaded (SAM) file (this is the default).

K$NDAM New directed (DAM) file.

K$NSGS New threaded (SAM) segment directory.

K$NSGD New directed (DAM) segment directory.

treename A specification of any file in any directory or subdirectory stored in any array treename packed two characters per word.

funit The number (1-62) of the file unit to be opened or deleted. funit is closed before any action is attempted.

chrpos A two-element integer array setup as follows:

chrpos(1) On entry, set to contain the first character in the array that is part of the treename, the count starting at 0. On exit, it will be pointing one past the last character that was part of the treename. A comma, new line, or carriage return will terminate the name, as will end of array. In case of error, chrpos(1) points one past the treename component that caused the error. Chrpos(1) is always modified by this subroutine, therefore, it must be set up before each call.

chrpos(2)	The number of characters in the treename array.
type	An integer variable that is set to the type of the file opened. type is set only on calls that open a file -- it is unmodified for other calls. Possible values of type are: 0 SAM file. 1 DAM file. 2 SAM segment directory. 3 DAM segment directory. 4 UFD.
code	An integer variable set to the return code. If no errors, code is 0.

TSRC$$ always closes the specified file unit then attaches to the user's home UFD before attempting any action. If the user's home UFD differs from his current UFD before calling TSRC$$, he will find himself attached to his home UFD following the call.

UNIT$A

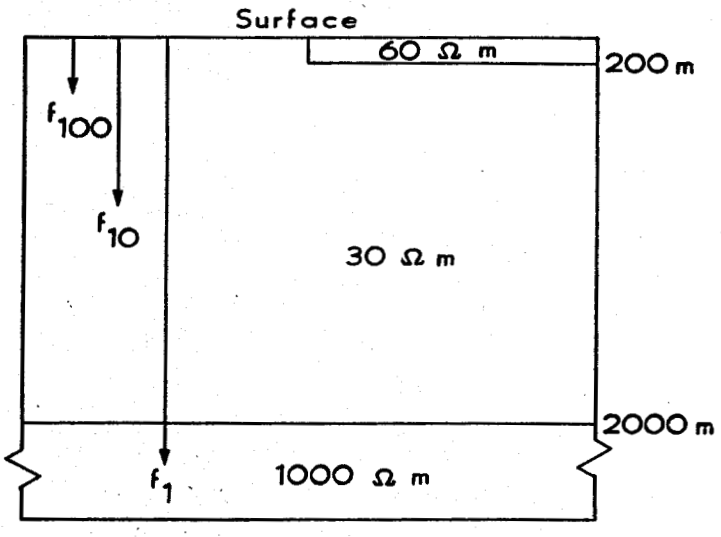
UNIT$A is a logical function which returns .TRUE. if the unit is open and .FALSE. if the unit is not open.

```
LOG = UNIT$A(unit)
```

unit	PRIMOS file unit. (Mode is INTEGER*2.)

SAMPLE TERMINAL SESSION

The following terminal session is an attempt to model the cross section below.



To define the mesh, it is helpful to look at the skin depths of the top layers for each of the frequencies we want to run. This helps to determine which frequencies can be used for a specific model and which frequencies will require a new mesh in order to conform to the rules of mesh design. The arrows shown on the cross section above indicate how deep one can expect to see at a given frequency.

SKIN DEPTHS
(1/6 skin depth is in parentheses)

freq.	res. =30 ohm-m	res. =60 ohm-m
100.	273 (50)	387 (64)
10.	866 (150)	1225 (200)
1.	2738 (500)	3870 (640)
0.1	8660 (1500)	12250 (2000)
0.01	27380 (5000)	38700 (6400)

Ideally, a separate mesh would be designed for each frequency, but in practice this is not necessary.

From the chart on skin depth versus frequency, it seems that we should have elements 50 m wide at the vertical contact of the surface block. Because of the range of frequencies, we can expect to encounter some numerical problems if we attempt to use the mesh for all frequencies. This is especially apparent at the low frequencies at the boundaries.

The purpose of this sample terminal session is to demonstrate how to use the program. It is not a lesson in MT modeling or mesh design.

```
OK, DATE

Thursday, March 26, 1981   12:41 PM

OK, TIME
  0'00   0'01   0'00
OK, MT2D
 *** [MT2D.REV3] ***

 TERMINL- Option: 0

       1  HARD COPY 80 COLUMN
       2  TEKTRONIX 4014
       3  CRT 80 COLUMN

 TERMINL- Option: 3


 MASTER-- Option: 0

       1  EXIT FROM SYSTEM
       2  INPUT NEW MODEL PARAMETERS
       3  EDIT MODEL PARAMETER FILE
       4  LIST MODEL PARAMETER FILE
       5  EXECUTE 2D MT FIN. ELE. PROG
       6  FILE MAIN. - SAVE AND RECALL
       7  PLOT MODELS AND RESULTS
       8  EXECUTE PHANTOM JOB

 MASTER-- Option: 2


MAXIMUM MESH SIZE               =  90 X  30
MAXIMUM NO. OF GROUPS           =  80
MAXIMUM NO. OF FREQUENCIES      =  10
MAXIMUM NO. OF BOUNDARY LAYERS =  10

TYPE "RETURN TO MASTER" TO EXIT INPUT FUNC.
 INPUT OWNER-PROJECT IDENT
 REV 3 TEST
 INPUT MODEL NAME
 SURFACE BLOCK
 INPUT IDX = 1 FOR LINE SOURCE
             2 FOR MT TE-MODE
             3 FOR MT TM-MODE
2
 INPUT # OF FREQUENCIES
5
 INPUT     5 FREQUENCIES
100.,10.,1.,0.1,0.01
 # OF X NODES (I) =
27
 # OF X ELEMENT GROUPS (I) =
19
```

```
+-------------------------------+
| TIME gives the accumulated    |
| connect,CPU,I/O times         |
| since LOGIN                   |
+-------------------------------+
```

```
 INPUT 19 X ELEMENTS GROUP #S (I)
1,1,1,1,1,1,1,1,1,9,1,1,1,1,1,1,1,1,1
*GROUP SUMMATION ERROR*
 INPUT 19 X ELEMENTS GROUP #S (I)
1,1,1,1,1,1,1,1,1,8,1,1,1,1,1,1,1,1,1
 INPUT 19 X GROUP SPACINGS (F)
100000.,30000.,10000.,3000.,1600.,800.,400.,200.,100.
50.,100.,200.,400.,700.,1600.,3000.,10000.,30000.,100000.
 # OF Z NODES (I) =
20
 # OF Z ELEMENT GROUPS (I) =
   (A GROUP MAY NOT OVERLAP THE AIR-EARTH INTERFACE)
18
 INPUT 18 Z ELEMENT GROUP #S (I)
1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1
 INPUT 18 Z GROUP SPACINGS (F)
30000.,10000.,3000.,1000.,300.,100.,30.
20.,40.,60.,80.,100.,200.,500.,1000.,5000.,10000.,30000.
 IS THIS A FLAT EARTH MODEL?
Y
 INPUT THE # OF AIR LAYERS
7
 SELECT NODES FOR PSEUDO-SECTION DISPLAY
 NODES FOR PSEUDO-SEC.(Y/N) - 2 THRU  26
 .........1.........2......
>  Y YYYY Y YY Y YYYY YY
 INPUT # OF RESISTIVITY CODES
4
 INPUT 4 RESISTIVITIES
 (RESISTIVITY OF AIR IS 1.0E+37)
60.,30.,1000.,1.E+38
 INPUT # LAYERS FOR LEFT AND RIGHT BOUNDARIES
2,3
 LEFT EDGE INTERFACE DEPTHS AND RESISTIVITIES
 INPUT   1  INTERFACE DEPTHS(METERS)
2000.
 INPUT   2  RESISTIVITIES
30.,1000.
 RIGHT EDGE INTERFACE DEPTHS AND RESISTIVITIES
 INPUT   2  INTERFACE DEPTHS(METERS)
200.,2000.
 INPUT   3  RESISTIVITIES
60.,30.,1000.
 DEFINE MESH AREA FOR INPUT
 X START NODE # (I) =
13
 X END NODE # (I) =
15
 Z START NODE # (I) =
7
 Z END NODE # (I) =
17
 NEW MEDIA CODE-TRIAG. OR SQUARE ELEMENTS (T OR S)?
S
```

No. of elements sums to more than (NODEX-1), so program prints an error message

```
 ROW  7 ELEMENT 13 TO 14
 ..
>44
 ROW  8 ELEMENT 13 TO 14
 ..
>21
 ROW  9 ELEMENT 13 TO 14
 ..
>21
 ROW 10 ELEMENT 13 TO 14
 ..
>21
 ROW 11 ELEMENT 13 TO 14
 ..
>21
 ROW 12 ELEMENT 13 TO 14
 ..
>22
 ROW 13 ELEMENT 13 TO 14
 ..
>22
 ROW 14 ELEMENT 13 TO 14
 ..
>22
 ROW 15 ELEMENT 13 TO 14
 ..
>22
 ROW 16 ELEMENT 13 TO 14
 ..
>33

 MASTER-- Option: 4


      LIST---- Option: 0

            1  RETURN TO MASTER OPTIONS
            2  LIST TITLES, MODE
            3  LIST GEOMETRY PARAMETERS
            4  LIST MEDIA CHARACTRISTCS
            5  LIST MESH CODE TRI. ELE.
            6  LIST MESH CODE SQR. ELE.
            7  LIST RESULTS
            8  LIST PSEUDO-SEC(MT-ONLY)
            9  SEND WORK FILE TO PRNTR

      LIST---- Option: 2

PROJECT NAME:
    REV 3 TEST
MODEL NAME:
    SURFACE BLOCK
MODE = TE
```

```
---------------------
! Note how little   !
! input is required !
! to define the mesh!
! code. See LIST    !
! option 6 below    !
! for mesh          !
---------------------
```

```
     LIST---- Option: 3

# OF X NODES =   27
# OF X ELEMENT GROUPS =   19
X ELEMENT GROUP #S
             1          1          1          1          1
             1          1          1          1          8
             1          1          1          1          1
             1          1          1          1
X ELEMENT GROUP SPACINGS IN A-UNITS
    100000.00  30000.00  10000.00   3000.00   1600.00
       800.00    400.00    200.00    100.00     50.00
       100.00    200.00    400.00    700.00   1600.00
      3000.00  10000.00  30000.00 100000.00
Type <CR> to continue (also RESET PAGE on Tek)

# OF Z NODES =   20
# OF Z ELEMENT GROUPS =   18
Z ELEMENT GROUP #S
             1          1          1          1          1
             1          1          1          1          1
             1          1          1          1          2
             1          1          1
Z ELEMENT GROUP SPACINGS IN A-UNITS
     30000.00  10000.00   3000.00   1000.00    300.00
       100.00     30.00     20.00     40.00     60.00
        80.00    100.00    200.00    500.00   1000.00
      5000.00  10000.00  30000.00
Type <CR> to continue (also RESET PAGE on Tek)

Z-NODE LOCATIONS OF AIR-EARTH INTERFACE
    8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
# AIR LAYERS AT LEFT AND RIGHT BOUNDRY
    7 7
 LEFT EDGE --INTERFACE DEPTHS AND COND.
  2000.000
     30.00   1000.00
 RIGHT EDGE--INTERFACE DEPTHS AND COND.
   200.000  2000.000
     60.00     30.00   1000.00

     LIST---- Option: 4

# MEDIA RESISTIVITIES
            4
MEDIA RESISTIVITY (MHO/METERS)
        60.         30.        0.10E 04    0.10E 39
# OF FREQUENCIES (HZ) =    5
FREQUENCIES =   100.0000   10.0000    1.0000    0.1000    0.0100

     LIST---- Option: 6


.........1.........2.....
```

```
4444444444444444444444444
4444444444444444444444444
4444444444444444444444444
4444444444444444444444444
4444444444444444444444444
4444444444444444444444444
4444444444444444444444444
2222222222222111111111111
2222222222222111111111111
2222222222222111111111111
2222222222222111111111111
2222222222222222222222222
2222222222222222222222222
2222222222222222222222222
2222222222222222222222222
3333333333333333333333333
3333333333333333333333333
3333333333333333333333333
3333333333333333333333333
.........1.........2.....
```

circled area was all that was needed to uniquely define the full mesh

LIST---- Option: 1

MASTER-- Option: 3

EDIT---- Option: 0

1 RETURN TO MASTER OPTIONS
2 EDIT MESH CODE ELEMENTS
3 EDIT MESH CODE ROWS
4 EDIT ENTIRE MESH CODE
5 EDIT # OF MEDIA CODES
6 EDIT MEDIA RESISTIVITY
7 EDIT MESH GEOMETRY
8 EDIT TITLES
9 EDIT MODE, L.S. POSITIONS
10 EDIT FREQUENCIES
11 EDIT EDGE DEPTH, RES.
12 EDIT PSUDO-SEC. NODES(MT)

EDIT---- Option: 7

An X-GROUP SPACING was entered wrong, so now it must be corrected

EMGEO--- Option: 0

1 RETURN TO EDITOR
2 EDIT # OF X NODES
3 EDIT # OF X ELEMENT GROUPS
4 EDIT X ELEMENT GROUP #S
5 EDIT X GROUP SPACINGS
6 EDIT # OF Z NODES

```
  7  EDIT # OF Z ELEMENT GROUPS
  8  EDIT Z ELEMENT GROUP #S
  9  EDIT Z GROUP SPACINGS
 10  EDIT AIR-EARTH INTERFACES

        EMGEO--- Option: 5

CHANGE-X GROUP SPACINGS
 1.000000E 05 3.000000E 04 1.000000E 04 3.000000E 03 1.600000E 03
 8.000000E 02 4.000000E 02 2.000000E 02 1.000000E 02 5.000000E 01
 1.000000E 02 2.000000E 02 4.000000E 02 7.000000E 02 1.600000E 03
 3.000000E 03 1.000000E 04 3.000000E 04 1.000000E 05
PARAMETER # (I)=14
#   14  OLD = 7.000000E 02  NEW =
800.
 1.000000E 05 3.000000E 04 1.000000E 04 3.000000E 03 1.600000E 03
 8.000000E 02 4.000000E 02 2.000000E 02 1.000000E 02 5.000000E 01
 1.000000E 02 2.000000E 02 4.000000E 02 8.000000E 02 1.600000E 03
 3.000000E 03 1.000000E 04 3.000000E 04 1.000000E 05
PARAMETER # (I)=0

        EMGEO--- Option: 1


     EDIT---- Option: 1


 MASTER-- Option: 5

 ** FINITE ELEMENT SUBROUTINE **
 ON FREQ #  1 -   100.0000 HZ.
 ON FREQ #  2 -    10.0000 HZ.
 ON FREQ #  3 -     1.0000 HZ.
 ON FREQ #  4 -     0.1000 HZ.
 ON FREQ #  5 -     0.0100 HZ.

 MASTER-- Option: 4                          ---------------
                                             ! now look at !
                                             ! results     !
     LIST---- Option: 8                      ---------------

APP. RES. (Y OR N)? Y
NEG.PHASE (Y OR N)? N
 Type <CR> to continue (also RESET PAGE on Tek)

                              APP. RES.      MODE = TE
 HORZ. DIST.     -6.3   -1.7   -0.9   -0.5   -0.3   -0.2   -0.1    0.0    0.1
FREQ  NODE#       4      6      7      8      9     11     13     14     16
  0.100E 03     31.3   31.3   31.3   31.5   32.4   34.3   37.3   40.6   46.6

  0.100E 02     30.1   30.0   30.2   30.6   31.0   31.5   32.2   32.9   34.1

  0.100E 01     32.6   33.0   33.2   33.3   33.4   33.6   33.8   34.1   34.6
```

```
  0.100E 00   150.4  150.8  150.9  151.0  151.3  151.7  152.4  153.3  154.7

  0.100E-01   346.6  486.4  497.2  496.5  499.6  499.6  499.3  501.5  504.3

 Type <CR> to continue (also RESET PAGE on Tek)

                                  APP. RES.      MODE = TE
 HORZ. DIST.      0.2    0.3    0.5    0.9    3.3    6.3
FREQ  NODE#     18     19     20     21     23     24
  0.100E 03     49.5   51.3   53.1   54.0   53.9   53.9

  0.100E 02     34.6   34.9   35.4   36.1   37.1   37.0

  0.100E 01     34.7   34.8   34.9   35.1   35.6   36.3

  0.100E 00    155.2  155.3  155.6  155.8  156.1  158.1

  0.100E-01    504.5  504.6  505.3  506.4  521.0  456.7

      LIST---- Option: 0

            1  RETURN TO MASTER OPTIONS
            2  LIST TITLES, MODE
            3  LIST GEOMETRY PARAMETERS
            4  LIST MEDIA CHARACTRISTCS
            5  LIST MESH CODE TRI. ELE.
            6  LIST MESH CODE SQR. ELE.
            7  LIST RESULTS
            8  LIST PSEUDO-SEC(MT-ONLY)
            9  SEND WORK FILE TO PRNTR

      LIST---- Option: 9

LONG OR SHORT LISTING (L OR S)? S
Your spool file, PRT008, is     4 records long    ! get hardcopy !

      LIST---- Option: 1


 MASTER-- Option: 6

*FILE MAINTENENCE*

      FMAIN--- Option: 0

            1  RETURN TO MASTER PROGRAM
            2  LIST MERGED FILE DIRECTORY
            3  SAVE WORK FILE ON MERGED FILE
            4  RESTORE WORK FILE FROM MERGED FILE
            5  DELETE AND PACK MERGED FILE
            6  SYSTEM PACK MERGE FILE
            7  INITIALIZE MERGE FILE
            8  SEND MERGE FILE CONTENTS TO PRINTER
```

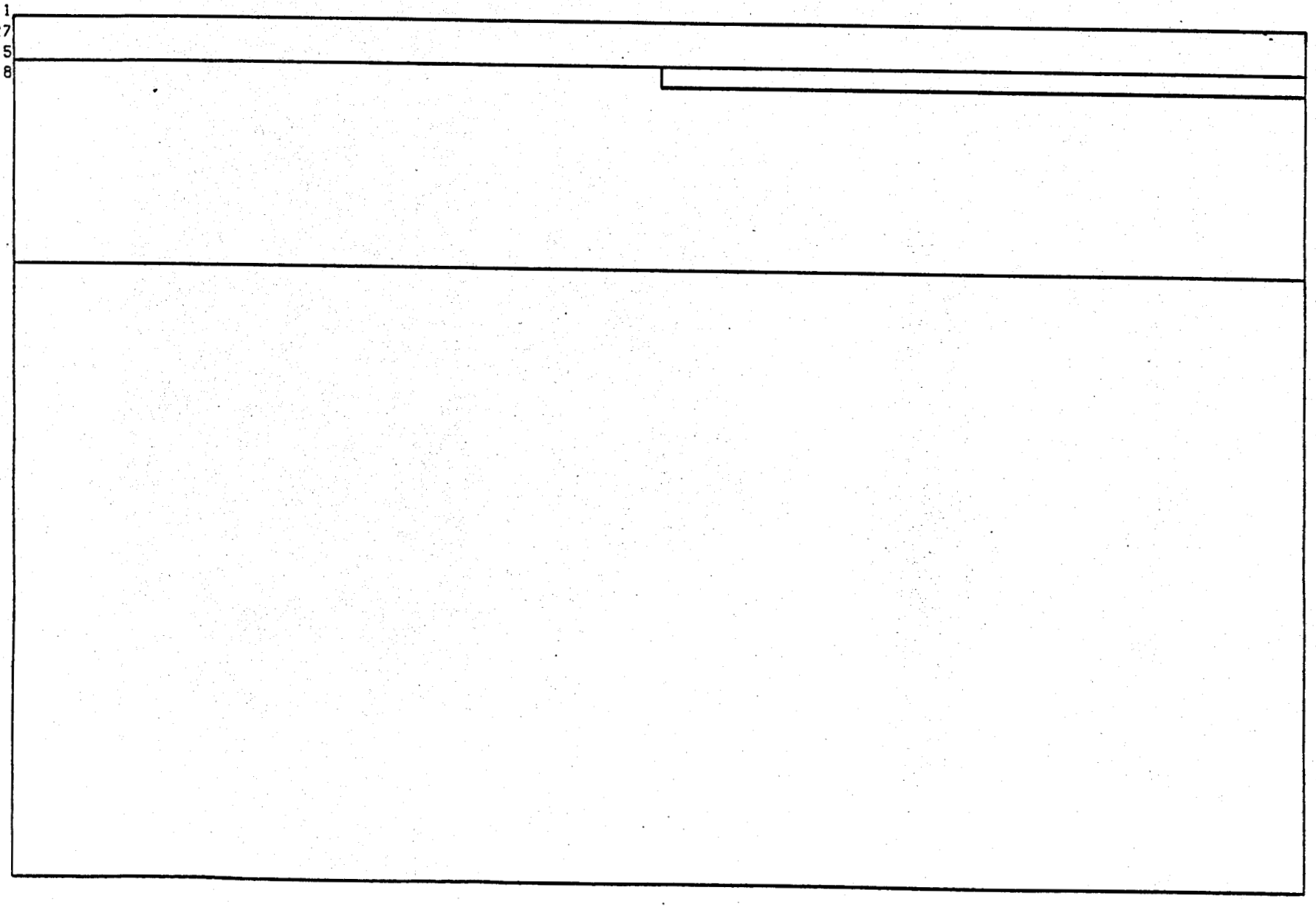
```
      FMAIN--- Option: 7

*INITIALIZE MERGE FILE* (VERIFY Y OR N)?Y
FILE IDENTIFIER (60A)= REV 3 TESTING MODELS

      FMAIN--- Option: 2

 REV 3 TESTING MODELS
MERGE FILE IS EMPTY

      FMAIN--- Option: 3

*SAVE WORK FILE*
WORK FILE SAVED AS SUBFILE#  1

      FMAIN--- Option: 2

 REV 3 TESTING MODELS
 *SUBFILE#  1
      REV 3 TEST
      SURFACE BLOCK

      FMAIN--- Option: 1


 MASTER-- Option: 7


      PLOT---- Option: 2

*CURRENT MODEL PLOT*
DO YOU WANT CROSS-SECTION?Y
DO YOU WANT MESH? N
 INPUT PLOT PAGE SIZE IN INCHES (X,Y)
11.,8.5

          DEVICE-- Option: 0

                1  TEKTRONIX GRAPHICS PLOT
                2  STATOS PLOT
                3  PRINTER (DEFAULTS TO #2)
                4  ZETA PEN PLOT
                5  UUCC CALCOMP PLOT

          DEVICE-- Option: 2

 START X NODE # (I)=1
 END X NODE # (I)=27
 START Z NODE # (I)=5
 END Z NODE # (I)=18
Statos Plot - Number of Vectors =       520
DO YOU WANT GRAPHS? Y
 INPUT GRAPH PAGE SIZE IN INCHES (X,Y)
```

```
+----------------------+
| new merge file,      |
| so must initialize   |
+----------------------+

+--------------+
| save model   |
+--------------+

+--------------+
| get plots    |
+--------------+
```

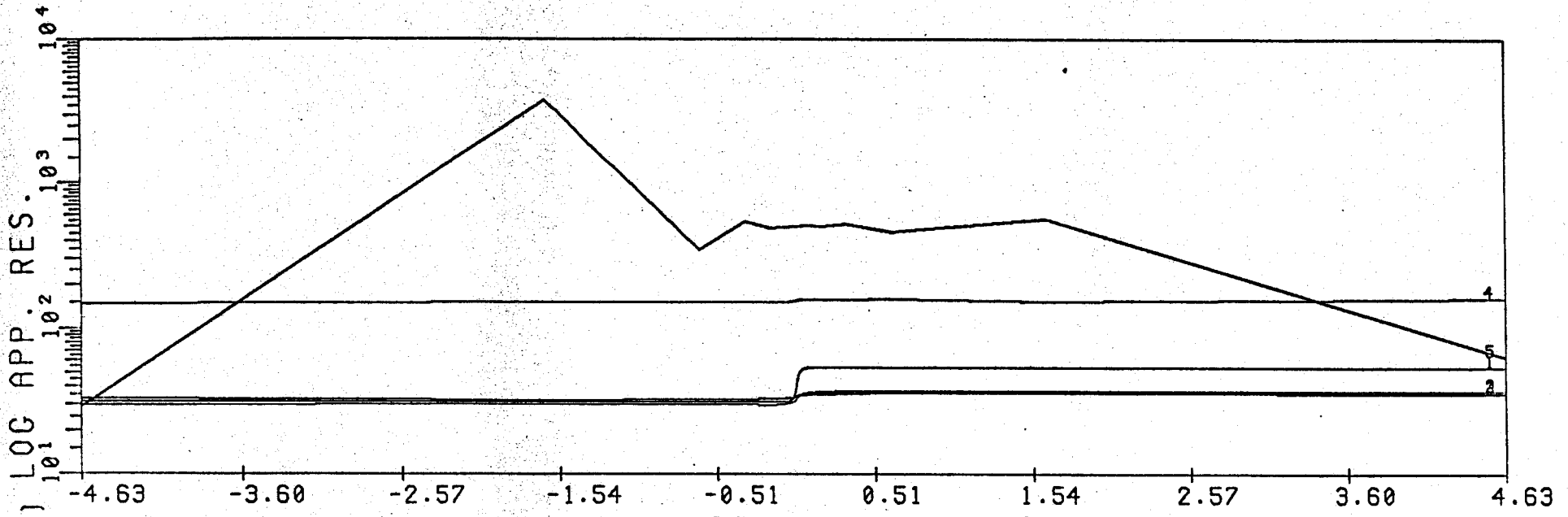
```
11.,8.5

            DEVICE-- Option: 2

 SELECT FREQ. FOR GRAPH (Y,N)
 .....
>YYYYY
Statos Plot - Number of Vectors =      1964

      PLOT----- Option: 1


 MASTER-- Option: 1

                                    ------------------
                                    ! see how much   !
OK, TIME                            ! time elapsed   !
  0'32  20'16   0'13                ! during session !
OK, COMO -END                       ------------------
```

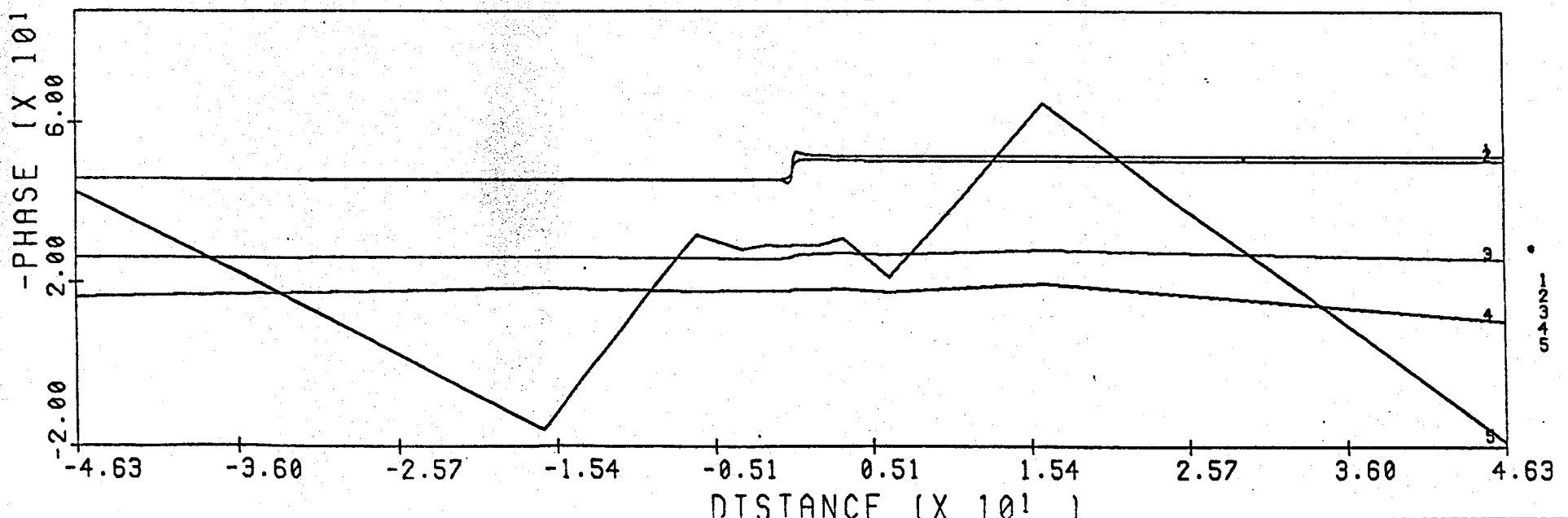
REV 3 TEST
SURFACE BLOCK
MODE = TE
DISTANCE = KM

IX1= 1
IX2= 27
IZ1= 5
IZ2= 18



REV 3 TEST
SURFACE BLOCK
MODE = TE
DISTANCE = KM





USER: MT2D

WORK-PRINT

```
**************************************************************************************************************************
**************************************************************************************************************************


M    M MMMMMM  MMM  MMMM
MM MM    M   M   M M   M
M M M    M       M M   M
M M M    M      M  M   M
M   M    M     M   M   M
M   M    M    M    M   M
M   M    M   MMMMM MMMM


M   M  MMM  MMMM  M   M        MMMM  MMMM   MMM  M   M MMMMM
M   M M   M M   M M   M        M   M M   M   M   MM  M   M
M M M M   M M   M M M M        M   M M   M   M   M M M   M
M M M M   M MMMM  MM    MMMMM MMMM  MMMM    M   M M M   M
M M M M   M M M   M M          M     M M     M   M M M   M
MM MM M   M M  M  M  M         M     M  M    M   M  MM   M
M   M  MMM  M   M M   M        M     M   M MMM  M   M   M


**************************************************************************************************************************
**************************************************************************************************************************

LABEL:  PRT008  -FORM        -COPIES   1

SPOOLED:  03/26/81 13:10
STARTED:  03/26/81 13:10, ON: CEN   BY: CEN
```

PROJECT NAME:
REV 3 TEST
MODEL NAME:
SURFACE BLOCK
MODE = TE
MEDIA RESISTIVITY (MHO/METERS)
60. 30. 0.10E 04 0.10E 39

HORZ. DIST.	-6.3	-1.7	-0.9	-0.5	-0.3	-0.2	-0.1	0.0	0.1	0.2	0.3	0.5	0.9	3.3	6.3
DEPTH NODE#	4	6	7	8	9	11	13	14	16	18	19	20	21	23	24
44430.00	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
14430.00	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4430.00	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
1430.00	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
430.00	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
130.00	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
30.00	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
0.00	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1
-20.00	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1
-60.00	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1
-120.00	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1
-200.00	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
-300.00	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
-500.00	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
-1000.00	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
-2000.00	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
-3000.00	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
-8000.00	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
-18000.00	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
-48000.00															

APP. RES.

HORZ. DIST.	-6.3	-1.7	-0.9	-0.5	-0.3	-0.2	-0.1	0.0	0.1	0.2	0.3	0.5	0.9	3.3	6.3
FREQ NODE#	4	6	7	8	9	11	13	14	16	18	19	20	21	23	24

0.100E 03	31.3	31.3	31.3	31.5	32.4	34.3	37.3	40.6	46.6	49.5	51.3	53.1	54.0	53.9	53.9
0.100E 02	30.1	30.0	30.2	30.6	31.0	31.5	32.2	32.9	34.1	34.6	34.9	35.4	36.1	37.1	37.0
0.100E 01	32.6	33.0	33.2	33.3	33.4	33.6	33.8	34.1	34.6	34.7	34.8	34.9	35.1	35.6	36.3
0.100E 00	150.4	150 8	150.9	151.0	151.3	151.7	152.4	153.3	154.7	155.2	155.3	155.6	155.8	156.1	158.1
0.100E-01	346.6	486.4	497.2	496.5	499.6	499 6	499.3	501.5	504.3	504.5	504 6	505.3	506.4	521.0	456.7

NEG. PHAZ

HORZ. DIST. FREQ NODE#	-6.3 4	-1.7 6	-0.9 7	-0.5 8	-0.3 9	-0.2 11	-0.1 13	0.0 14	0.1 16	0.2 18	0.3 19	0.5 20	0.9 21	3.3 23	6.3 24
0.100E 03	46.	46.	46.	46.	46.	46.	47.	49.	52.	53.	53.	53.	52.	52.	52
0.100E 02	46.	46.	46.	47.	47.	47.	48.	49.	50.	50.	51.	51.	51.	51.	51.
0.100E 01	26.	26.	26.	26.	26.	27.	27.	27.	27.	27.	27.	27.	27.	28.	27.
0.100E 00	18.	18.	18.	18.	18.	18.	18.	18.	18.	18.	18.	18.	18.	19.	18.
0.100E-01	32.	30.	30.	30.	30.	30.	30.	30.	30.	30.	30.	30.	30.	32.	22