2
MASTER

GRAV2D: AN INTERACTIVE 2½ DIMENSIONAL GRAVITY MODELING PROGRAM
(USER'S GUIDE AND DOCUMENTATION FOR REV.1)

Carleen Nutter

November, 1980

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

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

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

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

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

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

**GRAV2D: An Interactive 2 1/2 Dimensional
Gravity Modeling Program
(User's Guide and Documentation for Rev. 1)**

by
Carleen Nutter

DISCLAIMER
This book 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.

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

November, 1980

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

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

NOTICE

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

NOTICE

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

Table of Contents

ABSTRACT.....	1
INTRODUCTION.....	2
USER'S GUIDE.....	2
MASTER.....	2
INPUT.....	4
EDIT.....	9
LIST.....	10
EXECUTE.....	11
FILE MAINTENANCE.....	13
PLOT.....	13
DIRECT ACCESS FILE STRUCTURE.....	14
SYSTEM IMPLEMENTATION.....	14
ACKNOWLEDGMENTS.....	17
REFERENCES.....	18
APPENDIX I - PRIME FORTRAN EXTENSIONS.....	19
CODING statements.....	20
Formatted DECODE statement.....	20
List-directed DECODE statement.....	20
ENCODE statement.....	21
List Directed READ statement.....	21
Direct access READ and WRITE statements.....	22
Direct-access READ statements.....	23
Direct-access WRITE statements.....	24
NOBIG / BIG.....	24
PARAMETER statement.....	25
\$INSERT statement.....	25
APPENDIX II - PRIME SYSTEM ROUTINES.....	26
ATCH\$\$.....	27
ATTDEV.....	29
CLOS\$A.....	30
DELE\$A.....	30
ERRPR\$.....	30
EXIT.....	32
EXST\$A.....	32
OPEN\$A.....	32
SLEEP\$.....	33
SPOOL\$.....	34
SRCH\$\$.....	35

TIMDAT.....	37
TNOU.....	39
TNOUA.....	39
UNIT#A.....	39
APPENDIX III - DESCRIPTION OF NONLINEAR OPTIMIZATION METHOD.....	40
APPENDIX IV - SAMPLE TERMINAL SESSION #1.....	52
APPENDIX V - SAMPLE TERMINAL SESSION #2.....	79

LIST OF FIGURES

1. Initial guess model for terminal session #1.
(Plot generated by GRAV2D.) 72
2. Gravity model after search for terminal session #1.
(Plot generated by GRAV2D.) 74
3. Gravity model after inversion for terminal session #1.
(Plot generated by GRAV2D.) 76
4. Comparison of the three gravity models with
the theoretical model from terminal session
#1. (Plot computer-generated but not by
GRAV2D.) 77
5. Gravity model for terminal session #2. (Plot
generated by GRAV2D.) 86

ABSTRACT

GRAV2D is an interactive computer program used for modeling 2 1/2 dimensional gravity data. A forward algorithm is used to give the theoretical attraction of gravity intensity at a station due to a perturbing body given by the initial model. The resultant model can then be adjusted for a better fit by a combination of manual adjustment, one-dimensional automatic search, and Marquardt inversion.

GRAV2D has an interactive data management system for data manipulation and display built around subroutines to do a forward problem, a one-dimensional direct search and an inversion. These subroutines were originally separate batch-mode programs developed by John H. Snow at the Department of Geology and Geophysics, University of Utah.

This report is a user's guide and documentation for GRAV2D.

INTRODUCTION

GRAV2D is an interactive program for 2 1/2 D gravity modeling and is used for the interpretation of gravity anomalies. Models are composed of two-dimensional polygons with a finite, rather than infinite, width. A forward algorithm gives the theoretical attraction of gravity intensity at a station due to a perturbing body given by the initial model. The model can then be manually adjusted to obtain a better fit, or the program can be employed to do a one-dimensional direct search technique to automatically adjust the model until a good match is obtained between observed and theoretical curves. The search technique usually requires a large number of computations of the forward problem. After a good match is obtained with manual or automatic searching, an inversion can be done to further improve the model. The search method is restricted by a decreasing-resolution-with-depth problem that is inherent in potential field data. The Marquardt inversion requires a good starting point (low objective function) to be effective.

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

GRAV2D is a modification of programs (G2HSER, G2HINV) developed by John Snow while a graduate student at the Department of Geology and Geophysics, University of Utah. GRAV2D is essentially an interactive interface built around Snow's programs. The interactive interface was designed to facilitate data manipulation.

This user's guide will document how to use the interactive interface. The necessary input parameters will be described. The user is referred to Appendix III for a description of the mathematics.

USER'S GUIDE

MASTER

The program GRAV2D operates on an "options list" principle. The user has a variety of available options that allow him to perform the necessary I/O, data manipulation and computation to produce a new model and theoretical gravity values.

By selecting the appropriate options and responding to the prompts correctly, the user can work his way through the

modeling process.

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

GRAV2D

and encounters the first options query as follows:

TERMINL-OPTION # (I)=

The user inputs the appropriate option number to continue with the program. If the user doesn't know or can't remember the available options and their corresponding option numbers, he can obtain the options list by inputting a zero or carriage return, whereby the program outputs the options list to the terminal, and again requests an option number.

The options list for TERMINL- is

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

The user inputs the option number corresponding to the type of terminal being used.

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

MASTER--OPTION # (I)=

The available options under MASTER are

- 1 EXIT FROM SYSTEM
- 2 INPUT NEW DATA
- 3 EDIT DATA
- 4 LIST DATA AND RESULTS
- 5 XQT SEARCH/INVERSION PROGRAM
- 6 FILE MAIN. - SAVE AND RECALL
- 7 PLOT MODELS AND CURVES

The MASTER options are discussed separately in the following sections.

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

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

INPUT

The model is defined by specifying a number of polygons in the x-z plane where x-y-z is a normal right-hand coordinate system with z down, x to the right, and y out of the page. The y dimension is specified by the two strike lengths for that polygon. It is this finite length in the y direction that provides the 1/2 dimension for the 2 1/2 dimensional modeling.

Data is input from the keyboard. Optionally, the station, model and search data can be input from previously created disc files. Keyboard data is entered in free format (list directed READ). Formats for station, model and search data files are also free format, and data would appear in the file just as if the user were answering the interactive queries from the keyboard. See the sample terminal sessions for an example. When input is from data files, the input queries will still appear on the screen.

The following is a list of queries for input and the corresponding input parameters.

INPUT WORK FILE HEADER (80 CHAR.)

Variable name HEAD(20) is an alphanumeric identifier for the work file up to 80 characters.

DO YOU WANT TO INPUT STATION DATA FROM FILE?

Answer appropriately with "YES" or "NO".

If answer to previous query was "NO", input the station data from the keyboard, otherwise

INPUT FILENAME (16 CHAR)

Input the name of the file (up to 16 characters) from which the station data is to be read.

The next five queries are for station data.

INPUT PROFILE ID.

Variable name PROFID(20) is an alphanumeric identifier for the station data. It will be used as a title for the plot output.

INPUT CODE FOR UNITS: 1 = KILOMETERS
2 = METERS
3 = KILOFEET

Variable NUNITS is an integer corresponding to the appropriate distance units used in the program.

INPUT STATION NO. OR MATCH PT.

Variable NBASE is an integer referring to the station number or match point where the theoretical gravity will be set equal to the observed.

Match point at zero will calculate absolute residual.

INPUT NO. OF STATIONS

Variable NSTAT is an integer referring to the number of stations. The maximum number of stations is set to 500.

INPUT OBS. GRAVITY, DIST., ELEV. FOR n STATIONS

Input TD(i), XSTAT(i), ZSTAT(i), i=1, NSTAT

where

TD(i) is the observed gravity value at the ith station,
XSTAT(i) is the distance (in NUNITS) of the ith station from the origin (positive to the right). The first station is always the leftmost station of the profile,
ZSTAT(i) is the elevation above 0.0 (in NUNITS) of the ith station (positive up).

DO YOU WANT TO INPUT MODEL DATA FROM FILE?

Answer appropriately "YES" or "NO".

If answer to previous query is "NO", input the model data from keyboard, otherwise

INPUT FILENAME (16 CHAR)

input the filename of the file containing the data for the model.

This is the first query for the model data. (Note that this data set includes some search data.)

INPUT NO. OF POLYGONS

Variable NPOLY is an integer equal to the number of polygons in the model. The maximum number of polygons is set to 15. The maximum number of vertices per polygon is 33.

FOR POLYGON 1, INPUT:

OF SIDES(I), DENSITY CONTRAST, STRIKE OUT, STRIKE IN, DENSITY SEARCH#(I)

for polygon #1, input variables NSIDES(1), SUS(1), STRIK1(1), STRIK2(1), ISUS(1)

where

NSIDES(1) is an integer equal to the number of sides of polygon 1.

SUS(1) is a floating point number equal to the density contrast of polygon 1.

STRIK1(1) is a floating point number equal to the strike length "out of page" of polygon 1 (in NUNITS).

STRIK2(1) is a floating point number equal to the strike length "into page" of polygon 1 (in NUNITS).

ISUS(1) is an integer equal to the search number for the density contrast of polygon 1 (some number between 1 and the total number of search directions). Enter zero if no search is required on density contrast.

FOR n VERTICES, INPUT:

HORZ. COORD., VERT. COORD., VERT. SEARCH#(I), HORZ. SEARCH #(I)

XSTART(1,n), ZSTART(1,n), IVO(1,n), IVX(1,n) for n=1, NSIDES(1)+1

where

XSTART(1,n) is the horizontal coordinate (in NUNITS) of the nth vertex of the 1st polygon (positive to the right of the origin).

ZSTART(1,n) is the vertical coordinate (in NUNITS) of the nth vertex of the 1st polygon (positive downward).

IVD(1,n) is the search number for vertical movement on the nth vertex of the 1st polygon.

IVX(1,n) is the search number for the horizontal movement on the nth vertex of the 1st polygon.

Note: The vertices are ordered clockwise around the polygon.

Note: For NSIDES(1)+1 vertices, the last input vertex parameters should be the same as the first input vertex parameters, e.g., a closed polygon. Polygons should have all common vertices along mutual borders (see the model in the sample terminal session #2). Remember to change the vertices in both (all) polygons with common vertices when editing a shared vertex.

Note: The polygons next to the surface must have their end points extended beyond the boundaries and a trivial thickness, on the order of 10 meters (0.01 km), added just below the surface to add stability to the search and inversion procedures (see the sample terminal sessions at the end of this document).

Note: The model parameters may be adjusted in any order. For instance, the first search may be a set of vertices (1 or more) which are to be adjusted vertically. Then the IVD(L,LL)=1 for all those vertices which are to be moved in the first search. The second search could then be the density contrast of 1 or more of the polygons. Then ISUS(L)=2 for all those polygons, etc.

Note: For inversion, IVD and IVX specify which parameters will be solved for by inversion. Set IVD and/or IVX to zero to exclude, and non-zero to include in inversion.

FOR POLYGON 2 , INPUT:

OF SIDES(1), DENSITY CONTRAST, STRIKE OUT, STRIKE IN, DENSITY SEARCH#(1)

for polygon #2, input variables NSIDES(2), SUS(2), STRIK1(2), STRIK2(2), ISUS(2)

where

NSIDES(2) is an integer equal to the number of sides of polygon 2.

SUS(2) is a floating point number equal to the density contrast of polygon 2.

STRIK1(2) is a floating point number equal to the strike length "out of page" of polygon 2 (in NUNITS).

STRIK2(2) is a floating point number equal to the strike length "into page" of polygon 2 (in NUNITS).

ISUS(2) is an integer equal to the search number for the density contrast of polygon 2 (some number between 1 and the total number of search directions). Enter zero if no search is required on density contrast.

FOR n VERTICES, INPUT:

HORZ. COORD., VERT. COORD., VERT. SEARCH#(I), HORZ. SEARCH #(I)
XSTART(2,n), ZSTART(2,n), IVO(2,n), IVX(2,n) for
n=1, NSIDES(2)+1

where

XSTART(2,n) is the horizontal coordinate (in NUNITS) of the nth vertex of the 2nd polygon (positive to the right of the origin).

ZSTART(2,n) is the vertical coordinate (in NUNITS) of the nth vertex of the 2nd polygon (positive downward).

IVO(2,n) is the search number for vertical movement on the nth vertex of the 2nd polygon.

IVX(2,n) is the search number for the horizontal movement on the nth vertex of the 2nd polygon.

FOR POLYGON N, INPUT:

OF SIDES(I), DENSITY CONTRAST, STRIKE OUT, STRIKE IN, DENSITY SEARCH#(I)

for polygon #N, input variables NSIDES(N), SUS(N),
STRIK1(N), STRIK2(N), ISUS(N)

where

NSIDES(N) is an integer equal to the number of sides of polygon N.

SUS(N) is a floating point number equal to the density contrast of polygon N.

STRIK1(N) is a floating point number equal to the strike length "out of page" of polygon N (in NUNITS).

STRIK2(N) is a floating point number equal to the strike length "into page" of polygon N (in NUNITS).

ISUS(N) is an integer equal to the search number for the density contrast of polygon N (some number between 1 and the total number of search directions). Enter zero if no search is required on density contrast.

FOR n VERTICES, INPUT:

HORZ. COORD., VERT. COORD., VERT. SEARCH#(I), HORZ. SEARCH #(I)
XSTART(N,n), ZSTART(N,n), IVO(N,n), IVX(N,n) for
n=1, NSIDES(N)+1

where

XSTART(N,n) is the horizontal coordinate (in NUNITS) of the nth vertex of the Nth polygon (positive to the right of the origin).

ZSTART(N,n) is the vertical coordinate (in NUNITS) of the nth vertex of the Nth polygon (positive downward).

IVO(N,n) is the search number for vertical movement on the nth vertex of the Nth polygon

IVX(N,n) is the search number for the horizontal movement on the nth vertex of the Nth polygon.

This is the end of the model data.

DO YOU WANT TO INPUT SEARCH DATA FROM FILE?

Answer appropriately "YES" or "NO".

If answer to the previous query is "NO", input the search data from the keyboard, otherwise

INPUT FILENAME (16 CHAR)

input the name of the file where the search data resides.

This is the first query for the search data.

INPUT # ITERATIONS FOR INVERSION

Variable MAXIT is an integer equal to the maximum number of iterations of the inversion.

INPUT TOTAL # OF VERTEX SEARCH DIRECTIONS

Variable IORD is an integer equal to the total number of vertex search directions, both horizontal and vertical.

This must also be specified non-zero when doing inversion.

The maximum number of search directions is 50.

INPUT TOTAL # OF DENSITY CONTRAST SEARCH DIRS.

Variable ISS is an integer equal to the total number of density contrast search directions.

Note: If both IORD and ISS are zero, only the forward problem will be computed.

If $IORD+ISS > 0$, the next query will appear,

INPUT TOLERANCE, STEP SIZE FOR n SEARCH DIRECTIONS

(cannot be input from file - must be from keyboard)

variables XLIM(n), DELX(n) $n=1, IORD+ISS$

where

XLIM(n) is the tolerance for the nth search. This specifies to what degree of accuracy the positioning of the parameter should be found; given in NUNITS for vertex search and in gm/cc for density contrast search.

DELX(n) is the step size for the nth search direction (in same units as XLIM).

The search proceeds as follows:

The objective function is evaluated. All parameters for that search are adjusted by DELX and the function is reevaluated. If the objective function value improves, the step size is doubled for the next step. If there is no improvement, the step direction is reversed and the function reevaluated. The search then proceeds as long as function improvement is found. If a minimum cannot be found, the parameter will be reset at its initial position for the next search. If a vertex is adjusted past its horizontal limits, the vertex will be set at the limit and excluded from the search. These variables are used only by the automatic searching subroutine, not the inversion.

This is the end of the search data.

INPUT HOR. SCALE, VER. SCALE(CURVES), VER. SCALE(MODEL)

variables HSCALE, VSCALE, VBOD are floating point numbers used for scaling the plots,

where

HSCALE is the horizontal scale in NUNITS per inch for both the observed and computed curve plot and the model plot.

VSCALE is the vertical scale of the plot for the observed and computed anomalies in milligals/inch.

VBOD is the vertical scale of the model plot in NUNITS per inch.

Note: The maximum vertical scale for the curve plot can be determined from the observed anomaly. The maximum vertical scale for the model should be larger than the expected maximum depth of the vertices of the model.

After inputting the last variable, the program returns to the MASTER options.

EDIT

GRAV2D has an internal editor to provide limited editing capability under program control. Below is a list of EDIT options. Selecting EDIT options 3, 4 or 5 results in another level of option requests, such as ESTAT--- for editing station data. The user returns to the higher level options list by responding with 1 to the option request. All of the input parameters can be edited through this internal editor.

EDIT---OPTION # (I)=

- 1 RETURN TO MASTER OPTIONS
- 2 EDIT WORK FILE HEADER
- 3 EDIT STATION DATA
- 4 EDIT MODEL DATA
- 5 EDIT SEARCH/INVERSN DATA
- 6 EDIT PLOT DATA

ESTAT---OPTION # (I)=

- 1 RETURN TO EDIT OPTIONS
- 2 EDIT PROFILE ID.
- 3 EDIT UNITS OF DIST.
- 4 EDIT STATION #/MATCH PT.
- 5 EDIT NO. OF STATIONS
- 6 EDIT DISTANCES
- 7 EDIT ELEVATIONS
- 8 EDIT OBS. GRAVITY VALUES

EMODEL---OPTION # (I)=

- 1 RETURN TO EDIT OPTIONS
- 2 EDIT NO. OF POLYGON

- 3 EDIT # OF SIDES OF POLY.
- 4 EDIT DENSITY CONTRASTS
- 5 EDIT STRIKE LENGTH OUT
- 6 EDIT STRIKE LENGTH IN
- 7 EDIT HOR. COORDINATES
- 8 EDIT VERT. COORDINATES

ESEARCH-OPTION # (I)=

- 1 RETURN TO EDIT OPTIONS
- 2 EDIT # OF ITERATIONS
- 3 EDIT # OF VERTEX SEARCH DIR.
- 4 EDIT # OF DENSITY SEARCH DIR
- 5 EDIT DENSITY SEARCH #S
- 6 EDIT VERT. SEARCH #S(VERTEX)
- 7 EDIT HORZ. SEARCH #S(VERTEX)
- 8 EDIT TOLERANCES FOR SEARCH
- 9 EDIT STEP SIZES FOR SEARCH

After the editing is done and the option to return to MASTER has been selected, the user is asked

HAS THE WORK FILE CHANGED?

and if the user answers "YES", the date stored in the work file is changed to the current date and time to reflect the time of creation of the updated model. The date is obtained with the system routine TIMDAT.

The editor works by allowing the user to input a new value (in free format from the keyboard). If changing arrays, the user specifies the index (indices of 2D arrays) and then inputs the new array element. For 2D arrays the user must specify whether he is changing an element of an array, inserting a new element, or deleting an old one. He can insert and delete prior to changing the value specifying the length of the array, but should take care to remember to also edit the variable storing the length. For example, if the user inserts a new vertex in a polygon, he should also change the number of sides of that polygon. The printing routines will reflect the value that is currently stored in the variable containing the array length. If the index specified is less than zero (negative), the entire array is then input, one element at a time. An element is inserted after the one specified by the index. If one wants to insert before element (N,1), insert an element equal to (N,1) after (N,1) and then change the value of (N,1) to the value of the inserted element.

LIST

The list options are given below.

LIST----OPTION # (I)=

- 1 RETURN TO MASTER
- 2 LIST FILE HEADER, DATE
- 3 LIST STATION DATA
- 4 LIST MODEL DATA
- 5 LIST SEARCH DATA
- 6 LIST PLOT DATA
- 7 LIST ENTIR FILE CONTENTS
- 8 SEND WORK FILE TO PRNTR

All output is in 80 column format. Options 2 thru 7 print at the terminal. Option 8 prints the entire work file contents at the on-site printer. If the terminal type (from the TERMINL- options list) is not "hard copy", paging will be done so that the user can examine the printed information before it is scrolled off the screen. Typing a "carriage return" will cause the printing to continue. Option # 3 requests a starting and ending index for printing the station array so that only a portion can be printed. A starting and ending index of 0,0 will result in the entire array being printed.

EXECUTE

This option causes execution of Snow's programs. The forward problem, the 1D direct search, or the inversion can be done depending on the input parameters stored in the work file and the options selected. The output (calculated gravity value and new model) is written to the work file upon completion of the forward problem. The output from the direct search or inversion is stored only if the user answers the following query with a "YES".

DO YOU WANT TO SAVE THIS?

The execute options are given below.

XQT-----OPTION# (I)=

- 1 RETURN TO MASTER
- 2 XQT SEARCH
- 3 XQT INVERSION

The forward problem only is done by selecting option #2 with IORD and ISS both equal to zero.

Upon entering the search/inversion subroutines, the user can select certain output options. They are given below.

OUTPUT--OPTION# (I)=

- 1 NONE
- 2 PLOT
- 3 PRINT
- 4 PLOT AND PRINT

The options above allow the user to direct output during the execution of the program. It is most useful when the user wants to see what is happening during the execution. He can produce Tektronix plots of the curves and model or print the observed and calculated gravity values.

If the user selects OUTPUT-- options 3 or 4, another option request is made. It is

```
PRT.DEV-OPTION # (I)=
1  TERMINAL
2  PRINTER
```

and is used to determine where the print will appear. If the user selected options 2 or 4, another device selection option request is made. It is

```
DEVICE---OPTION # (I)=
1  TEKTRONIX GRAPHICS PLOT
2  STATOS PLOT
3  UUCS CALCOMP PLOT
```

and is used to determine where the plotting will appear.

Note: Any statos plot or calcomp plot will only be stored as a file of plot vectors until the proper PRIMOS command is input once the user has terminated GRAV2D.

If the user is doing the search or inversion, another option request is made if he has selected an OUTPUT- option other than #1. It is an option request to determine when during the iterations, the output will be produced. The available options are

```
WHEN----OPTION # (I)=
1  AFTER FINAL ITER
2  AFTER EACH ITER
3  AFTER SOME ITES
```

If WHEN---- option #3 is selected, the program queries the user after each iteration

DO YOU WANT OUTPUT THIS ITER.?

The program then gives the user the option of continuing with the search/inversion or stopping execution of the search/inversion and returning to the XQT--- options list.

If the user has selected DUPUT- option #1, he can examine the results by using MASTER options #4 (LIST) and #7 (PLOT).

When finished with the search or inversion, the user can either save the new model and calculated gravity values or retain the starting model by answering appropriately to the query

DO YOU WANT THIS SAVED?

FILE MAINTENANCE

This option allows the user to save his work files on a merge file (a catalog of files). This option contains utilities for manipulating the merge file. The merge file should be initialized (with FMAIN- option #6) before attempting to copy the first work file.

The available File Maintenance options are

FMAIN---OPTION # (I)=

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

PLOT

The plot option produces a plot of the calculated and observed gravity values and a plot of the model. Sample plots are included in the terminal sessions.

The plot options are

PLOT----OPTION # (I)=

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

Option #2 plots the model and gravity values in the work file. Option # 3 produces plots from the specified work files in the merge file.

The PLOT option produces a device selection option request to determine which plotting device is to be used. The device selection is the same as the one used in EXECUTE, called DEVICE-.

Status and Calcomp plots are generated but not plotted until the user terminates GRAV2D (by selecting MASTER- option #1) and types the proper PRIMOS command.

DIRECT ACCESS FILE STRUCTURE

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

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

GRAV2D-WORK is the name of the direct-access file that contains the current model parameters and the calculated gravity values. Most options work on data from this file and store results into this file. The two exceptions are MASTER Option #6 (File Maintenance) which provides a number of utilities that access the second direct-access file called GRAV2D-MERGE, and the PLOT Option #3 which plots models and curves from selected files stored on the merge file.

GRAV2D-MERGE is essentially a catalog of different work files. GRAV2D-WORK can be stored on the merge file. The work files on the merge file can be moved back into GRAV2D-WORK for manipulation with the program options.

The direct-access work file has 500 words per record and is 11 records long. The variables that are written to the work file are stored in COMMON blocks defined in the \$INSERT file. An entire record is written to the work file with a single write statement by specifying the beginning variable in the COMMON block associated with that record irregardless of mixed type within the COMMON block. The COMMON blocks are filled out with dummy arrays to insure correct record length.

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

SYSTEM IMPLEMENTATION

The PRIME 400 is a virtual memory machine. No attempt was made to optimize paging for this program.

GRAV2D uses 1344K words of memory (16 bits/word). Most floating point and integer numbers are 32 bits. Most system subroutines require short integer arguments (INTEGER*2). PRIME FORTRAN has two functions, INTL and INTS, that convert short integers to long (INTEGER*4) and long integers to short. Some variables in the program are double precision (REAL*8) to increase the accuracy of repeated calculations. A machine with a larger word size than 32 may get by with single precision.

The large size of the program is primarily a function of the maximum limits of the number of stations and model size. The \$INSERT file contains some very large COMMON blocks, G2HMAN and G2HIV, that are dependent on the following PARAMETER variables

```

NSTATS = IPWID = 500    - # of stations, also
                        width of direct
                        access file records
NSRDIR = 50            - # of search directions
NPOLYG = 15            - # of polygons in model
NVERTX = 33            - # of vertices per
                        polygon

```

These COMMON blocks store intermediate results for the inversion and search subroutines.

The amount of storage required by this program can be reduced considerably by decreasing the four PARAMETER variables mentioned above. They need only be changed in the \$INSERT file. (See \$INSERT and PARAMETER statement descriptions in Appendix I.) Changing the width of the direct-access file records will make previously created direct-access files unusable. The \$INSERT file contains some notes on changing these PARAMETER variables. The entire code in the \$INSERT file could be substituted for the \$INSERT GRAV2D.COM statements throughout the program if no equivalent mechanism exists at the site installing the program.

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

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

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

GRAV2D makes use of a device-independent plotting software package resident on the system for plotting on various graphic devices. This software library, UUPLT, is available from UURI/ESL. It uses Tektronix PLOT10 software to interface to the resident Tektronix 4014 graphics terminal. PLOT10 is available from Tektronix. It uses Varian Graphic's DATAPLOT III software to interface to the resident Varian STATOS electrostatic plotter. GRAV2D generates plot files that can be read by the University of Utah Computer Center's (UU/CC) CalComp plotting system. The CalComp plot files are written to magnetic tape and transferred to UU/CC for actual plotting.

The plotting subroutines in GRAV2D are very modular and could easily be replaced to conform to the plotting capabilities of another installation or be removed entirely.

The program uses five logical units for I/O.

- LU 1 - terminal read/write
- LU 8 - assigned to the work file
- LU 9 - assigned to the merge file
- LU 11 - assigned to temporary print files
- LU 13 - assigned to input files, if used

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

GRAV2D makes use of some system subroutines. See Appendix II for a list and description. Integer arguments are short integer unless otherwise stated. An equivalent system routine is probably available at any site. PRIME manual SUBROUTINES, P/N MAN3251-001 contains more detail if needed.

ACKNOWLEDGMENTS

The original program source code for GRAV2D.REV1 was developed by John H. Snow with help from Dr. Ralph T. Shuey, Robert F. Sawyer, and James A. Carter, all at the University of Utah. John Snow's thesis committee was Dr. Kenneth L. Cook (Chairman), Dr. Ralph T. Shuey, and Dr. Francis W. Christiansen.

The original program source code that was published in John Snow's M.S. thesis has been modified, expanded, and integrated into a modeling system by Carleen Nutter at the Earth Science Laboratory Division, University of Utah Research Institute.

Funding for the development of this software came from 1) The National Science Foundation (Grant Nos. DES 71-00922-A02 and GA-30182), the Mineral Leasing Fund of the University of Utah, the University of Utah Research Funds, and the University of Utah Department of Geology and Geophysics Special Funds for the contribution by John Snow; and 2) The Department of Energy / Division of Geothermal Energy - Idaho Operations Office (Contract No. DE-AC07-80ID 12079) for the contribution by Carleen Nutter.

John H. Snow has given the Earth Science Laboratory Division permission to publish Appendix I (pages 129 through 142) of his M.S. thesis which appears in this report as Appendix III. This Appendix explains the methodology of the program and we are grateful for John's permission to include it in this user's guide.

REFERENCES FOR APPENDIX III

- Al-Chalabi, M., 1971, Interpretation of gravity anomalies by non-linear optimization: Geophysical Prospecting, vol. 20, p. 1-16.
- Cady, J. W., 1977, Calculation of gravity and magnetic anomalies along profiles with end corrections and inverse solutions for density and magnetization: U.S. Geol. Survey Open-file report no. 77-463.
- Kuester, S. L., and Mize, J. H., 1973, Optimization techniques with Fortran: McGraw-Hill, New York, N.Y.
- Lanczos, C., 1961, Linear differential operators: D. Van Nostrand, Princeton, N.J.
- Shuey, R. T., 1977, Inversion theory, GGS 638, class notes and personal communication with J. H. Snow.
- Shuey, R. T., and Pasquale, A. S., 1973, End corrections in magnetic profile interpretation: Geophysics, v. 28, n. 3, p. 507-512.
- Talwani, M., Worzel, J. L., and Landisman, M., 1959, Rapid gravity computations for two-dimensional bodies with application to the Mendocino submarine fracture zone: Jour. Geophy. Research, v. 64, p. 49-59.

APPENDIX I
PRIME FORTRAN EXTENSIONS

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

CODING statements

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

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

Formatted DECODE statement

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

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

List-directed DECODE statement

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

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

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

ENCODE statement

ENCODE(c,f,a) list

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

List Directed READ statement

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

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

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

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

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

-357 100514 +12387

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

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

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

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

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

Direct access READ and WRITE statements

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

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

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

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

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

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

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

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

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

Direct-access READ statements

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

- u A long or short integer constant or variable whose value is the FORTRAN unit number.
- r The long or short integer expression whose value is the record number to be accessed.
- f The statement number of the format specified (optional).
- b The statement number to which control is transferred if a device or format error is encountered during transfer (optional).

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

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

Direct-access WRITE statements

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

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

NOBIG / BIG

-BIG (a parameter on the compilation statement)

BIG treats all dummy arrays as arrays that span segment boundaries. BIG forces the 64V mode and thus cannot be used in the 32R or 64R modes. If a dummy argument array may become associated with an array spanning a segment boundary (through a subroutine call or function reference), the compiler must be made aware of this by including BIG in the parameter list. The code generated will work whether or not the array actually spans segment boundaries.

PARAMETER statement

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

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

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

\$INSERT statement

\$INSERT insert-file

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

\$INSERT is used for:

- Insertion of COMMON specification into programs.
- Commonly used one-line functions.
- Data initialization statements.
- Parameter definitions, especially for the file management system, applications library, MIDAS, etc.

APPENDIX II

PRIME SYSTEM ROUTINES

The following material is taken from PRIME system manual PRIMOS SUBROUTINES REFERENCE GUIDE PDR 3621, dated March 1979, reproduced here with the permission of PRIME Computer, Inc.

This section is intended to describe the function of PRIME system routines used by GRAV2D. A non-PRIME site will have to substitute equivalent routines suitable to its operating environment. If the user requires more detail on system routines, he is directed to the appropriate PRIME manuals.

UURI/ESL's device-independent plotting software, UUPLT, is not documented in this publication, but is available as a separate document from ESL.

ATCH\$\$

ATCH\$\$ attaches to a UFD and, optionally, makes it the home UFD. In attaching to a directory, the subroutine ATCH\$\$ specifies where to look for the directory. ATCH\$\$ specifies a User File Directory (UFD) in the Master File Directory (MFD) on a particular logical disk, a sub-directory in the current UFD, or the home UFD as the target-directory of the ATCH\$\$ operation.

CALL ATCH\$(ufdnam, namlen, ldisk, passwd, key, code)

ufdnam The name of the UFD to be attached. If key=0 and ufdnam is the key K\$HOME, the home UFD is attached. If the reference subkey is K\$IMFD or K\$ICUR, ufdnam is either a Hollerith expression or the name of a three-word array that specifies a ufdname to attach to.

namlen The length in characters (1-32) of ufdnam. If ufdnam=K\$HOME, namlen is disregarded.

ldisk The number of the logical disk to be searched for ufdnam when key=K\$IMFD. The parameter, ldisk, must be a logical disk that is started up. Other values are:

K\$ALLD Search all started-up logical devices in logical device order, and attach to the UFD in which ufdnam appears in the MFD of the lowest numbered logical device.

K\$CURR Search the MFD of the disk currently attached.

passwd

A three-word array containing one of the passwords of ufdnam. Passwd can be specified as 0 if attaching to the home UFD. If the reference subkey is K\$IMFD or K\$ICUR, passwd is either a Hollerith expression (1 to 6 characters) or the name of a three-word array that specifies one of the passwords of ufdnam. If passwd is blank, it is specified as three words, each containing two blank characters.

key

Composed of two subkeys that are combined additively, a REFERENCE subkey and a SETHOME subkey. The REFERENCE subkey values are as follows:

K\$IMFD Attach to ufd in MFD on ldisk.

K\$ICUR Attach to ufdnam in current UFD (ufdnam is a subdirectory).

The SETHOME subkey, K\$SETH, may be added to the REFERENCE subkey, e.g., K\$IMFD+K\$SETH, which will set the current UFD to the home UFD after attaching. If the REFERENCE subkey is K\$ICUR, or if ufdnam is 0, ldisk is ignored, and ldisk is usually specified as 0.

code

An integer variable set to the return code.

To access files, the file system must be attached to some UFD. This implies that the file system has been supplied with the proper file directory name and either the owner or nonowner password, and the file system has found and saved the name and location of the file directory. The nonowner password may be given only under PRIMOS and PRIMOS III.

A BAD PASSWD error condition does not return to the user's program. PRIMOS command level is entered, and the user is not attached to any UFD. Other errors leave the attach point

unchanged.

ATTDEV

ATTDEV attaches specified devices by initializing both LUTBL, associating logical-device to physical device, and PUTBL, associating the logical-device to a specific unit or file of the device.

CALL ATTDEV(logical-device, physical-device, unit, buffer-size)

logical-device	The device-independent logical I/O unit for WRASC, RDASC, WRBIN, RDBIN, and FORTRAN READ and WRITE statements.
physical-device	The position in the device-type tables.
unit	The unit for multi-unit devices (e.g., for disk, file unit number).
buffer-size	The record size associated with the logical unit. Must be as large as maximum record size.

For the given logical-device, set the physical-device, unit, and buffer-size in the LUTBL, PUTBL, and RSTBL so that the logical unit has a current mapping.

Errors

If device is incorrect, ATTDEV returns the message: ATTDEV BAD UNIT (unit-number).

CLOS\$A

CLOS\$A is a logical function that closes the file open on unit. If the operation is successful, the function is **.TRUE.** and if unsuccessful, the function is **.FALSE.**

LOG = **CLOS\$A**(unit)

CALL CLOS\$A(unit)

unit

File unit.

unit is **INTEGER*2**.

DELE\$A

DELE\$A is a logical function which deletes the file named in name. If the operation is successful, the function is **.TRUE.** and if unsuccessful, the function is **.FALSE.**

LOG = **DELE\$A**(name,namlen)

CALL DELE\$A(name,namlen)

name

File name (may be a treename).

namlen

Length of name in characters.
(Mode is **INTEGER*2**, but the
type of name doesn't matter.)

ERRPR\$

ERRPR\$ interprets a return code and, if non-zero, prints a

standard message associated with the error return code, code, followed by optional user text.

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

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

K\$NRTN Exit to the system, never return to the calling program.

K\$SRTN Exit to the system, return to the calling program following an 'S' command.

K\$IRTN Return immediately to the calling program.

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

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

txtlen The length in characters of text.

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

namlen The length in characters of filnam.

EXIT

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

CALL EXIT

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

EXST\$A

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

LOG = EXST\$A(name,namlen)

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

OPEN\$A

OPEN\$A is a logical function that opens a file of the given name on unit. If the operation is successful, the function value is .TRUE. and if the operation is unsuccessful, the function value is .FALSE..

LOG = OPEN\$A(opnkey+typkey,name,namlen,unit)

CALL OPEN\$(opnkey+typkey, name, namlen, unit)

opnkey	A\$READ, open for reading (.NE. A\$WRIT or A\$RDWR); A\$WRIT, open for writing; A\$RDWR, open for reading and writing.
typkey	A\$SAMF, SAM file (.NE. A\$DAMF); A\$DAMF, DAM file.
name	File name (may be treename).
namlen	Length of name in characters.
unit	PRIMOS file unit.

All arguments are INTEGER*2 except name whose type does not matter.

SLEEP\$

SLEEP\$ suspends execution of a user process.

CALL SLEEP\$(interval)

interval	A variable in milliseconds for which execution is to be suspended (INTEGER*4).
----------	--

Execution of the user process is suspended for the specified interval. An interval $\leq$ 0 will result in an effective no-op. A QUIT and START from the user terminal will cause immediate return from the SLEEP\$ call.

SP00L\$

A user program can insert a file into the spool file directory by calling the SPOOL\$ subroutine from the applications program. This subroutine SPOOL\$ is in the SPOOL\$ library (R-mode) and VSP00\$ library (V-mode).

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

```
key          1 - copy named file into queue.
              2 - open file on unit info(2)
                  for writing.
```

```

name      File to be copied (key=1).
          Name to appear on banner
          (key=2).

```

```

namlen      Length of name,  in  characters
            (1-32).

```

```
info          Information array, 12 elements,
              as follows: "
```

```

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

```

```
buffer          Scratch buffer - this is used
                to set up control info and to
                copy the file to the spool
```

queue (key=1) - it must be at least 40 words long. Copy time is inversely proportional to buffer size. Nominal size is between 300-2000 words.

buflen	Length of buffer.
code	Return code (non-zero if file system error).

SRCH\$\$

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

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

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

K\$READ Open filnam for reading on funit.

K\$WRIT Open filnam for writing on funit.

K\$RDWR Open filnam for reading and writing on funit.

K\$CLOS Close file by filnam or by funit.

K\$DELE Delete file filnam.

K\$EXST Check on existence of filnam.

ref	A subkey modifying the action subkey as follows:
-----	--

K\$IUFD Search for filnam in the current UFD (this is the default).

K\$ISEG Perform the action specified by action on the file that is a segment directory entry in the directory open on file unit filnam.

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

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

newfil

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

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

K\$NDAM New directed (DAM) file.

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

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

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

filnam

Name of the file to be opened

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

namlen The length in characters (1-32) of filnam.

funit The number (1-15 under PRIMOS II) of the file unit to be opened or closed, or returned argument with K\$GETU key.

type An integer variable that is set to the type of the file opened. Type is set only on calls that open a file --- it is unmodified for other calls. Possible values of type are:

- 0 SAM file.
- 1 DAM file.
- 2 SAM segment directory.
- 3 DAM segment directory.
- 4 UFD.

code An integer variable set to the return code.

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

TIMDAT

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

CALL TIMDAT(array,num)

array

- (1) Two ASCII characters representing month.
Example: 11
- (2) Two ASCII characters representing day.
Example: 30
- (3) Two ASCII characters representing year.
Example: 75
- (4) Integer time in minutes since midnight.
- (5) Integer time in seconds.
- (6) Integer time in ticks.
- (7) Integer CPU time used in seconds.
- (8) Integer CPU time used in ticks (standard is 330 ticks/second).
- (9) Integer disk I/O time used in seconds.
(see note)
- (10) Integer disk I/O time used in ticks. (see note)
- (11) Integer number of ticks per second.
- (12) User number.
- (13-15) Six-character login name, left-justified.
Example: MSMITH

num

Num words of array are set. This routine does not return any useful information under PRIMOS II.

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

TNOU

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

CALL TNOU(buffer,count)

TNOUA

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

CALL TNOUA(buffer,count)

UNIT\$A

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

LOG = UNIT\$A(unit)

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

APPENDIX III
DESCRIPTION OF NONLINEAR OPTIMIZATION METHOD

by

John H. Snow

REFERENCE

Snow, J. H., 1978, Study of structural and tectonic patterns in south-central Utah as interpreted from gravity and aeromagnetic data, M.S. thesis, University of Utah.

Methods, such as those given by Talwani and others (1959) for 2-dimensional gravity modeling or by Shuey and Pasquale (1973) for 2-1/2-dimensional magnetic modeling, are commonly used for the interpretation of gravity and magnetic anomalies. Such methods simulate observed data by employing a forward algorithm which will give the theoretical attraction of gravity or magnetic intensity at some point (station) due to a perturbing body of polygonal cross section and of uniform density or magnetization. A theoretical anomaly is computed from some initial model, specified by the vertices $(x(i), z(i))$ of the polygon, a density or magnetic susceptibility contrast, and a strike length (either finite or infinite), and compared with the observed anomaly. The model is then adjusted, either manually or by an optimization or automatic procedure, until a good match is obtained between observed and theoretical curves.

Automatic modeling of gravity or magnetic data is a necessarily complicated problem for several reasons. 1) The forward problem is in general nonlinear and therefore usually requires a nonlinear optimization method to find a set of reasonable models (unless a good initial guess is available). 2) The number of parameters is usually large. For instance, if one were trying to find an optimum position, both horizontally and vertically for all the vertices of an eight-sided polygon (a relatively simple model), one would need to solve for sixteen parameters. 3) The non-uniqueness of potential field data can lead into an infinite number of "good fit" solutions, most of which are geologically unreasonable. 4) The model at best is only a much simplified approximation to the true subsurface structure.

Optimization techniques utilize an objective function such as

$$(1) \quad SSR(\bar{\theta}) = \sum_{i=1}^{NSTAT} (O_i - C_i)^2$$

where $SSR(\bar{\theta})$ is the sum of the differences between the

observed anomaly $O(i)$ and computed anomaly $C(i)$ squared. Given an initial model, the optimization procedure searches through a p -dimensional parameter space for a minimum of the objective function $SSR(\theta)$ by some type of iterative adjustment of the model parameters (θ). The behavior of the objective function in parameter space is complicated by the fact that the parameter space may contain many local minima which represent "good fit" solutions but are not realistic (Al-Chalabi, 1971).

The problem then is to find the minima which represent geologically reasonable models and to eliminate those minima which represent unreasonable models (many of which can give a solution within the tolerance of the data).

The simplest type of nonlinear optimization technique is just a manual adjustment of the model parameters until a good fit between observed and theoretical anomalies is obtained (usually referred to as the forward problem). The interpreter essentially eliminates unrealistic models by assuming an initial guess which is reasonable (either based on additional geologic and/or geophysical information, and/or an intuitive feel), and modifies that guess to fit the observed data.

Automated optimization methods such as a search technique do much the same thing. An initial guess is made and the objective function is evaluated. One or more of the parameters is then adjusted by either a systematic or random process and the objective function is re-evaluated. This usually requires a large number of computations of the forward problem. When a minimum is found the model is tested by some criteria to see if it is reasonable. By repeatedly evaluating different minima in the parameter space, a region which defines acceptable models can be found.

Other optimization methods, such as derivative methods, start from an initial point in parameter space and find the direction of steepest descent, (or direction toward a minima), of the objective function. The function can be linearized by representing it by a Taylor series expansion, and parameter corrections can then be calculated using linear inversion techniques. This process is repeated until a "good fit" solution is obtained. These methods require a starting point near an acceptable minimum to locate a reasonable model.

Al-Chalabi (1971) points out that a search method should be employed in the early stages of optimization to locate acceptable minimum and then an appropriate gradient method can be used to converge to the solution. The method presented here utilizes this two-phase approach, i.e., a 1-dimensional direct search to find an approximate solution, and a linearization method which further minimizes the

objective function (equation 1).

PHASE 1: 1-Dimensional Direct Search

A one-dimensional search technique was chosen for the phase 1 part of the optimization to minimize the number of forward problem computations and therefore reduce the computer costs. This method utilizes a 1-dimensional search algorithm (Kuester and Mize, 1973) which minimizes the objective function

$$(2) \quad SSR(\bar{\theta}) = \sum_{i=n_1}^{n_2} (O_i - C_i(\bar{\theta}))^2$$

where $n(1)$ and $n(2)$ define a data window, the width of which depends on the type of parameter (θ) being used (the data window will be discussed in more detail in a later section).

The forward algorithms for 2-dimensional and 2-1/2-dimensional gravity and 2-1/2-dimensional magnetic modeling as given by Talwani and others (1959), Cady (1977), and Shuey and Pasquale (1973) can be expressed as

$$(3) \quad C_i = \sum_{j=1}^{NPOLY} \phi_j \sum_{k=1}^{NSIDES(j)} Z_{ijk}(x_k, z_k)$$

where $\phi(j)$ represents either the density or susceptibility contrast of the j th polygon, and $Z_{ijk}(x(k), z(k))$ represents the analytic expression for the integral over one side of the j th polygon with respect to the i th station. The total theoretical anomaly at the i th station is the sum of the integrals over all sides of all the polygons in the model.

Equation 2 can then be expressed as

$$(4) \quad SSR(\bar{\theta}) = \sum_{i=n_1}^{n_2} \left(O_i - \sum_{j=1}^{NPOLY} \phi_j \sum_{n=1}^{NSIDES(j)} Z_{ijk} \right)^2$$

In the 1-dimensional search, a single parameter (either densities, susceptibilities, or certain vertices of the model) is incremented by a step size in some user-specified direction and the remaining parameters held fixed. Note that parameter, as used here, refers to some part of the model.

This may be a single vertex of a set, or a single density or susceptibility contrast or a set. The objective function is evaluated at each step until a minimum is found or the search goes beyond the desired range of search. When only the density or susceptibility is varied, equation 4 reduces to

$$(5) \quad SSR(\bar{\theta}) = \sum_{i=1}^{NSTAT} (O_i - (A_i + \sum_{j=1}^k \phi_j B_{ij}))^2$$

where $SSR(\bar{\theta})$ is the sum over all stations (NSTAT), and depends only on the density or susceptibility contrast $\phi(j)$. $A(i)$ represents the contribution of those polygons, which do not change, and $B(i,j)$ represents the contribution of the sum of the integrals around those k polygons whose densities or susceptibilities change.

When a vertex or set of vertices is varied, equation 3 becomes

$$(6) \quad SSR(\bar{\theta}) = \sum_{i=1}^{n_2} (O_i - (A_i + \sum_{j=1}^m \phi_j \sum_{k=1}^l Z_{ijk}))^2$$

where the objective function depends on those l integrals of the m polygons which change. $A(i)$ is the contribution of that part of the model which remains fixed and the remaining contribution is from the parts of the model which change and need to be recalculated at each step. Equations 5 and 6 were used to further reduce the computation time required for the search.

The Search Method

The search procedure used is a slight modification of an algorithm by Coggins (Kuester and Mize, 1973). The method finds an unconstrained minimum of a single parameter, nonlinear objective function, where the objective function is assumed to be unimodal (note: The program actually finds a maximum, which is just the negative of a minimum). The algorithm works as follows:

An initial starting point is chosen and the objective function evaluated. The parameter is then incremented by the step size DELX and the objective function re-evaluated. If function improvement is obtained, the step size is doubled for the next function evaluation. If improvement is not obtained the step direction is reversed.

When a minimum is found, the procedure will yield the last three values of the objective function which straddle the minimum. A quadratic equation is then curve-fitted to these

points. The derivative of the quadratic equation is then taken with respect to the parameter and set equal to zero. This equation can then be solved for the optimum parameter location.

Constraints were added to the search algorithm which stop the search when the step size exceeds 20 times its initial value to avoid unrealistic results when a minimum could not be found.

The Data Window

Because more than one minimum could be found along a given search direction (i.e., multimodal), the objective function $SSR(\theta)$ uses a restricted sum of squares defined by a data window of varying width. The width of the data window is based on the type of parameter under consideration.

When density or susceptibility is being varied, the entire data set is used for the objective function. The reasoning here is that in general this search direction will be unimodal due to the fact that if density or susceptibility are changed, only the amplitude of the anomaly is changed. For example, for a single polygon with fixed shape and some true density, the objective function has a single minimum at the true density. That is, for an anomaly calculated using the true model vertices but a density contrast higher or lower than the true density contrast, the anomaly will be larger or smaller than the true anomaly and the difference between them increased.

This is, however, not necessarily the case when the density is held fixed and the vertices of the polygon are adjusted. Al-Chalabi (1971) shows several good examples of gravity and magnetic models which have approximately the same density or susceptibility contrasts and give very similar anomalies, but have very dissimilar shapes.

When the entire data space is used to adjust, say, the depth of a single vertex, that vertex may be pushed to extreme positions. This occurs because, while moving the vertex only a reasonable amount may improve the fit in the vicinity of the vertex, an extreme move could create large residuals in the vicinity of the vertex but may give a lower total residual sum of squares for the overall data set. Therefore, when vertices are adjusted, only those stations which lie in the vicinity of the vertices are used.

The actual data window employed here is either twice the depth of the deepest vertex of that part of the model being adjusted, or the maximum and minimum horizontal coordinates of the vertices which come before and after that part of the model being changed, whichever provides the larger window. For example, if a single vertex $(x(i), z(i))$ were to be moved,

the maximum width of the data window would be either $|x(i+1) - x(i-1)|$ or $2 \cdot X \cdot z(i)$, whichever gives the larger window.

Model Constraints

Constraints were also placed on how much any search parameter in the model could be adjusted. As mentioned earlier, no parameter can be adjusted further than 20 times the initial step size for that search direction. Further constraints were placed on the vertice search directions. For horizontal adjustment in gravity modeling, no vertices can be moved past the fixed vertices before or after (in a clockwise sense) those vertices which are being moved. This then will allow only for vertical or lower angle faults to be adjusted, unless the entire fault plane is being moved. Magnetic modeling is the same except the user has the option to specify how much more to allow the vertices to move. Once a vertex has been moved past its limit, that vertex is fixed at the limit and deleted from further search.

Vertical adjustment was limited only in that no vertices were allowed to move above the surface, but they are not deleted from search.

PHASE 2: Marquardt Inversion

Once an approximate minimum is found, a derivative or linearization method can be used to adjust all the parameters simultaneously to obtain a final model.

The forward problem is linearized by expanding it into a Taylor series and neglecting higher order terms. From equation 3 we obtain

$$(7) \quad C_i^T = C_i^0 + \sum_{m=1}^M \frac{\partial C_j^0}{\partial \theta_m} \Delta \theta_m = \dots\dots\dots$$

where C_i^T is the theoretical anomaly, C_i^0 the initial estimate of C_i^T , the $\theta(m)$ are the parameters of the model, and the $\Delta \theta(m)$ the parameter corrections which, when added to the parameters of the initial model, will give a better approximation to the theoretical anomaly C_i^T . If we then substitute the observed anomaly $O(i)$ for the theoretical anomaly C_i^T , expand the function about the initial guess, and rearrange terms, equation 7 becomes

$$(8) \quad G_i = (O_i - C_i^0) = \sum_{m=1}^M \frac{\partial C_j^0}{\partial \theta_m} \Delta \theta_m$$

This then will give NSTAT (number of stations) equations in M (number of parameters) unknowns which can be solved using matrix inversion techniques.

Equation 8 can be represented in matrix notation as

$$(9) \quad \Delta \bar{G} = A \Delta \bar{\theta}$$

(the bar indicates a vector). The vector $\bar{G}$ is a residual vector of length NSTAT and $\Delta \bar{\theta}$ is a parameter correction vector of length M. A is an NSTAT X M Jacobian matrix given by

$$A = \begin{vmatrix} \frac{\partial C_1}{\partial \theta_1} & \dots & \frac{\partial C_1}{\partial \theta_m} \\ \vdots & & \vdots \\ \frac{\partial C_{\text{NSTAT}}}{\partial \theta_1} & \dots & \frac{\partial C_{\text{NSTAT}}}{\partial \theta_m} \end{vmatrix}$$

The elements of the matrix A are partial derivatives of the computed field $C(i)$ at each station i , with respect to a particular parameter $\theta(m)$ (either susceptibility or density contrasts, and horizontal and vertical positions of the vertices). The derivative with respect to either density or magnetization is obtained directly from equation 3 simply by replacing $\phi(j)$ with 1, or

$$(10) \quad \frac{\partial C_i}{\partial \phi_j} = \sum_{k=1}^{\text{NSIDE}(j)} Z_{ijk}$$

The derivative of $C(i)$ with respect to a vertex is obtained numerically as follows:

$$(11) \quad \frac{\partial C_i}{\partial x_{jk}} = \frac{C_i(x_{jk} + \Delta x) - C_i(x_{jk})}{\Delta x}$$

for horizontal derivatives, and

$$(12) \quad \frac{\partial C_i}{\partial z_{jk}} = \frac{C_i(z_{jk} + \Delta x) - C_i(z_{jk})}{\Delta x}$$

for vertical derivatives. Δx is a small increment in either the x (horizontal) and z (vertical) directions (a step of 0.1 was used for the gravity irregardless of the units of distance used, such as kilometers or kilofeet). For the magnetics, a step of 10 percent of the depth of the vertex was used. This was simply to avoid computer underflows when working with units such as meters (i.e., at a depth of 1000 meters moving a vertice in any direction by 0.1, would give a very small change in the magnetic field measured at the surface). Either way, the parameter corrections are virtually the same.

Equation 9 can then be solved by finding a general linear inverse to the Jacobian matrix A. Lanczos (1961) has shown that the matrix A can be decomposed into

$$(13) \quad A = U\Lambda V^T$$

(where the T indicates transpose), by the principles of eigenvector decomposition. The columns of U are M eigenvectors of length NSTAT associated with the columns of A, and the rows of V^T are M eigenvectors of length M associated with the rows of A. Λ is an M X M diagonal matrix containing the eigenvalues ($\lambda(i)$) associated with the matrix A. The inverse of A can then be easily obtained from

$$(14) \quad A^{-1} = V\Lambda^{-1}U^T$$

that is, by inverting the diagonal elements of Λ and pre-multiplying by V and post-multiplying by U^T .

Using the generalized inverse and solving for the parameter corrections $\Delta\bar{\theta}$, we obtain

$$(15) \quad \Delta\bar{\theta} = A^{-1}\Delta\bar{G}$$

A more convenient form of equation 15, in terms of regression analysis, is obtained from equation 9 by premultiplying both sides of the equation by A^T and then solving for the

parameter corrections,

$$(16) \quad \Delta \bar{\theta} = (A^T A)^{-1} A^T \Delta \bar{G}$$

In terms of the eigenvector decomposition this can be written as

$$(17) \quad \Delta \bar{\theta} = V \frac{1}{\Lambda} U^T \Delta \bar{G}$$

For small eigenvalues of the matrix A, the inverse can give rise to large changes in the parameters. This may drive the parameters outside the range of the linearization, (i.e., outside the range of the local minimum). A more stable form is given by the Marquardt step (Shuey, 1977), as

$$(18) \quad \Delta \bar{\theta} = (A^T A + (SSQ)I)^{-1} A^T \Delta \bar{G}$$

I is the identity matrix and SSQ is the estimate of the variance determined from the last value of the objective function $SSR(\bar{\theta})$ as

$$(19) \quad SSQ = SSR(\bar{\theta}) / (NSTAT - M)$$

Therefore, using the Marquardt step equation 17 becomes

$$(20) \quad \Delta \bar{\theta} = V \frac{\Lambda}{\Lambda^2 + (SSQ)I} U^T \Delta \bar{G}$$

The Linearization and Linear Regression Analysis

Once the model has been linearized, the usual statistical formulae of regression theory can be applied. Any statistical results derived from the linearization, however, are only valid to the extent that the linearization provides a good approximation to a realistic geologic model. That is, some models can be found which give good statistical results, but must be discounted because they are not geologically reasonable.

In linear regression theory, the statistical model can be represented as

$$\bar{Y} = X\bar{\beta} + \bar{\epsilon}$$

where $\bar{Y}$ is a vector of observations of length N , X is an $N \times M$ system matrix, $\bar{\beta}$ is a vector of the expected model parameters of length M , and $\bar{\epsilon}$ is a vector of length N which represents the observation error.

When no bias is present in the model, the estimate of β given by

$$b = (X^T X)^{-1} X^T Y$$

minimizes the sum of squares of the error $(\bar{\epsilon}^T \bar{\epsilon})$ between observed and theoretical data, and has minimum variance among all unbiased estimators of β . However, when bias is present, i.e., the model is not the true model (always the case with gravity and magnetic modeling), the estimate

$$b = (X^T X + \sigma^2)^{-1} X^T Y$$

has minimum variance plus (bias)² among all linear estimators (Shuey, 1977). σ^2 is the variance about the regression based on P degrees of freedom where

$$P = (\text{number of observations} - \text{number of parameters})$$

The variance is estimated by the sample variance SSQ given by equation 19.

The variance-covariance matrix of the parameters, in terms of the Marquardt inversion (equation 20), can be expressed as

$$(21) \quad \text{COV}(b) = \sum_{i=1}^m \frac{\sigma^2}{\lambda^2 + \sigma^2} V_i V_i^T$$

The variance-covariance matrix provides estimates for the variance or scatter of the parameters about their expected values. Parameter standard deviations are obtained from the square root of the diagonal terms of the variance-covariance. The parameter standard deviations represent a confidence region for which parameter values can be expected to be found for a given standard deviation in the data (SSQ).

The parameter correlation matrix can be obtained from the covariance matrix as

$$(22) \quad \text{COR}_{ij}(b) = \frac{\text{COV}_{ij}(\bar{b})}{\sqrt{\text{COV}_{ii}(\bar{b})} \sqrt{\text{COV}_{jj}(\bar{b})}}$$

The correlation matrix gives an indication of how independent each of the parameter solutions is with respect to the other parameters. For example, suppose one were modeling an anomaly over a single graben structure. The same anomaly could be obtained either by decreasing the density contrast and increasing the depth of the graben, or increasing the density contrast and decreasing the depth. These parameters would have a high negative correlation (where correlation ranges from -1 to 1, -1 indicates high negative correlation, 0 indicates no correlation, and 1 indicates high positive correlation).

APPENDIX IV

SAMPLE TERMINAL SESSION #1

The following terminal session demonstrates the use of the program from initial model through the inversion process. It also demonstrates the INPUT, EDIT, EXECUTE, LIST, PLOT, and FMAIN functions. The terminal session is summarized in the following steps:

- STEP 1 - Station distances are stored in a file, IN-STATIONS. A theoretical model is stored in file IN-MODEL. A forward problem will be calculated using the station distances and theoretical model to obtain theoretical gravity data. This data will be used as sample field data in STEP 2.
- STEP 2 - Use theoretical data from STEP 1 as observed data in new problem. Perturb the theoretical model from STEP 1 to obtain an initial guess. Normally STEP 1 would not be done, but actual field data would be used and the initial guess would be an attempt to describe the earth cross section where the data was obtained.
- STEP 3 - Compute a forward problem with the initial guess and observed gravity values to find the error. Obtain a plot (Fig. 1).
- STEP 4 - Do a 1D direct search. Note the change in the error and get a plot of new model (Fig. 2).
- STEP 5 - Do an inversion. Note error and get a plot (Fig. 3).
- STEP 6 - Store the newest model and results on the merge file.

A composite of the three plots is included for easier comparison (Fig. 4).

OK, SLIST IN-MODEL

1
 16, -.5, 10., 10., 0
 -100.000, 0.000, 0, 0
 118.500, 0.000, 0, 0
 118.500, 0.010, 0, 0
 16.25, 0.01, 0, 0
 15.5, 2., 0, 0
 13.5, 2., 0, 0
 12.0, 4.75, 0, 0
 9.5, 5., 0, 0
 8.50, 2.92, 0, 0
 7.5, 3.0, 0, 0
 6.5, .125, 0, 0
 5., .25, 0, 0
 4.5, 1.42, 0, 0
 2.5, 1.5, 0, 0
 2., .01, 0, 0
 -100.0, .01, 0, 0
 -100.0, .00, 0, 0

 | file containing model |
 | data in format required |
by INPUT

OK, SLIST IN-STATIONS
THEORETICAL MODEL FROM SNOW

1
 1
 90
 0., 0.0, 0.
 0., 0.2, 0.
 0., 0.4, 0.
 0., 0.6, 0.
 0., 0.8, 0.
 0., 1.0, 0.
 0., 1.2, 0.
 0., 1.4, 0.
 0., 1.6, 0.
 0., 1.8, 0.
 0., 2.0, 0.
 0., 2.2, 0.
 0., 2.4, 0.
 0., 2.6, 0.
 0., 2.8, 0.
 0., 3.0, 0.
 0., 3.2, 0.
 0., 3.4, 0.
 0., 3.6, 0.
 0., 3.8, 0.
 0., 4.0, 0.
 0., 4.2, 0.
 0., 4.4, 0.
 0., 4.6, 0.
 0., 4.8, 0.
 0., 5.0, 0.
 0., 5.2, 0.
 0., 5.4, 0.

 | file containing station |
 | data in format required |
 | by INPUT-observed gravity |
values are zeros

0., 5. 6, 0.
0., 5. 8, 0.
0., 6. 0, 0.
0., 6. 2, 0.
0., 6. 4, 0.
0., 6. 6, 0.
0., 6. 8, 0.
0., 7. 0, 0.
0., 7. 2, 0.
0., 7. 4, 0.
0., 7. 6, 0.
0., 7. 8, 0.
0., 8. 0, 0.
0., 8. 2, 0.
0., 8. 4, 0.
0., 8. 6, 0.
0., 8. 8, 0.
0., 9. 0, 0.
0., 9. 2, 0.
0., 9. 4, 0.
0., 9. 6, 0.
0., 9. 8, 0.
0., 10. 0, 0.
0., 10. 2, 0.
0., 10. 4, 0.
0., 10. 6, 0.
0., 10. 8, 0.
0., 11. 0, 0.
0., 11. 2, 0.
0., 11. 4, 0.
0., 11. 6, 0.
0., 11. 8, 0.
0., 12. 0, 0.
0., 12. 2, 0.
0., 12. 4, 0.
0., 12. 6, 0.
0., 12. 8, 0.
0., 13. 0, 0.
0., 13. 2, 0.
0., 13. 4, 0.
0., 13. 6, 0.
0., 13. 8, 0.
0., 14. 0, 0.
0., 14. 2, 0.
0., 14. 4, 0.
0., 14. 6, 0.
0., 14. 8, 0.
0., 15. 0, 0.
0., 15. 2, 0.
0., 15. 4, 0.
0., 15. 6, 0.
0., 15. 8, 0.
0., 16. 0, 0.
0., 16. 2, 0.

0., 16. 4, 0.
 0., 16. 6, 0.
 0., 16. 8, 0.
 0., 17. 0, 0.
 0., 17. 2, 0.
 0., 17. 4, 0.
 0., 17. 6, 0.
 0., 17. 8, 0.
 0., 18. 0, 0.

OK, GRAV2D

TERMINL-OPTION # (I)= 0
 1 HARD COPY 80 COLUMN
 2 TEKTRONIX 4014
 3 CRT 80 COLUMN

 | enter program and input |
 | 0 to obtain list of |
possible options

TERMINL-OPTION # (I)= 3
 *** GRAV2D MASTER ***
 MASTER--OPTION # (I)= 2
 INPUT WORK FILE HEADER (80 CHAR.)
 EXAMPLE OF TERMINAL SESSION
 DO YOU WANT TO INPUT STATION DATA FROM FILE?
 YES
 INPUT FILENAME (16 CHAR)
 IN-STATIONS
 INPUT PROFILE ID.
 INPUT CODE FOR UNITS: 1 = KILOMETERS
 2 = METERS
 3 = KILOFEET
 INPUT STATION NO. OR MATCH PT.
 INPUT NO. OF STATIONS
 INPUT OBS. GRAVITY, DIST., ELEV. FOR 90 STATIONS
 DO YOU WANT TO INPUT MODEL DATA FROM FILE?
 YES
 INPUT FILENAME (16 CHAR)
 IN-MODEL
 INPUT NO. OF POLYGONS
 FOR POLYGON 1, INPUT:
 # OF SIDES(I), DENSITY CONTRAST, STRIKE OUT, STRIKE IN, DENSITY SEARCH #(I)
 FOR 17 VERTICES, INPUT:
 HORZ. COORD., VERT. COORD., VERT. SEARCH #(I), HORZ. SEARCH #(I)
 DO YOU WANT TO INPUT SEARCH DATA FROM FILE?
 NO
 INPUT # ITERATIONS FOR INVERSION
 0
 INPUT TOTAL # OF VERTEX SEARCH DIRECTIONS
 0
 INPUT TOTAL # OF DENSITY CONTRAST SEARCH DIRS.
 0
 INPUT HOR. SCALE, VER. SCALE(CURVES), VER. SCALE(MODEL)
 3.
 30.
 1.75

MASTER--OPTION # (I)= 5
 XQT-----OPTION # (I)= 0
 1 RETURN TO MASTER
 2 XQT SEARCH
 3 XQT INVERSION

 : FORWARD PROBLEM :
 : STEP 1 :

XQT-----OPTION # (I)= 2
 OUTPUT--OPTION # (I)= 1
 THE TOTAL SUM OF SQUARES (SSR) = 104375.641
 0 SEARCHES LEFT

XQT-----OPTION # (I)= 1
 MASTER--OPTION # (I)= 4
 LIST----OPTION # (I)= 2

WORK FILE HEADER:

EXAMPLE OF TERMINAL SESSION

DATE: 07/02/80 15:12

THE TOTAL SUM OF SQUARES IS 104375.641

LIST----OPTION # (I)= 3

* STATION DATA *

THEORETICAL MODEL FROM SNOW

UNITS OF DISTANCE IS KILOM

STATION # OR MATCH PT IS 1

THE TOTAL SUM OF SQUARES IS 104375.641

INPUT RANGE OF STATIONS TO BE PRINTED

AS START INDEX, END INDEX

0,0 WILL PRINT ALL

0,0

Type <CR> to continue

STATION	DISTANCE TO STATION	ELEVATION	COMPUTED GRAVITY	OBSERVED GRAVITY	DIFFERENCE OBS-COMP
1	0.000	0.000	0.000	0.000	0.000
2	0.200	0.000	-0.294	0.000	0.294
3	0.400	0.000	-0.633	0.000	0.633
4	0.600	0.000	-1.027	0.000	1.029
5	0.800	0.000	-1.497	0.000	1.497
6	1.000	0.000	-2.067	0.000	2.067
7	1.200	0.000	-2.770	0.000	2.770
8	1.400	0.000	-3.664	0.000	3.664
9	1.600	0.000	-4.854	0.000	4.854
10	1.800	0.000	-6.562	0.000	6.562
11	2.000	0.000	-9.789	0.000	9.789
12	2.200	0.000	-14.024	0.000	14.024
13	2.400	0.000	-16.657	0.000	16.657
14	2.600	0.000	-18.590	0.000	18.590
15	2.800	0.000	-20.056	0.000	20.056
16	3.000	0.000	-21.173	0.000	21.173
17	3.200	0.000	-22.015	0.000	22.015
18	3.400	0.000	-22.627	0.000	22.627

Type <CR> to continue

19	3.600	0.000	-23.038	0.000	23.038
20	3.800	0.000	-23.256	0.000	23.256
21	4.000	0.000	-23.277	0.000	23.277
22	4.200	0.000	-23.081	0.000	23.081
23	4.400	0.000	-22.632	0.000	22.632

24	4.600	0.000	-21.883	0.000	21.883
25	4.800	0.000	-20.796	0.000	20.796
26	5.000	0.000	-19.474	0.000	19.474
27	5.200	0.000	-18.411	0.000	18.411
28	5.400	0.000	-17.937	0.000	17.937
29	5.600	0.000	-17.995	0.000	17.995
30	5.800	0.000	-18.517	0.000	18.517
31	6.000	0.000	-19.510	0.000	19.510
32	6.200	0.000	-21.074	0.000	21.074
33	6.400	0.000	-23.504	0.000	23.504
34	6.600	0.000	-27.095	0.000	27.095
35	6.800	0.000	-30.578	0.000	30.578
36	7.000	0.000	-33.522	0.000	33.522
37	7.200	0.000	-36.059	0.000	36.059
38	7.400	0.000	-38.283	0.000	38.283

Type<CR> to continue

39	7.600	0.000	-40.252	0.000	40.252
40	7.800	0.000	-42.009	0.000	42.009
41	8.000	0.000	-43.582	0.000	43.582
42	8.200	0.000	-44.992	0.000	44.992
43	8.400	0.000	-46.256	0.000	46.256
44	8.600	0.000	-47.387	0.000	47.387
45	8.800	0.000	-48.395	0.000	48.395
46	9.000	0.000	-49.288	0.000	49.288
47	9.200	0.000	-50.068	0.000	50.068
48	9.400	0.000	-50.743	0.000	50.743
49	9.600	0.000	-51.315	0.000	51.315
50	9.800	0.000	-51.786	0.000	51.786
51	10.000	0.000	-52.156	0.000	52.156
52	10.200	0.000	-52.428	0.000	52.428
53	10.400	0.000	-52.603	0.000	52.603
54	10.600	0.000	-52.680	0.000	52.680
55	10.800	0.000	-52.660	0.000	52.660
56	11.000	0.000	-52.545	0.000	52.545
57	11.200	0.000	-52.335	0.000	52.335
58	11.400	0.000	-52.031	0.000	52.031

Type<CR> to continue

59	11.600	0.000	-51.634	0.000	51.634
60	11.800	0.000	-51.147	0.000	51.147
61	12.000	0.000	-50.572	0.000	50.572
62	12.200	0.000	-49.912	0.000	49.912
63	12.400	0.000	-49.172	0.000	49.172
64	12.600	0.000	-48.355	0.000	48.355
65	12.800	0.000	-47.468	0.000	47.468
66	13.000	0.000	-46.517	0.000	46.517
67	13.200	0.000	-45.509	0.000	45.509
68	13.400	0.000	-44.448	0.000	44.448
69	13.600	0.000	-43.341	0.000	43.341
70	13.800	0.000	-42.188	0.000	42.188
71	14.000	0.000	-40.991	0.000	40.991
72	14.200	0.000	-39.742	0.000	39.742
73	14.400	0.000	-38.433	0.000	38.433
74	14.600	0.000	-37.047	0.000	37.047
75	14.800	0.000	-35.564	0.000	35.564

76	15.000	0.000	-33.955	0.000	33.955
77	15.200	0.000	-32.181	0.000	32.181
78	15.400	0.000	-30.176	0.000	30.176
Type<CR> to continue					
79	15.600	0.000	-27.931	0.000	27.931
80	15.800	0.000	-25.286	0.000	25.286
81	16.000	0.000	-22.079	0.000	22.079
82	16.200	0.000	-17.785	0.000	17.785
83	16.400	0.000	-13.204	0.000	13.204
84	16.600	0.000	-10.795	0.000	10.795
85	16.800	0.000	-9.042	0.000	9.042
86	17.000	0.000	-7.663	0.000	7.663
87	17.200	0.000	-6.535	0.000	6.535
88	17.400	0.000	-5.591	0.000	5.591
89	17.600	0.000	-4.785	0.000	4.785
90	17.800	0.000	-4.090	0.000	4.090

LIST----OPTION # (I)= 4

* MODEL DATA *

POLYGON NO. 1 DENSITY = -0.50000

STRIKE LENGTH 1 = 10.00000

STRIKE LENGTH 2 = 10.00000

SEARCH NO. = 0

X VERTICE	Z VERTICE	VERT. SEARCH NO.	HOR. SEARCH NO.
-100.000	0.000	0	0
118.500	0.000	0	0
118.500	0.010	0	0
16.250	0.010	0	0
15.500	2.000	0	0
13.500	2.000	0	0
12.000	4.750	0	0
9.500	5.000	0	0
8.500	2.920	0	0
7.500	3.000	0	0
6.500	0.125	0	0
5.000	0.250	0	0
4.500	1.420	0	0
2.500	1.500	0	0
2.000	0.010	0	0

Type <CR> to continue

-100.000	0.010	0	0
-100.000	0.000	0	0

Type <CR> to continue

LIST----OPTION # (I)= 5

* SEARCH DATA *

NO. OF ITERATIONS FOR INVERSION IS 0

TOTAL NO. OF VERTEX SEARCH DIRECTIONS IS 0

TOTAL NO. OF DENSITY CONTRAST SEARCH DIRECTIONS IS 0

LIST----OPTION # (I)= 6

* PLOT DATA *

HORIZONTAL SCALE (UNITS/INCH) = 3.00000

VERTICAL SCALE FOR CURVES (MILLIGALS/INCH) = 30.00000

VERTICAL SCALE FOR MODEL (UNITS/INCH) = 1.75000

LIST----OPTION # (I)= 1

MASTER--OPTION # (I)= 3

EDIT-----OPTION # (I): 3

```
ESTAT---OPTION # (I)= 0
```

- ```

1 RETURN TO EDIT OPTIONS
2 EDIT PROFILE ID.
3 EDIT UNITS OF DIST.
4 EDIT STATION #/MATCH PT.
5 EDIT NO. OF STATIONS
6 EDIT DISTANCES
7 EDIT ELEVATIONS
8 EDIT OBS. GRAVITY VALUES

```

ESTAT---OPTION # (I)= 8

## CHANGE-OBSERVED GRAVITY

[illegible]

PARAMETER # (I)=-1

```
1 OLD = 0.000000E+01 NEW =
0.
```

```
2 OLD = 0.000000E-01 NEW =
- .294
```

```
3 OLD = 0.000000E-01 NEW =
- .633
```

```
4 OLD = 0.000000E-01 NEW =
-1.029
```

```
5 OLD = 0.000000E-01 NEW =
-1.499
```

```
6 OLD = 0.000000E-01 NEW =
-2.067
```

```
7 OLD = 0.000000E-01 NEW =
-2.77
```

```
8 OLD = 0.000000E-01 NEW =
-3.664
```

```
9 OLD = 0.000000E-01 NEW =
-4.854
```

```
10 OLD = 0.000000E+01 NEW =
-6.562
```

```
11 OLD = 0.000000E-01 NEW =
-9.789
```

1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 2679, 26

```

1 STEP 2
2 Change observed
3 values from O's
4 to previously
5 calculated theo-
6 retical values

```

# 12	OLD = 0.000000E-01	NEW =
-14.024		
# 13	OLD = 0.000000E-01	NEW =
-16.657		
# 14	OLD = 0.000000E-01	NEW =
-18.590		
# 15	OLD = 0.000000E-01	NEW =
-20.056		
# 16	OLD = 0.000000E-01	NEW =
-21.173		
# 17	OLD = 0.000000E-01	NEW =
-22.015		
# 18	OLD = 0.000000E-01	NEW =
-22.627		
# 19	OLD = 0.000000E-01	NEW =
-23.038		
# 20	OLD = 0.000000E-01	NEW =
-23.256		
# 21	OLD = 0.000000E-01	NEW =
-23.277		
# 22	OLD = 0.000000E-01	NEW =
-23.081		
# 23	OLD = 0.000000E-01	NEW =
-22.632		
# 24	OLD = 0.000000E-01	NEW =
-21.883		
# 25	OLD = 0.000000E-01	NEW =
-20.796		
# 26	OLD = 0.000000E-01	NEW =
-19.474		
# 27	OLD = 0.000000E-01	NEW =
-18.411		
# 28	OLD = 0.000000E-01	NEW =
-17.937		
# 29	OLD = 0.000000E-01	NEW =
-17.995		
# 30	OLD = 0.000000E-01	NEW =
-18.517		
# 31	OLD = 0.000000E-01	NEW =
-19.510		
# 32	OLD = 0.000000E-01	NEW =
-21.074		
# 33	OLD = 0.000000E-01	NEW =
-23.504		
# 34	OLD = 0.000000E-01	NEW =
-27.095		
# 35	OLD = 0.000000E-01	NEW =
-30.578		
# 36	OLD = 0.000000E-01	NEW =
-33.522		
# 37	OLD = 0.000000E-01	NEW =
-36.059		
# 38	OLD = 0.000000E-01	NEW =
-38.283		

39 OLD = 0.000000E-01 NEW =
-40.252
40 OLD = 0.000000E-01 NEW =
-42.009
41 OLD = 0.000000E-01 NEW =
-43.582
42 OLD = 0.000000E-01 NEW =
-44.992
43 OLD = 0.000000E-01 NEW =
-46.256
44 OLD = 0.000000E-01 NEW =
-47.387
45 OLD = 0.000000E-01 NEW =
-48.395
46 OLD = 0.000000E-01 NEW =
-49.288
47 OLD = 0.000000E-01 NEW =
-50.068
48 OLD = 0.000000E-01 NEW =
-50.743
49 OLD = 0.000000E-01 NEW =
-51.315
50 OLD = 0.000000E-01 NEW =
-51.786
51 OLD = 0.000000E-01 NEW =
-52.156
52 OLD = 0.000000E-01 NEW =
-52.428
53 OLD = 0.000000E-01 NEW =
-52.603
54 OLD = 0.000000E-01 NEW =
-52.680
55 OLD = 0.000000E-01 NEW =
-52.66
56 OLD = 0.000000E-01 NEW =
-52.545
57 OLD = 0.000000E-01 NEW =
-52.335
58 OLD = 0.000000E-01 NEW =
-52.031
59 OLD = 0.000000E-01 NEW =
-51.634
60 OLD = 0.000000E-01 NEW =
-51.147
61 OLD = 0.000000E-01 NEW =
-50.572
62 OLD = 0.000000E-01 NEW =
-49.912
63 OLD = 0.000000E-01 NEW =
-49.172
64 OLD = 0.000000E-01 NEW =
-48.355
65 OLD = 0.000000E-01 NEW =
-47.468

```

66 OLD = 0.000000E-01 NEW =
-46.517
67 OLD = 0.000000E-01 NEW =
-45.509
68 OLD = 0.000000E-01 NEW =
-44.448
69 OLD = 0.000000E-01 NEW =
-43.341
70 OLD = 0.000000E-01 NEW =
-42.188
71 OLD = 0.000000E-01 NEW =
-40.991
72 OLD = 0.000000E-01 NEW =
-39.742
73 OLD = 0.000000E-01 NEW =
-38.433
74 OLD = 0.000000E-01 NEW =
-37.047
75 OLD = 0.000000E-01 NEW =
-35.564
76 OLD = 0.000000E-01 NEW =
-33.955
77 OLD = 0.000000E-01 NEW =
-32.181
78 OLD = 0.000000E-01 NEW =
-30.196
79 OLD = 0.000000E-01 NEW =
-27.931
80 OLD = 0.000000E-01 NEW =
-25.286
81 OLD = 0.000000E-01 NEW =
-22.079
82 OLD = 0.000000E-01 NEW =
-17.785
83 OLD = 0.000000E-01 NEW =
-13.204
84 OLD = 0.000000E-01 NEW =
-10.795
85 OLD = 0.000000E-01 NEW =
-9.042
86 OLD = 0.000000E-01 NEW =
-7.663
87 OLD = 0.000000E-01 NEW =
-6.535
88 OLD = 0.000000E-01 NEW =
-5.591
89 OLD = 0.000000E-01 NEW =
-4.785
90 OLD = 0.000000E-01 NEW =
-4.090

```

ESTAT---OPTION # (I)= 1

EDIT---OPTION # (I)= 1

HAS THE WORK FILE CHANGED?

Y

MASTER--OPTION # (I)= 4

LIST----OPTION # (I)= 3

* STATION DATA *

THEORETICAL MODEL FROM SNOW

UNITS OF DISTANCE IS KILOM

STATION # OR MATCH PT IS 1

THE TOTAL SUM OF SQUARES IS 104375.641

INPUT RANGE OF STATIONS TO BE PRINTED

AS START INDEX, END INDEX

0,0 WILL PRINT ALL

0,0

Type <CR> to continue

STATION	DISTANCE TO STATION	ELEVATION	COMPUTED GRAVITY	OBSERVED GRAVITY	DIFFERENCE OBS-COMP
1	0.000	0.000	0.000	0.000	0.000
2	0.200	0.000	-0.294	-0.294	0.294
3	0.400	0.000	-0.633	-0.633	0.633
4	0.600	0.000	-1.027	-1.027	1.027
5	0.800	0.000	-1.499	-1.499	1.499
6	1.000	0.000	-2.067	-2.067	2.067
7	1.200	0.000	-2.770	-2.770	2.770
8	1.400	0.000	-3.664	-3.664	3.664
9	1.600	0.000	-4.854	-4.854	4.854
10	1.800	0.000	-6.562	-6.562	6.562
11	2.000	0.000	-9.789	-9.789	9.789
12	2.200	0.000	-14.024	-14.024	14.024
13	2.400	0.000	-16.657	-16.657	16.657
14	2.600	0.000	-18.590	-18.590	18.590
15	2.800	0.000	-20.056	-20.056	20.056
16	3.000	0.000	-21.173	-21.173	21.173
17	3.200	0.000	-22.015	-22.015	22.015
18	3.400	0.000	-22.627	-22.627	22.627

Type<CR> to continue

19	3.600	0.000	-23.038	-23.038	23.038
20	3.800	0.000	-23.256	-23.256	23.256
21	4.000	0.000	-23.277	-23.277	23.277
22	4.200	0.000	-23.081	-23.081	23.081
23	4.400	0.000	-22.632	-22.632	22.632
24	4.600	0.000	-21.883	-21.883	21.883
25	4.800	0.000	-20.796	-20.796	20.796
26	5.000	0.000	-19.474	-19.474	19.474
27	5.200	0.000	-18.411	-18.411	18.411
28	5.400	0.000	-17.937	-17.937	17.937
29	5.600	0.000	-17.995	-17.995	17.995
30	5.800	0.000	-18.517	-18.517	18.517
31	6.000	0.000	-19.510	-19.510	19.510
32	6.200	0.000	-21.074	-21.074	21.074
33	6.400	0.000	-23.504	-23.504	23.504
34	6.600	0.000	-27.095	-27.095	27.095
35	6.800	0.000	-30.578	-30.578	30.578
36	7.000	0.000	-33.522	-33.522	33.522
37	7.200	0.000	-36.059	-36.059	36.059
38	7.400	0.000	-38.283	-38.283	38.283

Type<CR> to continue

39	7.600	0.000	-40.252	-40.252	40.252
40	7.800	0.000	-42.009	-42.009	42.009
41	8.000	0.000	-43.582	-43.582	43.582
42	8.200	0.000	-44.992	-44.992	44.992
43	8.400	0.000	-46.256	-46.256	46.256
44	8.600	0.000	-47.387	-47.387	47.387
45	8.800	0.000	-48.395	-48.395	48.395
46	9.000	0.000	-49.288	-49.288	49.288
47	9.200	0.000	-50.068	-50.068	50.068
48	9.400	0.000	-50.743	-50.743	50.743
49	9.600	0.000	-51.315	-51.315	51.315
50	9.800	0.000	-51.786	-51.786	51.786
51	10.000	0.000	-52.156	-52.156	52.156
52	10.200	0.000	-52.428	-52.428	52.428
53	10.400	0.000	-52.603	-52.603	52.603
54	10.600	0.000	-52.680	-52.680	52.680
55	10.800	0.000	-52.660	-52.660	52.660
56	11.000	0.000	-52.545	-52.545	52.545
57	11.200	0.000	-52.335	-52.335	52.335
58	11.400	0.000	-52.031	-52.031	52.031
Type<CR> to continue					
59	11.600	0.000	-51.634	-51.634	51.634
60	11.800	0.000	-51.147	-51.147	51.147
61	12.000	0.000	-50.572	-50.572	50.572
62	12.200	0.000	-49.912	-49.912	49.912
63	12.400	0.000	-49.172	-49.172	49.172
64	12.600	0.000	-48.355	-48.355	48.355
65	12.800	0.000	-47.468	-47.468	47.468
66	13.000	0.000	-46.517	-46.517	46.517
67	13.200	0.000	-45.509	-45.509	45.509
68	13.400	0.000	-44.448	-44.448	44.448
69	13.600	0.000	-43.341	-43.341	43.341
70	13.800	0.000	-42.188	-42.188	42.188
71	14.000	0.000	-40.991	-40.991	40.991
72	14.200	0.000	-39.742	-39.742	39.742
73	14.400	0.000	-38.433	-38.433	38.433
74	14.600	0.000	-37.047	-37.047	37.047
75	14.800	0.000	-35.564	-35.564	35.564
76	15.000	0.000	-33.955	-33.955	33.955
77	15.200	0.000	-32.181	-32.181	32.181
78	15.400	0.000	-30.196	-30.196	30.196
Type<CR> to continue					
79	15.600	0.000	-27.931	-27.931	27.931
80	15.800	0.000	-25.286	-25.286	25.286
81	16.000	0.000	-22.079	-22.079	22.079
82	16.200	0.000	-17.785	-17.785	17.785
83	16.400	0.000	-13.204	-13.204	13.204
84	16.600	0.000	-10.795	-10.795	10.795
85	16.800	0.000	-9.042	-9.042	9.042
86	17.000	0.000	-7.663	-7.663	7.663
87	17.200	0.000	-6.535	-6.535	6.535
88	17.400	0.000	-5.591	-5.591	5.591
89	17.600	0.000	-4.785	-4.785	4.785
90	17.800	0.000	-4.090	-4.090	4.090


```

LIST-----OPTION # (I)= 1
MASTER--OPTION # (I)= 3
EDIT-----OPTION # (I)= 0
 1 RETURN TO MASTER OPTIONS
 2 EDIT WORK FILE HEADER
 3 EDIT STATION DATA
 4 EDIT MODEL DATA
 5 EDIT SEARCH/INVERSN DATA
 6 EDIT PLOT DATA

```

```

EDIT-----OPTION # (I)= 4
 EMODEL--OPTION # (I)= 0
 1 RETURN TO EDIT OPTIONS
 2 EDIT NO. OF POLYGON
 3 EDIT # OF SIDES OF POLY.
 4 EDIT DENSITY CONTRASTS
 5 EDIT STRIKE LENGTH OUT
 6 EDIT STRIKE LENGTH IN
 7 EDIT HOR. COORDINATES
 8 EDIT VERT. COORDINATES

```

```

 EMODEL--OPTION # (I)= 7 | change model |
CHANGE-HORZ. COORDINATES | to initial guess |

```

```

-1.000000E 02 1.185000E 02 1.185000E 02 1.625000E 01 1.550000E 01
 1.350000E 01 1.200000E 01 9.500000E 00 8.500000E 00 7.500000E 00
 6.500000E 00 5.000000E 00 4.500000E 00 2.500000E 00 2.000000E 00
-1.000000E 02-1.000000E 02

```

CHANGE, ADD, DELETE OR RETURN?CHANGE

PARAMETER # (I, J)=-1, -1

```

1 1 OLD =-1.000000E 02 NEW =
-100.
1 2 OLD = 1.185000E 02 NEW =
118.5
1 3 OLD = 1.185000E 02 NEW =
118.5
1 4 OLD = 1.625000E 01 NEW =
16.75
1 5 OLD = 1.550000E 01 NEW =
15.75
1 6 OLD = 1.350000E 01 NEW =
12.5
1 7 OLD = 1.200000E 01 NEW =
11.25
1 8 OLD = 9.500000E 00 NEW =
9.5
1 9 OLD = 8.500000E 00 NEW =
8.5
1 10 OLD = 7.500000E 00 NEW =
7.25
1 11 OLD = 6.500000E 00 NEW =
6.37
1 12 OLD = 5.000000E 00 NEW =
5.

```

```

1 13 OLD = 4.500000E 00 NEW =
4.
1 14 OLD = 2.500000E 00 NEW =
2.
1 15 OLD = 2.000000E 00 NEW =
1.5
1 16 OLD = -1.000000E 02 NEW =
-100.
1 17 OLD = -1.000000E 02 NEW =
-100.
-1.000000E 02 1.185000E 02 1.185000E 02 1.675000E 01 1.575000E 01
1.250000E 01 1.125000E 01 9.500000E 00 8.500000E 00 7.250000E 00
6.370000E 00 5.000000E 00 4.000000E 00 2.000000E 00 1.500000E 00
-1.000000E 02 -1.000000E 02
CHANGE, ADD, DELETE OR RETURN? RETURN
 EMODEL--OPTION # (I)= 8
CHANGE-VERT. COORDINATES
0.000000E-01 0.000000E-01 1.000000E-02 1.000000E-02 2.000000E 00
2.000000E 00 4.750000E 00 5.000000E 00 2.920000E 00 3.000000E 00
1.250000E-01 2.500000E-01 1.420000E 00 1.500000E 00 1.000000E-02
1.000000E-02 0.000000E-01
CHANGE, ADD, DELETE OR RETURN? CHANGE
PARAMETER # (I, J)=-1, -1
1 1 OLD = 0.000000E-01 NEW =
0.
1 2 OLD = 0.000000E-01 NEW =
0.
1 3 OLD = 1.000000E-02 NEW =
1 4 OLD = 1.000000E-02 NEW =
1 5 OLD = 2.000000E 00 NEW =
1.25
1 6 OLD = 2.000000E 00 NEW =
1.25
1 7 OLD = 4.750000E 00 NEW =
4.
1 8 OLD = 5.000000E 00 NEW =
4.
1 9 OLD = 2.920000E 00 NEW =
2.
1 10 OLD = 3.000000E 00 NEW =
2.
1 11 OLD = 1.250000E-01 NEW =
1 12 OLD = 2.500000E-01 NEW =
1 13 OLD = 1.420000E 00 NEW =
2.
1 14 OLD = 1.500000E 00 NEW =
2.
1 15 OLD = 1.000000E-02 NEW =
1 16 OLD = 1.000000E-02 NEW =
1 17 OLD = 0.000000E-01 NEW =
0.
0.000000E-01 0.000000E-01 1.000000E-02 1.000000E-02 1.250000E 00
1.250000E 00 4.000000E 00 4.000000E 00 2.000000E 00 2.000000E 00
1.250000E-01 2.500000E-01 2.000000E 00 2.000000E 00 1.000000E-02

```

```

1. 0000000E-02 0. 0000000E-01
CHANGE, ADD, DELETE OR RETURN?RETURN
 EMODEL--OPTION # (I)= 0
 1 RETURN TO EDIT OPTIONS
 2 EDIT NO. OF POLYGON
 3 EDIT # OF SIDES OF POLY.
 4 EDIT DENSITY CONTRASTS
 5 EDIT STRIKE LENGTH OUT
 6 EDIT STRIKE LENGTH IN
 7 EDIT HOR. COORDINATES
 8 EDIT VERT. COORDINATES

 EMODEL--OPTION # (I)= 1
EDIT----OPTION # (I)= 0
 1 RETURN TO MASTER OPTIONS
 2 EDIT WORK FILE HEADER
 3 EDIT STATION DATA
 4 EDIT MODEL DATA
 5 EDIT SEARCH/INVERSN DATA
 6 EDIT PLOT DATA

EDIT----OPTION # (I)= 5
 ESEARCH-OPTION # (I)= 0
 1 RETURN TO EDIT OPTIONS
 2 EDIT # OF ITERATIONS
 3 EDIT # OF VERTEX SEARCH DIR.
 4 EDIT # OF DENSITY SEARCH DIR
 5 EDIT DENSITY SEARCH #S
 6 EDIT VERT. SEARCH #S(VERTEX)
 7 EDIT HORZ. SEARCH #S(VERTEX)
 8 EDIT TOLERANCES FOR SEARCH
 9 EDIT STEP SIZES FOR SEARCH

 ESEARCH-OPTION # (I)= 2
CHANGE--# ITERATIONS
OLD= 0 NEW=
4

 ESEARCH-OPTION # (I)= 3
CHANGE--# VERTX SEARCH DIR. S
OLD= 0 NEW=
7

 ESEARCH-OPTION # (I)= 6
CHANGE--VERT. SEARCH #S
 0 0 0 0 0
 0 0 0 0 0
 0 0 0 0 0
 0 0

CHANGE, ADD, DELETE OR RETURN?CHANGE
PARAMETER # (I, J)=-1, -1
1 1 OLD = 0 NEW =
0
1 2 OLD = 0 NEW =
0
1 3 OLD = 0 NEW =

```

```

0
1 4 OLD = 0 NEW =
0
1 5 OLD = 0 NEW =
6
1 6 OLD = 0 NEW =
6
1 7 OLD = 0 NEW =
3
1 8 OLD = 0 NEW =
3
1 9 OLD = 0 NEW =
3
1 10 OLD = 0 NEW =
3
1 11 OLD = 0 NEW =
0
1 12 OLD = 0 NEW =
0
1 13 OLD = 0 NEW =
1
1 14 OLD = 0 NEW =
1
1 15 OLD = 0 NEW =
0
1 16 OLD = 0 NEW =
0
1 17 OLD = 0 NEW =
0

```

```

0 0 0 0 6
6 3 3 3 3
0 0 1 1 0
0 0

```

CHANGE, ADD, DELETE OR RETURN?RETURN
 ESEARCH-OPTION # (I)= 7
 CHANGE-HORZ. SEARCH #S

```

0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0

```

CHANGE, ADD, DELETE OR RETURN?CHANGE
 PARAMETER # (I, J)=-1, -1

```

1 1 OLD = 0 NEW =
0
1 2 OLD = 0 NEW =
0
1 3 OLD = 0 NEW =
0
1 4 OLD = 0 NEW =
7
1 5 OLD = 0 NEW =
7
1 6 OLD = 0 NEW =
7

```

```

1 7 OLD = 0 NEW =
5
1 8 OLD = 0 NEW =
4
1 9 OLD = 0 NEW =
4
1 10 OLD = 0 NEW =
0
1 11 OLD = 0 NEW =
0
1 12 OLD = 0 NEW =
0
1 13 OLD = 0 NEW =
0
1 14 OLD = 0 NEW =
2
1 15 OLD = 0 NEW =
2
1 16 OLD = 0 NEW =
0
1 17 OLD = 0 NEW =
0

```

```

 0 0 0 7 7
 7 5 4 4 0
 0 0 0 2 2
 0 0

```

CHANGE, ADD, DELETE OR RETURN?CHANGE

PARAMETER # (I,J)=1,6

```

1 6 OLD = 7 NEW =
5

```

```

| change incorrectly |
typed number

```

```

 0 0 0 7 7
 5 5 4 4 0
 0 0 0 2 2
 0 0

```

CHANGE, ADD, DELETE OR RETURN?RETURN

ESEARCH-OPTION # (I)= 0

- 1 RETURN TO EDIT OPTIONS
- 2 EDIT # OF ITERATIONS
- 3 EDIT # OF VERTEX SEARCH DIR.
- 4 EDIT # OF DENSITY SEARCH DIR
- 5 EDIT DENSITY SEARCH #S
- 6 EDIT VERT. SEARCH #S(VERTEX)
- 7 EDIT HORZ. SEARCH #S(VERTEX)
- 8 EDIT TOLERANCES FOR SEARCH
- 9 EDIT STEP SIZES FOR SEARCH

ESEARCH-OPTION # (I)= 8

CHANGE-TOLERANCES

```

0.000000E-01 0.000000E-01 0.000000E-01 0.000000E-01 0.000000E-01
0.000000E-01 0.000000E-01

```

PARAMETER # (I)=-1

```

1 OLD = 0.000000E-01 NEW =
1.
2 OLD = 0.000000E-01 NEW =

```

1.
3 OLD = 0.000000E-01 NEW =
1.
4 OLD = 0.000000E-01 NEW =
1.
5 OLD = 0.000000E-01 NEW =
1.
6 OLD = 0.000000E-01 NEW =
1.
7 OLD = 0.000000E-01 NEW =
1.

ESEARCH-OPTION # (I)= 9

CHANGE-STEP SIZES

0.000000E-01 0.000000E-01 0.000000E-01 0.000000E-01 0.000000E-01
0.000000E-01 0.000000E-01

PARAMETER # (I)=-1

1 OLD = 0.000000E-01 NEW =
2 OLD = 0.000000E-01 NEW =
3 OLD = 0.000000E-01 NEW =
4 OLD = 0.000000E-01 NEW =
5 OLD = 0.000000E-01 NEW =
6 OLD = 0.000000E-01 NEW =
7 OLD = 0.000000E-01 NEW =

ESEARCH-OPTION # (I)= 1

EDIT----OPTION # (I)= 1

HAS THE WORK FILE CHANGED?

Y

MASTER--OPTION # (I)= 4

LIST----OPTION # (I)= 4

* MODEL DATA *

POLYGON NO. 1 DENSITY = -0.50000

STRIKE LENGTH 1 = 10.00000

STRIKE LENGTH 2 = 10.00000

SEARCH NO. = 0

X VERTICE	Z VERTICE	VERT. SEARCH NO.	HOR. SEARCH NO.
-100.000	0.000	0	0
118.500	0.000	0	0
118.500	0.010	0	0
16.750	0.010	0	7
15.750	1.250	6	7
12.500	1.250	6	5
11.250	4.000	3	5
9.500	4.000	3	4
8.500	2.000	3	4
7.250	2.000	3	0
6.370	0.125	0	0
5.000	0.250	0	0
4.000	2.000	1	0
2.000	2.000	1	2
1.500	0.010	0	2

Type <CR> to continue

-100.000	0.010	0	0
-100.000	0.000	0	0

Type <CR> to continue

LIST----OPTION # (I)= 5
 * SEARCH DATA *
 NO. OF ITERATIONS FOR INVERSION IS 4
 TOTAL NO. OF VERTEX SEARCH DIRECTIONS IS 7
 TOTAL NO. OF DENSITY CONTRAST SEARCH DIRECTIONS IS 0
 Type <CR> to continue

	TOLERANCE	STEP SIZE
1	1.00000	0.25000
2	1.00000	0.25000
3	1.00000	0.25000
4	1.00000	0.25000
5	1.00000	0.25000
6	1.00000	0.25000
7	1.00000	0.25000

LIST----OPTION # (I)= 1
 MASTER--OPTION # (I)= 3
 EDIT----OPTION # (I)= 5
 ESEARCH-OPTION # (I)= 3
 CHANGE-# VERTX SEARCH DIR. S
 OLD= 7 NEW=
 0

ESEARCH-OPTION # (I)= 1
 EDIT----OPTION # (I)= 1
 HAS THE WORK FILE CHANGED?

Y
 MASTER--OPTION # (I)= 5
 XQT-----OPTION # (I)= 2
 OUTPUT--OPTION # (I)= 1
 THE TOTAL SUM OF SQUARES (SSR) = 8283.779
 0 SEARCHES LEFT

XQT-----OPTION # (I)= 1
 MASTER--OPTION # (I)= 7
 PLOT----OPTION # (I)= 0
 1 RETURN TO MASTER OPTIONS
 2 PLOT CURRENT MODEL
 3 PLOT MULTIPLE SAVED MODELS

PLOT----OPTION # (I)= 2
 CURRENT MODEL PLOT
 DEVICE--OPTION # (I)= 0
 1 TEKTRONIX GRAPHICS PLOT
 2 STATOS PLOT
 3 UCC CALCOMP PLOT

DEVICE--OPTION # (I)= 2
 # OF VECTORS= 2364
 PLOT----OPTION # (I)= 1
 MASTER--OPTION # (I)= 3
 EDIT----OPTION # (I)= 5
 ESEARCH-OPTION # (I)= 3
 CHANGE-# VERTX SEARCH DIR. S
 OLD= 0 NEW=

POLYGON	DENSITY
NO.	CONTRAST
1	-0.50000

FIGURE 1 INITIAL GUESS

7

ESEARCH-OPTION # (I)= 1
 EDIT----OPTION # (I)= 1
 HAS THE WORK FILE CHANGED?

N

MASTER--OPTION # (I)= 5	----- ! STEP 4 ! -----
XQT-----OPTION # (I)= 2	
OUTPUT--OPTION # (I)= 1	
THE TOTAL SUM OF SQUARES (SSR) =	8283.779
0 SEARCHES LEFT	
THE TOTAL SUM OF SQUARES (SSR) =	6935.303
6 SEARCHES LEFT	
THE TOTAL SUM OF SQUARES (SSR) =	6008.625
5 SEARCHES LEFT	
THE TOTAL SUM OF SQUARES (SSR) =	2553.403
4 SEARCHES LEFT	
THE TOTAL SUM OF SQUARES (SSR) =	2258.706
3 SEARCHES LEFT	
THE TOTAL SUM OF SQUARES (SSR) =	992.981
2 SEARCHES LEFT	
THE TOTAL SUM OF SQUARES (SSR) =	1026.183
1 SEARCHES LEFT	
THE TOTAL SUM OF SQUARES (SSR) =	908.666
0 SEARCHES LEFT	

DO YOU WANT THIS SAVED?

YES

XQT-----OPTION # (I)= 1
 MASTER--OPTION # (I)= 7
 PLOT----OPTION # (I)= 2
 CURRENT MODEL PLOT
 DEVICE--OPTION # (I)= 2
 # OF VECTORS= 2362
 PLOT----OPTION # (I)= 1
 MASTER--OPTION # (I)= 5
 XQT-----OPTION # (I)= 0
 1 RETURN TO MASTER
 2 XQT SEARCH
 3 XQT INVERSION

XQT-----OPTION # (I)= 3	
OUTPUT--OPTION # (I)= 1	
THE TOTAL SUM OF SQUARES IS	908.664
4 ITERATIONS LEFT	
THE TOTAL SUM OF SQUARES IS	48.659
3 ITERATIONS LEFT	
THE TOTAL SUM OF SQUARES IS	24.613
2 ITERATIONS LEFT	
THE TOTAL SUM OF SQUARES IS	17.496
1 ITERATIONS LEFT	
THE TOTAL SUM OF SQUARES IS	17.164
0 ITERATIONS LEFT	

DO YOU WANT THIS SAVED?

YES

ORIGINAL = ○
NEW = ×

POLYGON NO.	DENSITY CONTRAST
1	-0.50000

FIGURE 2 GRAVITY MODEL AFTER SEARCH

```

XQT-----OPTION # (I)= 1
MASTER--OPTION # (I)= 6
FILE MAINTENANCE
FMAIN---OPTION # (I)= 2
TEST FILES FOR DEBUGGING G2HSER
*SUBFILE# 1
SOUTH LINE - HILL AIRFORCE BASE - WITH D.P.
HILL AIRFORCE BASE SOUTH LINE
*SUBFILE# 2
NORTH LINE - HILL AIRFORCE BASE - WITH D.P.
HILL AIRFORCE BASE NORTH LINE
*SUBFILE# 3
TEST OF RONS FILE
BALTAZOR L-3 1ST CUT
*SUBFILE# 4
EXAMPLE ONE
HILL AIRFORCE BASE SOUTH LINE
*SUBFILE# 5
TEST OF AUTOMATIC SEARCH ON THEORETICAL MODEL
THEORETICAL MODEL FROM SNOW
Type <CR> to continue
*SUBFILE# 6
TEST OF INVERSION ON THEORETICAL MODEL AFTER SEARCH
THEORETICAL MODEL FROM SNOW
*SUBFILE# 7
M2 INV
LINE M2 GRAV
FMAIN---OPTION # (I)= 0
1 RETURN TO MASTER PROGRAM
2 LIST MERGED FILE DIRECTORY
3 SAVE WORK FILE ON MERGED FILE
4 RESTORE WORK FILE FROM MERGED FILE
5 DELETE AND PACK MERGED FILE
6 INITIALIZE MERGE FILE
7 SEND MERGE FILE CONTENTS TO PRINTER

FMAIN---OPTION # (I)= 3
SAVE WORK FILE
WORK FILE SAVED AS SUBFILE# 8
FMAIN---OPTION # (I)= 1
MASTER--OPTION # (I)= 7
PLOT----OPTION # (I)= 2
CURRENT MODEL PLOT
DEVICE--OPTION # (I)= 2
OF VECTORS= 2293
PLOT----OPTION # (I)= 1
MASTER--OPTION # (I)= 1

```

POLYGON NO.	DENSITY CONTRAST
1	-0.50000

FIGURE 3 GRAVITY MODEL AFTER INVERSION

FIGURE 4 COMPARISON OF 3 PREVIOUS GRAVITY MODELS
WITH THEORETICAL MODEL

OK, T
 2'48 8'11 0'27
 OK, COMO -END

```

! Time since LOGIN !
! explained below !

```

Note Time since LOGIN

2 hrs. 48 min. of connect time
 8 min. 11 sec. of CPU time
 0 min. 27 sec. of disc I/O time

Approx. breakdown

STEP 3 (forward problem) = 17 sec. CPU time
 STEP 4 (search) = 185 sec. CPU time
 STEP 5 (inversion) = 160 sec. CPU time

APPENDIX V
SAMPLE TERMINAL SESSION #2

This terminal session uses a multiple-polygon
model.

OK, T
4'06 2'54 0'28
OK, SLIST STATIONS. SOUTH
HILL SOUTH LINE

1

26

150

-1.16, .15, 0.0
-1.90, .30, 0.0
-3.17, .61, 0.0
-4.56, .91, 0.0
-5.91, 1.22, 0.0
-6.40, 1.37, 0.0
-6.88, 1.52, 0.0
-7.83, 1.83, 0.0
-8.59, 2.13, 0.0
-9.01, 2.29, 0.0
-9.41, 2.44, 0.0
-10.22, 2.74, 0.0
-11.03, 3.05, 0.0
-11.42, 3.20, 0.0
-12.07, 3.50, 0.0
-12.50, 3.81, 0.0
-12.76, 4.12, 0.0
-12.81, 4.27, 0.0
-12.83, 4.57, 0.0
-12.71, 4.88, 0.0
-12.47, 5.18, 0.0
-12.22, 5.49, 0.0
-11.83, 5.79, 0.0
-11.49, 6.10, 0.0
-11.27, 6.40, 0.0
-10.99, 6.71, 0.0
-11.00, 7.01, 0.0
-10.96, 7.16, 0.0
-10.95, 7.31, 0.0
-11.01, 7.62, 0.0
-11.27, 7.93, 0.0
-11.51, 8.23, 0.0
-11.70, 8.53, 0.0
-11.82, 8.84, 0.0
-11.97, 8.99, 0.0
-11.91, 9.14, 0.0
-11.91, 9.30, 0.0
-11.82, 9.60, 0.0
-11.78, 9.75, 0.0
-11.75, 9.91, 0.0
-11.79, 10.21, 0.0
-11.89, 10.52, 0.0
-12.07, 10.67, 0.0
-12.49, 10.97, 0.0
-12.83, 11.12, 0.0
-13.34, 11.28, 0.0
-13.53, 11.43, 0.0

-14.38, 11.74, 0.0
-14.85, 11.89, 0.0
-15.31, 12.04, 0.0
-15.81, 12.19, 0.0
-16.35, 12.34, 0.0
-16.88, 12.50, 0.0
-17.22, 12.65, 0.0
-17.99, 12.80, 0.0
-18.63, 12.95, 0.0
-19.25, 13.11, 0.0
-19.91, 13.26, 0.0
-20.60, 13.41, 0.0
-21.26, 13.56, 0.0
-21.94, 13.72, 0.0
-22.59, 13.87, 0.0
-23.28, 14.02, 0.0
-23.96, 14.17, 0.0
-24.68, 14.33, 0.0
-25.30, 14.48, 0.0
-26.01, 14.63, 0.0
-26.84, 14.78, 0.0
-27.39, 14.93, 0.0
-28.02, 15.09, 0.0
-28.55, 15.24, 0.0
-29.15, 15.39, 0.0
-29.67, 15.54, 0.0
-30.23, 15.70, 0.0
-30.76, 15.85, 0.0
-31.18, 16.00, 0.0
-31.68, 16.15, 0.0
-32.06, 16.31, 0.0
-32.46, 16.46, 0.0
-32.83, 16.61, 0.0
-33.15, 16.76, 0.0
-33.47, 16.92, 0.0
-33.78, 17.07, 0.0
-34.00, 17.22, 0.0
-34.26, 17.37, 0.0
-34.48, 17.53, 0.0
-34.65, 17.68, 0.0
-34.78, 17.83, 0.0
-34.86, 17.98, 0.0
-34.97, 18.14, 0.0
-35.03, 18.29, 0.0
-35.03, 18.44, 0.0
-35.11, 18.59, 0.0
-35.08, 18.74, 0.0
-34.98, 18.90, 0.0
-34.94, 19.05, 0.0
-34.84, 19.20, 0.0
-34.77, 19.36, 0.0
-34.68, 19.51, 0.0
-34.54, 19.66, 0.0
-34.34, 19.81, 0.0

-34.23,19.96,0.0
-34.14,20.12,0.0
-33.96,20.27,0.0
-33.77,20.42,0.0
-33.45,20.57,0.0
-33.08,20.73,0.0
-32.74,20.88,0.0
-32.51,21.03,0.0
-32.16,21.28,0.0
-31.94,21.49,0.0
-31.67,21.64,0.0
-31.32,21.79,0.0
-30.27,22.25,0.0
-29.88,22.40,0.0
-29.47,22.56,0.0
-29.07,22.71,0.0
-28.68,22.86,0.0
-28.22,23.01,0.0
-27.68,23.16,0.0
-27.18,23.32,0.0
-26.70,23.47,0.0
-26.16,23.62,0.0
-25.57,23.77,0.0
-24.95,23.93,0.0
-24.32,24.08,0.0
-23.79,24.23,0.0
-23.29,24.38,0.0
-22.80,24.54,0.0
-22.20,24.69,0.0
-21.59,24.84,0.0
-21.00,24.99,0.0
-20.39,25.15,0.0
-19.62,25.30,0.0
-18.89,25.45,0.0
-18.29,25.61,0.0
-17.55,25.76,0.0
-16.82,25.92,0.0
-15.71,26.06,0.0
-14.73,26.18,0.0
-13.76,26.37,0.0
-13.35,26.52,0.0
-12.95,26.66,0.0
-10.08,27.58,0.0
-8.46,27.89,0.0
-5.78,28.13,0.0
-4.22,28.37,0.0
-3.67,28.54,0.0
-2.76,28.78,0.0

OK, SLIST MODEL3

5

23,-.50,10.,10.,17

-2000.,0.000,0.0

2000.,0.000,0.0

2000. . . 001, 0, 0
29. 200, . 001, 0, 0
28. 660, . 3500, 1, 51
28. 20, . 450, 0, 0
28. 000, . 550, 0, 0
27. 40, . 630, 0, 0
26. 8, . 32, 0, 0
24. 8, . 480, 0, 0
24. 6, . 175, 0, 0
20. 7, . 175, 0, 0
20. 0, . 75, 0, 0
16. 2, . 75, 0, 0
15. 7, . 15, 0, 0
13. 3, . 16, 0, 0
12. 5, . 41, 0, 0
-2000. . . 001, 0, 0
-2000. . . 000, 0, 0
29, -. 46, 10, . 10, . 18
12. 5, . 41, 0, 0
13. 3, . 160, 0, 0
15. 7, . 150, 0, 0
16. 2, . 75, 0, 0
20. 0, . 75, 0, 0
20. 7, . 175, 0, 0
24. 6, . 175, 0, 0
24. 8, . 480, 0, 0
26. 8, . 320, 0, 0
27. 400, . 630, 15, 16
26. 800, . 950, 13, 14
25. 60, . 950, 0, 0
20. 6, . 950, 0, 0
20. 2, 1. 30, 0, 0
16. 2, 1. 3, 0, 0
15. 8, . 85, 0, 0
15. 00, . 650, 0, 0
12. 80, . 650, 0, 0
11. 300, . 750, 11, 8
10. 850, . 750, 7, 8
9. 200, . 850, 7, 8
9. 000, . 950, 0, 0
8. 100, . 900, 0, 0
7. 700, . 780, 5, 6
6. 000, . 860, 5, 6
4. 400, 1. 18, 1, 4
2. 950, . 970, 1, 4
2. 150, . 750, 20, 30
16, -. 42, 10, . 10, . 19
11. 300, . 750, 11, 8
12. 80, . 650, 0, 0
15. 0, . 650, 0, 0
15. 80, . 85, 0, 0
16. 20, 1. 3, 0, 0
20. 2, 1. 3, 0, 0
20. 6, . 95, 0, 0

25. 600, . 950, 13, 14
 25. 200, 1. 55, 13, 14
 20. 8, 1. 55, 0, 0
 20. 4, 1. 75, 0, 0
 16. 4, 1. 75, 0, 0
 16. 0, 1. 45, 0, 0
 13. 500, 1. 350, 11, 12
 12. 8, . 90, 0, 0
 11. 30, . 900, 0, 0
 11. 30, . 750, 0, 0
 12, -. 39, 10. , 10. , 20
 13. 500, 1. 350, 11, 12
 16. 0, 1. 45, 0, 0
 16. 4, 1. 75, 0, 0
 20. 4, 1. 75, 0, 0
 20. 8, 1. 55, 0, 0
 25. 200, 1. 550, 13, 14
 23. 600, 1. 900, 13, 14
 21. 40, 2. 100, 0, 0
 21. 20, 2. 400, 0, 0
 16. 4, 2. 40, 0, 0
 16. 0, 2. 20, 0, 0
 14. 600, 2. 200, 11, 12
 13. 500, 1. 350, 11, 12
 9, -. 36, 10. , 10. , 21
 14. 6, 2. 20, 0, 0
 16. 0, 2. 20, 0, 0
 16. 400, 2. 400, 11, 12
 21. 20, 2. 400, 0, 0
 21. 40, 2. 10, 0, 0
 23. 600, 1. 900, 13, 14
 23. 500, 2. 3, 0, 0
 21. 300, 2. 800, 9, 10
 16. 600, 2. 900, 9, 10
 14. 600, 2. 200, 11, 12

OK, GRAV2D

TERMINL-OPTION # (I)= 3

*** GRAV2D MASTER ***

MASTER--OPTION # (I)= 2

INPUT WORK FILE HEADER (80 CHAR.)

EXAMPLE TWO

DO YOU WANT TO INPUT STATION DATA FROM FILE?

YES

INPUT FILENAME (16 CHAR)

STATIONS. SOUTH

INPUT PROFILE ID.

INPUT CODE FOR UNITS: 1 = KILOMETERS

2 = METERS

3 = KILOFEET

INPUT STATION NO. OR MATCH PT.

INPUT NO. OF STATIONS

INPUT OBS. GRAVITY, DIST., ELEV. FOR 150 STATIONS

DO YOU WANT TO INPUT MODEL DATA FROM FILE?

```

YES
INPUT FILENAME (16 CHAR)
MODEL3
INPUT NO. OF POLYGONS
FOR POLYGON 1 , INPUT:
OF SIDES(I), DENSITY CONTRAST, STRIKE OUT, STRIKE IN, DENSITY SEARCH #(I)
FOR 24 VERTICES, INPUT:
HORZ. COORD., VERT. COORD., VERT. SEARCH #(I), HORZ. SEARCH #(I)
FOR POLYGON 2 , INPUT:
OF SIDES(I), DENSITY CONTRAST, STRIKE OUT, STRIKE IN, DENSITY SEARCH #(I)
FOR 30 VERTICES, INPUT:
HORZ. COORD., VERT. COORD., VERT. SEARCH #(I), HORZ. SEARCH #(I)
FOR POLYGON 3 , INPUT:
OF SIDES(I), DENSITY CONTRAST, STRIKE OUT, STRIKE IN, DENSITY SEARCH #(I)
FOR 17 VERTICES, INPUT:
HORZ. COORD., VERT. COORD., VERT. SEARCH #(I), HORZ. SEARCH #(I)
FOR POLYGON 4 , INPUT:
OF SIDES(I), DENSITY CONTRAST, STRIKE OUT, STRIKE IN, DENSITY SEARCH #(I)
FOR 13 VERTICES, INPUT:
HORZ. COORD., VERT. COORD., VERT. SEARCH #(I), HORZ. SEARCH #(I)
FOR POLYGON 5 , INPUT:
OF SIDES(I), DENSITY CONTRAST, STRIKE OUT, STRIKE IN, DENSITY SEARCH #(I)
FOR 10 VERTICES, INPUT:
HORZ. COORD., VERT. COORD., VERT. SEARCH #(I), HORZ. SEARCH #(I)
DO YOU WANT TO INPUT SEARCH DATA FROM FILE?
NO
INPUT # ITERATIONS FOR INVERSION
0
INPUT TOTAL # OF VERTEX SEARCH DIRECTIONS
0
INPUT TOTAL # OF DENSITY CONTRAST SEARCH DIRS.
0
INPUT HOR. SCALE, VER. SCALE(CURVES), VER. SCALE(MODEL)
5., 17.5, 3.
MASTER--OPTION # (I)= 5
XGT-----OPTION # (I)= 2
OUTPUT--OPTION # (I)= 1
THE TOTAL SUM OF SQUARES (SSR) = 11.402
0 SEARCHES LEFT
XGT-----OPTION # (I)= 1
MASTER--OPTION # (I)= 7
PLOT-----OPTION # (I)= 2
CURRENT MODEL PLOT
DEVICE--OPTION # (I)= 2
OF VECTORS= 3412
PLOT-----OPTION # (I)= 1
MASTER--OPTION # (I)= 1

```

ORIGINAL = ○
NEW = ×

POLYGON NO.	DENSITY CONTRAST
1	-0.50000
2	-0.46000
3	-0.42000
4	-0.39000
5	-0.36000

FIGURE 5 GRAVITY MODEL

OK, T
4'12 5'12 0'45
OK, COMO -END