ORKUSTOFNUN
NATIONAL ENERGY AUTHORITY
GEOTHERMAL DIVISION

# One-dimensional Inversion of Central Loop Transient Electro-Magnetic Soundings

Computer Program, Description and User's Guide

Knútur Árnason

OS-88055/JHD-29 B November 1988

ORKUSTOFNUN - National Energy Authority
Geothermal Division
Grensásvegur 9
IS 108 Reykjavík
Iceland

Verknr.:440300

# One-dimensional Inversion of Central Loop Transient Electro-Magnetic Soundings

Computer Program, Description and User's Guide

Knútur Árnason

OS-88065/JHD-29 B November 1988

## ABSTRACT

This report describes a computer program for one-dimensional inversion of central loop transient electro-magnetic resistivity soundings. It is also meant as a user's guide to the program. A short description of the inversion method is followed by step by step instructions of how to run the program. The input and output files are described as well as plotting utilities for graphic display of the results.

CONTENTS

# 1. INTRODUCTION

The following is a user's guide and a short description of the program TINV (Transient electro-magnetic INVersion) for one-dimensional inversion of central loop transient electro-magnetic resistivity soundings. TINV is a non-linear least-squares inversion program using a Levenberg-Marquardt inversion algorithm together with a fast forward routine based on the linear filter method. The Levenberg-Marquardt inversion algorithm used in the program is described by H. K. Johansen (1977). The theoretical basis for the forward algorithm is described by K. Árnason (1988)

The inversion program TINV assumes that the apparent resistivity data is collected with equipment where the current in the transmitter loop is turned off linearly from its maximum value to zero and that the time values, at which the apparent resistivity values are given, are measured in seconds after the current has become zero ($t=0$ is at the end of the turn-off ramp). The program also assumes that the data is collected with a circular transmitter loop. If this is not the case the actual transmitter loop is simulated by a circular loop having the same area.

The program reads measured apparent resistivity data from an input file and prompts for an initial guess for a one-dimensional resistivity model. It iteratively adjusts the resistivity model to minimize the difference between the measured and the calculated apparent resistivity values. The results are written into two files, an output list file that can be typed on the screen and printed as a hard copy and an output plot file that can be plotted both on the screen and as a hard copy on a printer or a plotter.

The source code consists of 16 modules, the main program TINV.FOR and 15 subroutines. The modules are listed in Table 1. The program is written in standard FORTRAN 77 and the executable file TINV.EXE was made by compiling the source files by the Microsoft FORTRAN Optimizing Compiler, version 4.01, and linking by the Microsoft Object Linker. The program can be run on IBM PC or PC-compatible computers with an 8087 coprocessor. The program can be made to run on main frame computers by compiling and linking the FORTRAN source files but the plotting utilities (described below) only run on IBM PC and PC-compatibles.

In order to run the inversion program, the file TINV.EXE must be either in the working directory or a directory specified in the PATH-list. To plot the results the files SPLOT.BAT (screen plot), PPLOT.BAT (paper plot), RES.GRD (grid file) and

| MODULE | FUNCTION |
|---|---|
| TINV | Main program; inversion algorithm |
| TRDT | Reads input data from file and terminal |
| J1FLT | Stores J1 Hankel transform filter |
| COSFLT | Stores cosine transform filter |
| TFWD | Forward algorithm; calculates apparent resistivity curve from a given model |
| YCAL | Calculates frequency domain response for a given frequency |
| RECUR | Calculates kernel function for frequency domain response |
| SPLINE | Calculates second derivatives of frequency domain responses |
| SPLINT | Performs cubic spline interpolation between calculated frequency domain responses |
| TRF | Integrates sharp step response from t to t+TOFF to get ramped step response |
| CHSQ | Calculates chi-square sum |
| TDR | Calculates partial derivative matrix |
| SVDC | Performs a singular value decomposition on the partial derivative matrix |
| ORDW | Orders eigenvalues in increasing order |
| NEWP | Calculates increments to be added to model parameters to get a new model |
| WROT | Writes out results into output files |

**Table 1.** The modules of the program TINV

CENTERED.SYM (plot symbols) must be in the working directory and the files VIEW.EXE (screen plot) and PLOT.EXE (paper plot) must be either in the working directory or a directory specified in the PATH-list.

A slightly modified version of the forward routine of the inversion program has been made into a separate program called TEM (the executable file is TEM.EXE). TEM computes late time apparent resistivity curve corresponding to a one-dimensional resistivity model, read from the terminal and for time values equally distributed on log-scale with five points per power of e, over an interval specified by the user. The results are written into an output file (that has the right format for an input file for TINV) and a plot file that can be plotted in the same way as the output plot files from the inversion program TINV. The program TEM is a by-product of the inversion program, meant for simple model calculations, and will not be discussed further.

## 2. THE STRUCTURE OF THE PROGRAM

The backbone of the program TINV is a general non-linear least-squares inversion algorithm of the Levenberg-Marquardt type. The inversion algorithm is supplemented by routines for data input, TRDT, and output, WROT, a forward routine, TFWD, that calculates apparent resistivity values from a given one-dimensional resistivity model and a routine, TDR, that calculates the partial derivatives of the apparent resistivity with respect to the model parameters. The general structure of the program is shown on Figure 2.1. The least-squares algorithm, the forward routine, TFWD, and the partial derivative routine, TDR, will now be discussed briefly.

### 2.1 The inversion algorithm

The program TINV, like most inversion programs, works in such a way that it reads the measured data points (apparent resistivity curve) and prompts for a starting model. The interpreter guesses, by visual inspection of the data curve, the number of layers and initial model parameters i.e. the resistivity values and thicknesses of the layers. Each model parameter can either taken to be a free or a fixed parameter. The program iteratively adjusts the values of the free model parameters to get the best fit between the measured curve and the curve calculated from the model. It is important to realize that the program does not change the number of layers during the iteration process. It is therefore in most cases necessary to try models with different numbers of layers to find the model that best fits the data. It should also be kept in mind that the model resulting from the iterative inversion can depend on the initial guess. A poor

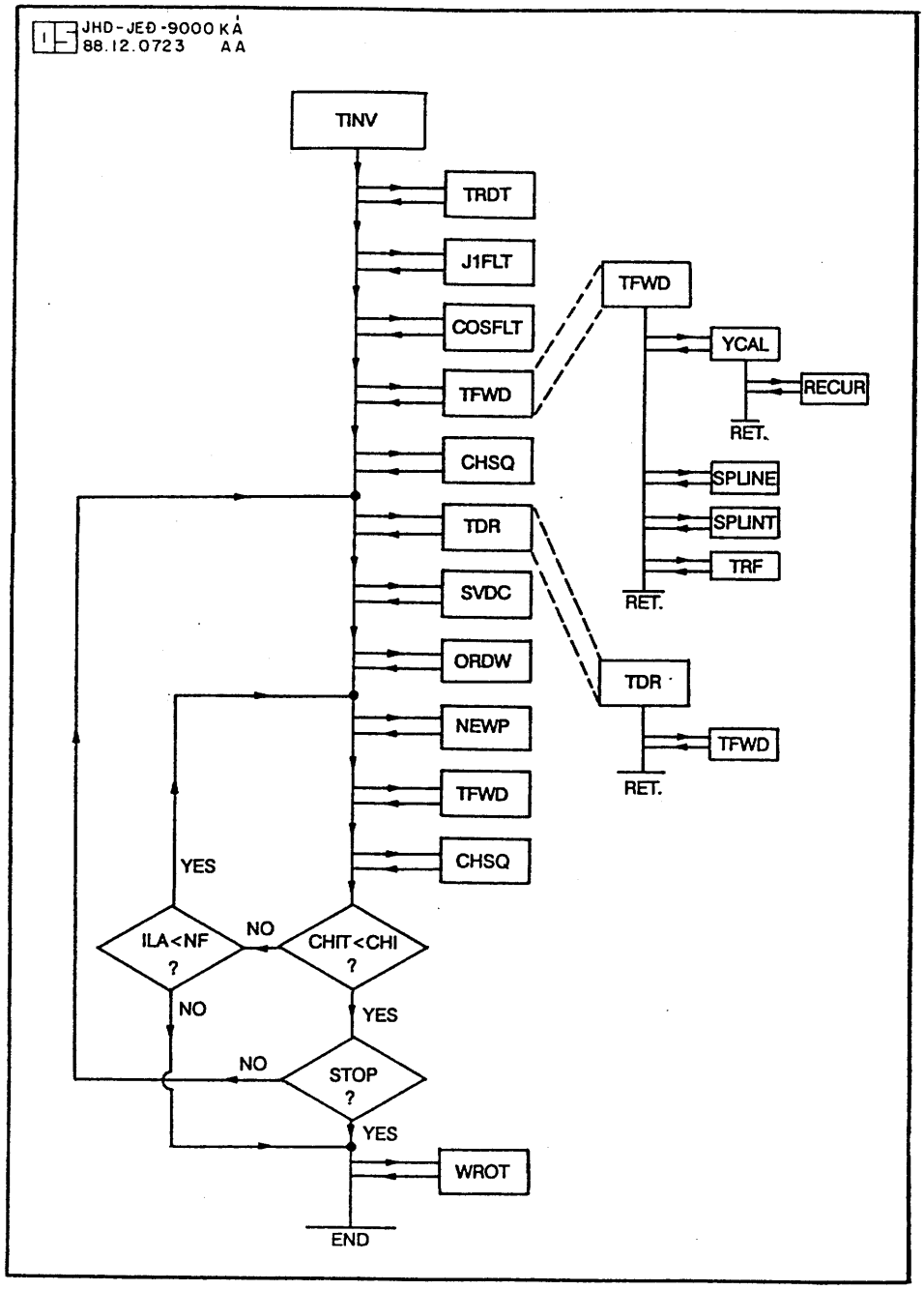


**Figure 2.1** Structure of inversion program

guess can lead the inversion process astray.

In the the inversion algorithm, all computations are done with data and model parameters on logarithmic form, that is to say $(\ln(t),\ln(\rho_a))$ is used instead of $(t,\rho_a)$ and the model parameters are kept as $P(i)=\ln(p(i))$ where the p(i) stand for the resistivity values and layer thicknesses. This is done because the non-linearity in the dependence

of the apparent resistivity on the model parameters is not as severe in the logarithmic as in the linear representation. The logarithmic representation furthermore prevents the occurrence of non-physical negative model parameters.

The quality of the fit between the measured and the calculated apparent resistivity values, calculated by the subroutine TFWD, is measured by the chi-square sum, CHI, which is calculated by the subroutine CHSQ. CHI is given, in terms of the natural logarithms of the measured and calculated apparent resistivity values, by the following formula:

$$CHI = \left[\frac{1}{ND}\sum_{1}^{ND}[(\ln(\rho_{am}) - \ln(\rho_{ac}))\cdot WPM]^2\right]^{1/2}$$

where ND is the number of data points and WPM is a weight factor (to be discussed later). The lower the CHI is, the better is the fit. If CHI is less than 0.1 it can be interpreted as the average fractional difference between the measured and the calculated values.

The program always keeps the best model obtained and each iteration cycle starts with the determination of a temporary model to be tried next. To determine the temporary model, the partial derivatives of the apparent resistivity values, with respect to the free model parameters, is calculated by the subroutine TDR. The partial derivative matrix, **A**, is decomposed, by the singular value decomposition routine SVDC, into a product of an orthogonal data eigenvector matrix, **U**, a diagonal eigenvalue matrix, **L**, and the transpose of an orthogonal parameter eigenvector matrix, **V**:

$$\mathbf{A} = \mathbf{U}\cdot\mathbf{L}\cdot\mathbf{V}^t.$$

Letting the logarithms of the free model parameters and the calculated apparent resistivity values be represented by the vectors **p** and **q**, respectively, a small variation **dp**, of the model parameters results in a variation **dq**, of the calculated apparent resistivity values, given as:

$$\mathbf{dq} = \mathbf{A}\cdot\mathbf{L}\cdot\mathbf{V}^t\cdot\mathbf{dp}.$$

To get the increments to be added to the model parameter vector **p** in order to get the temporary model to be tried next, the vector **dq**, in the equation above, is taken to be the difference between the measured and calculated apparent resistivity values. The equation is then "inverted" in the subroutine NEWP by multiplying **dq** by a "damped" inverse of the partial derivative matrix. The damping is performed by adding a

Marquardt parameter to the eigenvalues. This damping is necessary because if the partial derivative matrix is nearly singular, one or more of the eigenvalues are very small, and an undamped inversion of the equation would result in unreasonably large increments dp. The Marquardt parameter, XLA, is taken to be equal to one of the eigenvalues which have been ordered in an increasing order in the subroutine ORDW. The ordered eigenvalues are numbered by the index ILA which runs from 1 to NF, the number of free parameters in the model.

The first temporary model tried in each iteration step is obtained by adding the increments resulting from the damped inversion of the above equation, using the smallest eigenvalue (but not smaller than 0.01, the index ILA shows which eigenvalue is used) as a Marquardt parameter. If the chi-square sum, CHIT, for the temporary model is less than CHI then the temporary model is kept as the best model and a new iteration cycle is started. If CHIT is higher than CHI, another temporary model, obtained by inversion of the above equation with increased damping with a higher eigenvalue (ILA increased by one), is tried. This is continued until CHIT becomes lower than CHI or ILA gets higher than NF, in which case the iteration process is terminated.

There are four stop checks in the program. The iteration process terminates if:

a) The parameter ILA gets higher than NF. This is called no-convergence. If the program terminates on this stop check it does not necessarily mean that an acceptable fit to the measured data was not obtained. It simply states that the program could not further improve the fit.
b) The average fractional difference, CHI, between the measured and the calculated apparent resistivity values becomes less than $10^{-3}$ i.e. an average difference less than 0.1%. This is called CHI-convergence.
c) The fractional decrease in CHI in the last iteration is less than $10^{-5}$. This is called DCHI-convergence.
d) The program has performed the number of iterations that the operator asked for and he does not wish to perform more iterations. This stop check is called maximum of iterations.

A CHI-convergence is seldom obtained in inversion of real measured data and the iteration is usually terminated on no-convergence or DCHI-convergence (no further improvement) or on maximum of iterations.

The forward routine TFWD calculates the late time apparent resistivity values at time values equally distributed on logarithmic scale with five points per power of e. The inversion algorithm demands therefore that the data have this distribution. If the measured data do not fulfill this condition, an interpolation, on log-log scale, is performed (in the data input subroutine TRDT) in order to get data points, equally spaced in logarithm of time, with five points per power of e.

A different weight can be given to the data points in the inversion. The weighing is controlled by the weight parameter RW ($-1 \leq RW \leq 1$). Using RW=1, the data points are weighed with $WPM \propto \ln(\rho_a)$. This causes data points with higher apparent resistivity values to have more weight in the inversion. For WR=-1 the data points are weighed with $WPM \propto 1/\ln(\rho_a)$ giving the lower apparent resistivity values the higher weight. For WR in between -1 and 1 the data points are weighed with $WPM \propto (\ln(\rho_a))^{WR}$ giving weights in between the above extreme cases. For WR=0 all data points have the same weight. The different modes of weighings can be used to put emphasis on different parts of the measured data curve. Normally it is recommended to give all data points similar weight and hence to take WR close to 0, giving $WPM \approx 1$.

## 2.2 The forward algorithm

The forward algorithm TFWD calculates late time apparent resistivity values for a given one-dimensional resistivity model, a given transmitter loop radius, RP, and turn-off time, TOFF. The turn-off time is the time it takes the transmitter to turn the current in the transmitter loop linearly off from its maximum value to zero. The calculations are done in four steps. The first step is to calculate the frequency domain response i.e. the induced voltage in a receiver coil at the centre of a circular transmitter loop into which an alternating current, $I_0 e^{i\omega t}$, is transmitted. The next step is to calculate, from the frequency domain response, the transient voltage response due to a sharp current step in the transmitter loop. The third step is to take into account the turn-off time and calculate the response of a ramped step function with the ramp length TOFF. The fourth and final step is to calculate the late time apparent resistivity from the ramped step response.

The frequency domain response is calculated, in the subroutine YCAL, as a function of the frequency, $\omega$, of the transmitted current and the radius, r (=RP), of the

transmitter loop according to the following formula:

$$V(\omega, r) = (-i\omega)\Phi(\omega, r) I_0 e^{i\omega t} \quad ; \quad \Phi(\omega, r) = \int_0^\infty K(\omega, \lambda) J_1(\lambda r) d\lambda.$$

The kernel function $K(\omega,\lambda)$, calculated in the subroutine RECUR, contains the model parameters. The Hankel transform integral above is calculated by the use of a digital filter. The filter is transferred through a common block from the subroutine J1FLT which is called by the main program. The frequency domain response has typically to be determined at about 100 frequency values equally spaced on log-scale with five points per power of e. It is very time consuming to calculate each frequency domain response value by the numerical integration described above. Fortunately, the frequency domain response is a well behaved function of the frequency. It is therefore, in most cases, sufficient to calculate the frequency domain response in the above described manner at every 2nd to 4th point. The response values at the intermediate points can then be determined by a spline between the calculated values. Use is made of this in the forward routine TFWD in order to reduce the running time of the program. The frequency domain response is calculated by a numerical integration, in the subroutine YCAL, at every MMth frequency value where MM is an integer supplied by the user. The response at the intermediate frequency values is obtained by a cubic spline performed by the subroutines SPLINE and SPLINT. How densely the response has to be calculated by numerical integration depends on the resistivity model and the radius of the transmitter loop. The higher the resistivity is, the smaller the MM must be. It is not recommended to take MM to be higher than 4, unless the model contains a thick layer of very low resistivity (less than 1 Ωm) at a depth not greater than about three times the transmitter loop radius. If MM is taken too high, the calculated late time apparent resistivity will show oscillations at late time values. If this is observed, MM has to be decreased.

The next step is to calculate the time domain response due to a sharp step function transmitted into the transmitter loop. The response is given by the following integral:

$$V_{step}(r, t) = \frac{2I}{\pi} \int_0^\infty Re\Phi(r, \omega)\cos(\omega t) d\omega$$

where I is the transmitted current strength and $\Phi$ is the Hankel transform integral for the frequency domain response. The cosine transform integral is turned into a convolution integral by changing to logarithmic variables and the integration is performed

numerically by using a digital filter. The cosine transform filter is transferred through a common block from the subroutine COSFLT which was called by the main program. The forward routine, TFWD, calculates the time domain response for time values equally spaced on log-scale, for five values per power of e, by shifting the kernel function relative to the filter and performing the convolution summation.

The third step is to take into account the fact that the current turn-off is not described by a sharp step but rather by a ramped step function with the ramp length given by the turn-off time, TOFF. The ramped step response is obtained by convolving the derivative of the current function with the sharp step response. This is given by the following integral:

$$V(r,t) = \frac{1}{TOFF} \int_{t}^{t+TOFF} V_{step}(r, \tau) d\tau.$$

This integral is calculated numerically in the subroutine TRF.

The fourth and final step in the forward algorithm is to calculate the late time apparent resistivity values from the ramped step response. This is done at the end of the routine TFWD, and according to the formula:

$$\rho_a(r,t) = \frac{\mu_0}{4\pi} \left[ \frac{2A_s A_r I}{5t^{5/2} V(r,t)} \right]^{2/3}$$

where $A_s$ and $A_r$ are the effective areas of the transmitter and receiver loops, respectively (area times number of turns) and $\mu_0$ is the magnetic permeability of vacuum. The above formula defines the late time apparent resistivity, as a function of time after the current turn-off (t=0 is at the end of the turn-off ramp), in terms of the induced voltage in the receiver loop at the centre of a transmitter loop of radius r (=RP).

## 2.3 The partial derivative algorithm

The partial derivatives of the logarithm of the apparent resistivity values with respect to the free logarithmic model parameters are calculated by the subroutine TDR. A small increment is added to the free logarithmic parameters, one at the time, and the corresponding logarithmic apparent resistivity values calculated by the forward routine TFWD. The derivatives are then calculated as the difference between the logarithmic apparent resistivity values for the shifted and unshifted models, divided by the increment. The result is returned as the partial derivative matrix **A**.

## 3. RUNNING THE PROGRAM

In order to run the inversion program, the file TINV.EXE must either be in the working directory or a directory specified in the PATH-list. The measured apparent resistivity curve is read from an input file. The input file must have one data point in each line. A data point consists of a pair of numbers separated by a comma (,), a space(s) or a tab. The first number is a time value, t, and the second number is the corresponding measured late time apparent resistivity value, $\rho_a$.

The following is an example of an input data file:

```
0.89000E-04  526.9
0.10900E-03  422.8
0.14000E-03  320.7
0.17700E-03  249.8
0.21900E-03  199.6
0.28000E-03  157.6
0.35500E-03  124.3
0.44300E-03  101.8
0.56300E-03   82.0
0.71200E-03   67.7
0.87600E-03   55.7
0.10900E-02   47.2
0.14000E-02   39.1
0.17700E-02   33.3
0.21900E-02   29.1
0.28000E-02   24.6
0.35500E-02   21.1
0.44300E-02   18.4
0.56300E-02   16.0
0.71200E-02   14.1
0.87600E-02   12.4
0.10900E-01   11.2
0.14000E-01   10.1
0.17650E-01    9.5
0.21950E-01    9.0
0.28000E-01    8.5
0.35450E-01    8.2
0.44250E-01    7.9
0.56300E-01    7.9
0.71200E-01    8.1
```

When a data file containing the measured apparent resistivity curve has been created, having the above described format, the inversion program is run by going through the following steps. (All user's responses discussed below are to be followed by striking the return key):

1. The inversion program is started by typing TINV at the system prompt.

2. The program prompts for input file name. This is answered by typing the name of the file containing the measured data curve to be inverted.

3. The program asks for the weight parameter RW discussed above. This is answered by typing a number in the interval -1 to 1. It is generally recommended to use RW close to 0.

4. The program asks for the turn-off time, TOFF. This is answered by typing the turn-off time in seconds.

5. The program asks for the effective radius, RP, of the transmitter loop. If the transmitter loop used for collecting the data was not circular, the radius of a circular loop with the same area as the loop used is typed.

6. The program asks how densely the frequency domain response is to be calculated by numerical integration. This is answered by typing an integer for the parameter MM, causing the frequency domain response to be calculated at every MMth frequency value and be determined by cubic spline at intermediate frequency values. It is normally not recommended to take MM to be higher than 4 (see discussion above).

7. The program asks for the guessed initial model. It asks for the number of layers which is answered by typing the guessed number of layers. Next it asks for the model parameters for each layer (resistivity in Ohmm and thickness in m). Each parameter can either be a free parameter to be adjusted by the program or a fixed parameter not to be adjusted. For a free parameter the guessed value is typed. For parameters to be held fixed the corresponding parameter values are followed by a comma and an asterisk (e.g. 235,*).

8. The program prompts for the number of iterations to be performed. This is answered by typing the desired number of iterations.

9. Finally the program prompts for names of output list file and an output plot file into which the results from the iterative inversion are written. When these file names have been given, the program starts the iteration process.

During the iterations the program writes on the terminal the iteration number ITR, the parameter ILA that tells which of the ordered eigenvalues is used as Marquardt parameter and its value XLA. It writes the best model parameters obtained so far (resistivity values, rho and layer thicknesses, d) and the temporary model parameters (rhot and dt) to be tried next. It also writes the lowest CHI-value obtained and the temporary CHIT obtained from the temporary model. If CHIT is lower than CHI then the temporary model is kept as the best model and the program proceeds to the next iteration with a new temporary model. If CHIT is higher than CHI another temporary model is tried. This is repeated until CHIT becomes lower than CHI or ILA becomes higher than NF, the number of free parameters in the model and the program terminates on the criterion of no-convergence. In addition to the above listed parameters, the program also writes on the terminal, each time the forward routine TFWD is called, the number of the lowest and the highest frequency values for which the frequency domain response is calculated. These values are written merely for psychological reasons so that the operator can see that the program working. The minimum value should be considerably higher than 1 and the maximum value lower than 283.

If the iteration process has not stopped on the no-convergence, the CHI-convergence or the DCHI-convergence stop checks (see above) and the number of iterations specified (in step 8 above) has been performed, the program pauses and asks if the iteration process is to be continued. If this question is answered with N (no), the iteration process is terminated on maximum number of iterations. If it is answered with Y (yes), the program asks how many more iterations are to be performed.

# 4. OUTPUT FILES

The results from the inversion are written into two output files, the output list file and the output plot file. The content of these files will now be discussed shortly.

## 4.1 The output list file

The following is an example of an output list file:

```
 EFFECTIVE LOOP RADIUS, RP=169.3 m
 TURN-OFF TIME, TOFF=.240E-03 s
 WEIGHT PARAMETER, RW=  .00


INITIAL MODEL PARAMETERS:
rho: 1000.00    50.00     2.00     8.00
  d:  100.00    50.00   100.00
CHI= .49883E+00


ITR= 1  ILA= 2  CHI= .3910E+00  DCHI= .2161E+00
rho:  413.58     1.84     8.22    11.30
  d:  144.76    15.50   466.19


ITR= 2  ILA= 3  CHI= .1215E+00  DCHI= .6894E+00
rho:   86.66     6.66     4.34    14.42
  d:  112.20    19.67   145.75


ITR= 3  ILA= 2  CHI= .1060E+00  DCHI= .1270E+00
rho:  109.27    12.58     6.07     9.80
  d:  100.70    57.55   285.70


ITR= 4  ILA= 1  CHI= .1594E-01  DCHI= .8497E+00
rho:  128.61     9.30     4.88    12.17
  d:   98.65    67.27   290.69


ITR= 5  ILA= 1  CHI= .1118E-01  DCHI= .2988E+00
rho:  132.33     9.48     4.79    12.56
  d:   98.61    68.17   258.00


ITR= 6  ILA= 1  CHI= .1109E-01  DCHI= .7392E-02
rho:  132.48     9.45     4.76    12.45
  d:   98.64    68.83   255.69

*** THE PROGRAM TERMINATED  AFTER  7 ITRERATIONS ***

ISTOP=2 * MAX ITERATIONS *
```

FINAL CHI-SQ. SUM IS CHI= .1109E-01

THE FINAL MODEL PARAMETERS ARE:

rho: 132.26 9.43 4.76 12.39
d: 98.72 68.98 254.65

DATA EIGENVECTORS:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | -.095 | .061 | -.421 | .282 | -.534 | .450 | -.064 |
| 2 | -.135 | .072 | -.367 | .212 | -.228 | .057 | .014 |
| 3 | -.167 | .081 | -.315 | .143 | .011 | -.215 | .063 |
| 4 | -.188 | .087 | -.275 | .084 | .141 | -.311 | .071 |
| 5 | -.198 | .092 | -.242 | .033 | .181 | -.265 | .046 |
| 6 | -.202 | .095 | -.207 | -.017 | .181 | -.158 | .010 |
| 7 | -.206 | .099 | -.166 | -.069 | .179 | -.053 | -.023 |
| 8 | -.212 | .105 | -.117 | -.123 | .184 | .026 | -.046 |
| 9 | -.218 | .112 | -.065 | -.174 | .178 | .099 | -.062 |
| 10 | -.220 | .117 | -.016 | -.210 | .136 | .187 | -.075 |
| 11 | -.220 | .119 | .031 | -.226 | .075 | .252 | -.076 |
| 12 | -.218 | .118 | .075 | -.218 | .010 | .258 | -.055 |
| 13 | -.215 | .112 | .110 | -.185 | -.058 | .208 | -.015 |
| 14 | -.213 | .103 | .139 | -.134 | -.117 | .104 | .040 |
| 15 | -.210 | .090 | .162 | -.078 | -.166 | -.012 | .096 |
| 16 | -.205 | .074 | .176 | -.028 | -.205 | -.092 | .131 |
| 17 | -.198 | .056 | .183 | .009 | -.222 | -.131 | .139 |
| 18 | -.192 | .035 | .181 | .040 | -.209 | -.156 | .125 |
| 19 | -.187 | .012 | .174 | .072 | -.169 | -.180 | .092 |
| 20 | -.180 | -.010 | .164 | .109 | -.127 | -.174 | .032 |
| 21 | -.171 | -.031 | .155 | .154 | -.085 | -.151 | -.046 |
| 22 | -.163 | -.051 | .147 | .194 | -.031 | -.103 | -.140 |
| 23 | -.156 | -.071 | .136 | .221 | .036 | -.030 | -.229 |
| 24 | -.148 | -.095 | .120 | .241 | .094 | .036 | -.272 |
| 25 | -.139 | -.124 | .098 | .246 | .138 | .088 | -.243 |
| 26 | -.132 | -.154 | .075 | .231 | .164 | .124 | -.144 |
| 27 | -.125 | -.185 | .053 | .199 | .170 | .150 | -.005 |
| 28 | -.118 | -.217 | .027 | .153 | .164 | .163 | .133 |
| 29 | -.111 | -.247 | -.001 | .099 | .147 | .166 | .218 |
| 30 | -.104 | -.276 | -.028 | .033 | .106 | .138 | .264 |
| 31 | -.098 | -.302 | -.052 | -.048 | .031 | .068 | .289 |
| 32 | -.092 | -.326 | -.077 | -.134 | -.048 | -.007 | .244 |
| 33 | -.086 | -.344 | -.102 | -.217 | -.117 | -.070 | .132 |
| 34 | -.081 | -.355 | -.119 | -.270 | -.162 | -.123 | -.127 |
| 35 | -.074 | -.355 | -.124 | -.283 | -.187 | -.178 | -.593 |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

PARAMETER EIGENVECTORS:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | -.140 | .104 | -.590 | .278 | -.577 | .458 | -.033 |
| 2 | -.246 | -.083 | .651 | -.305 | -.326 | .553 | -.055 |
| 3 | -.140 | -.935 | -.157 | .108 | .205 | .156 | -.062 |
| 4 | -.003 | -.056 | -.056 | -.217 | -.237 | -.292 | -.897 |
| 5 | -.947 | .160 | -.079 | -.003 | .179 | -.198 | .016 |
| 6 | -.058 | -.161 | .400 | .644 | -.455 | -.425 | .087 |
| 7 | .022 | .232 | .184 | .596 | .472 | .394 | -.423 |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

PARAMETER EIGENVALUES:

.446E+01 .224E+01 .101E+01 .591E+00 .322E+00 .201E+00 .700E-01

```
CORRELATION MATRIX:

1  1.000
2   .821  1.000
3   .277   .449  1.000
4   .098   .171   .529  1.000
5  -.923  -.942  -.376  -.130  1.000
6  -.278  -.588  -.799  -.311   .429  1.000
7   .251   .396   .844   .857  -.326  -.623  1.000
     1      2      3      4      5      6      7
```

| I | T | Rhoam | Rhoac | WPM |
|---|---|---|---|---|
| 1 | .82724E-04 | 570.44 | 565.67 | 1.00 |
| 2 | .10104E-03 | 459.09 | 460.93 | 1.00 |
| 3 | .12341E-03 | 368.63 | 373.34 | 1.00 |
| 4 | .15073E-03 | 296.43 | 300.71 | 1.00 |
| 5 | .18411E-03 | 239.65 | 241.32 | 1.00 |
| 6 | .22487E-03 | 194.59 | 193.86 | 1.00 |
| 7 | .27465E-03 | 160.55 | 157.31 | 1.00 |
| 8 | .33546E-03 | 131.54 | 129.61 | 1.00 |
| 9 | .40973E-03 | 109.22 | 108.11 | 1.00 |
| 10 | .50045E-03 | 91.19 | 90.85 | 1.00 |
| 11 | .61125E-03 | 76.68 | 76.68 | 1.00 |
| 12 | .74659E-03 | 64.74 | 64.85 | 1.00 |
| 13 | .91188E-03 | 54.03 | 55.03 | 1.00 |
| 14 | .11138E-02 | 46.44 | 47.11 | 1.00 |
| 15 | .13604E-02 | 39.95 | 40.57 | 1.00 |
| 16 | .16616E-02 | 34.77 | 35.00 | 1.00 |
| 17 | .20294E-02 | 30.54 | 30.33 | 1.00 |
| 18 | .24788E-02 | 26.74 | 26.41 | 1.00 |
| 19 | .30276E-02 | 23.39 | 23.13 | 1.00 |
| 20 | .36979E-02 | 20.57 | 20.37 | 1.00 |
| 21 | .45166E-02 | 18.19 | 18.04 | 1.00 |
| 22 | .55166E-02 | 16.19 | 16.09 | 1.00 |
| 23 | .67379E-02 | 14.52 | 14.43 | 1.00 |
| 24 | .82297E-02 | 12.89 | 13.03 | 1.00 |
| 25 | .10052E-01 | 11.63 | 11.83 | 1.00 |
| 26 | .12277E-01 | 10.66 | 10.82 | 1.00 |
| 27 | .14996E-01 | 9.92 | 9.99 | 1.00 |
| 28 | .18316E-01 | 9.41 | 9.33 | 1.00 |
| 29 | .22371E-01 | 8.96 | 8.82 | 1.00 |
| 30 | .27324E-01 | 8.55 | 8.47 | 1.00 |
| 31 | .33373E-01 | 8.28 | 8.23 | 1.00 |
| 32 | .40762E-01 | 8.01 | 8.10 | 1.00 |
| 33 | .49787E-01 | 7.90 | 8.03 | 1.00 |
| 34 | .60810E-01 | 7.97 | 8.00 | 1.00 |
| 35 | .74274E-01 | 8.14 | 8.04 | 1.00 |

The program begins by writing the effective transmitter loop radius, RP, the turn-off time, TOFF, and the weight parameter, RW, as well as the initial model and the corresponding chi-square sum, CHI. During the iteration process the program writes for each iteration the iteration number, ITR, the Marquardt parameter counter, ILA, the chi-square sum, CHI, for the model obtained in the present iteration, the fractional decrease DCHI of the chi-square sum between the present and the last iteration and the model obtained in the present iteration step. Model parameters that are held fixed are followed by an asterisk. When the iteration process is finished the program writes the number of iterations performed and the stop check on which the program stopped. Then it writes the final chi-square sum and the best model obtained.

Next the program writes out information on how changes in the model parameters affect the calculated apparent resistivity values. This is described by three matrices. **U** is an NDxNF matrix whose column vectors are the data eigenvectors listed (as columns) in the output list file (ND and NF are the number of data points and free model parameters). **L** is an NFxNF diagonal matrix whose diagonal elements are the parameter eigenvalues written in the list file. **V** is an NFxNF orthogonal matrix whose column vectors are the listed parameter eigenvectors (as columns).

In order to bring out the significance of these matrices we think of the resistivity model as an NF dimensional vector **p**. If no parameter is fixed, the first NL components of this vector are the natural logarithms of the resistivity values of the layers (NL is the number of layers in the model, NF=2*NL-1 if no parameter is fixed). The remaining NL-1 components are the natural logarithms of the layer thicknesses. Fixed parameters are not included in the vector **p**. For example, if we take a three layered model and fix the resistivity of the second layer then NF=4 and $p_1$ and $p_2$ are the natural logarithms of the resistivity values of the first and the third layer and $p_3$ and $p_4$ are the natural logarithms of the thicknesses of the first and the second layer. In the same way we think of the natural logarithm of the calculated apparent resistivity values as an ND dimensional vector **q**. If we change the model vector **p** by a small amount **dp** then the calculated apparent resistivity vector will be changed by **dq** according to the equation:

$$\mathbf{dq} = \mathbf{A} \cdot \mathbf{L} \cdot \mathbf{V}^t \cdot \mathbf{dp}$$

where $\mathbf{V}^t$ means the transpose of the matrix **V**.

From this equation the significance of the data and parameter eigenvectors and the eigenvalues can be deduced. In the first place we see that the apparent resistivity increments **dq** are proportional to the eigenvalues. The first parameter eigenvector shows which model parameters have the strongest association with the first eigenvalue which is normally the highest one. The degree of association is shown by the absolute value of the components of the eigenvector, the higher the absolute value the greater the contribution of the corresponding model parameter. Likewise the eigenvector corresponding to the smallest eigenvalue shows which model parameters have the least influence on the calculated apparent resistivity. The model parameters that are most strongly associated to the highest eigenvalue are the most reliable ones whereas the parameters associated to the smallest eigenvalue are the most uncertain

parameters in the final model. The relative contribution of the parameter eigenvectors to the different apparent resistivity values is described by the data eigenvectors.

In the four layer example above it is seen that the thickness of the first layer is the best determined model parameter because the eigenvector corresponding to the highest eigenvalue has the absolute value of the fifth component close to one (0.947) while the other components are relatively small. The resistivity value of the third layer is also well determined because the eigenvector corresponding to the second highest eigenvalue has the absolute value of the third component close to one (0.935). It is also seen that the resistivity value of the fourth layer is not well determined because the absolute value of the ~~fifth~~ FOURTH component of all the eigenvectors is small except for the eigenvector corresponding to the smallest eigenvalue.

The program writes a correlation matrix into the output list file. This matrix indicates how the different model parameters are correlated to each other. If an off-diagonal element of the correlation matrix has absolute value close to one, it means that the corresponding model parameters are correlated. If an off-diagonal element is close to minus one, then the sum of the corresponding logarithmic parameters and hence the product of the linear parameters is a well defined parameter. If on the other hand an off-diagonal element is close to one then the difference between the corresponding logarithmic model parameters and hence the ratio of the linear parameters is a well defined parameter. There is a weak multiplicative correlation between the first and the fifth model parameters (resistivity and thickness of the first layer; $Corr_{5,1} = -0.923$) in the above example. There is also a weak multiplicative correlation between the second and fifth model parameters (resistivity of the second layer and the thickness of the first layer; $Corr_{5,2} = -0.942$). Two model parameters are not strongly correlated unless the corresponding correlation matrix element is very close to one ($\geq 0.99$).

Finally the program writes into the output list file the time values, T, for the data points, the measured apparent resistivity values, Rhoam, the apparent resistivity values, Rhoac, calculated from the final model and the weight factors for the data points, WPM.

## 4.2 The output plot file

The inversion program writes the measured data points, the calculated apparent resistivity values, the final model and the value of the chi-square sum into the output plot file. The content of this file can be plotted both on the terminal and as a hard copy on a printer or a plotter. To plot the results the files SPLOT.BAT (screen plot), PPLOT.BAT (paper plot), RES.GRD (grid file) and CENTERED.SYM (plot symbols) must be in the working directory and the files VIEW.EXE (screen plot) and PLOT.EXE (paper plot) must be either in the working directory or a directory specified in the PATH-list.

The measured apparent resistivity values are plotted against the square root of time, measured in microseconds (i.e. $\sqrt{t} \cdot 1000$ where t is in seconds), as small circles on a double logarithmic plot and the calculated apparent resistivity curve is drawn as an unbroken line. The resistivity model is displayed both numerically as resistivity values (Ohmm) and layer thicknesses (m) and also as a histogram where the x-axis shows the depth in meters and the y-axis the resistivity values. The value of the chi-square sum is also displayed on the plot. The plot is marked by a station identification which is identical to that part of the output plot file name which is in front of the point (.). If e.g. the output plot file is given the name HT05.PLT, then the plot will be marked as STATION: HT05.

To plot the results on the terminal, simply type SPLOT followed by the name of the output plot file and press return. This initiates a command procedure that appends the plot file to the grid file RES.GRD and plots the results on the screen by the program VIEW.EXE. The plot can be zoomed in by striking the + key and shifted both in horizontal and vertical directions by striking the arrow keys. To return to the DOS prompt, strike the Esc-key, then q and return.

To plot a hard copy on a printer or a plotter, simply type PPLOT followed by the name of the output plot file and return. This initiates a command procedure that appends the plot file to the grid file and plots the results by the program PLOT.EXE. The plot program asks if the plot is to be shifted, and if confirmed, how much in each direction. The first time a hard copy is produced or if the output device is changed, it may be necessary to reset the output device specification. This is done by changing the working directory to the directory containing the program PLOT.EXE and typing

PLOT/I. The plot program displays the current output device specification and asks if it is to be changed. If this is answered positively, it displays a list of possible output devices and the appropriate choice can be made and saved by following a step by step procedure conducted by the plot program.

An example of an output plot, plotted on a printer, is shown on Figure 4.1.

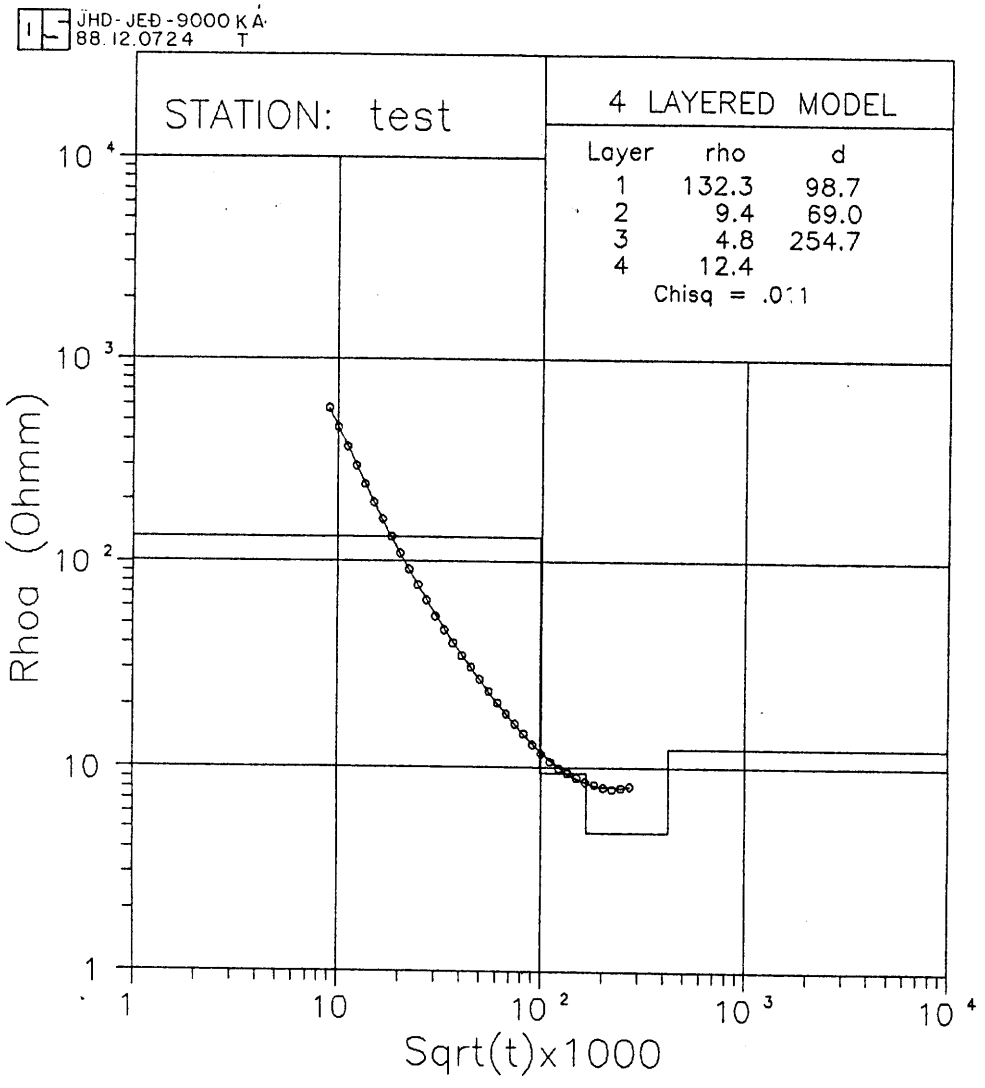


**Figure 4.1.** An output plot from TINV

## REFERENCES

Johansen, H. K., 1977: A man/computer interpretation system for resistivity soundings over a horizontally stratified earth. Geophysical Prospecting 25, pp. 667-691.

Árnason, K., 1988: Transient electro-magnetic soundings over a horizontally stratified earth. In preparation.