IPINV: A Two-Dimensional Dipole-Dipole Resistivity
Modeling and Inversion Program
(User's Guide and Documentation for Rev. 1)

MASTER

by
Alan G. Tripp
and

Terry J. Kilbuck

January 1981

Work performed under Contract No. DE-AC07-80ID12079

EARTH SCIENCE LABORATORY
University of Utah Research Institute
Salt Lake City, Utah

Prepared for
U.S. Department of Energy
Division of Geothermal Energy

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of 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. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

IPINV: A Two-Dimensional Dipole-Dipole Resistivity
Modeling and Inversion Program
(User's Guide and Documentation for Rev. 1)

by

Alan C. Tripp

and

Terry J. Killpack

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

January 1981

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

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

ACKNOWLEDGMENTS.....	1
INTRODUCTION.....	2
PROGRAM ORGANIZATION.....	3
Model Definition.....	3
Figure 1 IPINV Network (all cells).....	5
Figure 2 IPINV Network (middle).....	6
Figure 3 IPINV Network (bricks).....	7
PROGRAM USAGE.....	8
Prime 400 Environment.....	8
Input File Format.....	8
Record number 1.....	8
Record number 2 (conditional).....	9
Record number 3.....	9
Record number 4.....	10
Record number 5.....	11
Record number 6.....	12
Record number 7.....	12
Record number 8.....	12
Record number 9.....	13
Record number (9+1) through (9+NRES).....	13
Record number (10+NRES).....	14
Record number (11+NRES).....	14
Record number (10+NRES+1 through 10+NRES+NBLOC).....	14
Record number (11+NRES+NBLOC).....	15
Record number (12+NRES+NBLOC) (conditional).....	15
Appendix A Sample Execution.....	17
Sample Input.....	18
Record number 1.....	19
Record number 2.....	20
Record number 3.....	20
Record number 4.....	20
Record number 5.....	20
Record number 6.....	21
Record number 7.....	21
Record number 8.....	21
Record number 9.....	21
Record number 10 through 26.....	21
Record number 27.....	22
Record number 28.....	22
Record number 29 through 43.....	22

Record number 44.....	22
Record number 45.....	22
Sample Output.....	23
Appendix B PRIMOS Subroutines.....	33
ERRPR\$.....	35
EXIT.....	36
SRCH\$\$.....	36
TNOU.....	39
TSRC\$\$.....	39
Appendix C Program Conversion.....	42
Memory Requirements.....	43
Prime computer extensions.....	43
\$INSERT statement.....	43
NOBIG / BIG.....	44
List-Directed READ statement.....	44
System Subroutines.....	45
REFERENCES.....	47

ACKNOWLEDGMENTS

The network analogy forward algorithm for IPINV was originally programmed and implemented by Dr. Charles M. Swift. A significant amount of the original source code developed by Dr. Swift remains in the forward problem of the IPINV program. The inversion process was implemented by Alan C. Tripp while he was a Ph.D. candidate in the Department of Geology and Geophysics at the University of Utah.

INTRODUCTION

IPINV.REV1 is a batch program that is capable of inverting resistivity data to two-dimensional models for a dipole-dipole array. The forward problem is computed using a transmission surface analogy.

Resistivity data are frequently gathered over conductivity structures which are approximately two-dimensional. Several computer algorithms that compute the forward response of two dimensional structures have been developed and used in routine data interpretation (Madden, 1971; Rijo, 1977; Dey and Morrison, 1976). An interpreter using these forward algorithms can have difficulties in determining the size and direction of the parameter perturbations necessary for convergence of the trial-and-error forward modeling. Pelton et al., (1978) optimized the step-size in parameter space for a single two-dimensional body in a host rock by utilizing the technique of linearized least-squares. The partial derivatives necessary for the least squares inversion were calculated by referencing a forward response data file and interpolating between models. This method provides cheap, accurate interpretations of a given simple model type, but breaks down if the interpretation requires a complicated model.

IPINV.REV1 is capable of inverting resistivity data to two-dimensional models of arbitrary complexity. The program was written by Alan C. Tripp while a graduate student in the Department of Geology and Geophysics at the University of Utah. The details of the theory and methodology of the program are given in Tripp et al., (in prep.). The two-dimensional forward modeling routine is based on the transmission surface analogy (Madden, 1972). The inversion algorithm is a linearized least-squares technique. Step-size stabilization is provided by either the Box-Kanemaesu (1972) method or by using a Marquardt step. The program uses log derivatives to increase the rate of convergence.

PROGRAM ORGANIZATION

IPINV is a batch mode modeling and inversion program. The model for the forward problem is defined by constants in the program source code and by user input. The user input enters the program through a user-defined input file which is created and modified using the system text editor. The input file information controls the forward modeling and inversion processes. The computed results are written to an output file that can be read with the text editor or printed on a line printer.

Model Definition

The network analogy mesh is fixed in the IPINV program at 67 horizontal nodes and 12 vertical nodes. There are 68 horizontal cells (one more than the number of nodes) and 13 vertical cells. A cell is a rectangular area bounded by four adjacent nodes and does not contain any other nodes within its boundaries.

The cells around the border of the network are of zero dimension. The branch impedance across the zero dimension cells is a termination impedance. It is used to reduce the reflections at the boundary of the network. The termination impedances that are used are only approximate values to those required to terminate energy reflections at the boundary of the transmission surface.

Figure 1 shows the full IPINV network. The upper left-hand corner of the network corresponds to the node numbers (1,1). The lower right-hand corner of the network corresponds to node numbers (67,12). The area of the network in Figure 1 that is outlined in a thick rectangle is reproduced in Figure 2 at a larger scale. The upper left-hand corner of the network shown in Figure 2 corresponds to node numbers (4,1). The lower right-hand corner corresponds to node numbers (64,12). Figure 2 also shows the location of the transmitter and receiver electrodes of the dipole-dipole array. The electrode positions are noted by small triangular symbols at various node positions along the top of the network. The electrodes are placed one dipole spacing apart. The numbers running down the left side of the network in Figure 2 are the vertical default cell dimensions of the network expressed as multiples of the uniform dipole spacing. It is possible to change the vertical cell dimensions via user input, but it is not possible to change the number of cells in the vertical direction.

Figure 3 shows the "bricks" that are used in the IPINV network. The bricks define the area of the network that may be used in the inversion process. Each brick is assigned an inversion flag by user input. If the flag is 0, the

corresponding brick is not used in the inversion process. If the flag is non-zero, it is a sequential brick number. Inversion bricks are counted left to right and then top to bottom. The bricks are grouped together to form "blocks" for the inversion process. All of the bricks of a given block must contain the same resistivity code. A block may contain up to seven bricks. Blocks are defined by specifying the brick numbers via user input. The brick and block mechanism has been introduced into the program to provide an easy method of defining the cross section units that are to be inverted. Inversion is done only on block size cross section units.

The resistivity model is defined through user input by assigning a resistivity in ohm-meters to sequential code numbers. Each cell in the mesh is then assigned a code number that corresponds to the resistivity of the cell. Inversion bricks are defined and assigned a resistivity code number. The inversion blocks are defined by specifying the index numbers of the bricks that are contained in each block. A block can only be constructed in rectangular fashion using contiguous bricks and by using only the bricks outlined in Figure 3.

FIGURE 1 IPINV NETWORK (ALL CELLS)

THE NETWORK ENCLOSED IN THE BOX IS SHOWN IN FIGURE NUMBER 2

FIGURE 2 IPINV NETWORK (MIDDLE)

FIGURE 3 IPINV NETWORK (BRICKS)

PROGRAM USAGE

Prime 400 Environment

IPINV.REV1 is operational on the UURI/ESL Prime 400 computer system. The program is executed by typing the system command IPINV. To use IPINV on the Prime system, a user must log in to a user file directory, and then create an input file called IPINV.INPUT using the system text editor. When the input file is completed, the program can be executed by typing the system command, IPINV. After execution is completed, the user will find a file called IPINV.OUTPUT in the user file directory. This is the output file from the program. It can be examined with the text editor, or it can be sent to the line printer using the SPOOL command. If the user desires to change the input model, the IPINV.INPUT file must be modified or recreated, and the program executed again. Execution of the program will delete the file IPINV.OUTPUT if it exists and create a new output file each time the program is executed. If the user desires to save the output data, the output file name of the previous execution must be changed to some other name before a new execution is invoked.

Input File Format

The input file for IPINV is a FORTRAN-formatted sequential file. The name of the input file is IPINV.INPUT. The program automatically searches for a file named IPINV.INPUT and opens it for reading on logical unit number 8. The logical unit number may be changed by changing the source line in the IPINV main program where the unit number LUI is defined.

Record number 1

NIT, ISI, NOPT, MQT, NNSEP, NDERP
(Free Format)

NIT - (integer) is the number of iterations allowed in the inversion.

ISI - (integer) is a flag variable. If ISI has the value of 1, the program uses a Gaussian step-size truncation. If the value of ISI is 2, the program uses Box-Kanemaesu step. If ISI is 1 and MQT (see below) is 1, the program uses a Marquardt step.

NOPT - (integer) is a flag variable that is used to control the forward versus the inversion process. If NOPT is 1, the program computes a forward response only. If the value of NOPT is something other than 1, the forward and inverse problems are computed to NIT (see above) iterations. If NOPT is 1, NIT must also be 1.

MQT - (integer) is the Marquardt step flag. If MQT is 1, the program uses the Marquardt step. If MQT is 1, then ISI (see above) must also be 1. If MQT is a value other than 1, then the program uses the value of ISI (see above) to determine the mode of stepping.

NNSEP - (integer) is the number of dipole separations for inversion. NNSEP must be less than or equal to 10.

NDERP - (integer) is the derivative flag.
 if NDERP = 1 Use regular derivatives
 if NDERP = 2 Use Log derivatives
 else Use Log derivatives

Record number 2 (conditional)

Record number 2 is conditional on the value of MQT (see record number 1). If MQT is equal to 1 (Marquardt step), then record 2 must be included. If MQT is not equal to 1, then record 2 is omitted from the input file.

DIAGAD
 (Free Format)

DIAGAD - (real) is the Marquardt factor for Marquardt stepping.

Record number 3

Record number 3 contains the flags that are used for diagnostic printing of the brick derivatives.

NDF

(Free Format)

NDF(1) - (integer) is the flag for the first brick that is being used for inversion. If NDF(1) equals 0, do not print the derivatives for brick number 1. If NDF(1) equals 1, print the derivatives.

NDF(2) - (integer) is the flag for the second brick that is being used for inversion. If NDF(2) is 0, do not print the derivatives for brick number 2. If NDF(2) is 1, print the derivatives.

NDF(3) - etc.

Record number 4

Record number 4 consists of real variables that represent the observed resistivities in ohm-meters in the dipole-dipole survey. The observed pseudosection is fixed in horizontal extent to consist of the values between the -6 to +6 electrode positions. The depth of the pseudosection is governed by the NNSEP variable that is described in record 1. The numbers are input from left to right and from top to bottom in the pseudosection. The number of pseudosection resistivities that are to be entered is calculated by the formula $((21-NNSEP)*NNSEP)/2$.

The pseudosection location numbers are shown in the diagram below.

-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
NNSEP												
1	1	2	3	4	5	6	7	8	9	10		
2		11	12	13	14	15	16	17	18	19		
3			20	21	22	23	24	25	26	27		
4				28	29	30	31	32	33	34		
5					35	36	37	38	39	40		
6						41	42	43	44	45		
7							46	47	48	49		
8								50	51	52		
9									53	54		
10											55	

OBS
(Free Format)

OBS(1) - (real) is the observed resistivity in

the left-hand position of the top row
in the observed resistivity
pseudosection.

OBS(2) - (real) is the value to the right of
OBS(1) in the top row of the
pseudosection.

OBS(11) - (real) is the left observed value of
the second row of the observed
pseudosection.

etc.

Record number 5

Record number 5 consists of the data weights for the
observed resistivity values. The data weights have a
one-to-one correlation with the observed resistivity
values. If the observed resistivity value in a given
position of the pseudosection has a "reasonable" value, the
corresponding data weight is given a value of 1.0. If the
observed resistivity value in a given position of the
pseudosection has an "unreasonable" value, the
corresponding data weight is given a value of 0.0.
Observed resistivity values that have zero data weights are
excluded from the inversion process.

OOBS
(Free Format)

OOBS(1) - (real) is the data weight associated
with OBS(1).

OOBS(2) - (real) is the data weight associated
with OBS(2).

etc.

Record number 6

VVAR
(Free Format)

VVAR - (real) is the a priori estimated variance of the data. If it is not possible to estimate the variance, VVAR may be set to 0.0, and the program will estimate the variance from the square error.

Record number 7

LPR, NBL, NBLOC
(Free Format)

LPR - (integer) is 0, indicating no printing of potentials, or it is the node number of the location where potentials will be printed before converting to lambda space. This print option is used for debugging purposes.

NBL - (integer) is the number of bricks that are to be inverted. This variable must have a value less than or equal to 25.

NBLOC - (integer) is the number of blocks. This variable must be less than or equal to 25. A block is a group of bricks that are grouped together for the inversion process. Blocks are defined by user input that follows in subsequent records.

Record number 8

NRES, IDELV
(Free Format)

NRES - (integer) is the number of distinct cross section units. There will be a distinct resistivity associated with each unit. Each block is considered a distinct unit.

IDELV - (integer) is the vertical dimension flag.
If IDELV = 0, the default vertical dimensions of the cel

ls

in the network are
considered "reasonabl

e".

If IDELV = 1, the dimensions of th

e

cells are read.

Record number 9

TEXT
(8A4)

TEXT - (alphanumeric) is an identifying text that describes the model. It must be less than or equal to 32 characters.

Record number (9+1) through (9+NRES)

There are NRES read statements that read the cross section unit code numbers and the associated resistivities. They all use the code format shown below.

II,RESX
(Free Format)

II - (integer) is the array of the cross section resistivity code number.

RESX - (real) is the corresponding unit resistivity in ohm-meters.

The unit code numbers and corresponding unit resistivities for the inversion blocks are listed first, followed by the background unit numbers and the corresponding resistivities.

Record number (10+NRES)

This record contains the unit numbers for the network cells between the 8th and 60th horizontal nodes and between the 1st and 12th vertical nodes. The cell resistivity code numbers are read from left to right and from top to bottom. The cells that make up the middle 16 dipole spacings in the network are coded with resistivity code numbers. The program extends the code to the left and right boundaries of the network automatically. The user is not required to code the cells that are beyond the 16 dipole spacings in the middle of the network.

```
((ICODE(I,J), J=1,52), I=1,11)
(26I2)
```

The input file for this record consists of 22 lines of 26 integers each. Each pair of lines defines one row of the network. The integer unit numbers are each two digits long.

Record number (11+NRES)

This record contains the brick parameter flags. If the flags are 0, the resistivity of the associated brick is held fixed during the inversion process. If the flag is greater than 0 it represents the number of the cross section unit to which the brick belongs. There are 76 bricks in the network. The numbers in these parameter flags are associated with bricks from left to right and top to bottom in the network.

```
(NPAR(I), I=1,76)
(Free Format)
```

Record number (10+NRES+1 through 10+NRES+NBLOC)

A record is read for each block in the resistivity section. These records contain the brick indices for each block. The blocks may contain up to seven bricks. If a block contains less than seven bricks, the index numbers should be zero filled.

```
(NBRIC(I,J), J=1,7)
(7I2)
```

NBRIC(1,J) - (integer) are the seven brick indices for block number 1.

NBRIC(2,J) - (integer) are the seven brick indices for block number 2.

etc.

Record number (11+NRES+NBLOC)

This record contains the block inversion flags. The flags are stored in ascending order. If NITSS(I) is 0, the Ith block resistivity is constrained. If the inversion flag is 1, the resistivity of the corresponding block is unconstrained.

(NITSS(I), I=1,NBLOC)
(40I2)

NITSS(1) - (integer) is the block inversion flag for block number 1.

NITSS(2) - (integer) is the block inversion flag for block number 2.

NITSS(NBLOC) - (integer) is the block inversion flag for block number NBLOC.

Record number (12+NRES+NBLOC) (conditional)

This record is conditional on the value of IDELV (see record number 8). If IDELV is 0, then record number 12+NRES+NBLOC is omitted. If IDELV is 1, then this record is inserted and contains the new network spacings for the vertical direction in the network. There are a total of 12

spacings. The spacings are ordered from the top of the network to the bottom. DELV(1) and DELV(13) are constants and are equal to zero.

(DELV(I), I=2,12)
(free format)

DELV(2) - (real) is the vertical spacing of the nodes in the top row of cells in the network.

DELV(3) - (real) is the vertical spacing of the nodes in the second row of cells in the network.

DELV(12) - (real) is the vertical spacing of the nodes in the eleventh and bottom row of cells in the network.

Appendix A
Sample Execution

Sample Input

An example of the execution of IPINV.REV1 is provided for the user in this appendix. The program requires a sequential formatted input file called IPINV.INPUT. A listing of the sample input file is provided below. There are 77 lines of formatted data in 52 columns in the sample input file that is used.

Col. --12345678901234567890123456789012345678901234567890

[illegible]

Each time a READ statement in the program is executed a record is read from this file. Some records use more than one line of the input file. There are 44 records in this sample input file.

Record number 1 requires line number 1. NIT receives the value of 2. The number of iterations allowed in the inversion is 2. ISI receives the value of 1. The program uses a Gaussian step-size truncation. NOPT receives the value of 0. The value of 0 allows the computation of both a forward and inverse problem. MQT receives the value of

0. Marquardt step is turned off and the program uses the value of ISI to determine the step mode. NNSEP receives the value of 6. The number of dipole separations for inversion is 6. This will require 45 observed resistivity values (see record number 4 below) and 45 weights (see record number 5 below). NDERP (the derivative flag) receives the value of 2. A value of 2 tells the program to use log derivatives in the inversion.

Record number 2

The conditional record number 2 is omitted because the value of MQT in the previous record was set to 0.

Record number 3

Record number 3 requires line number 2 in the input file. Record number 3 is the array of brick derivative printing flags. All the flags receive the value of 0, which turns off the printing of all the derivatives for all the bricks.

Record number 4

Record number 4 requires lines 3 through 8. This record is the observed resistivity values of the dipole-dipole pseudosection. The number of separations was set at a value of 6 (see record number 1 above). A separation of 6 requires 45 resistivity values for the pseudosection.

Record number 5

Record number 5 requires lines 9 through 14 in the input file. This record contains the data weights for the observed resistivities that were read in the previous record. The weights are set with a value of 0 or 1. All of the data weights have the value of 1 in this sample problem. This means that all the resistivity values are assumed to be correct and all are used in the inversion process.

Record number 6

Record number 6 requires line number 15 in the input file. VVAR receives a value of 0. This indicates that it is not possible to estimate the a priori variance of the data. The program will use an estimated variance from the square error.

Record number 7

Record number 7 requires line number 16 in the input file. LPR is the first variable in the record, and it receives the value of 0. A value of 0 indicates that there is to be no printing of the potentials at transmitter locations. NBL receives the value of 15. Thus 15 bricks will be required to construct the inversion blocks. NBLOC receives the value of 15. Thus 15 blocks will be inverted.

Record number 8

Record number 8 requires line number 17 in the input file. NRES receives the value of 17. This means that there are 17 distinct resistivity units in the cross section of the network. IDELV receives the value of 0 which tells the program to accept the default vertical cell dimensions as set by the program rather than reading in new values.

Record number 9

Record number 9 requires line number 18 of the input file. This record is an identifier for the sample model that is being processed.

Record number 10 through 26

Records 10 through 26 require lines 19 through 35 respectively. These records contain the resistivity values of each distinct resistivity unit in the cross section and a code number for each resistivity. In the sample input shown index number 1 corresponds to a resistivity of 113.7 ohm-meters. Each cell of the network will be coded with index numbers of 1 through 17 which correspond to the resistivities that are read in records 10 through 26. There are 17 resistivity records read in this part of the file, a number which corresponds to the value of NRES in

record number 8 above.

Record number 27

The reading of this record requires lines 36 through 57 in the input file. These lines are the cell code numbers of records 10 to 26. The cell code numbers are stored in the input file as 22 lines of 26 two-digit integer numbers. Each row of the network uses 2 lines of the input file for cell codes.

Record number 28

The reading of record number 28 requires lines 58 through 61 of the input file. The numbers that are read in this record are the brick parameter flags. In the sample input file shown above, brick number 24 is defined as the first brick whose resistivity is not held fixed and it is defined as cross section unit number 1.

Record number 29 through 43

The reading of records 29 through 43 requires lines 62 through 76 in the input file. The values that are read in these records contain the brick indices for the bricks contained in each block. In the sample input shown above, each block is made up of 1 brick. Block number 1 has unit code number 1, which corresponds to brick number 24. Block number 2 has unit code number 2 which corresponds to brick number 25, etc.

Record number 44

The reading of record number 44 requires line number 77 in the input file. In the sample input shown above, all the blocks have unconstrained resistivities in the inversion process because all the block inversion flags are 1.

Record number 45

The conditional record number 45 is omitted because the value of IDELV in record number 8 is 0.

Sample Output

The output from IPINV is written to a formatted sequential file called IPINV.OUTPUT. The program automatically searches for a file called IPINV.OUTPUT and if found, the file is deleted and a new one created. If IPINV.OUTPUT doesn't exist, a new file is created. The output file is opened on logical unit number 9. It is possible to change the logical unit number by changing the statement in the IPINV main program where LUD is defined.

The sample output for the input file described above is given in the following pages in reduced form.

.)

•

)

)

NBRIC VALUES 1' FOLLOW:

DATA CHANGE VECTOR IS :

PERCENT RESIDUAL FOR MODEL 1 IS 5.1

DERIVATIVES OF DATA W. R. T. BLOCK 1 RESISTIVITY ARE :

DERIVATIVES OF DATA W. R. T. BLOCK 2 RESISTIVITY ARE :

DERIVATIVES OF DATA W. R. T. BLOCK 3 RESISTIVITY ARE :

DERIVATIVES OF DATA W. R. T. BLOCK 4 RESISTIVITY ARE :

-0.00976	-0.02886	-1.51359	16.85945	6.71405	0.64093	-0.79391	-0.01760
-0.02360	-0.00776	-0.17341	-0.65497	7.95438	-0.90765	0.27581	4.60211
-1.34881	-0.03668	-0.05884	-2.15329	2.52100	-1.41632	-0.27702	0.08764

5.44669	-1.67400	-0.05426	7.64894	-0.66685	-0.15470	-1.22115	-0.50199
5.93001	-1.91477	-2.31916	-0.05888	-0.57687	-1.06543	-0.91843	6.35569
-0.19492	-0.22082	-0.43583	-0.95741	-1.19769			

DERIVATIVES OF DATA W.R.T. BLOCK 5 RESISTIVITY ARE :

-0.00833	-0.01310	-0.28096	3.39571	3.68359	4.27893	-4.05987	-0.07386
-0.08063	-0.02386	-0.09857	-0.17946	3.81128	-3.70044	-0.65514	22.35257
-6.32161	-0.14284	-0.19268	-0.73656	2.07342	-4.30814	-0.69239	-3.05517
22.90729	-7.30911	-0.20033	8.42641	-2.07084	-0.45008	-2.59211	-2.83659
22.18826	-7.95065	-7.46999	-2.18356	-1.55715	-2.05552	-2.64066	21.88825
-0.62065	-0.64759	-1.13895	-1.70615	-2.53428			

DERIVATIVES OF DATA W.R.T. BLOCK 6 RESISTIVITY ARE :

-0.86467	1.11146	5.09141	3.14528	-0.20429	-0.05856	-0.11669	-0.01192
-0.03012	-0.01399	6.62884	2.77244	2.06122	6.78086	0.00898	-0.60551
-0.28107	-0.03190	-0.08661	8.34154	0.04977	0.38305	2.29335	0.76305
-0.87851	-0.45968	-0.05628	-0.72532	-0.77864	-0.02228	14.15287	1.18036
-1.11762	-0.63674	-2.93822	-0.24341	-0.35429	15.11923	1.58930	-1.32741
-0.86949	-1.38656	-0.48023	16.09578	1.96958			

DERIVATIVES OF DATA W.R.T. BLOCK 7 RESISTIVITY ARE :

-0.23333	-0.28129	3.64412	3.38699	0.99870	-0.09210	-0.16018	-0.01471
-0.03441	-0.01565	-0.96945	2.44551	2.99595	4.06656	0.61169	-0.70016
-0.36063	-0.03813	-0.09739	8.88600	1.03218	1.07401	1.01434	4.58844
-0.87911	-0.55952	-0.06558	2.75695	0.10509	0.10326	4.89412	5.58311
-0.98706	-0.74516	-0.12110	-0.03411	0.08206	4.77094	6.46386	-1.06123
-0.19788	-0.31793	-0.05372	4.73072	7.25659			

DERIVATIVES OF DATA W.R.T. BLOCK 8 RESISTIVITY ARE :

-0.11550	-0.20785	0.58765	4.45883	3.41982	-0.19077	-0.35143	-0.02584
-0.05184	-0.02167	-1.11885	0.53077	4.34132	5.61204	1.33185	-1.18769
-0.71330	-0.06245	-0.14097	2.42004	1.73795	3.24656	1.22443	7.07165
-1.30670	-1.01590	-0.10209	5.02620	1.04243	0.60044	4.80879	7.30326
-1.29397	-1.27114	2.50161	0.18730	2.15198	4.25414	7.41730	-1.25600
0.46204	0.71905	1.84195	3.86598	7.57380			

DERIVATIVES OF DATA W.R.T. BLOCK 9 RESISTIVITY ARE :

-0.05455	-0.07848	-0.10743	2.70461	4.46089	0.38101	-0.72362	-0.05398
-0.09693	-0.03747	-0.50071	0.02390	3.59333	2.35914	1.62825	3.83207
-1.28046	-0.12130	-0.25364	0.52414	1.86143	0.78020	0.20862	7.97672
5.19838	-1.61307	-0.18778	6.92800	0.17730	-0.11559	0.29849	7.80866
6.21316	-1.83687	0.34465	-0.08109	-0.86634	0.10247	7.57808	7.04861
-0.32932	-0.49955	-0.78997	0.02350	7.47865			

DERIVATIVES OF DATA W.R.T. BLOCK 10 RESISTIVITY ARE :

-0.03074	-0.03859	-0.16759	1.16010	3.27238	1.44855	-1.12660	-0.10794
-0.15062	-0.05038	-0.28963	-0.10698	2.01048	2.82997	0.98277	10.35279
-1.53459	-0.21372	-0.36857	-0.33616	1.26338	1.70285	-0.05781	3.82063
11.77572	-1.45941	-0.30235	5.52591	0.65849	-0.22144	-1.26010	3.10681
12.34386	-1.27070	2.16531	-0.12317	-1.47314	-1.42402	2.57299	12.84269
-0.48300	-0.72759	-1.32837	-1.45900	2.27052			

DERIVATIVES OF DATA W.R.T. BLOCK 11 RESISTIVITY ARE :

-0.12186	-0.04180	0.26196	0.42019	-0.10389	-0.03033	-0.07595	-0.00812
-0.02094	-0.00955	-0.03763	0.33927	1.20610	1.33339	-0.04571	-0.37833
-0.18297	-0.02145	-0.05930	1.45070	0.70855	2.00293	0.52282	-0.24778
-0.55562	-0.29656	-0.03735	2.30954	0.90288	0.59359	3.56813	-0.21296
-0.70417	-0.40630	2.61246	0.24097	3.39954	3.88546	-0.15063	-0.82822
0.67898	1.30724	3.26609	4.17633	-0.07971			

DERIVATIVES OF DATA W.R.T. BLOCK 12 RESISTIVITY ARE :

-0.00222	-0.00137	-0.00020	0.01672	0.01369	-0.00125	-0.00606	-0.00079
-0.00226	-0.00113	0.01381	0.01367	0.03340	0.10975	0.01227	-0.01262
-0.01510	-0.00217	-0.00660	0.09691	0.01870	0.06898	0.05180	0.12173
-0.01654	-0.02506	-0.00387	0.07152	0.01322	0.02499	0.39404	0.16847

-0.01841	-0.03481	0.01516	0.00353	0.17313	0.45647	0.21340	-0.01885
-0.00104	0.02435	0.17960	0.51333	0.25384			

DERIVATIVES OF DATA W.R.T. BLOCK 13 RESISTIVITY ARE :

-0.12967	-0.15456	-0.10964	1.32495	0.34760	-0.12094	-0.32190	-0.02991
-0.06528	-0.02752	-1.11603	-0.08191	2.24966	4.71440	0.20010	-1.23003
-0.70131	-0.07280	-0.17450	-0.32611	1.23976	4.84898	1.60171	1.30501
-1.62318	-1.03257	-0.11812	4.46019	2.16717	1.34556	8.61687	1.46791
-1.85359	-1.30303	6.98119	0.56555	6.56844	8.27076	1.56681	-1.99153
1.81790	2.77702	5.83940	7.85715	1.63566			

DERIVATIVES OF DATA W.R.T. BLOCK 14 RESISTIVITY ARE :

-0.01173	-0.01085	0.00212	0.12519	0.03131	-0.00384	-0.03154	-0.00389
-0.01039	-0.00585	-0.08986	0.00899	0.27697	0.41187	0.04691	0.09139
-0.01703	-0.00712	-0.02617	0.07934	0.20075	0.45961	0.11265	0.39769
0.35805	0.04890	-0.00812	0.96562	0.25579	0.08727	0.48373	0.55920
0.66609	0.14017	1.07775	0.04245	0.27972	0.24028	0.68863	0.94990
0.16936	0.11952	0.03736	0.04488	0.78530			

DERIVATIVES OF DATA W.R.T. BLOCK 15 RESISTIVITY ARE :

-0.02306	-0.02444	-0.06655	0.10945	0.28163	0.00726	-0.16857	-0.02049
-0.04108	-0.01709	-0.21073	-0.05881	0.22216	1.37216	0.21899	0.24761
-0.19998	-0.03675	-0.09273	-0.28075	0.16325	1.34240	0.54897	1.36840
0.61468	-0.07522	-0.04241	0.80923	0.66491	0.42280	2.75841	1.34197
0.98316	0.11773	2.53865	0.18786	1.90350	2.11019	1.22492	1.29396
0.67502	0.82505	1.31110	1.51688	1.10716			

THE GAUSS PARAMETER CHANGES FOR THE BLOCKS OF MODEL 1 ARE:

BLOCK NO. PARAMETER CHANGE

1	-0.08767
2	-0.09389
3	-0.11647
4	0.03311
5	-0.15575
6	0.91115
7	0.04003
8	-1.27938
9	0.04659
10	0.94406
11	-0.20423
12	2.66941
13	0.89048
14	0.85180
15	-0.24933

THE CONSTRAINED PARAMETER CHANGES FOR THE BLOCKS OF MODEL 1 ARE:

1	-0.08767
2	-0.09389
3	-0.11647
4	0.03311
5	-0.15575
6	0.91115
7	0.04003
8	-1.27938
9	0.04659
10	0.94406
11	-0.20423
12	2.66941
13	0.89048
14	0.85180
15	-0.24933

PARAMETER CORRELATION MATRIX FOLLOWS

1.00E 00							
-1.98E-01	1.00E 00						
1.33E-01	-3.83E-01	1.00E 00					
-6.07E-01	-7.79E-02	-2.66E-01	1.00E 00				
2.93E-01	1.45E-01	2.89E-01	-7.75E-01	1.00E 00			
-7.00E-01	4.57E-01	-1.48E-02	1.03E-01	1.36E-01	1.00E 00		
-1.15E-02	-7.04E-01	9.51E-02	5.15E-01	-6.17E-01	-5.16E-01	1.00E 00	
2.80E-01	3.52E-01	-7.72E-02	-6.65E-01	7.68E-01	1.29E-01	-8.25E-01	1.00E 00
-1.64E-01	-1.30E-01	-1.89E-01	4.54E-01	-6.57E-01	9.33E-05	5.64E-01	-8.41E-01
1.00E 00							
1.65E-03	-1.01E-01	5.75E-02	1.99E-01	-3.90E-01	-3.27E-01	2.26E-01	-6.52E-02
-2.86E-01	1.00E 00						
6.57E-01	1.48E-02	3.38E-02	-6.67E-01	5.03E-01	-2.47E-01	-5.01E-01	6.97E-01
-4.62E-01	-1.28E-01	1.00E 00					
5.24E-02	4.11E-01	-1.14E-01	6.18E-02	1.49E-02	-1.70E-01	-3.24E-01	2.79E-01
-4.17E-01	3.91E-01	1.21E-01	1.00E 00				
-4.66E-01	-1.25E-01	9.62E-02	5.82E-01	-6.02E-01	8.67E-02	6.42E-01	-8.62E-01
6.68E-01	1.34E-01	-9.17E-01	-2.29E-01	1.00E 00			
2.85E-02	2.83E-01	4.20E-01	-4.80E-01	6.80E-01	4.06E-01	-6.83E-01	5.98E-01
-6.14E-01	-2.38E-01	3.24E-01	2.36E-01	-4.43E-01	1.00E 00		
5.30E-02	-2.02E-01	-4.69E-01	2.20E-01	-1.03E-01	-1.40E-01	1.17E-01	7.31E-02
5.60E-02	-2.01E-01	2.79E-01	-2.00E-01	-3.93E-01	-3.87E-01	1.00E 00	

THE BLOCK RESISTIVITY STD DEV. ARE:

0.33750	0.44458	0.38724	0.56915	0.49220	0.78141	2.26586	3.43749
1.97089	0.83446	4.30231	19.67195	3.52044	8.65302	3.38539	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.00000							

FOLLOWING APP. RES. FOR ITER. 2 AND NSWITCH 1
 THE VALUE OF STEPS IS 0.21353

RESISTIVITY CODE -	NUMBER	RESISTIVITY
	1	111.6
	2	103.4
	3	103.8
	4	106.7
	5	104.7
	6	42.8
	7	18.2
	8	9.5
	9	22.0
	10	41.5
	11	66.9
	12	2383.4
	13	14.2
	14	284.5
	15	61.3
	16	100.0
	17	10.0

RESISTIVITY SECTION

92.6410828	19.7410698	19.6982346	20.7465019	50.1379623	58.6365891	55.2511520	16.4065247	36.3965149	17.4676552
98.9193726	85.2730255	83.7625122	83.7835693	111.1423187	24.7153015	40.3927002	78.2085571	17.3642578	37.2449341
55.1776199	11.6530609	89.2958679	52.9977722	21.2171173	121.0842133	101.0693665	98.4087067	97.7991486	25.8544922
43.9390182	118.1650543	52.3807526	14.9434090	39.5456314	14.7224808	54.5061493	93.7391357	59.2601395	21.8782425
127.1028442	104.7046356	100.8409882	46.2231445	117.4334412	69.9728394	52.0569992	17.2142029	10.4054260	82.5946503
53.9117432	92.7314758	63.3154678	22.0038986	130.3295593	106.5946198	118.6907806	67.0559540	70.8132477	61.3630219

4. 5992374 56. 9005737 85. 9830170 53. 6693878 92. 9058075 66. 9215698 22. 3421021 134. 3421936 66. 5491333 71. 6277008
80. 3015442 16. 9695740 26. 0176849 56. 3336792 90. 0699921 54. 5488586 94. 0263977 70. 3751221 22. 9190826
6-----5-----4-----3-----2-----1-----0-----1-----2-----3-----4-----5-----6

92 19 19 20 50 58 55 16 36 17 98 85 83 83
111 24 40 78 17 37 55 11 89 52 21 121 101 98 97
25 43 118 52 14 39 14 54 93 59 21 127 104 100
46 117 69 52 17 10 82 53 92 63 22 130 106
118 67 70 61 4 56 85 53 92 66 22 134
66 71 80 16 26 56 90 54 94 70 22

DATA CHANGE VECTOR IS :

-0. 0259499 0. 0395137 0. 0245531 -0. 0452155 -0. 0504022 0. 0516459 -0. 0110143 -0. 0038884 -0. 0012083 -0. 0008571
0. 0565915 0. 0511334 0. 0120706 0. 0803730 -0. 0593692 0. 0600435 -0. 0075622 -0. 0103389 -0. 0031832 0. 0423993
0. 0103702 0. 0628996 0. 0912047 -0. 0522424 0. 0626086 -0. 0027097 -0. 0128817 0. 0413075 0. 0383126 0. 0275302
0. 1213335 -0. 0570930 0. 0633184 0. 0013332 0. 0602904 -0. 0220528 0. 0404624 0. 1181229 -0. 0627602 0. 0634495
-0. 0041169 -0. 0084374 0. 0287297 0. 1117357 -0. 0654075 0. 0000000 0. 0000000 0. 0000000 0. 0000000 0. 0000000
0. 0000000 0. 0000000 0. 0000000 0. 0000000 0. 0000000 0. 0000000 0. 0000000 0. 0000000 0. 0000000 0. 0000000

PERCENT RESIDUAL FOR MODEL 2 IS 4.2

DERIVATIVES OF DATA W.R.T. BLOCK 1 RESISTIVITY ARE :

-1. 49204 2. 49716 4. 84961 17. 58311 -1. 94682 -0. 02973 -0. 03293 -0. 00307
-0. 00774 -0. 00359 11. 11700 -0. 71604 -3. 10042 12. 88531 -0. 56763 -0. 30608
-0. 07910 -0. 00822 -0. 02233 -2. 86460 -1. 05850 -5. 10218 3. 08074 -3. 08365
-0. 43955 -0. 12955 -0. 01456 -2. 87031 -0. 94521 -1. 39439 16. 76109 -3. 27618
-0. 55857 -0. 18019 -2. 41573 -0. 22557 -7. 51997 17. 15438 -3. 43315 -0. 66710
-0. 58857 -1. 16747 -7. 40043 17. 75616 -3. 60224

DERIVATIVES OF DATA W.R.T. BLOCK 2 RESISTIVITY ARE :

-0. 12835 -1. 01371 15. 80563 6. 96743 2. 16653 -0. 08939 -0. 03294 -0. 00201
-0. 00433 -0. 00187 -4. 40572 4. 90940 0. 15065 -0. 53383 0. 81714 -0. 66728
-0. 06962 -0. 00506 -0. 01207 14. 10484 -0. 24897 -0. 74767 -0. 70656 5. 26458
-0. 81257 -0. 10362 -0. 00856 -0. 90837 -0. 18290 -0. 20688 -4. 66416 6. 04098
-0. 90593 -0. 13492 -0. 50732 -0. 03850 -1. 07413 -5. 14029 6. 73267 -0. 98684
-0. 10532 -0. 19343 -1. 02932 -5. 56467 7. 38905

DERIVATIVES OF DATA W.R.T. BLOCK 3 RESISTIVITY ARE :

-0. 02431 -0. 15237 1. 58116 7. 73728 17. 22799 -0. 64870 -0. 13522 -0. 00434
-0. 00712 -0. 00265 -0. 75746 0. 86879 1. 17303 0. 38742 3. 10630 -4. 11941
-0. 24931 -0. 00979 -0. 01863 3. 37198 -0. 21921 -0. 25794 -0. 16340 13. 05501
-4. 56652 -0. 33133 -0. 01533 -1. 46348 -0. 13830 -0. 04987 -1. 11058 11. 98052
-4. 69522 -0. 39796 -0. 49491 -0. 01346 -0. 24382 -1. 20407 11. 13629 -4. 81640
-0. 04505 -0. 05732 -0. 21255 -1. 27971 10. 72877

DERIVATIVES OF DATA W.R.T. BLOCK 4 RESISTIVITY ARE :

-0. 00965 -0. 02770 -1. 45613 16. 32406 6. 47515 0. 66837 -0. 83924 -0. 01903
-0. 02552 -0. 00836 -0. 16728 -0. 61588 7. 75951 -1. 16957 0. 32411 4. 76437
-1. 43017 -0. 03967 -0. 06358 -1. 99330 2. 44638 -1. 55578 -0. 29172 0. 41035
5. 60725 -1. 77955 -0. 05869 7. 45447 -0. 71873 -0. 16851 -1. 25672 -0. 15357
6. 07735 -2. 03941 -2. 50450 -0. 06433 -0. 62481 -1. 08226 -0. 55703 6. 49378
-0. 21347 -0. 24114 -0. 47069 -0. 96250 -0. 82562

DERIVATIVES OF DATA W.R.T. BLOCK 5 RESISTIVITY ARE :

-0. 00919 -0. 01419 -0. 27849 3. 51768 3. 58929 4. 36221 -4. 08735 -0. 07873
-0. 08735 -0. 02602 -0. 10857 -0. 17816 3. 96409 -4. 03866 -0. 70045 22. 97369
-6. 36777 -0. 15286 -0. 20932 -0. 73739 2. 15564 -4. 45626 -0. 77458 -3. 24861
23. 64413 -7. 36680 -0. 21503 8. 83116 -2. 11372 -0. 49774 -2. 94652 -3. 00338
22. 98931 -8. 01755 -7. 63021 -0. 20139 -1. 75383 -2. 34986 -2. 78561 22. 74547
-0. 68237 -0. 72336 -1. 29300 -1. 95928 -2. 66612

DERIVATIVES OF DATA W.R.T. BLOCK 6 RESISTIVITY ARE :

-0.86421	1.16408	5.14666	2.98810	-0.11136	-0.05478	-0.11555	-0.01173
-0.02979	-0.01379	6.93697	2.77576	1.65957	6.73635	0.05028	-0.58414
-0.27911	-0.03141	-0.08561	8.47568	-0.15989	-0.13797	2.29918	1.01657
-0.85355	-0.45736	-0.05544	-1.28131	-1.04111	-0.13040	14.49260	1.46763
-1.09146	-0.63458	-3.64706	-0.29845	-0.74483	15.53874	1.90482	-1.30126
-1.02638	-1.63014	-0.82304	16.58215	2.31154			

DERIVATIVES OF DATA W.R.T. BLOCK 7 RESISTIVITY ARE :

-0.22603	-0.29518	3.79895	3.16589	1.01423	-0.08695	-0.16191	-0.01484
-0.03486	-0.01583	-1.02798	2.46872	2.87392	3.82060	0.61532	-0.67799
-0.36727	-0.03856	-0.09870	8.86761	0.96152	0.99925	0.96764	4.63456
-0.87074	-0.57320	-0.06643	2.50190	0.07384	0.10164	4.80593	5.63010
-0.99950	-0.76664	-0.22614	-0.03488	0.13566	4.74893	6.50673	-1.09479
-0.20303	-0.30542	0.02838	4.75813	7.29443			

DERIVATIVES OF DATA W.R.T. BLOCK 8 RESISTIVITY ARE :

-0.12571	-0.23205	-0.40243	4.90569	3.20977	-0.22008	-0.37997	-0.02809
-0.05736	-0.02445	-1.26563	0.37948	4.91154	7.25179	1.21451	-1.44743
-0.78166	-0.06870	-0.15725	1.79800	1.95624	4.69778	1.65412	6.33336
-1.67139	-1.12734	-0.11324	9.55758	1.56965	0.92934	6.60732	6.51793
-1.74918	-1.42413	3.88155	0.30534	3.47814	5.98251	6.60549	-1.78485
0.79138	1.23079	3.03597	5.35491	6.73545			

DERIVATIVES OF DATA W.R.T. BLOCK 9 RESISTIVITY ARE :

-0.05488	-0.07879	-0.04969	2.51313	4.52968	0.39235	-0.73562	-0.05277
-0.09457	-0.03676	-0.51381	0.05770	3.43418	2.13183	1.62757	3.89623
-1.29078	-0.11833	-0.24768	0.63663	1.80789	0.67098	0.15184	7.88740
5.27914	-1.61418	-0.18302	6.87427	0.15438	-0.13154	0.00393	7.69515
6.31155	-1.82800	0.33620	-0.08277	-0.94556	-0.19314	7.45850	7.16549
-0.32693	-0.51314	-0.87707	-0.26851	7.36119			

DERIVATIVES OF DATA W.R.T. BLOCK 10 RESISTIVITY ARE :

-0.03005	-0.03761	-0.13189	1.25947	3.02482	1.49462	-1.08825	-0.10899
-0.15306	-0.05108	-0.28969	-0.08116	2.13083	2.37472	0.89143	10.78805
-1.45520	-0.21603	-0.37444	-0.24097	1.33468	1.38882	-0.27125	3.52890
12.32385	-1.34542	-0.30586	5.88362	0.56255	-0.35268	-2.16416	2.87880
12.96683	-1.12699	2.01507	-0.16848	-2.00484	-2.23417	2.39611	13.52958
-0.60992	-0.92576	-1.78314	-2.20125	2.12692			

DERIVATIVES OF DATA W.R.T. BLOCK 11 RESISTIVITY ARE :

-0.12063	-0.03129	0.23650	0.47690	-0.08640	-0.03355	-0.08544	-0.00895
-0.02298	-0.01042	0.06543	0.33736	1.23043	1.54401	-0.03388	-0.42129
-0.20535	-0.02359	-0.06495	1.47753	0.70957	2.09841	0.60076	-0.14329
-0.61909	-0.33217	-0.04101	2.32044	0.91828	0.61819	4.13035	-0.08378
-0.78516	-0.45447	2.65050	0.24223	3.56928	4.49393	0.00010	-0.92425
0.67753	1.31738	3.42937	4.82682	0.09029			

DERIVATIVES OF DATA W.R.T. BLOCK 12 RESISTIVITY ARE :

-0.00105	-0.00061	-0.00155	0.01243	0.00891	-0.00089	-0.00385	-0.00049
-0.00139	-0.00069	0.01126	0.00743	0.01980	0.07363	0.00755	-0.00962
-0.00957	-0.00133	-0.00404	0.05560	0.00975	0.04340	0.03405	0.07450
-0.01316	-0.01586	-0.00237	0.03588	0.00703	0.01569	0.25933	0.10263
-0.01540	-0.02199	0.00451	0.00189	0.10915	0.30024	0.12964	-0.01668
-0.00181	0.01339	0.11341	0.33753	0.15398			

DERIVATIVES OF DATA W.R.T. BLOCK 13 RESISTIVITY ARE :

-0.11860	-0.14743	-0.12471	1.34663	0.32013	-0.11997	-0.30460	-0.02726
-0.05868	-0.02421	-1.08689	-0.09813	2.16026	4.63337	0.17841	-1.25013
-0.65620	-0.06548	-0.15498	-0.41846	1.18353	4.56234	1.55454	1.18067
-1.63595	-0.95346	-0.10498	4.33680	2.03007	1.25065	8.49901	1.31149
-1.84792	-1.18847	6.65408	0.52170	6.18527	8.06814	1.38102	-1.96441
1.69683	2.57927	5.43568	7.56825	1.42426			

DERIVATIVES OF DATA W.R.T. BLOCK 14 RESISTIVITY ARE :

-0.01083	-0.01059	0.00110	0.13520	0.01798	-0.00460	-0.02404	-0.00313
-0.00890	-0.00522	-0.08863	0.00656	0.27341	0.40140	0.03442	0.08196
-0.00231	-0.00555	-0.02259	0.06107	0.19493	0.43749	0.10010	0.30496
0.34148	0.06760	-0.00596	0.93897	0.24289	0.07728	0.40721	0.43807
0.64158	0.16028	1.03022	0.03768	0.23424	0.16061	0.54583	0.91771
0.15176	0.09946	0.00798	-0.03438	0.62668			

DERIVATIVES OF DATA W.R.T. BLOCK 15 RESISTIVITY ARE :

-0.02545	-0.02784	-0.06739	0.15086	0.26373	0.00883	-0.16477	-0.02179
-0.04476	-0.01869	-0.24143	-0.05931	0.28021	1.50838	0.20970	0.29187
-0.18686	-0.03980	-0.10220	-0.28510	0.20009	1.44500	0.58661	1.33365
0.71282	-0.04823	-0.04697	0.98346	0.71280	0.44203	2.96750	1.32008
1.13199	0.16056	2.74178	0.19487	1.99897	2.26188	1.21257	1.48457
0.70288	0.85673	1.37440	1.61278	1.10018			

THE GAUSS PARAMETER CHANGES FOR THE BLOCKS OF MODEL 2 ARE:

BLOCK NO. PARAMETER CHANGE

1	-0.10523
2	-0.06975
3	-0.10374
4	0.04930
5	-0.12575
6	0.78234
7	-0.09760
8	-0.75755
9	-0.24306
10	0.86689
11	-0.31399
12	0.77046
13	1.11281
14	0.41989
15	-0.18346

THE CONSTRAINED PARAMETER CHANGES FOR THE BLOCKS OF MODEL 2 ARE:

1	-0.10523
2	-0.06975
3	-0.10374
4	0.04930
5	-0.12575
6	0.78234
7	-0.09760
8	-0.75755
9	-0.24306
10	0.86689
11	-0.31399
12	0.77046
13	1.11281
14	0.41989
15	-0.18346

PARAMETER CORRELATION MATRIX FOLLOWS

1.00E 00								
-1.58E-01	1.00E 00							
1.67E-01	-3.01E-01	1.00E 00						
-6.19E-01	-1.10E-01	-3.92E-01	1.00E 00					
3.30E-01	1.48E-01	4.61E-01	-7.95E-01	1.00E 00				
-7.83E-01	3.90E-01	-6.09E-02	2.69E-01	-5.02E-02	1.00E 00			
5.55E-02	-7.96E-01	3.61E-03	4.34E-01	-4.95E-01	-4.53E-01	1.00E 00		
3.31E-01	3.66E-01	9.79E-02	-7.00E-01	7.44E-01	-7.94E-02	-7.06E-01	1.00E 00	
-7.49E-02	7.83E-02	-5.05E-01	2.70E-01	-4.94E-01	2.08E-01	1.18E-01	-5.33E-01	1.00E 00

-1.43E-01	-2.52E-01	1.42E-02	4.34E-01	-6.47E-01	-2.20E-01	5.16E-01	-4.62E-01
-2.10E-01	1.00E 00						
6.95E-01	1.73E-02	1.42E-01	-6.59E-01	4.61E-01	-4.25E-01	-3.62E-01	6.62E-01
-1.49E-01	-3.90E-01	1.00E 00					
-7.89E-03	4.60E-01	-5.68E-04	3.48E-02	4.39E-02	-1.41E-01	-3.43E-01	2.37E-01
-4.11E-01	1.91E-01	1.04E-03	1.00E 00				
-5.22E-01	-1.38E-01	-6.54E-02	6.19E-01	-5.92E-01	2.85E-01	5.14E-01	-8.62E-01
3.66E-01	4.82E-01	-9.04E-01	-1.35E-01	1.00E 00			
4.54E-02	2.86E-01	5.48E-01	-4.90E-01	6.74E-01	2.55E-01	-6.11E-01	5.28E-01
-4.97E-01	-4.78E-01	2.66E-01	3.00E-01	-3.98E-01	1.00E 00		
7.49E-02	-1.97E-01	-4.35E-01	1.93E-01	-9.39E-02	-1.37E-01	1.28E-01	1.30E-01
9.97E-02	-2.01E-01	2.99E-01	-2.79E-01	-4.20E-01	-3.83E-01	1.00E 00	

THE BLOCK RESISTIVITY STD DEV. ARE:

0.34524	0.45826	0.40128	0.60788	0.48635	0.77304	1.88038	2.30730
1.28304	0.93743	4.08976	31.56208	3.75181	8.96150	3.13985	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.00000							

Appendix B
PRIMOS Subroutines

IPINV uses a few PRIMOS IV subroutines to manage the input and output files. PRIME Computer Inc. has given permission to provide a description of the system routines that are used so that the program can be converted to other computer systems and other system routines substituted for the ones described in this appendix.

ERRPR\$

ERRPR\$ interprets a return code and, if 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).

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.

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)

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.

Appendix C
Program Conversion

IPINV.REV1 is operational on a Prime 400 computer system at the University of Utah Research Institute. It should be possible to convert the program to run on other systems. The program could be converted to most main frame systems such as those manufactured by UNIVAC, CDC, and IBM. The program could also be run on minicomputer systems that have the capability of spanning 64k memory segments. The Prime computer accomplishes this by using virtual memory. It may be difficult to implement the program on PDP 11 minicomputers because of the large data arrays that are contained in the common blocks.

Memory Requirements

IPINV.REV1 uses approximately 12k words (24k bytes) of memory for procedure (instructions) and approximately 100k words (200k bytes) for data storage. This would require approximately 56k words on a UNIVAC or an IBM system.

Prime computer extensions

There are a number of Prime extensions that are used in the code of IPINV.REV1. Some of the extensions are found on other computer systems and some are peculiar to Prime. The program has data arrays that are greater than 64k bytes. Thus the program will not run on minicomputers that do not have the capability of addressing across segment boundaries. The \$INSERT command is a convenience that is used on the Prime system. It is possible to make the program operable on other systems by substituting the actual code that is inserted by the \$INSERT command. There are several system routines that are used in the program, and a description of the routines is given in Appendix B.

\$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.

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 a segment boundary.

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 parenthesis, 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 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.

System Subroutines

The conversion of the IPINV program to another computer system will require the substitution of several system subroutines that provide file handling and program termination. Descriptions of the PRIMOS IV subroutines that are used are provided in Appendix B. The PRIMOS IV subroutines are TNOU, SRCH\$\$, TSRC\$\$, ERRPR\$, and EXIT. In some systems the call to EXIT should be replaced with the STOP statement in the source code. The other PRIMOS IV routines are called by three subroutines (OPNRED, OPNNEW, and CLOSEF), which open and close data files on logical

unit numbers. The program can be implemented on another system by substituting new routines for OPNRED, OPNNEW, and CLOSEF, or by substituting new routines for TNOU, SRCH\$\$, TSRC\$\$, and ERRPR\$.

REFERENCES

- Box, G. E. P., and Kanemaesu, H., 1972, Topics in model building, part II: on non-linear least squares: Tech. Rept. no. 321, Dept. of Statistics, University of Wisconsin, Madison, Wisconsin.
- Dey, A., and Morrison, H. F., 1976, Resistivity modeling for arbitrarily shaped two dimensional structures, part I: theoretical formulation: Lawrence Berkeley Laboratory report 5223, Berkeley, California.
- Madden, T. R., 1971, The resolving power of geoelectric measurements for delineating resistive zones within the crust, in The structure and physical properties of the earth's crust, Washington, D. C., AGU, Geophysical Monograph Series, v. 14, ed. J. G. Heacock, p. 95-105.
- Madden, T. R., 1972, Transmission systems and network analogies to geophysical forward and inverse problems: Report no. 72-3, Dept. of Geology and Geophysics, M.I.T., Cambridge, Mass.
- Pelton, W. H., Rijo, L., and Swift, C. M., Jr., 1978, Inversion of two-dimensional resistivity and induced-polarization data: Geophysics, v. 43, p. 788-803.
- Rijo, L., 1977, Modeling of electrical and electromagnetic data: Ph.D. dissertation, University of Utah, Salt Lake City, Utah.
- Tripp, A. C., Hohmann, G. W., and Swift, C. M., Jr., (in prep.), Two Dimensional Resistivity Inversion: to be submitted to Geophysics.