MASTER

DOCUMENTATION AND ANALYSIS OF THE SCHLUMBERGER INTERACTIVE I-D INVERSION PROGRAM SLUMB

by
STEWART SANDBERG

Work performed under Contract No.

DE-AC07-79ET/27002

Department of Geology and Geophysics

**University of Utah
Salt Lake City, Utah (USA)**

September 1979

Prepared for
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.

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 employees, 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 or the U.S. Department of Energy to the exclusion of others that may be suitable.

DOCUMENTATION AND ANALYSIS
OF THE SCHLUMBERGER
INTERACTIVE 1-D
INVERSION PROGRAM
SLUMB

by
STEWART SANDBERG

July, 1979

TABLE OF CONTENTS

	Page
I. USING THE PROGRAM	1
A. Introduction	1
1. Non-standard FORTRAN	1
2. Underflow suppression	2
3. Conversion to a system without a graphics output device	3
B. Instructions on how to use the program interactively	4
1. Practical considerations	5
a) Iterations	5
b) How good are the parameters	6
2. How to run the program (data formats)	7
C. Explanation of Output	11
1. Output on the Tektronix screen	13
2. Abbreviated Output	14
II. THEORY	15
A. How the program works (inversion theory)	15
1. The forward problem	15
2. The inverse problem	16
B. Schlumberger resistivity theory	19
III. TESTS MADE ON THE PROGRAM	25
A. Discussion of the effect of noise	25

B. Effect of using the wrong model for inversion	31
IV. DOCUMENTATION	34
A. Program Subroutine descriptions	34
B. Program listing	37
V. REFERENCES	70
VI. EXAMPLE	71
A. Input data cards	71
B. Regular Output	73
C. Tektronix Output	79
D. Abbreviated output	81

I. USING THE PROGRAM

A. Introduction

This computer program is designed to accept field data from a Schlumberger resistivity array and invert it in terms of a one-dimensional layered geoelectrical model. Because the inverse problem is non-linear, an initial guess model is required. This input model can be obtained by traditional curve matching, or by repeated use of a forward algorithm.

This program was written in FORTRAN V in 1976 by Luiz Rijo. It was developed on a UNIVAC 1108 computer, operating under EXEC. 8 (level 31). The program size is approximately 16 K decimal words. (The UNIVAC has a word size of 36 bits.) The program is designed for interactive use using a Tektronix 4013 Graphics Display terminal. (Conversion to run in batch mode without this special terminal is simple and instructions are given in section I-A-3.)

For an example of application of the program, see Rijo et al. (1977).

1. Non-Standard FORTRAN

This program makes use of the PARAMETER statement available on the University of Utah UNIVAC 1108 extended FORTRAN V compiler. This statement is used in cases where the values of certain integer constants are the only changes made to the FORTRAN text between compilations (e.g. the dimensions of certain arrays). The PARAMETER statement allows a symbolic name to be assigned to the constant. Consequently, only the PARAMETER statement need be revised instead of all the constants throughout the program. Subsequent to its definition, the PARAMETER variable may be

used only where an integer constant may be used and must not be redefined in the source program.

If the `PARAMETER` statement is not available to you, replace all `PARAMETER` variables by their integer constants in the source program.
e.g.,

```

      .
      .
      .
PARAMETER NN = 40, MM = 15

```

```
COMMON/BLK2/FLOG(NN),F(NN),YLOG(NN),Y(NN),PLOG(MM),P(MM)
      .
      .
      .

```

can be replaced with

```

      .
      .
      .
COMMON/BLK2/FLOG(40),F(40),YLOG(40),Y(40),PLOG(15),P(15)
      .
      .
      .

```

2. Underflow Suppression

The program frequently calculates values too small ($<10^{-38}$) to be correctly represented in the 36 bits of a UNIVAC 1108 word. This results

in an underflow and an error exit from the program.

As an option, the University of Utah UNIVAC 1108 operating system provides for avoiding an error exit due to underflows. The offending variable is set to zero, and the program continues. The number of underflows encountered during execution is noted upon termination of the job. This option is implemented by using option U on the @XQT statement. Option F is used on the @XQT statement to suppress not only the underflow error exit but also the counting of underflow occurrences by the system thereby saving some computer time.

If your installation does not provide for underflow suppression, DOUBLE PRECISION calculations may be necessary.

3. Conversion to a System Without a Graphics Output Device

It is trivial to convert the program to run in a batch mode. First, seven FORTRAN statements in MAIN need to be removed. These are:

<u>Statement number</u>	<u>Statement</u>
25	DATA IOFF, ION/017,016/
26	CALL INITT(30)
27	CALL TOUTPT (IOFF)
28	CALL CHSYNC
29	WRITE(6,200)
30	200 FORMAT(1X, '?')
139	CALL FINITT(0.,0.)

Second, remove subroutines DRAWX and LGPLT2.

Throughout the program, output is written on device 3 by the Fortran

statement WRITE(3,#). When running in batch mode these statements should be replaced with WRITE(6,#) or PRINT#.

Subroutine LGPLT2 must be replaced with an appropriate line printer plotting routine, or a routine to print the values of the parameters in the argument list of LGPLT2.

LGPLT2 calls a subroutine ICSSCU to draw a smooth curve between the points plotted. This subroutine is in a commercial software package from IMSL, and is detailed at the end of section IV-B.

B. Instructions on How to Use the Program Interactively

This program is designed to use a Tektronix 4013 Graphics Display terminal under "Terminal Control System" (Tektronix software package). The program produces a log-log plot of apparent resistivity versus AB/2 distance on the display terminal. Additional output is written to a temporary data file (I/O UNIT #3) which can then be sent to a line printer or examined on the terminal with the editor.

The theory of least-squares ridge regression is discussed in Rijo et al. (1977). However, some practical aspects of using this program were not mentioned.

First, it is important to begin the inversion with an initial guess which roughly matches the data. If the initial guess model is poor, the algorithm will diverge. A good initial guess may be found through rough curve matching or by using a computer forward solution and manually varying the layered model parameters.

Second, an inversion should be continued until X^2_v , the sum of the squared residuals divided by the degrees of freedom, asymptotes to a value

comparable to the estimated data variance (see Practical Considerations).

1. Practical Considerations

a) Iterations

When inverting an earth model obtained by curve-matching or forward manual iteration, it is best to specify 4 or 5 iterations for ITMAX on the first data card. This doesn't mean 4 or 5 iterations will be done however.

The last iteration is done for statistical purposes only. The parameters are not changed between the second-to-last and the last iteration. Therefore, specifying N iterations gives you N-1 iterations.

There is a situation where less than N-1 iterations can take place. In the subroutine YINVRT, if the new X^2v calculated from the new parameters is greater than the old X^2v calculated from the previous iterations parameters, then the old parameters are kept (and the old X^2v), and an iteration is counted. (In one particular case, 4 iterations were specified and only 1 iteration actually took place.) This is important to keep in mind.

If the regular output is specified, check the end of the printout where the parameters are listed for each iteration. If the parameters don't change every iteration, then fewer iterations have actually been done than were specified. On the abbreviated output, check to be sure that $RCSQ(X^2v)$ decreases with each iteration (except the last one where it is the same as the second-to-last). If this isn't true, then fewer iterations were done than specified.

b) How good are the parameters

Putting a high number of iterations on the first data card doesn't necessarily improve the accuracy of the resulting inverted parameters. The mathematical inversion used in this program converges very rapidly to a solution.

There is another way of exiting from the program other than completing the specified number of iterations. The reduced chi-square parameter is

$$RCSQ = \chi^2_v = \frac{1}{N-NP} \sum_{i=1}^N (\rho_{a\text{observed}} - \rho_{a\text{calculated}})^2$$

where

N = number of data observations

NP = number of parameters

$\rho_{a\text{observed}}$ = apparent resistivity for the i th $AB/2$ distance observed in the field

$\rho_{a\text{calculated}}$ = apparent resistivity for the i th $AB/2$ distance calculated from the inverted parameters

When

$$\left| \frac{\chi^2_{y_{m-1}}}{\chi^2_{y_m}} - 1 \right| < 0.01 \quad (\text{where } m \text{ is the iteration number}),$$

then 2 more iterations will take place and the program will end. This is a very conservative convergence criteria most of the time. A recommended alternative is to specify 4 or 5 iterations and check to see if convergence has occurred. Convergence is indicated by an abrupt decrease in RCSQ in the

first 2 or 3 iterations and minor decreases, each iteration afterward.

The data fit is given by RCSQ. Let

$$\epsilon = \left\{ \log_{10}^{-1} \left[(RCSQ)^{1/2} \right] - 1 \right\} \cdot 100$$

then ϵ is the percent error in the data fit. For example, if $RCSQ = 5.64 \times 10^{-4}$ then $\epsilon = 5.6$ percent error. This is a very good percentage error since good field data is usually accurate to about 5%. Percent error less than about 5% results from fitting noise in the data.

Parameters with large standard deviations usually have large correlation coefficients. Parameters with large correlation coefficients have standard deviations much larger than their actual uncertainties. Thin conductive layers are highly correlated between resistivity and thickness. Inman (1975) gives a very good analysis of this effect. When two parameters are highly correlated, it is usually not possible to resolve them independently. One thing that can be done is to fix one parameter (resistivity perhaps) and invert to find the other (thickness). When drill hole data or other independent sources of geologic data are available, parameters can be better resolved.

Another factor to be taken in to account is the validity of assuming a layered earth model. Most data is taken over 3-dimensional structures that we try to interpret with 1-dimensional earth models. Noise is in the data that could be called 3-dimensional or wrong model noise. If this 3-dimensional data is fit very accurately by a 1-dimensional model (with no geologic constraints), then most likely the interpretation is worthless.

2. How to Run the Program

After signing on the terminal and typing the RUN card, type the following:

@ASG,A filename.

@@ON F

@XQT,OF filename.absolute element name

When the screen clears and a question mark appears, type in the data cards. There are usually 6 kinds of data cards. The formats and descriptions are as follows:

CARD 1: read ITMAX, MODEL, N, MQT, IG, IWTFAC, IVES, ICOEFS
format (20I4)

ITMAX = the maximum number of iterations

MODEL = the number of layers in the model (maximum = 8)

N = the number of observations (maximum = 40)

MQT = 1 for ridge regression. If MQT = 0, then the Newton-Raphson technique is used for the inversion. Ridge regression is the preferred method.

IG = 1 normally. If the parameters are to be restricted to a maximum variation, then IG = 2 and the log of the maximum variation will be input along with the initial guess parameters.

IWTFAC = 0 normally. If the data is to be weighted, then

IWTFAC = 1 and the vector of weights SNORM(J), J=1,40 is read in on the next card with format (40F2.0).

IVES = an integer identifier for the plot. Also this is used to control the output. If IVES>10 then the regular

output is obtained. If $IVES \leq 10$ then an abbreviated output is generated instead. Examples of these are in the section on EXPLANATION OF OUTPUT.

ICOEFS = 0 normally. There is a provision for reading a filter with 436 coefficients rather than the normal 64 coefficients from subroutine RDCOEF. If ICOEFS = 1 then 436 coefficients will be read from file 2, format (4E16.7). (See Rijo (1977))

CARD 2: If IWTFAC = 1 then read SNORM(J), J = 1,40
format(40F2.0)
otherwise: read V(J), J = 1,N
format(10F8.0)

V(J) = the jth measured voltage drop.

CARD 3: read XI(J), J = 1,N
format(10F8.0)

XI(J) = the jth transmitted current.

CARD 4: read F(I), I = 1,N
format(16F5.0)

F(I) = the ith AB/2 distance

CARD 5: read XMN(I) , I = 1,N
format(16F5.0)

XMN(I) = the ith MN distance

CARD 6: read P(J), J = 1, 2*MODEL - 1

format(10F8.0)

P(J) = the jth model parameter value (initial guess parameter vector). Resistivities are first, followed by thicknesses. (e.g. $\rho_1, \rho_2, \rho_3, t_1, t_2$)

If IG = 2, read PVAR(J), J = 1, 2*MODEL - 1

format(10F8.0)

PVAR(J) = the maximum log variation permitted for the jth model parameter (used when a parameter constraint is desired).

A data file could also be used instead of typing all the field data every time an inversion is attempted. A data file is set up as follows:

Type:

@ASG,CP DATA1.

@EDIT DATA1.

INPUT

Type in data cards

hit return key

type EXIT

Then, when running the program, after typing:

```
@XQT,U filename.absolute element name
```

the screen clears and the question mark appears. Then type:

```
@ADD DATA1.
```

An example of properly formatted input data cards is in section VI-A. of this report.

When the plot is finished, a copy can be made (if using a terminal with a copier attached). Then, to finish the program a carriage return must be typed. When the program has exited, the output can be retrieved by typing:

```
@SUSPEND
```

```
@DATA,L 3.
```

```
@END
```

```
@RESUME,P GEOPHY
```

This will send the output to the line printer in room 319 WBB. The output may be examined on the terminal by entering:

```
@EDIT,R 3.
```

Edit mode is now engaged and the regular edit commands may be used.

C. Explanation of Output

Examples of the regular and abbreviated outputs from a typical inversion follow in section VI-B, -C, and -D of this report. In the regular output, 6 of the subroutines write data out. The program is only allowed to print at 4 different times. The first printing comes from the main program just after the data are read and the initial guess parameters are

entered. The AB/2 distance, MN, current, voltage, calculated apparent resistivity and data weights are listed first for all observations. next, the initial guess parameters (in the order in which they are read) are printed along with $\log_{10}$ parameters, variances of parameters, max. and min. values (calculated at max variation if entered).

The second printing is during the second-to-last iteration. The value of the marquardt factor (DIAGAD) is printed, the determinant of the system matrix $(\bar{A}^T \bar{A} + k \bar{I})$, and the parameter correlation matrix are printed from XLSQ. SUBVRT then lists the parameters from the previous iteration versus the current parameters, and then for each AB/2 value, the observed versus calculated apparent resistivity and the difference are listed. LGPLT2¹ then writes no. of data points (N1), no. of unique AB/2 values (N), and then lists the data that are plotted. ($X1 = \log_{10} AB/2$, $Y1 = (\log_{10} \rho_a) - 1$, $Y2 = (\log_{10} \rho_{a_{new}}) - 1$) SUBVRT then writes the value of reduced CHI-SQUARE for the old parameters (YLSO), and then the new (YLSE). YINVRT lists the following: log parameter change ($\log_{10} \Delta \bar{P} = \log_{10} (\bar{A}^T \bar{A} + k \bar{I})^{-1} \bar{A}^T \Delta \bar{G}$), log parameters ($\log_{10} P$), log deviation ($\sqrt{YLSO} \sum_{i=1}^{NP} \sqrt{COV}$), fractional change (ΔP), Parameters(P), standard deviation ($\sigma = \log_{10}^{-1} \text{SIGMA}$) and pct deviation ($\frac{\sigma}{P} \times 100$). Parameters are then listed with low and high values, corresponding to one standard deviation (in log space).

The third printing is during the last iteration. This printing is the same as the second printing except for the following: observed versus calculated resistivity is omitted and SANDR writes the resolution matrix $(\bar{R} = \bar{H} \bar{A} = \{(\bar{A}^T \bar{A} + k \bar{I})^{-1} \bar{A}^T\} \bar{A})$, and the information density matrix

¹LGPLT2 draws the plot on the TEKTRONIX screen.

($\bar{S} = \bar{A}\bar{H}$, should be diagonally dominant). The remaining output is the same as the 2nd printing except one more iteration has taken place. (Also the Marquardt factor has been set equal to zero so $\Delta\bar{P} = (\bar{A}^T\bar{A})^{-1}\bar{A}^T\bar{A}\bar{G}$.)

The fourth printing is done in the MAIN program after YINVRT has returned control to MAIN. A listing of data for each iteration is made: Reduced chi-squared ($RCSQ = X^2v$), DIAGAD (MARQUARDT factor k), XCM, and XCT (variables which control the ridge regression). The next data printed are as follows (column by column): $RCSQ(X^2v)$, first parameter (ρ_1), PCT (% deviation), second parameter (ρ_2), PCT (% deviation), third parameter (ρ_3), PCT (% deviation), fourth parameter (t_1), PCT, etc. until all parameters are done. Notice that PCT for iteration 5 is the same number as PCT DEVIATION from the third printing. Also notice that the parameter values printed on the plot screen are from the third-from-last iteration.

1. Output on the Tektronix Screen

The subroutine LGPLT2 draws on the TEKTRONIX terminal screen. A copy of the screen output follows in section VI-C corresponding to the previous output.

In the upper left corner is the title: VES - 11. The initials VES stand for vertical electrical sounding, while 11 is the integer identifier for the plot that was entered on data card 1 for the variable name IVES.

The next thing down from the title is a list of the inverted parameters. RHO stands for the apparent resistivity of each layer (in ohm-meters) and H is layer thickness (in meters). These parameters are from the third-from-last iteration as can be seen by comparing them to those listed at the end of the computer printout corresponding to iteration

number.

The plot is a log-log plot of $AB/2$ distance versus apparent resistivity. The dots are the field data points, and the theoretical curve is calculated from the inverted parameters and interpolated using the cubic spline IMSL library subroutine ICSSCU.

2. Abbreviated Output

The abbreviated output includes results with a minimum of statistics. This output was specified by setting $IVES \leq 10$ on the first input data card. In the example, 251 lines of regular output were condensed into 119 lines.

II. THEORY

A. How The Program Works (Inversion Theory)

The inversion of geophysical field data involves two steps. The first step is solving the forward problem, i.e. calculating the response of an assumed model. The second step is the inverse problem, which is mathematical adjustment of the model so that the response fits the field data.

1. The Forward Problem (See Rijo et al., 1977)

The potential (voltage) at a distance r from a point current source I on the surface of a layered earth is given by (see theoretical development):

$$V(r, \vec{P}) = \frac{I}{2\pi} \int_0^{\infty} R(\lambda, \vec{P}) J_0(\lambda r) d\lambda$$

where $\vec{P}$ is the vector of parameters (resistivities and thicknesses of the horizontal layers). The Schlumberger apparent resistivity is then

$$\rho_a = \frac{k}{I} [V(r_1, \vec{P}) - V(r_2, \vec{P})]$$

where

$$r_1 = \frac{AB}{2} - \frac{MN}{2}$$

$$r_2 = \frac{AB}{2} + \frac{MN}{2}$$

$$k = 2\pi \left(\frac{1}{AM} - \frac{1}{BM} - \frac{1}{AN} + \frac{1}{BN} \right)^{-1}$$

If we take the potential integral and substitute:

$$\lambda = e^{-y}$$

and

$$r = e^x$$

then the potential is:

$$V(r, \vec{P}) = \frac{-I}{2\pi r} \int_{-\infty}^{\infty} R(y, \vec{P}) e^{x-y} J_0(e^{x-y}) dy$$

which is in the form of a convolution integral

$$V(r, \vec{P}) = \int_{-\infty}^{\infty} f(y) g(x-y) dy$$

where $R(y, P)$ can be considered as an input function $f(y)$, and $e^x J_0(e^x)$ as the impulse response of a stationary filter. If $R(y, \vec{P})$ is evaluated at N discrete points, we can approximate $V(r, \vec{P})$ by a finite sum

$$V(r, \vec{P}) \approx \frac{I}{2\pi r} \sum_{j=1}^N c_j R(\ln r - \eta_j, \vec{P})$$

where N_j are the abscissas of the stationary filter coefficients c_j .

This program uses a 61-point filter to evaluate $V(r, P)$ with adequate accuracy and minimal computer cost.

2. The Inverse Problem (See Rijo et al., 1977)

Usually in resistivity data, the error is a fixed percentage of ρ_a , the apparent resistivity. Therefore it is convenient to work with $\log_{10} \rho_a$ instead. (For example, data with 5% random error in voltage ΔV is $.05R\Delta V + \Delta V \approx \text{data}$ where R is a random number with normal distribution and

standard deviation = 1. Then

$$\log_{10}(\text{data}) = \log_{10}\Delta V + \log_{10}(.05R+1) = \log_{10}\Delta V + \epsilon$$

where ϵ is the error which is now additive. This relation was used in testing the program. See discussion of random noise.)

Let $\vec{G}$ be the vector of log apparent resistivities. And, let $\vec{P}$ be the vector of log parameters. Then

$$\vec{G} = \vec{A}\vec{P} + \epsilon$$

where $A_{ij} = \frac{\partial G_i}{\partial P_j} \bigg|_{\vec{P}_0}$

Now since $\vec{A}$ is overdetermined (there are more observations than parameters) and if $\vec{P}_0$ (initial guess vector of parameters) is close to the best-fitting vector of log parameters $\vec{P}$, then

$$\vec{P} - \vec{P}_0 = (\vec{A}^T\vec{A})^{-1}\vec{A}^T\Delta\vec{G} = \Delta\vec{P} \quad (1)$$

Now, $\vec{A}^T\vec{A}$ usually has small (≈ 0 as far as the computer is concerned) eigenvalues for many geophysical situations. This makes $(\vec{A}^T\vec{A})^{-1}$ unstable (maybe nonexistent). Then using (1) to iterate onto the solution $\vec{P}$ is not stable. In order to improve the situation this computer program uses ridge regression.

Equation (1) is modified as follows:

$$\Delta\vec{P} = (\vec{A}^T\vec{A} + K\vec{I})^{-1}\vec{A}^T\Delta\vec{G} \quad (2)$$

where $\vec{I}$ is the identity matrix and K (The Marquardt factor) is a small positive number. Actually $\vec{A}^T\vec{A}$ is scaled so that it has ones down the

diagonal. If K is very small then equation (2) approaches equation (1). If K is large, then equation (2) used to modify the parameters is called the gradient method. The gradient method always converges to the solution $\vec{P}$ but is usually too slow. The method of varying K at each iteration is called ridge regression. Convergence is speeded up as K decreases (up to a certain point where the solution begins to diverge due to instability). Therefore, there is an optimum value of K which is different for every problem and iteration.

In this computer program, K is called DIAGAD. It starts out being 10^{-4} and increases or decreases according to a goodness of fit criterion called reduced chi-squared:

$$\chi^2_r = (1/N-NP) \sum_{k=1}^N (\log \rho_a - \log \rho_{a \text{ new}})^2$$

where N is the number of data observations, NP is the number of parameters, ρ_a is the apparent resistivity calculated from the data observations, and $\rho_{a \text{ NEW}}$ is the apparent resistivity calculated from the current parameter values. If χ^2_r decreases after new parameters are calculated with DIAGAD = 10^{-4} , then DIAGAD gets decreased to 10^{-5} . If χ^2_r increases, then DIAGAD is increased.

When the algorithm converges, $\Delta \vec{P} \rightarrow 0$ and an estimate of the data variance is given by χ^2_r from the final parameter vector $\vec{P}$. The covariance matrix is then

$$\text{cov}(\vec{P}) = \chi^2_r (\bar{A} \bar{A})^{-1}$$

The correlation matrix is determined by normalizing with respect to the diagonal elements of $\text{cov}(\vec{P})$.

$$[\text{cor}(\vec{P})]_{ij} = \frac{[\text{cov}(\vec{P})]_{ij}}{[\text{cov}(\vec{P})]_{jj}^{1/2} [\text{cov}(\vec{P})]_{ii}^{1/2}}$$

Also, in the computer program, $\bar{A}$ is calculated using the first forward difference approximation. This is done in subroutine NUDERV with the increment being 1% of the parameter value.

B. Schlumberger Resistivity Theory

Suppose a point current source electrode is at the surface of an isotropic homogeneous halfspace. The current flows radially outward perpendicular to a hemispherical surface of radius r .

Since the hemisphere has surface area $2\pi r^2$, the current density J_r (current per unit area) is

$$J_r = I/2\pi r^2$$

at a distance r from the current source I . From the constitutive relation $\vec{J} = \sigma \vec{E}$ (ohm's law), and the relationship between electric field and potential (voltage) $\vec{E} = -\nabla V$ (in our case $E_r = -\partial V/\partial r$), we have $V = I\rho/2\pi r$ where $(\rho = 1/\sigma)$.

Now, if we consider a pair of current sources, one putting current into the ground (+I) and one removing current from the ground (-I), and we have two detecting electrodes, between which we will measure a potential

difference (voltage). The array looks like the following:

This is the Schlumberger array layout with the distance between I and $-I$ equal to AB and the distance between P_1 and P_2 equal to MN .

Now suppose the earth is an isotropic homogeneous halfspace with resistivity ρ . The potential (voltage) observed at the point P_1 is (remember potential is additive from each current source):

$$V(P_1) = \frac{I\rho}{2\pi} \left(\frac{AB}{2} - \frac{MN}{2} \right)^{-1} - \frac{I\rho}{2\pi} \left(\frac{AB}{2} + \frac{MN}{2} \right)^{-1}$$

$$= \frac{I\rho}{2\pi r^*}$$

where

$$\frac{1}{r^*} = \frac{1}{\frac{AB}{2} - \frac{MN}{2}} - \frac{1}{\frac{AB}{2} + \frac{MN}{2}}$$

The potential $V(P_2)$ observed at the point P_2 is likewise

$$V(P_2) = I\rho/2\pi r^+, \text{ where } 1/r^+ = -1/r^*$$

Then the potential difference (measured voltage) between the two points P_1 and P_2 is:

$$\Delta V = V(P_1) - V(P_2) = I\rho/\pi r^*$$

Now look at a more complicated geometry:

The halfspace is now two layers. The top layer has resistivity ρ_1 , and the bottom layer has resistivity ρ_2 . We want to know the potential everywhere in the halfspace. The potential in each layer can be thought of as consisting of a primary potential (potential of a homogeneous halfspace with resistivity equal to that in the upper layer), and a secondary potential (perturbation due to bottom layer).

$$V' = V_p + V_{s1} \quad (\text{potential in top layer})$$

$$V'' = V_p + V_{s2} \quad (\text{potential in bottom halfspace})$$

As before we can solve for V_p in this geometry.

$$\begin{aligned} V_p &= \frac{I\rho_1}{2\pi} (z^2 + r^2)^{-\frac{1}{2}} \\ &= \frac{I\rho_1}{2\pi} \int_0^\infty J_0(\lambda r) e^{-\lambda z} d\lambda \\ r &= \sqrt{x^2 + y^2} \end{aligned}$$

where J_0 is the zeroth order Bessel function of the first kind and we have used a Laplace transform identity. The reason for doing this is to match with the general solutions for the secondary potentials found from EM theory applied to the electrostatic case (D.C. current) with no θ dependence for the point electrode.

$$V_{s1} = \int_0^{\infty} [f_1(\lambda)e^{-\lambda z} + g_1(\lambda)e^{\lambda z}] J_0(\lambda r) d\lambda$$

$$V_{s2} = \int_0^{\infty} [f_2(\lambda)e^{-\lambda z} + g_2(\lambda)e^{\lambda z}] J_0(\lambda r) d\lambda$$

Now, in order to solve for the 4 unknowns f_1 , f_2 , g_1 , and g_2 , we use 3 boundary conditions:

- (1) V_{s2} due to the point electrode goes to zero as $z \rightarrow \infty$
- (2) continuity of potentials across interface $V' = V''|_{z=d_1}$
- (3) normal current densities across interfaces continuous

When these boundary conditions are imposed, we find

$$g_2 = 0$$

$$f_1 = g_1 = \frac{I\rho_1}{2\pi} \left(\frac{-u_1 e^{-2\lambda d_1}}{1 + u_1 e^{-2\lambda d_1}} \right)$$

$$f_2 = \frac{I\rho_1}{2\pi} \left[\frac{-u_1 (e^{-2\lambda d_1} + 1)}{1 + u_1 e^{-2\lambda d_1}} \right]$$

where

$$u_1 = \frac{\rho_1 - \rho_2}{\rho_1 + \rho_2}$$

We now substitute in and the resulting secondary potential at $z=0$ (we only care about measuring at the surface of the earth) is

$$V_{s1} = \frac{I\rho_1}{2\pi} \int_0^{\infty} \frac{-2u_1 e^{-\lambda d_1}}{1 + u_1 e^{-2\lambda d_1}} J_0(\lambda r) d\lambda$$

then the total is

$$V' = \frac{I\rho_1}{2\pi} \int_0^\infty \frac{1 - u_1 e^{-2\lambda d_1}}{1 + u_1 e^{-2\lambda d_1}} J_0(\lambda r) d\lambda$$

In a similar fashion we can solve for the potential over an N-layered earth using 2N boundary conditions to solve for the 2N unknowns $f_1, g_1, f_2, g_2, \dots, f_N, g_N$.

The result is

$$V_N = (I\rho_1/2\pi) \int_0^\infty R(\lambda, \vec{P}) J_0(\lambda r) d\lambda = V(r, \vec{P})$$

where

$$R(\lambda, \vec{P}) = k_{12\dots N} = \frac{1 - u_{12\dots N} e^{-2\lambda d_1}}{1 + u_{12\dots N} e^{-2\lambda d_1}}$$

$$u_{12\dots N} = \frac{\rho_1 - \rho_2 k_{23\dots N}}{\rho_1 + \rho_2 k_{23\dots N}}$$

$$k_{(m-1)m\dots N} = \frac{1 - u_{(m-1)m\dots N} e^{-2\lambda d_{m-1}}}{1 + u_{(m-1)m\dots N} e^{-2\lambda d_{m-1}}}$$

$$u_{(m-1)m\dots N} = \frac{\rho_{m-1} - \rho_m k_{m(m+1)\dots N}}{\rho_{m-1} + \rho_m k_{m(m+1)\dots N}}$$

$$K_{(N-1)N} = \frac{1 - U_{(N-1)N} e^{-2\lambda d_{N-1}}}{1 + U_{(N-1)N} e^{-2\lambda d_{N-1}}}$$

$$U_{(N-1)N} = \frac{\rho_{N-1} - \rho_N}{\rho_{N-1} + \rho_N}$$

This is the integral in section II-A-1. that calculates the potential (voltage) at a distance r from a point current source I on the surface of an N-layered earth. $\vec{P}$ is the vector of parameters:

$$\vec{P} = (\rho_1, \rho_2, \rho_3, \dots, \rho_N, d_1, d_2, \dots, d_{N-1}) \quad \underline{2N-1 \text{ parameters}}$$

III. TESTS MADE ON THE PROGRAM

A. Discussion of the Effect of Noise

Three models were inverted to investigate the effects of random noise in the data. Random noise (added to theoretical data) is an approximation to real data noise.

Five percent random noise was used because a good field crew can take Schlumberger data which is accurate to within five percent. A forward program generated a set of $AB/2$ distances versus apparent resistivity ρ_a . A second program added noise to ρ_a and then generated field data (voltage, current, and MN distance). The noise was added as follows:

1. A set of pseudo-random numbers were generated with mean=0 and $\sigma=1$.
2. Apparent resistivity was then modified $\rho_a(\text{NEW})_i = .05R_i\rho_a(\text{OLD})_i + \rho_a(\text{OLD})_i$ where R_i is the random number corresponding to the i th data point.

The first model investigated was a three-layer earth:

$$t_1 = t_2 = 100\text{m}$$

The input and output from this inversion is the sample output in this

report (section VI). The initial guess vector of parameters was ten percent lower than the actual parameters (for example, $\rho_1 = 90\Omega\text{-m}$ and $t_1 = 90\text{m}$). With no noise added and after 5 iterations, the parameters were:

limits*		correlation matrix				
$\rho_1 = 100.$	(100.,100.)	ρ_1	1.0			
$\rho_2 = 14.4$	(12.3,16.9)	ρ_2	.29	1.0		
$\rho_3 = 1050.$	(982.,1120.)	ρ_3	.15	.47	1.0	
$t_1 = 98.4$	(96.3,100.)	t_1	-.35	-.98	-.39	1.0
$t_2 = 139.$	(118.,165.)	t_2	.29	1.0	.50	-.98 1.0
			ρ_1	ρ_2	ρ_3	t_1 t_2

With five percent noise added and the same initial guess vector of parameters:

		correlation matrix				
$\rho_1 = 99.1$	(97.4,101.)	ρ_1	1.0			
$\rho_2 = 18.8$	(9.04,39.2)	ρ_2	.32	1.0		
$\rho_3 = 1980.$	(652.,6030.)	ρ_3	.25	.49	1.0	
$t_1 = 99.0$	(84.3,116.)	t_1	-.37	-.95	-.32	1.0
$t_2 = 190.$	(86.2,417.)	t_2	.33	1.0	.53	-.94 1.0
			ρ_1	ρ_2	ρ_3	t_1 t_2

As can be seen, the random noise destroyed the ability to "see" down to the basement. The data fit (see section 11-B-1.-b How good are

*limits - High and low values corresponding to $\log^{-1}_{10} (\log_{10} \text{PARAMETER} + \log_{10} \sigma)$ and $\log^{-1}_{10} (\log_{10} \text{PARAMETER} - \log_{10} \sigma)$.

the parameters) for noiseless data was $\epsilon = .54\%$. For the noisy data $\epsilon = 5.62\%$. These data fit parameters indicate that convergence has been good and more iterations will only end up fitting the noise and not the data.

Another factor to be considered is the high correlations between some of the parameters. Inman (1975, p. 804) points out that if the correlations are small, then the standard deviation is a good measure of the uncertainty of each parameter. However, if the correlation is large (correl. coeff. ≈ 1), then the standard deviations will be larger than the actual uncertainties. When the correlation coefficient is +1, then only the ratio of thickness to resistivity can be determined (longitudinal conductance). When the coefficient is -1, then only the resistivity times the thickness can be determined (transverse resistance). Thin conductive layers often have correlation coefficients close to +1.

In this model, high positive correlation is observed between t_2 and ρ_2 . This means that only $t_2/\rho_2 = s$ can be resolved, but not each parameter individually. High negative correlation between $t_1-\rho_2$, and t_1-t_2 is also observed. These high correlations can be an indicator that the problem should be reparameterized.

An easier way to determine parameters that are highly correlated is to use geologic information to fix one parameter (ρ_2 probably), then let the other one be inverted. This geologic information could be well log data, EM sounding data, or lab resistivity determinations on outcrop samples.

The second model was a four-layer earth:

$t_1 = 30m$	$\rho_1 = 20 \Omega\text{-m}$
$t_2 = 30m$	$\rho_2 = 160 \Omega\text{-m}$
$t_3 = 20m$	$\rho_3 = 10 \Omega\text{-m}$
	$\rho_4 = 900 \Omega\text{-m}$

The initial guess was 10% lower for each parameter (same as the first model). With no noise added and after 5 iterations:

limits		correlation matrix							
$\rho_1 = 20.4$	(20.3,20.4)	ρ_1	1.0						
$\rho_2 = 79.7$	(56.0,113.)	ρ_2	.22	1.0					
$\rho_3 = 9.42$	(.0589,1500.)	ρ_3	-.04	.04	1.0				
$\rho_4 = 917.$	(890.,946)	ρ_4	.06	-.31	.22	1.0			
$t_1 = 27.8$	(25.8,30.1)	t_1	.30	.98	-.03	-.29	1.0		
$t_2 = 44.3$	(25.2, 78.1)	t_2	-.17	-.92	-.41	.24	-.87	1.0	
$t_3 = 17.3$	(.102,2930.)	t_3	-.03	.07	1.0	.21	.00	-.45	1.0
			ρ_1	ρ_2	ρ_3	ρ_4	t_1	t_2	t_3

$$RCSQ = 1.008 \times 10^{-5}$$

$$\epsilon = .73\%$$

With the data fit error so low, there is no better resolution by increasing the number of iterations. With 5% random noise added as before, and the same initial guess,

		correlation matrix			
$\rho_1 = 20.1$	(19.7,20.6)	ρ_1	1.0		
$\rho_2 = 72.3$	(16.3,320.)	ρ_2	.25	1.0	
$\rho_3 = 10.3$	(1.03×10^{-5} , 1.03×10^8)	ρ_3	.16	.51	1.0

$$\rho_4 = 1030. \quad (766., 1380.)$$

$$t_1 = 27.9 \quad (17.1, 45.7)$$

$$t_2 = 67.7 \quad (1.32, 3460.)$$

$$t_3 = 15.7 \quad (1.57 \times 10^{-5}, 1.57 \times 10^8)$$

ρ_4	.06	-.41	-.24	1.0			
t_1	.33	.98	.55	-.38	1.0		
t_2	-.22	-.96	-.70	.47	-.93	1.0	
t_3	.17	.55	1.0	-.26	.59	-.74	1.0
	ρ_1	ρ_2	ρ_3	ρ_4	t_1	t_2	t_3

$$\text{RCSQ} = 6.835 \times 10^{-4}$$

$$\mathcal{E} = 6.20\%$$

The large standard deviations on ρ_3 and t_3 are due to the 1.0 correlation coefficient. Other large correlation coefficients occur between ρ_2 and t_2 (can only resolve $\rho_2 \times t_2$), between t_1 and ρ_2 (only t_1/ρ_2), between t_2 and t_1 (only $t_2 \times t_1$), and a fairly large value of $-.74$ between t_2 and t_3 . The third layer is not resolved by the data and the random error that was added only made the resolution poorer. The second layer was a thin resistive layer which made resolution fairly difficult as well. The resolution was decreased significantly with the addition of noise.

The third model also had four layers:

The initial guess was 10% lower than the above values.

The noise decreased the resolution on t_2 and ρ_4 most significantly. Except for the resolution of layer 3 (a thin conductive layer), the data seems to fit quite well.

In conclusion, the effect of noise in the data is to decrease resolution. The amount of decrease in resolvability appears greater in correlated parameters than in uncorrelated ones.

B. Effect of Using the Wrong Model for Inversion

Only one example of inversion to the wrong model was done. The model used was the same model as model 3 of the last section on noise in the data.

$t_1 = 20m$	$\rho_1 = 100 \Omega\text{-m}$
$t_2 = 100m$	$\rho_2 = 50 \Omega\text{-m}$
$t_3 = 20m$	$\rho_3 = 10 \Omega\text{-m}$
	$\rho_4 = 1000 \Omega\text{-m}$

The data were inverted to a three layer model to try and eliminate the thin conductive layer at depth (layer 3). The initial guess parameters were:

$$\rho_1^0 = 90 \Omega\text{-m}$$

$$\rho_2^0 = 45 \Omega\text{-m}$$

$$\rho_3^0 = 900 \Omega\text{-m}$$

$$t_1^0 = 18m$$

$$t_2^0 = 90m$$

The noise-free data were inverted first. The result after 5 iterations was:

$\rho_1 = 100.$ (99.8,100.)
 $\rho_2 = 49.2$ (48.9,49.5)
 $\rho_3 = 1080.$ (1050.,1110.)
 $t_1 = 21.$ (20.8,21.3)
 $t_2 = 197.$ (195.,199.)

correlation matrix

ρ_1	1.0				
ρ_2	.18	1.0			
ρ_3	.07	.35	1.0		
t_1	-.39	-.76	-.20	1.0	
t_2	.18	.87	.69	-.65	1.0
	ρ_1	ρ_2	ρ_3	t_1	t_2

$$\text{RCSQ} = 6.022 \times 10^{-6}$$

$$\mathcal{E} = .57\%$$

The data with 5% noise were then inverted with the same initial guess. After 5 iterations:

$\rho_1 = 98.8$ (96.9,101.)
 $\rho_2 = 50.3$ (47.4,53.4)
 $\rho_3 = 1200.$ (892.,1610.)
 $t_1 = 20.4$ (17.9,23.2)
 $t_2 = 203.$ (183.,225.)

ρ_1	1.0				
ρ_2	.18	1.0			
ρ_3	.08	.35	1.0		
t_1	-.39	-.75	-.19	1.0	
t_2	.18	.87	.69	-.64	1.0
	ρ_1	ρ_2	ρ_3	t_1	t_2

$$\text{RCSQ} = 5.799 \times 10^{-4}$$

$$\mathcal{E} = 5.70\%$$

These results can be compared with the four-layer inversion results of

the last section. After 5 iterations, the data with no noise added was fit only slightly better with four layers than with three ($\epsilon = .50\%$ for 4 layers, and $\epsilon = .57\%$ for 3 layers). With noise added, after 5 iterations the three layer model fit better than the four layer model ($\epsilon = 5.70\%$ with 3 layers, and $\epsilon = 6.17\%$ with 4 layers). The problem with high correlation coefficients and corresponding low resolution with four layer interpretation has been eliminated by using three layers. The highest correlation coefficient in the three layer interpretation is .87 between ρ_2 and t_2 . This signifies a thin conductive layer. The standard deviations are quite small compared to the four layer case.

The conclusion is that with noise in the data (field data), sometimes the wrong model will fit the data best. However, the wrong model is probably close to the correct model interpretation. Since we probably seldom deal with data from a truly one-dimensional layered earth, the interpretation must be constrained geologically anyway. The results from a one-dimensional resistivity interpretation together with results from another geophysical method interpretation are more meaningful than resolving the difference between two similar models with one data set. This must be kept in mind when using this program to interpret data.

IV. DOCUMENTATION

A. Program Subroutine Descriptions

The program consists of 14 subroutines and a main program. These subroutines and the interconnections between them can be shown by the following diagram. The boxes represent the subroutines with the name of each inside the box. The arrows indicate which subroutine calls which other subroutine.

The following is a list of things done by each subroutine in roughly the sequence that they are first called. For details consult program listing.

MAIN - Reads in control data, and field data. Converts data to $\log_{10}$ data, reads data weights (if any), calculates apparent resistivities from field data, and writes all this out. Reads the initial guess parameter vector $P(J)$, takes $\log_{10} P(J)$, and writes these out. Parameters to remain unchanged in inversion are fixed and YINVRT is called to start inversion. Finally, the data from each inversion iteration is written out and the program stops.

- FIX - Removes from vector of parameters those parameters which are fixed in inversion, or replaces in the vector fixed parameters depending on the value of IFU.
- RDCOEF - Data used by RESIST. This data consists of filter abscissas and coefficients for evaluating the resistivity integral. (See theoretical explanation).
- YINVRT - Main driving program for inversion. Stores data from each iteration in SUMUP (I,J) to be printed out in MAIN. Uses a modified Marquardt algorithm to vary the Marquardt factor in the ridge regression expression (See theoretical development.) Uses reduced chi-squared criteria to stop iterations. Writes out on last two iterations log parameter change, log parameters, log deviation, fractional change, parameters, standard deviation, % deviation.

- CDELTY - Calculates reduced chi-squared given a set of parameters and original data.

$$RCHISQ = \frac{1}{N-NP} \sum_{k=1}^N (\rho_a - \rho_{a_{NEW}})^2$$

where N = no. of observations in field data

NP = no. of parameters (2 x (no. layers) - 1)

ρ_a = apparent resistivity calculated from field data (observed).

$\rho_{a_{NEW}}$ = apparent resistivity calculated from current version of parameters.

- CY - Calculates apparent resistivity for current parameter values ($\rho_{a_{NEW}}$).

$$\rho_{a_{NEW}} = \frac{2\pi(V_1 - V_2)}{(\frac{1}{R_1} - \frac{1}{R_2})}$$

where V_1 and V_2 are voltages observed over layered earth with current parameters ($\rho_1, \rho_2, \dots, h_1, h_2, \dots$) calculated in RESIST.

$$R_1 = \frac{AB}{2} - \frac{MN}{2}$$

$$R_2 = \frac{AB}{2} + \frac{MN}{2}$$

- RESIST - Calculates voltages on a layered earth about a point electrode. Uses a convolution method. (See theoretical development for reference)
- RIDGE - Calculates Marquardt factor to be added to diagonal of normalized matrix $\bar{A}^T \bar{A}$
- SUBVRT - Calculates new reduced chi-square, new theoretical observables, and new parameters.

NUDERV - Calculates system matrix from numerical derivative using log parameters.

The system matrix

$$[\bar{A}]_{ij} = \frac{\partial p_{ai}}{\partial P_j} \quad \text{where } P_j = \log_{10} p_j$$

$$\frac{\partial p_{ai}}{\partial P_j} = \frac{\partial p_{ai}}{\partial [\log_{10} p_j]} = \frac{\partial p_{ai}}{\partial p_j} \frac{\partial p_j}{\partial [\log_{10} p_j]}$$

and

$$\frac{\partial p}{\partial [\log_{10} p]} = \frac{\partial (10^{\log_{10} p})}{\partial [\log_{10} p]} = (\ln 10) p$$

then

$$\frac{\partial p_{ai}}{\partial P_j} = (\ln 10) p_j \frac{\partial p_{ai}}{\partial p_j}$$

Now we approximate the derivative by the first forward difference

$$\frac{\partial p_{ai}}{\partial P_j} \approx \frac{p_{ai}(\log_{10}[p_j + \Delta p_j]) - p_{ai}(\log_{10} p_j)}{\Delta p_j}$$

and $\Delta p = mp$ where $m = .01$ for 1% of parameter

$$\frac{\partial p_{ai}}{\partial P_j} \approx \frac{p_{ai}(\log_{10}[p_j(1+m)]) - p_{ai}(P_j)}{mp_j}$$

$$\frac{\partial p_{ai}}{\partial P_j} \approx \frac{p_{ai}(P_j + \log_{10}[1+m]) - p_{ai}(P_j)}{m p_j}$$

then

$$[\bar{A}]_{ij} \approx \frac{(\ln 10) p_j}{.01 p_j} [p_{ai}(P_j + .004) - p_{ai}(P_j)]$$

$$A_{ij} = \frac{p_{ai}(P_j + .004) - p_{ai}(P_j)}{.004}$$

inis is the way that the system matrix is calculated in NUDERV.

- XLSQ - Computes parameter covariance matrix, parameter standard deviation, and parameter correlation matrix.
- MAINE - Inverts $(A^T A + kI)$ symmetric matrix.
- SANDR - Computes pseudo inverse $H = (A^T A + kI)^{-1} A^T$. Also computes and writes out resolution matrix $R = HA$, and information density matrix $S = AH$.
- LGPLT2 - Draws plot and writes on Tektronix screen. Also writes on output no. of data points, no. of unique points $AB/2$, $\log_{10} AB/2$, $(\log_{10} \rho_a) - 1$, $\log_{10} AB/2$, $(\log_{10} \rho_{aNEW}) - 1$. Calls a library function ICSSCU which puts a smooth cubic spline along the data points before plotting the curve.
- DRAWX - Plots ρ_a and $AB/2$ on Tektronix screen. Called by LGPLT2.

B. Program Listing

A listing of the program follows in approximately the order in which the subroutines are called. Comment cards explain the main developments in the program.

<u>SUBROUTINE</u>	<u>PAGE</u>
MAIN.	39
FIX	43
RDCOEF.	45
YINVRT.	46
CDELTY.	50
CY.	51
RESIST.	52

RIDGE	53
SUBVRT.	55
NUDERV.	57
XLSQ.	58
MAINE	61
SANDR	62
LGPLT2.	64
DRAWX	68
ICSSCU	69

```

1.      C LEAST SQUARES INVERSION PROGRAM FOR SCHLUMBERGER DATA
2.      C ITMAX=NO. OF ITERATIONS
3.      C MODEL=NO. OF LAYERS
4.      C N=NO. OF OBSERVATIONS
5.      C MQT=0 NEWTCN-RAPHSON
6.      C MQT=1 GRADIENT-STARTER
7.      C IG=1 INPUT INITIAL GUESS
8.      C IG=2 INPUT INITIAL GUESS PLUS LOG OF MAX VARIATION
9.      C IWTFAC=1 INPUT SNORM
10.     C IVES=INTEGER IDENTIFIER FOR PLOT. ALSO THE PROGRAM HAS BEEN
11.     C MODIFIED SUCH THAT FOR IVES<=10, AN ABBREVIATED OUTPUT IS
12.     C OBTAINED. FOR IVES >10, THE REGULAR OUTPUT IS OBTAINED.
13.     C ICOEFS = 0 USE 61 COEFS IN RESIST
14.     C           = 1 USE 436 COEFS IN RESIST
15.     C           PARAMETER NN=40,MM=15
16.     C           COMMON/BLK2/FLOG(NN),F(NN),YLOG(NN),Y(NN),PLOG(MM),P(MM)
17.     C           COMMON/BLK5/YC(NN),DY(NN),A(NN,MM),SNORM(NN),X(MM),SIGMA(MM)
18.     C           COMMON/BLK7/PI(MM),PVAR(MM),IFIX,PMIN(MM),PMAX(MM)
19.     C           COMMON/BLK10/ITMAX,MODEL,N,NP,MQT,IPRT,IG,ILOG,IWTFAC,ICOEFS
20.     C           COMMON/BLK15/LAST,IPRS
21.     C           COMMON/BLK20/SUMUP(20,35),ITER
22.     C           COMMON/BLK32/XMN(40),IMN,XI(40),V(40)
23.     C           COMMON/BLKTK/IVES,RHOH(MM)
24.     C           COMMON/BLKCF/AO(436),WO(436)
25.     C           DATA IOFF,ION/017,016/
26.     C           CALL INITT(30)
27.     C           CALL TOUTPT(IOFF)
28.     C           CALL CHSYNC
29.     C           WRITE(6,200)
30.     C           200 FORMAT(1X,'?')
31.     C           READ 1 ITMAX,MODEL,N,MQT,IG,IWTFAC,IVES,ICOEFS
32.     C           IF(ICOEFS.EQ.1) READ(2,400) (WO(I),I=1,436),(AO(I),I=1,436)
33.     C
34.     C READ THE FILTER COEFFICIENTS
35.     C
36.     C IF(ICOEFS.EQ.0) CALL RDCOEFF
37.     C IF(ICOEFS.NE.1 .OR. ICOEFS.NE.0) WRITE(3,1) ICOEFS
38.     C 400 FORMAT(4(1PE16.7))
39.     C IPRT=1
40.     C ILOG =1

```

```

41.      IFIX=0
42.      1  FORMAT(20I4)
43.      C
44.      C  WEIGHTING FACTORS FOR DATA
45.      C
46.      IF(IWTFAC.EQ.1) READ 120 (SNORM(J),J=1,40)
47.      120  FORMAT(40F2.0)
48.      IPRS=0
49.      IF(IPRT.GE.3) IPRS=1
50.      IPR=IPRT
51.      NO=1
52.      NF=N
53.      C
54.      C  READING IN THE VOLTAGE AND CURRENT DROPS, AND NEXT THE
55.      C  AB/2 AND MN DISTANCES.
56.      C
57.      READ 2 (V(J),J=NO,NF)
58.      READ 2(XI(J),J=NO,NF)
59.      2  FORMAT(10F8.0)
60.      READ 3 (F(I),I=NO,NF)
61.      READ 3 (XMN(I),I=NO,NF)
62.      3  FORMAT(16F5.0)
63.      WRITE(3,4)
64.      4  FORMAT(1H ,16X,'AB/2',15X,'MN',15X,'I',16X,'V',13X,'OBSRES',11X,
65.      *'WEIGHT')
66.      C
67.      C  APPARENT RESISTIVITY IS CALCULATED FOR EACH DATA POINT AND
68.      C  LOG APP. RESIS. IS USED ALONG WITH LOG AB/2.
69.      C
70.      DO 5 J=NO,NF
71.      XR=XMN(J)/2.
72.      G=2./(F(J)-XR)-2./(F(J)+XR)
73.      Y(J)=(6.283185307*V(J))/(G*XI(J))
74.      YLOG(J)=ALOG10(Y(J))
75.      FLOG(J)=ALOG10(F(J))
76.      WRITE(3,6) J,F(J),XMN(J),XI(J),V(J),Y(J),SNORM(J)
77.      5  CONTINUE
78.      6  FORMAT(1H ,I4,6F17.3)
79.      WRITE(3,10)
80.      10  FORMAT(///)

```

```

81.      NP=2*MODEL-1
82.      IF(IG.GT.0) READ 2 (P(J),J=1,NP)
83.      DO 11 J=1,NP
84.      11 RHOH(J) = P(J)
85.      IF(IG.EQ.2) READ 2 (PVAR(J),J=1,NP)
86.      IF(IPRT.LT.2) WRITE(3,7)
87.      7  FORMAT(1H0,'INITIAL GUESS'/12X,'P',15X,'PLOG',13X,'PVAR',13X,
88.      *'PMIN',13X,'PMAX')
89.      DO 8 J=1,NP
90.      PLOG(J)=ALOG10(P(J))
91.      PI(J)=PLOG(J)
92.      IF(IG.EQ.2) PMIN(J)=PI(J)-PVAR(J)
93.      IF(IG.EQ.2) PMAX(J)=PI(J)+PVAR(J)
94.      IF(IPRT.LT.2) WRITE(3,9) P(J),PLOG(J),PVAR(J),PMIN(J),PMAX(J)
95.      IF(IG.EQ.2.AND.PVAR(J).LT.1.E-10) IFIX=IFIX+1
96.      8  CONTINUE
97.      9  FORMAT(1H ,1P6E17.3)
98.      IF(IFIX.EQ.0) GO TO 17
99.      NPI=NP
100.     C
101.     C  NP IS CEANGED TO UNIQUE NUMBER
102.     C
103.      CALL FIX(PLOG,NP,IFU)
104.      NP=NPL
105.      CALL FIX(PMIN,NP,IFU)
106.      NP=NPL
107.      CALL FIX(PMAX,NP,IFU)
108.     17 CONTINUE
109.      CALL YINVRT(FLOG,YLOG,PLOG)
110.     C
111.     C  IF ABBREVIATED OUTPUT IS DESIRED THEN SKIP DOWN
112.     C
113.      IF(IVES.LE.10)GO TO 21
114.      WRITE(3,60)
115.      DO 18 K=1,ITER
116.      K1=K-1
117.      WRITE(3,70) K1,(SUMUP(K,L),L=1,4)
118.     18 CONTINUE
119.     70 FORMAT(1H ,I3,1P4E14.4)
120.     60 FORMAT(1H1,/, ' ITER',6X,'RCSQ',10X,'DIAGAL',8X,'XCM',11X,'XCT')

```



```
121.      II=FLOAT(NP)/5.+1.
122.      DO 15 I=1,II
123.      WRITE(3,13)
124.      12 CONTINUE
125.      13 FORMAT(1H ,/, ' ITER',6X,'RCSQ',8X,5(13X,'PCT',5X))
126.      JO=5*(2*I-1)
127.      JF=JC+9
128.      DO 15 K=1,ITER
129.      K1=K-1
130.      WRITE(3,15) K1,SUMUP(K,1),(SUMUP(K,J),J=JO,JF)
131.      15 FORMAT(1H ,I3,1PE14.4,5(1PE13.4,0PF8.1))
132.      GO TO 20
133.      21 WRITE(3,61)
134.      61 FORMAT(1H0, ' ITER',6X,'RCSQ',10X,'DIAGAD')
135.      DO 22 K=1,ITER
136.      K1=K-1
137.      22 WRITE(3,70)K1,(SUMUP(K,L),L=1,2)
138.      20 CONTINUE
139.      CALL FINITT(0.,0.)
140.      STOP
141.      END
```

```

1.      SUBROUTINE FIX(PLOG,NP,IFU)
2.      C
3.      C   SUBROUTINE FIX REMOVES FROM VECTOR OF PARAMETERS THOSE
4.      C   PARAMETERS WHICH ARE FIXED IN INVERSION, OR REPLACES IN
5.      C   THE VECTOR FIXED PARAMETERS, DEPENDING ON THE VALUE OF IFU.
6.      C
7.      C   PARAMETER NN=40,MM=15
8.      C   COMMON/BLK7/PI(MM),PVAR(MM),IFIX
9.      C   DIMENSION PLOG(MM),PNEW(MM)
10.     C   IF(IFU.EQ.-1) GO TO 30
11.     C FIX PARAMETERS
12.     C   L=0
13.     C
14.     C   REORDER PARAMETERS SO UNDETERMINED ONES ARE FIRST.
15.     C
16.     C   DO 20 J=1,NP~
17.     C   IF(PVAR(J).GT.1.E-10) GO TO 15
18.     C   L=L+1
19.     C   GO TO 20
20.     15   K=J-L
21.     C   PLOG(K)=PLOG(J)
22.     20   CONTINUE
23.     C   IFU=0
24.     C
25.     C   NUMBER OF PARAMETERS TO BE DETERMINED IS CHANGED
26.     C
27.     C   NP=NP-IFIX
28.     C   RETURN
29.     C UNFIX PARAMETERS
30.     30   NP=NP+IFIX
31.     C   L=0
32.     C   DO 40 J=1,NP
33.     C   IF(PVAR(J).GT.1.E-10) GO TO 35
34.     C   PNEW(J)=PI(J)
35.     C   L=L+1
36.     C   GO TO 40
37.     35   K=J-L
38.     C   PNEW(J)=PLOG(K)
39.     40   CONTINUE
40.     C   IFU=0

```

F I X (1)

DATE 06/20/79

41.		DO 50 J=1,NP
42.	50	PLOG(J)=PNEW(J)
43.		RETURN
44.		END

```

1.      SUBROUTINE RDCOEF
2.      COMMON/BIKCF/AO(436),WO(436)
3.      C      FILTER ABSCISSA AND COEFF FOR EXP(X)*J0(EXP(X)). DELX=LN10/10
4.      DATA (AO(I),I=1,61)/
5.      *-6.8348046E 00,-6.6045461E 00,-6.3742876E 00,-6.1440291E 00,
6.      *-5.9137706E 00,-5.6835121E 00,-5.4532536E 00,-5.2229951E 00,
7.      *-4.9927366E 00,-4.7624781E 00,-4.5322196E 00,-4.3019611E 00,
8.      *-4.0717026E 00,-3.8414441E 00,-3.6111856E 00,-3.3809271E 00,
9.      *-3.1506686E 00,-2.9204101E 00,-2.6901516E 00,-2.4598931E 00,
10.     *-2.2296346E 00,-1.9993761E 00,-1.7691176E 00,-1.5388591E 00,
11.     *-1.3086006E 00,-1.0783421E 00,-8.4808358E-01,-6.1782508E-01,
12.     *-3.8756658E-01,-1.5730808E-01, 7.2950416E-02, 3.0320892E-01,
13.     * 5.3346742E-01, 7.6372592E-01, 9.9398442E-01, 1.2242429E 00,
14.     * 1.4545014E 00, 1.6847599E 00, 1.9150184E 00, 2.1452769E 00,
15.     * 2.3755354E 00, 2.6057939E 00, 2.8360524E 00, 3.0663109E 00,
16.     * 3.2965694E 00, 3.5268279E 00, 3.7570864E 00, 3.9873449E 00,
17.     * 4.2176034E 00, 4.4478619E 00, 4.6781204E 00, 4.9083789E 00,
18.     * 5.1386374E 00, 5.3688959E 00, 5.5991544E 00, 5.8294129E 00,
19.     * 6.0596714E 00, 6.2899299E 00, 6.5201884E 00, 6.7504469E 00,
20.     * 6.9807054E 00/
21.     DATA (WO(I),I=1,61)/
22.     * 7.3260937E-04, 5.6326423E-04, 1.3727237E-04, 7.5331222E-04,
23.     * 3.5918326E-04, 1.0500608E-03, 7.1530982E-04, 1.5160070E-03,
24.     * 1.2841617E-03, 2.2497985E-03, 2.1906186E-03, 3.4076782E-03,
25.     * 3.6321245E-03, 5.2376028E-03, 5.9212519E-03, 8.1315877E-03,
26.     * 9.5527062E-03, 1.2708615E-02, 1.5305589E-02, 1.9941086E-02,
27.     * 2.4396626E-02, 3.1333652E-02, 3.8683065E-02, 4.9127993E-02,
28.     * 6.0824806E-02, 7.6314344E-02, 9.3928346E-02, 1.1545027E-01,
29.     * 1.3868663E-01, 1.6248847E-01, 1.8114332E-01, 1.8424433E-01,
30.     * 1.5556741E-01, 6.8592481E-02, -8.8339029E-02, -2.8819226E-01,
31.     * -3.5565260E-01, -5.6288677E-02, 4.8186942E-01, -5.1516453E-02,
32.     * -2.6102989E-01, 2.1416490E-01, -9.4490687E-02, 2.6196370E-02,
33.     * -5.1097828E-04, -6.6032948E-03, 7.5193619E-03, -6.7854344E-03,
34.     * 5.8044372E-03, -4.9354894E-03, 4.2323106E-03, -3.6733648E-03,
35.     * 3.2266260E-03, -2.8649137E-03, 2.5677680E-03, -2.3202655E-03,
36.     * 2.1115187E-03, -1.9334662E-03, 1.7800248E-03, -1.6465436E-03,
37.     * 1.3468317E-03/
38.     RETURN
39.     END

```

```

1.      SUBRCUTINE YINVRT(F,Y,P)
2.      C
3.      C      MAIN DRIVER FOR THE INVERSION
4.      C
5.      PARAMETER NN=40,MM=15
6.      DIMENSION F(NN),Y(NN),P(MM),PL(MM),TL(MM),TH(MM)
7.      COMMON/BLK5/YC(NN),DY(NN),A(NN,MM),SNORM(NN),X(MM),SIGMA(MM)
8.      COMMON/BLK7/PI(MM),PVAR(MM),IFIX
9.      COMMON/BLK10/ITMAX,MODEL,N,NP,MQT,IPRT,IG,ILOG,IWTFAC,ICOEFS
10.     COMMON/BLK15/LAST
11.     COMMON/BLK20/SUMUP(20,35),ITER
12.     COMMON/BLKTK/IVES,RHOH(MM)
13.     CALL CDELTY(F,Y,P,YL)
14.     C
15.     C      THE MATRIX SUMUP STORES DATA THAT IS PRINTED LATER IN MAIN
16.     C
17.     SUMUP(1,1)=YL
18.     DO 1 J=1,NP
19.     J2=2*J+3
20.     1      SUMUP(1,J2)=10.**P(J)
21.     XCT=-2.
22.     XCM=4.
23.     LAST=0
24.     C
25.     C      I IS THE ITERATION COUNTER
26.     C
27.     DO 10 I=1,ITMAX
28.     WRITE(3,26) I
29.     26      FORMAT(1X,/,10(' '),/, 'ITERATION NUMBER',I4,/,10(' '))
30.     I1=I+1
31.     ITER=I1
32.     IF(MQT.EQ.1) XCM=XCM/2.
33.     IFLAG=0
34.     IF(MQT.EQ.0.OR.LAST.EQ.2) GO TO 2
35.     IPRTL=IPRT
36.     IPRT=3
37.     C
38.     C      GET MARQUARDT FACTOR
39.     C
40.     CALL RIDGE(XCT,XCM,F,Y,P,DIAGAD,YLSE,YL,IFLAG)

```

```

41.      IPRT=IPRTL
42.      2      IF(LAST.EQ.2)DIAGAD=0.0
43.      IF(LAST.GT.0.AND.IPRT.LT.2)IPRT=0
44.      C
45.      C      GET NEW PARAMETERS AND NEW REDUCED CHI-SQUARE
46.      C
47.      IF(IFLAG.EQ.0.OR.LAST.GT.0) CALL SUBVRT(F,Y,P,DIAGAD,YLSE)
48.      SUMUP(I1,1)=YLSE
49.      SUMUP(I1,2)=DIAGAD
50.      SUMUP(I1,3)=XCM
51.      SUMUP(I1,4)=XCT
52.      C
53.      C      IF LAST=2 OLD PARAMETERS ARE KEPT. IF OLD CHI-SQUARE IS
54.      C      BIGGER THAN NEW CHI-SQUARE THEN NEW PARAMETERS ARE USED.
55.      C
56.      IF(LAST.EQ.2) GO TO 14
57.      IF(YLSE.IT.YL) GO TO 25
58.      MQT=1
59.      14      DO 15 JJ=1,NP
60.      15      P(JJ)=PL(JJ)
61.      25      CONTINUE
62.      IF(IPRT.EQ.0.AND.IVES.GT.10) WRITE(3,7)
63.      7      FORMAT(1H ,//,2X,'LOG PARAM CHANGE',3X,'LOG PARAMETERS',4X,'LOG DE
64.      *VIATION',2X,'FRACTIONAL CHANGE',4X,'PARAMETERS',5X,'STD DEVIATION'
65.      *,4X,'PCT DEVIATION')
66.      DO 8 I=1,NP
67.      C
68.      C      RESTRAIN LOG(PARAMETERS) TO WITHIN +OR- 30
69.      C
70.      IF(P(L) .GT. 30. ) P(L)=30.
71.      IF(P(L) .LT. -30.) P(L)=-30.
72.      IF(X(L) .GT. 30.) X(L) = 30.
73.      IF(X(L).LT.-30.) X(L)=-30.
74.      PL(L)=P(L)
75.      C
76.      C      ANTIP=LINEAR PARAMETER, ANTIS=PARAMETER STANDARD DEVIATION,
77.      C      PCT=PERCENT DEVIATION.
78.      C
79.      ANTIP=EXP(2.30258*P(L))
80.      ANTIX=EXP(2.30258*X(L))-1.

```

```

81.      IF(SIGMA(L).GT.6.0)SIGMA(L)=6.0
82.      ANTIS=(EXP(2.30258*SIGMA(L))-1.)*ANTIP
83.      PCT=ANTIS*100./ANTIP
84.      L2=3+2*L
85.      L3=L2+1
86.      C
87.      C   STORE RESULTS AND CONTINUE WITH INVERSION
88.      C
89.      SUMUP(I1,L2)=ANTIP
90.      SUMUP(I1,L3)=PCT
91.      IF(IPRT.EQ.0.AND.IVES.GT.10) WRITE(3,9) X(L),P(I),SIGMA(L),ANTIP,A
92.      *NTIP,ANTIS,PCT
93.      CRUD=P(L)-SIGMA(L)
94.      TL(L)=EXP(2.30258*CRUD)
95.      CRUD=P(L)+SIGMA(L)
96.      TH(L)=EXP(2.30258*CRUD)
97.      RHOH(L) = ANTIP
98.      8   CONTINUE
99.      9   FORMAT(1H ,1PE14.2,1PE17.2,0PF16.2)
100.     IF(IPRT.EQ.0)WRITE(3,12)
101.     DO 16 L=1,NP
102.     NR=(NP+1)/2
103.     ANTIF=EXP(2.30258*P(L))
104.     IF(L.GT.NR)GO TO 27
105.     IF(IPRT.EQ.0)WRITE(3,13)L,ANTIP,TL(L),TH(L)
106.     GO TO 16
107.     27 JJ=L-NR
108.     IF(IPRT.EQ.0)WRITE(3,28)JJ,ANTIP,TL(L),TH(L)
109.     16 CONTINUE
110.     12 FORMAT(1H ,//,2X,'HIGH AND LOW PARAMETER VALUES CORRESPONDING',//,2
111.     *X,'TO ONE STANDARD DEVIATION IN LOG SPACE',//,2X,'PARAM. NAME ',3X
112.     *,'PARAMETER',3X,'(LOW VALUE,HIGH VALUE)')
113.     13 FORMAT(1H , 'RESIS(' ,I2,')' ,8X,E9.3,4X,'(' ,E9.3,' ,',E9.3,')')
114.     28 FORMAT(1H , 'THICK(' ,I2,')' ,8X,E9.3,4X,'(' ,E9.3,' ,',E9.3,')')
115.     IF(IPRT.EQ.0) WRITE(3,11)
116.     11  FORMAT(1H ,//,130('*'),/,130('*'))
117.     PC=ABS(YL/YLSE-1.)
118.     IF(LAST.EQ.2)SUMUP(I1,1)=SUMUP(I,1)
119.     IF(LAST.EQ.2)RETURN
120.     IF(LAST.EQ.1)LAST=2

```

Y I N V R T (10)

DATE 06/20/79

```
121.      IF(LAST.EQ.2)GO TO 10
122.      C
123.      C      CRITERION TO STOP ITERATIONS IS EITHER (1) ITERATION COUNTER
124.      C      APPROACHES ITMAX, OR (2) IF RED.CHI-SQ. OLD IS APPROX. = TO
125.      C      RED.CHI-SQ. NEW.
126.      C
127.      IF(PC.LT.1.E-02.AND.LAST.NE.2) LAST=1
128.      IF(I.EQ.(ITMAX-2))LAST=1
129.      IF(YLSE.LT.YL)YL=YLSE
130.      10      CONTINUE
131.      RETURN
132.      END
```



```
1.      SUPROUTINE CDELTY(F,Y,P,RCHISQ)
2.      C
3.      C      CALCULATES REDUCED CHI-SQUARE
4.      C
5.      PARAMETER NN=40,MM=15
6.      DIMENSION F(NN),Y(NN),P(MM)
7.      COMMON/BLK5/YC(NN),DY(NN),A(NN,MM),SNORM(NN),X(MM),SIGMA(MM)
8.      COMMON/BLK32/XMN(40),IMN
9.      COMMON/BLK10/ITMAX,MODEL,N,NP,MQT,IPRT,IG,ILOG,IWTFAC,ICOEFS
10.     YLSE=0.0
11.     DO 10 K=1,N
12.     IMN=K
13.     CALL CY(F(K),P,YC(K))
14.     DY(K)=Y(K)-YC(K)
15.     C
16.     C      IF THERE ARE WEIGHTING FACTORS FOR THE DATA THEN DY(K)=W*DY(K)
17.     C
18.     IF(SNORM(1).GT.1.E-10) DY(K)=DY(K)*SNORM(K)
19.     10 YLSE=YLSE+DY(K)*DY(K)
20.     RCHISQ=YLSE/FLOAT(N-NP)
21.     RETURN
22.     END
```

```

1.      SUBRCUTINE CY(FLOG,PLOG,YC)
2.      C
3.      C   CALCULATES APPARENT RESISTIVITY FOR CURRENT PARAMETER VALUES.
4.      C   FLOG=LOG(AB/2),PLOG=LOG(PARAMETER)
5.      C
6.      PARAMETER NN=40,MM=15
7.      DIMENSION P(MM),PLOG(MM),RESIS(MM),THICK(MM),R(2),V(2)
8.      COMMON/BLK7/PI(MM),PVAR(MM),IFIX
9.      COMMON/BLK10/ITMAX,MODEL,N,NP,MQT,IPRT,IG,ILOG,IWTFAC,ICOEFS
10.     COMMON/BLK32/XMN(40),IMN
11.     IFU=-1
12.     IF(IFIX.NE.0) CALL FIX(PLOG,NP,IFU)
13.     DO 5 J=1,NP
14.     5   P(J)=10.**PLOG(J)
15.     XL=10.**FLOG
16.     XR=XMN(IMN)/2.
17.     NR=2
18.     R(1)=XL-XR
19.     R(2)=XL+XR
20.     C
21.     C   G IS GEOMETRIC FACTOR
22.     C
23.     G=2./((XL-XR)-2./((XL+XR)
24.     DO 10 I=1,MODEL
25.     J=MODEL+I
26.     RESIS(I)=P(I)
27.     10  THICK(I)=P(J)
28.     CALL RESIST(MODEL,NR,RESIS,THICK,R,V,ICOEFS)
29.     DV=2.*(V(1)-V(2))
30.     YC=6.283185307*DV/G
31.     IF(ILOG.EQ.1)YC=ALOG10(YC)
32.     IF(IFIX.NE.0) CALL FIX(PLOG,NP,IFU)
33.     RETURN
34.     END

```

R E S I S T (4)

DATE 06/20/79

```

1.      SUBROUTINE RESIST(N,NR,RESIS,THICK,R,V,ICOEFS)
2.      C RESIST CALCULATES VOLTAGES ON A LAYERED EARTH ABOUT A POINT ELECTRODE.
3.      C USES A CONVOLUTION METHOD (EG KOEFORD GEOPHYS PROSP. V20P406). FILTER COEFF
4.      C FROM W. ANDERSON (U.S.G.S. DENVER). N= NO OF LAYERS. NR= DISTANCES AT WHICH
5.      C VOLTAGES ARE REQUIRED. RESIS CONTAINS RESISTIVITIES OF LAYERED STRUCTURE.
6.      C THICK CONTAINS LAYER THICKNESSES. R CONTAINS DISTANCES AT WHICH VOLTAGES ARE
7.      C REQUIRED. V CONTAINS VOLTAGES. PROGRAM TAKES ARBITRARY STRUCTURES AND NO OF
8.      C OBSERVATION POINTS. ***** NOTE RESISTIVITIES IN RESIS ARE DESTROYED. *****
9.      C ACCURACIES ARE PROBABLY OF THE ORDER OF +- 2 IN THE 4TH SIGNIFIGENT DIGIT OF V
10.     REAL V(1),RESIS(1),THICK(1),R(1)
11.     COMMON/PIKCF/AO(436),WO(436)
12.     NCFS = 61
13.     IF(ICOEFS.EQ.1) NCFS = 436
14.     P1=RESIS(1)
15.     NN=N-1
16.     DO30I=1,NN @NORMALISE RESISTIVITIES
17.     30 RESIS(I)=RESIS(I)/RESIS(I+1)
18.     DO100KR=1,NR @FIND APP RESIST FOR EACH SEPERATION
19.     PAA=0.0
20.     DD=ALOG(R(KR))
21.     DO2J=1,NCFS @ FIND 'REFLECTION COEFF' FOR EACH LAMBDA AND * WITH FILTER
22.     G=DE-AO(J) @FIND AFSCISSA VALUE
23.     G=EXP(-G)
24.     TK=1.0
25.     LL=N-1
26.     LI=N
27.     DO11I=1,LL
28.     LI=LI-1
29.     TR=RESIS(LI)
30.     TMU=(TR-TK)/(TR+TK)
31.     E=2.0*G*THICK(LI)
32.     TEX=EXP(-E)
33.     TMU=TMU*TEX
34.     1 TK=(1.0-TMU)/(1.0+TMU)
35.     2 PAA=PAA+TK*WO(J)
36.     PAA=PAA*P1
37.     100 V(KR)=PAA/(6.283185307*R(KR))
38.     RETURN
39.     END

```

R I D G E (1)

DATE 06/20/79

```

1.      SUBROUTINE RIDGE(XCT,XCM,F,YP,P,DIAGAD,YLSE,YL,IFLAG)
2.      C
3.      C      CALCULATES MARQUARDT FACTOR TO BE ADDED TO DIAGONAL=DIAGAD
4.      C
5.      PARAMETER NN=40,MM=15
6.      DIMENSION F(NN),YF(NN),P(MM),PL(MM),X(3),Y(3)
7.      XK=XCT-2.*XCM
8.      IFLAG=0
9.      DO 5 L=1,MM
10.     PL(L)=P(L)
11.     C
12.     C      CALCULATES REDUCED CHI-SQUARE FOR 3 VALUES OF DIAGAD
13.     C      DETERMINED BY THE QUANTITIES XCT AND XK.
14.     C
15.     DO 10 I=1,3
16.     XK=XK+XCM
17.     DIAGAD=10.**XK
18.     X(I)=XK
19.     CALL SUBVRT(F,YP,P,DIAGAD,Y(I))
20.     IF(Y(1).GT.YL) GO TO 6
21.     XCT=XK
22.     IF(XCM.LT.0.5) XCM=XCM*3.
23.     YLSE=Y(1)
24.     IFLAG=1
25.     C
26.     C      IF REDUCED CHI-SQUARE IS LESS THAN FOR PREVIOUS ITERATION RETURN
27.     C
28.     RETURN
29.     6      CONTINUE
30.     DO 8 L=1,MM
31.     P(L)=PL(L)
32.     10     CONTINUE
33.     IF(Y(2).GT.Y(1).OR.Y(2).GT.Y(3)) GO TO 12
34.     IFLAG=0
35.     C
36.     C      FIT PARABOLA TO REDUCED CHI-SQUARE VERSUS DIAGAD,AND PICK VALUE OF
37.     C      DIAGAD FOR WHICH CHI-SQUARE IS MINIMUM.
38.     C
39.     A1=Y(1)/((X(1)-X(2))*(X(1)-X(3)))
40.     A2=Y(2)/((X(2)-X(1))*(X(2)-X(3)))

```

R I D G E (1)

DATE 06/20/79

```
41.      A3=Y(3)/((X(3)-X(1))*(X(3)-X(2)))
42.      XCT=(A1*(X(2)+X(3))+A2*(X(1)+X(3))+A3*(X(1)+X(2)))/(2.*(A1+A2+A3))
43.      GO TO 13
44.      12      IF(Y(1).LE.Y(3))XCT=XCT-XCM
45.      IF(Y(1).GT.Y(3))XCT=XCT+XCM
46.      IF(XCM.IT.0.5)XCM=XCM*3.
47.      13      DIAGAD=10.**XCT
48.      RETURN
49.      END
```

```

1.      SUBROUTINE SUBVRT(F,Y,P,DIAGAD,YLSE)
2.      C
3.      C   CALCULATES NEW CHI-SQUARE, NEW THEORETICAL OBSERVABLES, AND
4.      C   NEW PARAMETERS.
5.      C
6.      PARAMETER NN=40,MM=15
7.      DIMENSION F(NN),Y(NN),P(MM),AY(NN),AYC(NN),AF(NN),ADY(NN)
8.      COMMON/BLK5/YC(NN),DY(NN),A(NN,MM),SNORM(NN),X(MM),SIGMA(MM)
9.      COMMON/BLK7/PI(MM),PVAR(MM),IFIX,PMIN(MM),PMAX(MM)
10.     COMMON/BLK10/ITMAX,MOIEL,N,NP,MQT,IPRT,IG,ILOG,IWTFAC,ICOEFS
11.     COMMON/BLK15/LAST
12.     COMMON/BLK32/XMN(40),IMN,XI(40),V(40)
13.     COMMON/BLKTK/IVES,REOH(MM)
14.     NO=1
15.     NF=N
16.     C
17.     C   CALCULATES REDUCED CHI-SQUARE FOR OLD PARAMETERS
18.     C
19.     CALL CDELTY(F,Y,P,YLSO)
20.     C
21.     C   CALCULATE SYSTEM MATRIX
22.     C
23.     CALL NUDERV(F,YC,P,A,NO,NF)
24.     C
25.     C   INVERT SYSTEM MATRIX AND GET PARAMETER CHANGE
26.     C
27.     CALL XLSC(N,NP,A,DY,X,SNORM,YLSO,SIGMA,DIAGAD,IPRT)
28.     IF(IFRT.EQ.0.AND.IVES.GT.10) WRITE(3,1)
29.     1   FORMAT(1H ,//,4X,'OLD PARAMS',3X,'NEW PARAMS')
30.     DO 2 L=1,NP
31.     PO=P(L)
32.     C
33.     C   CALCULATE NEW PARAMETERS
34.     C
35.     P(L)=P(L)+X(L)
36.     IF(IG.LT.2) GO TO 9
37.     IF(P(L).LT.PMIN(L)) P(L)=PMIN(L)
38.     IF(P(L).GT.PMAX(L)) P(L)=PMAX(L)
39.     9   CONTINUE
40.     PN=P(L)

```

```
41.      C
42.      C      PRINT PARAMETER CHANGE
43.      C
44.      IF(IPRT.EQ.0.AND.IVES.GT.10) WRITE(3,6) PO,PN
45.      2      CONTINUE
46.      IF(LAST.EQ.2) RETURN
47.      C
48.      C      GET REDUCED CHI-SQUARE FOR NEW PARAMETERS
49.      C
50.      CALL CDELTY(F,Y,P,YLSE)
51.      IF(IPRT.GT.0) GO TO 5
52.      IF(IYES.GT.10)WRITE(3,3)
53.      3      FORMAT(1H0,9X,'AB/2',7X,'DELRES',5X,'OBSRES',5X,'CALRES')
54.      DO 4 K=1,N
55.      AY(K)=10.**Y(K)
56.      AYC(K)=10.**YC(K)
57.      ADY(K)=AY(K)-AYC(K)
58.      AF(K)=10.**F(K)
59.      IF(IYES.GT.10)WRITE(3,4) K,AF(K),ADY(K),AY(K),AYC(K)
60.      4      FORMAT(1H ,I4,1P7E11.2)
61.      IH = NP - MODEL + 2
62.      CC      CALL LCGPLT(AF,AY,N,AF,AYC,N,XMN,IYES,RHOH(1),RHOH(IH),MODEL)
63.      CALL LGFIT2(AF,AY,N,AYC,IYES,RHOH(1),RHOH(IH),MODEL)
64.      5      CONTINUE
65.      IF(IYES.GT.10)GO TO 8
66.      RETURN
67.      8 IF(IPRT.EQ.0) WRITE(3,7) YLSO,YLSE
68.      6      FORMAT(1H ,1P2E13.5)
69.      7      FORMAT(1H0,'RED CHI SQ OLD PARMS=',1PE11.5,5X,'RED CHI SQ NEW PARM
70.      *S=',1PE11.5,/,130(' '))
71.      RETURN
72.      END
```

SANFERRG*SLUPE(1).NUDERV

```
1      SUPERROUTINE NUDERV(F,Y,P,A,NO,NF)
2      C
3      C      NUMERICAL DERIVATIVE USING LOG PARAMETERS TO GET SYSTEM MATRIX
4      C
5      C      PARAMETER NN=40,MM=15
6      C      DIMENSION F(NN),Y(NN),P(MM),P0(MM),A(NN,MM)
7      C      COMMON/BLK10/ITMAX,MODEL,N,NP,MQT,IPRT,IG,ILOG,IWTFAC,ICOEFS
8      C      COMMON/PLK32/XMN(40),IMN
9      C
10     C      XD USED FOR LOG DERIVS IS 1% OF THE PARAMETER
11     C
12     C      XI=.004
13     C      DO 1 K=1,NP
14     1      P0(K)=P(K)
15     C      DO 10 I=NO,NF
16     C      IMN=I
17     C
18     C      J IS INDEX ON PARAMETER
19     C
20     C      DO 10 J=1,NP
21     C      F(J)=P0(J)+XD
22     C
23     C      GET THEORETICAL VALUE OF PARAMETER PLUS DIFFERENCE
24     C
25     C      CALL CY(F(I),P,YPX)
26     C      F(J)=P0(J)
27     C
28     C      CALCULATE NUMERICAL DERIVATIVE
29     C
30     10    A(I,J)=(YPX-Y(I))/XD
31     C      RETURN
32     C      END
```

<*>


```

1.      SUBRCUTINE XLSQ(M,N,A,DELTAC,DELTAP,SNORM,XSQ,SIGMA,DIAGAD,IPRT)
2.      C
3.      C   COMPUTES PARAMETER COVARIANCE MATRIX,PARAMETER STANDARD DEVIATION,
4.      C   AND PARAMETER CORRELATION MATRIX.
5.      C
6.      PARAMETER NN=15,MM=40
7.      REAL A(MM,NN),ATA(NN,NN),INV(NN,NN),DELTAP(NN),
8.      *DELTAC(MM),W(NN),WORK(MM),SNORM(MM),SIGMA(MM),H(NN,MM),B(NN)
9.      COMMON/BLKTK/IVES,RHOH(MM)
10.     DATA R/1HR/,B/15*1HT/
11.     ZERO=0.0
12.     SUM=ZERO
13.     IF(SNORM(1).LT.1.E-10) GO TO 5
14.     DO 4 I=1,M
15.     DO 4 J=1,N           G WEIGHT SYSTEM MATRIX
16.     4 A(I,J)=A(I,J)*SNORM(I)
17.     COMPUTE DIAGONAL ELEMENTS OF ATA
18.     5 DO 7 J=1,N
19.     SUM=0.0
20.     DO 6 I=1,M
21.     6 SUM=SUM+A(I,J)*A(I,J)
22.     7 W(J)=1.0/(SQRT(SUM))
23.     COMPUTE WEIGHTED A MATRIX FOR PASS INTO SANDR
24.     DO 8 I=1,M
25.     DO 8 J=1,N
26.     8 A(I,J)=A(I,J)*W(J)
27.     CCMPUTE ATA WITH MARQUARDT WEIGHTS AND TAKE ADVANTAGE OF SYMMETRY
28.     N1=N-1
29.     DO 20 K=1,N1
30.     IST=K+1
31.     DO 15 J=IST,N
32.     SUM=0.0
33.     DO 11 I=1,M
34.     11 SUM=SUM+A(I,K)*A(I,J)
35.     ATA(J,K)=SUM
36.     15 ATA(K,J)=SUM
37.     20 CONTINUE
38.     DO 25 I=1,N
39.     25 ATA(I,I)=1.0+DIAGAD
40.     IF(IPRT.EQ.0) WRITE(3,26) DIAGAD

```

```

41.      26  FORMAT(1H ,//,1X,'DIAGAD=',1PE8.2)
42.      27  CONTINUE
43.      CALL MAINE(N,ATA,INV,DET)
44.      IF(IPRT.EQ.0) WRITE(3,504) DET
45.      COMPUTE PARAMETER CHANGE FROM GENERALIZED INVERSE
46.      CALL SANDR(INV,A,M,N,H,IPRT,IVES)
47.      DO 35 I=1,N
48.      SUM=0.0
49.      DO 30 K=1,M
50.      30  SUM=SUM+H(I,K)*DELTAC(K)
51.      35  DELTAP(I)=SUM*W(I)
52.      DO 52 I=1,N
53.      DO 51 J=1,I          @ PARAMETER COVARIANCE MATRIX
54.      51  INV(I,J)=INV(I,J)*W(I)*W(J)
55.      52  CONTINUE
56.      FAC=SQRT(XSQ)
57.      DO 54 I=1,N          @ PARAMETER STANDARD DEVIATION
58.      SUM=SQRT(INV(I,I))
59.      54  SIGMA(I)=SUM*FAC
60.      IF(IPRT.EQ.0) WRITE(3,508)
61.      IF(IPRT.NE.0) RETURN
62.      NR=(N+1)/2
63.      NT=N-NR
64.      DO 56 I=1,N          @ PARAMETER CORRELATION MATRIX
65.      DO 55 J=1,I
66.      55  WORK(J)=INV(I,J)/(SQRT(INV(I,I)*INV(J,J)))
67.      IF(I.LT.NR) GO TO 57
68.      P(I)=R
69.      WRITE(3,1000) I,(WORK(J),J=1,I)
70.      GO TC 56
71.      57  IT=I-NR
72.      WRITE(3,1001) IT,(WORK(J),J=1,I)
73.      56  CONTINUE
74.      WRITE(3,1002)(P(J),J,J=1,NR),(B(K+NR),K,K=1,NT)
75.      504 FORMAT(20X,//,' DETERMINANT OF SYSTEM MATRIX=',1PE8.2,/)
76.      508 FORMAT(1H , 'PARAMETER CORRELATION MATRIX')
77.      1000 FORMAT(1H , 'RESIS(',I2,')',15F6.2)
78.      1001 FORMAT(1H , 'THICK(',I2,')',15F6.2)
79.      1002 FORMAT(1H ,9X,15(1X,A1, '(' ,I2, ') '))
80.      RETURN

```

X L S Q (8)

DATE 06/20/79

E1.

END

1.		SUPROUTINE MAINE(N,PP,AA,DET)	
2.	C		
3.	C	INVERTS (ATA+KI) SYMMETRIC MATRIX	
4.	C	A IS THE INVERSE OF B	
5.	C		
6.		PARAMETER NN=40,MM=15	
7.		REAL*8 A(MM,MM),B(MM,MM),R(MM),C(MM),DEL,DETERM	
8.		REAL AA(MM,MM),BB(MM,MM)	
9.		DETERM=1.	63.
10.		DO 10 I=1,N	64.
11.		DO 10 J=1,N	65.
12.		B(I,J)=DBLE(BB(I,J))	
13.	10	A(I,J)=0.D0	
14.		DO 40 L=1,N	67.
15.		DEL=B(L,L)	68.
16.		DO 30 I=1,L	69.
17.		C(I)=0.	70.
18.		R(I)=0.	71.
19.		DO 20 J=1,L	72.
20.		C(I)=C(I)+A(I,J)*B(J,L)	73.
21.	20	R(I)=R(I)+B(L,J)*A(J,I)	74.
22.	30	DEL=DEL-B(L,I)*C(I)	75.
23.		DETERM=DETERM*DEL	76.
24.		C(L)=-1.	77.
25.		R(L)=-1.	78.
26.		DO 40 I=1,L	79.
27.		C(I)=C(I)/DEL	80.
28.		DO 40 J=1,L	81.
29.	40	A(I,J)=A(I,J)+C(I)*R(J)	82.
30.		DET=SNGL(DETERM)	
31.		DO 45 I=1,N	
32.		DO 45 J=1,N	
33.	45	AA(I,J)=SNGL(A(I,J))	
34.		RETURN	83.
35.		END	84.

C RETURN SINGLE PRECISION

```
1.      SUBRCUTINE SANDR(INV,A,M,N,H,IPRT,IVES)
2.      COMPUTES THE GENERALIZED INVERSE, RESOLUTION MATRIX
3.      C      AND INFORMATION DENSITY MATRIX
4.      C      THE LAST TWO ARE NOT PASSED BACK
5.      PARAMETER NN=15,MM=40
6.      REAL A(MM,NN),INV(NN,NN),H(NN,MM),DUM(MM)
7.      COMPUTE GENERALIZED INVERSE (ATA-1)(AT)=H
8.      COMMON/BLK15/LAST,IPRS
9.      DO 5 I=1,N
10.     DO 5 J=1,M
11.     SUM=0.0
12.     DO 4 K=1,N
13.     4 SUM=SUM+INV(I,K)*A(J,K)
14.     5 H(I,J)=SUM
15.     IF(LAST.NE.2.OR.IPRS.NE.0) RETURN
16.     IF(IVES.IE.10)RETURN
17.     COMPUTE THE RESOLUTION MATRIX R=HA
18.     WRITE(3,100)
19.     DO 21 I=1,N
20.     DO 20 J=1,I
21.     SUM=0.0
22.     DO 19 K=1,M
23.     19 SUM=SUM+H(I,K)*A(K,J)
24.     20 DUM(J)=SUM
25.     WRITE(3,200),(DUM(J),J=1,I)
26.     21 CONTINUE
27.     COMPUTE INFORMATION DENSITY MATRIX S=AH
28.     WRITE(3,101)
29.     DO 31 I=1,M
30.     DO 30 K=1,I
31.     SUM=0.0
32.     DO 29 J=1,N
33.     29 SUM=SUM+A(I,J)*H(J,K)
34.     30 DUM(K)=SUM
35.     WRITE(3,300) (DUM(K),K=1,I)
36.     31 CONTINUE
37.     WRITE(3,99)
38.     99 FORMAT(//)
39.     100 FORMAT(1H , 'RESOLUTION MATRIX')
40.     101 FORMAT(//,1X, 'INFORMATION DENSITY MATRIX')
```

S A N D R (3)

DATE 06/20/79

```
41.      200  FORMAT(1H ,20F6.3)
42.      300  FORMAT(1H ,26F5.2)
43.      RETURN
44.      END
```

```

1.      SUBROUTINE LGPLT2(X1,Y1,N1,Y2,IT,RHO,H,NR)
2.      C
3.      C LGPLT2 DOES THE PLOTTING ON THE TEKTRONIX 4013
4.      C THE PLCT CONTAINS THE RESULTS OF THE THIRD-FROM-LAST ITERATION
5.      C X1=AB/2,Y1=RHOA,N1=N,Y2=INVERTED RHOA,IT=IVES,NR=MODEL
6.      C
7.      DIMENSION X1(N1),Y1(N1),Y2(N1)
8.      DIMENSION AX1(27),AY1(27),DUMX(27),DUMY(27)
9.      DIMENSION RHO(NR),H(1)
10.     DIMENSION DY(27),A(27),B(26,3),WK(225)
11.     C WORKS FOR N1 LESS THAN OR EQUAL TO 27
12.     C AND X(I) = X(I+1) PUT NOT X(I)=X(I+1)=X(I+2)
13.     DATA IOFF,ION/017,016/
14.     DATA S,DY,IAE/0.0,27*1.0,26/
15.     CALL TOUTFT(ION)
16.     CALL BELL
17.     CALL ERASE
18.     CALL VWINDO(0.,3.,0.,3.)
19.     CALL SWINDO(334,614,77,626)
20.     C DRAW BOX AND CYCLE LINES
21.     CALL MCVEA(0.,0.)
22.     CALL DRAWA(0.,3.)
23.     CALL DRAWA(3.,3.)
24.     CALL DRAWA(3.,0.)
25.     CALL DRAWA(0.,0.)
26.     CALL MOVEA(1.,0.)
27.     CALL DRAWA(1.,3.)
28.     CALL MOVEA(2.,0.)
29.     CALL DRAWA(2.,3.)
30.     CALL MOVEA(0.,1.)
31.     CALL DRAWA(3.,1.)
32.     CALL MOVEA(0.,2.)
33.     CALL DRAWA(3.,2.)
34.     C LABEL AXIS AND PRINT TITLE
35.     CALL MOVEA(0.,3.)
36.     CALL TOUTFT(IOFF)
37.     CALL CHSYNC
38.     WRITE(6,200) IT
39.     200 FORMAT(/17X,'10000',/2X,'VES - ',I3,////////,
40.     # 18X,'1000',//////////,19X,'100',//////////,

```

```

41.      # 20X,'10',/,24X,'1.0',10X,'10.0',9X,
42.      # '100.0',9X,'1000.0')
43.      CALL TCUTPT(ION)
44.      C   TAKE LOG OF FUNCTION 1 AND POINT PLOT
45.      DO 10 I=1,N1
46.      AX1(I) = ALOG10(X1(I))
47.      AY1(I) = ALOG10(Y1(I)) - 1.
48.      10 CONTINUE
49.      DO 20 I=1,N1
50.      CALL DRAWX(AX1(I),AY1(I))
51.      20 CONTINUE
52.      C   NOW DO FUNCTION 2
53.      I = 0
54.      N = 0
55.      4 I=I+1
56.      N=N+1
57.      IF(I.EQ.N1) GO TO 3
58.      IF(X1(I).EQ.X1(I+1)) GO TO 6
59.      3 DUMX(N) = X1(I)
60.      DUMY(N) = Y2(I)
61.      GO TO 5
62.      6 DUMX(N) = X1(I)
63.      DUMY(N) = (Y2(I) + Y2(I+1))/2.
64.      I = I+1
65.      5 IF(I.LT.N1) GO TO 4
66.      CONTINUE
67.      DO 8 I=1,N
68.      DUMX(I) = ALOG10(DUMX(I))
69.      DUMY(I) = ALOG10(DUMY(I)) - 1.
70.      8 CONTINUE
71.      IF(N.LT.4) GO TO 30
72.      C
73.      C   IF THE APPREVIATED OUTPUT IS DESIRED (IT<10) THEN THESE
74.      C   NEXT WRITE STATEMENTS ARE SKIPPED.
75.      C
76.      IF(IT.LE.10)GO TO 403
77.      WRITE(3,400)
78.      WRITE(3,401) N1,N
79.      WRITE(3,402) (AX1(KK),AY1(KK),DUMX(KK),DUMY(KK),KK=1,N1)
80.      402 FORMAT(/1X,'***** OUTPUT FROM PLOT SUBROUTINE *****')

```



```

81.      401 FORMAT(3X,'N1=',I4,5X,'N=',I4,/,4X,'X1',8X,'Y1',8X,'X1',8X,'Y2')
82.      402 FORMAT(4F10.2)
83.      C
84.      403 CALL ICSSCU(DUMX,DUMY,DY,N,S,A,B,IAP,WK,IER)
85.      C
86.      HSTEP = 0.05
87.      CALL MOVEA(DUMX(1),DUMY(1))
88.      DO 9 I=1,N
89.      XI = DUMX(I)
90.      X = XI
91.      16 X = X + HSTEP
92.      XBAR = X - XI
93.      SY=((B(I,3)*XBAR + B(I,2))*XBAR + B(I,1))*XBAR + A(I)
94.      CALL DRAWA(X,SY)
95.      IF(X.LT.DUMX(I+1)) GO TO 16
96.      9 CONTINUE
97.      GO TO 33
98.      30 CALL MOVEA(DUMX(1),DUMY(1))
99.      IF(N.LT.2) GOTO 33
100.     DO 31 I=2,N
101.     CALL DRAWA(DUMX(I),DUMY(I))
102.     31 CONTINUE
103.     33 CONTINUE
104.     C      FLCT FINISHED
105.     CALL MOVEA(0.,2.)
106.     CALL BELI
107.     CALL TOUTFT(IOFF)
108.     CALL CHSYNC
109.     WRITE(6,201)
110.     NRH = NR - 1
111.     DO 15 I=1,NRH
112.     15 WRITE(6,202) RHO(I),H(I)
113.     WRITE(6,202) RHO(NRH)
114.     201 FORMAT(1X,' RHO      H',/)
115.     202 FORMAT(1X,F7.2,1X,F6.1)
116.     C
117.     C      ENTERING AN INTEGER MAKES PROGRAM CONTINUE
118.     C
119.     READ(5,100) ICONT
120.     100 FORMAT(I5)

```

L G P L T 3 (4)

DATE 06/20/79

121.

122.

RETURN

END

D R A W X

DATE 06/20/79

```
1.      SUBROUTINE DRAWX(X,Y)
2.      CALL MOVEA(X,Y)
3.      SX = X - 0.0063
4.      SY = Y - 0.0063
5.      CALL MOVEA(SX,SY)
6.      CALL DRAWR(0.0125,0.0125)
7.      SY = Y + 0.0063
8.      CALL MCVEA(SX,SY)
9.      CALL DRAWR(0.0125,-0.0125)
10.     RETURN
11.     END
```

ICSSCU

Subroutine in the IMSL library from

IMSL
Sixth Floor, GNB Building
7500 Bellaire Boulevard
Houston, Texas U.S.A. 77036

ICSSCU places a smooth cubic spline along a given set of data points.

SUBROUTINE ICSSCU(X,F,DF,NX,SM,Y,C,IC,WK,IER)

where

- NX = number of elements in X,F,DF,Y. NX must be .GE. 2.
- X = vector of length NX containing the abscissae of the NX data points (X(I),F(I)). X must be ordered so that X(I) .LT. X(I+1).
- F = vector of length NX containing the ordinates (or function values) of the NX data points
- DF = vector of length NX. DF(I) is the relative weight of data point i.
- SM = a non-negative number which controls the extent of smoothing. The spline function S is determined such that the sum from 1 to NX of $((S(X(I))-F(I))/DF(I))^{**2}$.LE. SM, where equality holds unless S describes a straight line.
- Y,C = spline coefficients (output). Y is a vector of length NX. C is an NX-1 by 3 matrix. The value of the spline approximation at T is $S(T) = ((C(I,3)*D+C(I,2))*D+C(I,1))*D+Y(I)$ where X(I) .LE. T .LT. X(I+1) and D = T-X(I).
- IC = row dimension of matrix C in the calling program. IC must be .GE. NX-1
- WK = work area vector of length greater than or equal to 7*NX+14.
- IER = error parameter.
 - IER = 129, IC is less than NX-1
 - IER = 130, NX is less than 2
 - IER = 131, input abscissae are not ordered

reference for spline algorithm: Reinsch, Christian H.,
"Smoothing by Spline Functions", Numerische Mathematik,
10(3)1967, 177-183

V. REFERENCES

- Inman, J.R., 1975, Resistivity inversion with ridge regression: Geophysics, v. 40, p. 798-817.
- Rijo, L., Pelton, W.H., Feitosa, E.C., and Ward, S. H., 1977, Interpretation of apparent resistivity data from Apodi Valley, Rio Grande Do Norte, Brazil: Geophysics, v. 42, p. 811-822.
- Ward, S. H., 1967, Mining Geophysics Volume II: Published by the Society of Exploration Geophysicists, P.O. Box 3098, Tulsa, Oklahoma, p. 87-94.

VI. EXAMPLE

A. Input Data Cards

END DATA.

B. Regular Output

GLATA, L 3.
DATA UBR1 S73MOD 07/01/79 19:16:45 (0)

		AB/2	MN	I	V	OPSRES	WEIGHT
1.							
2.							
3.	1	1.000	1.000	.050	2.122	100.000	.000
4.	2	1.470	1.000	.050	.836	100.340	.000
5.	3	2.150	1.000	.050	.362	99.566	.000
6.	4	3.160	1.000	.050	.163	99.843	.000
7.	5	4.640	1.000	.050	.075	99.877	.000
8.	6	6.810	1.000	.050	.034	99.727	.000
9.	7	10.000	1.000	.050	.016	99.653	.000
10.	8	14.600	1.000	.100	.015	99.745	.000
11.	9	21.540	1.000	.200	.014	99.792	.000
12.	10	31.620	1.000	.400	.013	99.546	.000
13.	11	46.420	1.000	.800	.012	98.486	.000
14.	12	68.130	1.000	1.600	.010	95.600	.000
15.	13	100.000	2.000	3.200	.018	88.545	.000
16.	14	146.800	4.000	3.200	.014	73.718	.000
17.	15	215.400	16.000	3.200	.019	54.130	.000
18.	16	316.200	32.000	3.200	.014	41.610	.000
19.	17	464.000	64.000	3.200	.013	44.209	.000
20.	18	681.000	128.000	3.200	.017	60.217	.000
21.	19	1000.000	128.000	3.200	.011	85.704	.000
22.	20	1468.000	256.000	3.200	.015	120.563	.000
23.	21	2154.000	512.000	3.200	.020	171.822	.000
24.							
25.							
26.							
27.							

INITIAL GUESS

	P	PLOG	PVAR	PMIN	PMAX
28.					
29.					
30.	9.000+01	1.954+00	0.000	0.000	0.000
31.	9.000+00	9.542-01	0.000	0.000	0.000
32.	9.000+02	2.954+00	0.000	0.000	0.000
33.	9.000+01	1.954+00	0.000	0.000	0.000
34.	9.000+01	1.954+00	0.000	0.000	0.000
35.					

ITERATION NUMBER 1

ITERATION NUMBER 2

ITERATION NUMBER 3

ITERATION NUMBER 4

DIAGAD- 3.04-06

56. DETERMINANT OF SYSTEM MATRIX= 6.00-05

57.
58.
59. PARAMETER CORRELATION MATRIX

60. RESIS(1) 1.00
61. RESIS(2) .29 1.00
62. RESIS(3) .15 .47 1.00
63. THICK(1) -.34 -.98 -.38 1.00
64. THICK(2) .29 1.00 .49 -.98 1.00
65. R(1) R(2) R(3) T(1) T(2)
66.

67.
68. OLD PARAMS NEW PARAMS
69. 2.00090+00 2.00089+00
70. 1.15833+00 1.15555+00
71. 3.02088+00 3.02036+00
72. 1.99280+00 1.99317+00
73. 2.14458+00 2.14170+00

	AF/2	DELRES	OPSRRES	CALRES
74. 0				
75. 1	1.00+00	-1.98-01	1.00+02	1.00+02
76. 2	1.47+00	1.81-01	1.00+02	1.00+02
77. 3	2.15+00	-5.48-01	9.96+01	1.00+02
78. 4	3.16+00	-2.20-01	9.98+01	1.00+02
79. 5	4.64+00	-1.29-01	9.99+01	1.00+02
80. 6	6.81+00	-2.21-01	9.97+01	9.99+01
81. 7	1.00+01	-2.35-01	9.97+01	9.99+01
82. 8	1.47+01	-7.05-02	9.97+01	9.98+01
83. 9	2.15+01	1.14-01	9.98+01	9.97+01
84. 10	3.16+01	2.33-01	9.95+01	9.93+01
85. 11	4.64+01	2.40-01	9.85+01	9.82+01
86. 12	6.81+01	3.35-01	9.56+01	9.53+01
87. 13	1.00+02	6.28-01	8.85+01	8.79+01
88. 14	1.47+02	1.81-01	7.37+01	7.35+01
89. 15	2.15+02	-3.75-01	5.41+01	5.45+01
90. 16	3.16+02	5.87-02	4.16+01	4.16+01
91. 17	4.64+02	2.17-01	4.42+01	4.40+01
92. 18	6.81+02	1.98-01	6.02+01	6.00+01
93. 19	1.00+03	-4.05-01	8.57+01	8.61+01
94. 20	1.47+03	-1.44+00	1.21+02	1.22+02
95. 21	2.15+03	1.58+00	1.72+02	1.70+02

96.
97. ***** OUTPUT FROM PLOT SUBROUTINE *****

	N1= 21	N= 21		
	X1	Y1	X1	Y2
100.	.00	1.00	.00	1.00
101.	.17	1.00	.17	1.00
102.	.33	1.00	.33	1.00
103.	.50	1.00	.50	1.00
104.	.67	1.00	.67	1.00
105.	.83	1.00	.83	1.00
106.	1.00	1.00	1.00	1.00
107.	1.17	1.00	1.17	1.00
108.	1.33	1.00	1.33	1.00
109.	1.50	1.00	1.50	1.00
110.	1.67	.99	1.67	.99
111.	1.83	.98	1.83	.98
112.	2.00	.95	2.00	.94

113.	2.17	.87	2.17	.87
114.	2.33	.73	2.33	.74
115.	2.50	.62	2.50	.62
116.	2.67	.65	2.67	.64
117.	2.83	.78	2.83	.78
118.	3.00	.93	3.00	.94
119.	3.17	1.08	3.17	1.09
120.	3.33	1.24	3.33	1.23

121. ORED CHI SQ OLD PARMS= 5.53953-06

RED CHI SQ NEW PARMS= 5.54045-06

	LOG PARAM CHANGE	LOG PARAMETERS	LOG DEVIATION	FRACTIONAL CHANGE	PARAMETERS	STD DEVIATION	PCT DEVIATION
124.							
125.							
126.							
127.	-6.24-06	2.00+00	6.96-04	-1.44-05	1.00+02	1.61-01	.16
128.	-2.78-03	1.16+00	6.82-02	-6.38-03	1.44+01	2.45+00	17.00
129.	-5.25-04	3.02+00	2.89-02	-1.21-03	1.05+03	7.21+01	6.87
130.	3.65-04	1.99+00	9.00-03	8.41-04	9.84+01	2.06+00	2.09
131.	-2.88-03	2.14+00	7.09-02	-6.60-03	1.39+02	2.47+01	17.74
132.							
133.							

134. HIGH AND LOW PARAMETER VALUES CORRESPONDING
135. TO ONE STANDARD DEVIATION IN LOG SPACE

PARAM. NAME	PARAMETER	(LOW VALUE, HIGH VALUE)
136.		
137.	RESIS(1)	.100+03 (.100+03, .100+03)
138.	RESIS(2)	.144+02 (.123+02, .168+02)
139.	RESIS(3)	.105+04 (.682+03, .112+04)
140.	THICK(1)	.984+02 (.963+02, .100+03)
141.	THICK(2)	.139+03 (.118+03, .164+03)
142.		

146. *****
147. *****
148. ITERATION NUMBER 5
149. *****

150.
151.
152. DIACAD= 0.00
153.

154.
155. DETERMINANT OF SYSTEM MATRIX= 5.88-05
156.

157. RESOLUTION MATRIX

158.	1.000
159.	1.000
160.	-.000 1.000
161.	.000 -.000 1.000
162.	.000 .000 .000 1.000
163.	-.000 -.000 -.000 -.000 1.000
164.	
165.	

166. INFORMATION DENSITY MATRIX


```

224.
225.
226.
*****
227.
*****
228.
229. 1
230. ITER RCSQ DIAGAD XCM XCT
231. 0 3.5009-03 0.0000 0.0000 0.0000
232. 1 7.7395-06 1.0000-04 2.0000+00 -4.0000+00
233. 2 6.1832-06 1.0000-05 1.0000+00 -5.0000+00
234. 3 5.5395-06 3.1623-06 5.0000-01 -5.5000+00
235. 4 5.5404-06 3.0425-06 2.5000-01 -5.5168+00
236. 5 5.5404-06 0.0000 1.2500-01 -5.5168+00
237. ITER RCSQ PCT PCT PCT PCT
PCT 0 3.5009-03 9.0000+01 .0 9.0000+00 .0 9.0000+02 .0 9.0000+01 .0 9.0000+01
.0
239. 1 7.7395-06 1.0006+02 4.0 1.1439+01 8273.8 9.2864+02 262.7 1.0038+02 36.8 1.0862+02 880
4.0
240. 2 6.1832-06 1.0019+02 .2 1.4247+01 44.0 1.0330+03 7.1 9.8898+01 3.1 1.3773+02 4
5.1
241. 3 5.5395-06 1.0021+02 .2 1.4399+01 19.1 1.0492+03 7.2 9.8356+01 2.3 1.3950+02 1
9.9
242. 4 5.5404-06 1.0021+02 .2 1.4399+01 17.0 1.0492+03 6.9 9.8356+01 2.1 1.3950+02 1
7.7
243. 5 5.5404-06 1.0021+02 .2 1.4399+01 17.2 1.0492+03 6.9 9.8356+01 2.1 1.3950+02 1
7.9
244.
245. ITER RCSQ PCT PCT PCT PCT
PCT 0 3.5009-03 0.0000 .0 0.0000 .0 0.0000 .0 0.0000 .0 0.0000
.0
247. 1 7.7395-06 0.0000 .0 0.0000 .0 0.0000 .0 0.0000 .0 0.0000
.0
248. 2 6.1832-06 0.0000 .0 0.0000 .0 0.0000 .0 0.0000 .0 0.0000
.0
249. 3 5.5395-06 0.0000 .0 0.0000 .0 0.0000 .0 0.0000 .0 0.0000
.0
250. 4 5.5404-06 0.0000 .0 0.0000 .0 0.0000 .0 0.0000 .0 0.0000
.0
251. 5 5.5404-06 0.0000 .0 0.0000 .0 0.0000 .0 0.0000 .0 0.0000
.0
END DATA.

```

C. Tektronix Output

UES - 11

10000

RHO

H

1000

100.21 98.4

14.40 139.5

1049.24

100

10

1.0

10.0

100.0

1000.0

D. Abbreviated Output

GEATA, L 3.
DATA UER1 S73MOD 07/01/79 19:22:28 (2)

		AB/2	MN	I	V	OPSRES	WEIGHT
1.	0						
2.							
3.	1	1.000	1.000	.050	2.122	100.000	.000
4.	2	1.470	1.000	.050	.836	100.340	.000
5.	3	2.150	1.000	.050	.362	99.566	.000
6.	4	3.160	1.000	.050	.163	99.843	.000
7.	5	4.640	1.000	.050	.075	99.877	.000
8.	6	6.810	1.000	.050	.034	99.727	.000
9.	7	10.000	1.000	.050	.016	99.653	.000
10.	8	14.680	1.000	.100	.015	99.745	.000
11.	9	21.540	1.000	.200	.014	99.792	.000
12.	10	31.620	1.000	.400	.013	99.546	.000
13.	11	46.420	1.000	.800	.012	98.486	.000
14.	12	68.130	1.000	1.600	.010	95.600	.000
15.	13	100.000	2.000	3.200	.018	88.545	.000
16.	14	146.800	4.000	3.200	.014	73.718	.000
17.	15	215.400	16.000	3.200	.019	54.130	.000
18.	16	316.200	32.000	3.200	.014	41.610	.000
19.	17	464.000	64.000	3.200	.013	44.209	.000
20.	18	681.000	128.000	3.200	.017	60.217	.000
21.	19	1000.000	128.000	3.200	.011	85.704	.000
22.	20	1468.000	256.000	3.200	.015	120.563	.000
23.	21	2154.000	512.000	3.200	.020	171.822	.000
24.							
25.							
26.							
27.							
28.							

INITIAL GUESS

	P	PLOG	PVAR	PMIN	PMAX
29.					
30.	9.000+01	1.954+00	0.000	0.000	0.000
31.	9.000+00	9.542-01	0.000	0.000	0.000
32.	9.000+02	2.954+00	0.000	0.000	0.000
33.	9.000+01	1.954+00	0.000	0.000	0.000
34.	9.000+01	1.954+00	0.000	0.000	0.000
35.					

ITERATION NUMBER 1

ITERATION NUMBER 2

ITERATION NUMBER 3

ITERATION NUMBER 4

DIAGAD= 3.04-06

56. DETERMINANT OF SYSTEM MATRIX= 6.00-05

57.
58.
59. PARAMETER CORRELATION MATRIX

60. RESIS(1) 1.00
61. RESIS(2) .29 1.00
62. RESIS(3) .15 .47 1.00
63. THICK(1) -.34 -.98 -.38 1.00
64. THICK(2) .29 1.00 .49 -.98 1.00
65. R(1) R(2) R(3) T(1) T(2)
66.

67.
68. HIGH AND LOW PARAMETER VALUES CORRESPONDING
69. TO ONE STANDARD DEVIATION IN LOG SPACE

70. PARAM. NAME PARAMETER (LOW VALUE,HIGH VALUE)
71. RESIS(1) .100+03 (.100+03, .100+03)
72. RESIS(2) .144+02 (.123+02, .168+02)
73. RESIS(3) .105+04 (.982+03, .112+04)
74. THICK(1) .984+02 (.963+02, .100+03)
75. THICK(2) .139+03 (.118+03, .164+03)
76.
77.

78. *****

79. *****

80.

81. *****

82. ITERATION NUMBER 5

83. *****

84. *****

85.

86. DIAGAD= 0.00

87.

88.

89. DETERMINANT OF SYSTEM MATRIX= 5.00-05

90.

91.

92. PARAMETER CORRELATION MATRIX

93. RESIS(1) 1.00
94. RESIS(2) .29 1.00
95. RESIS(3) .15 .47 1.00
96. THICK(1) -.35 -.98 -.39 1.00
97. THICK(2) .29 1.00 .50 -.98 1.00
98. R(1) R(2) R(3) T(1) T(2)
99.

100.

101. HIGH AND LOW PARAMETER VALUES CORRESPONDING
102. TO ONE STANDARD DEVIATION IN LOG SPACE

103. PARAM. NAME PARAMETER (LOW VALUE,HIGH VALUE)
104. RESIS(1) .100+03 (.100+03, .100+03)
105. RESIS(2) .144+02 (.123+02, .169+02)
106. RESIS(3) .105+04 (.982+03, .112+04)
107. THICK(1) .984+02 (.963+02, .100+03)
108. THICK(2) .139+03 (.118+03, .165+03)
109.

110.

111.

112.

113.	0 ITER	RCSQ	DIAGAD
114.	0	3.5009-03	0.0000
115.	1	7.7395-06	1.0000-04
116.	2	6.1832-06	1.0000-05
117.	3	5.5395-06	3.1623-06
118.	4	5.5404-06	3.0425-06
119.	5	5.5404-06	0.0000

. END DATA.