UNITED STATES DEPARTMENT OF THE INTERIOR

GEOLOGICAL SURVEY

Hypermag--An Interactive, Two-Dimensional Gravity and Magnetic Modeling Program

by

Richard W. Saltus

U.S. Geological Survey

Denver, CO 80225

Richard J. Blakely

U.S. Geological Survey

Menlo Park, CA 94025

Open-File Report 83-241

1983

This report is preliminary and has not been reviewed for conformity with U.S. Geological Survey editorial standards. Use of brand names in this report is for the sake of description only, and does not constitute endorsement by the USGS. Although this program has been tested, the USGS makes no guarantee of correct results.

Memorandum

Date March 24, 1983

To: Staff Production Controller, Geologic Div.

From: Chief, Office of Scientific Publications

Subject: New USGS open-file report

The following report was authorized by Henry Spall for the Director

on 3/8/83 for release in the open files: (9730-00916)

TITLE: Hypermag--an interactive, two-dimensional gravity and magnetic modeling program

AUTHOR(S): Richard W. Saltus and Richard J. Blakely

CONTENTS: 91 p., _____ over-size sheets (i.e., larger than 8½ x 11 inches)

Map scale: _____

Depositories:

(NC) USGS Library, Room 4A100, 12201 Sunrise Valley Dr., Reston, VA 22092
(Da) USGS Library, 1526 Cole Blvd. at West Colfax Ave., Golden, CO
(Mail address: Stop 914, Box 25046, Federal Center, Denver, CO 80225)
(M) USGS Library, 345 Middlefield Rd., Menlo Park, CA 94025

PRICE:	ORIGINATING OFFICE	
Fiche $ _____	Name: CTR (Stop 902), Denver, CO	Release date: APRIL 1983
Paper $ _____	Address: _____	Area: _____
	Telephone: FTS- 234-7367	OFR No. 83-241

Abstract

Program Hypermag is a Fortran program designed to facilitate construction of two-dimensional gravity or magnetic models. It features interactive and graphical features, is written in Fortran 77, and references only a few external subroutines. It is presently operational on the USGS Honeywell/Multics computer.

Contents

Introduction

Hypermag is a program for constructing two-dimensional gravity or magnetic models. It is interactive in nature and therefore requires little knowledge of the computer system and its editing facilities. Graphical input and output vastly reduces the amount of time normally spent by the user between calculations of edited models. In addition to its own interactively-acquired input, Hypermag will also accept input prepared for other USGS programs (e.g., 2mag , 2dgrav3, and saki). At time of writing, a copy of Hypermag was located at >online>Csc>RSaltus>hypermag on the USGS Multics computer in Denver.

This report describes Hypermag version 1.3 as of 1/24/83.

Theory

Hypermag uses standard approximations for the causative bodies: that they are infinitely extended in one horizontal direction and that their cross-sectional shape is polygonal. Gravity calculations are made within subroutine tgrav which uses the numerical technique of Talwani and others (1959). The gravity equations are from Grant and West (1965), page 289. Magnetic calculations are performed by subroutine ribbon2 using equations from Talwani and Heirtzler (1964).

The program will accept models with up to 50 bodies, each having up to 50 corners. The user may enter an observed profile to be plotted along with the calculated curve. The profile is limited to 1000 points. A linear regional trend may be removed from the observed if desired.

Program Usage

1. General

Program Hypermag is highly interactive; i.e., it leads the user through the model construction in a stepwise manner by a user-program conversation. Default answers are available for most of the questions, and are indicated within brackets at the end of a question. The default will be used if the user presses the return key without entering an answer. When specific responses are expected to a question, the choices are indicated in parentheses at the end of the question, with different responses delimited by a slash (/). For example, the question,

"Body command (a/c/d/h/q) [h]?"

indicates that one of the letters a, c, d, h, or q is expected as a user response and that the letter "h" is the default answer.

The program is organized in a simple structure (Figure 1). To get from a subsystem back to the next higher level, the user enters "q" or "//".

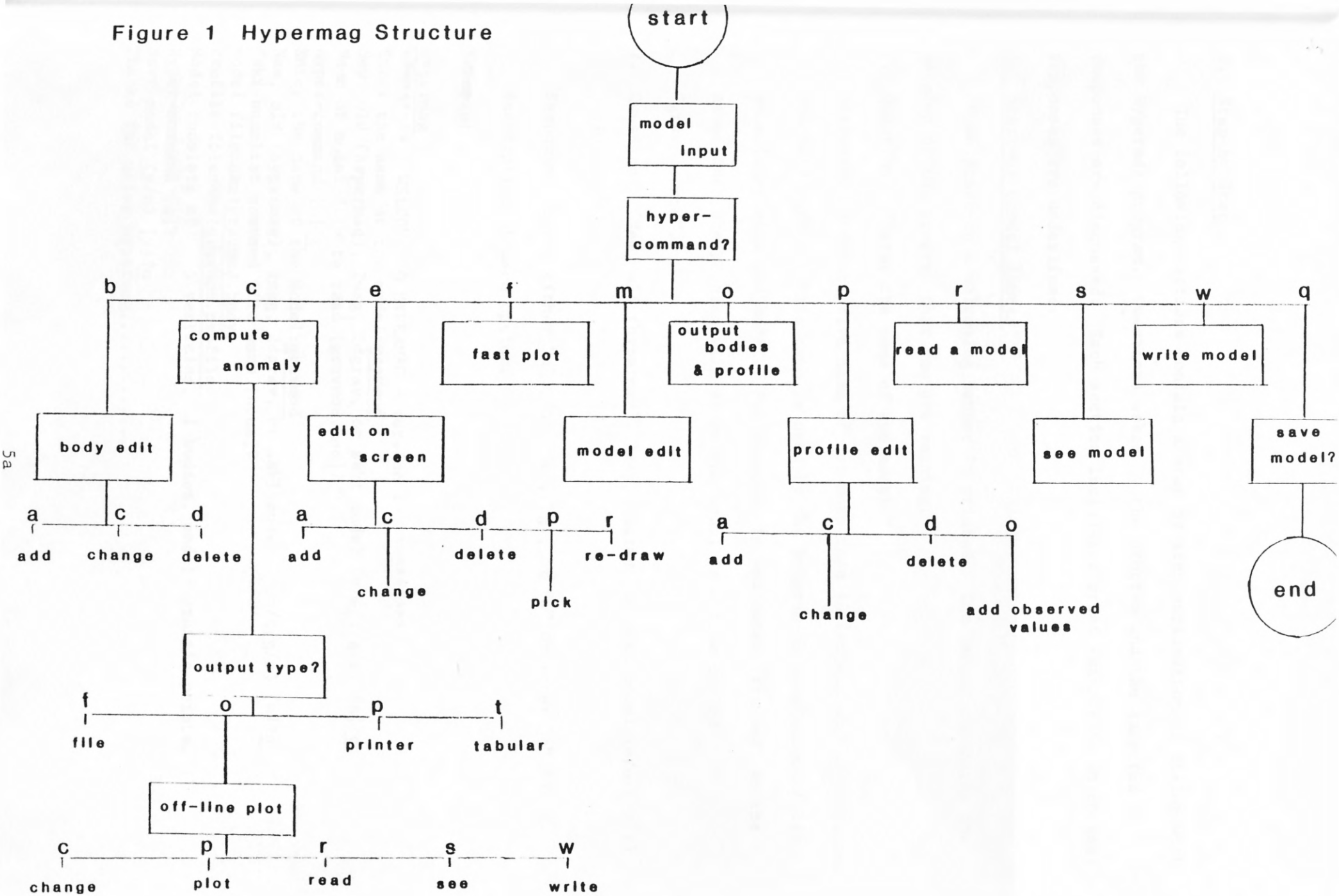
Whenever the letter "h" is indicated as a possible response, this means a short explanatory message is available (see the examples below). This help message will briefly describe the options available at that point.

When a list of items is being edited (e.g. in the profile editing routine, or the off-line plot parameters routine), a negative number indicates start with that value and continue, a positive number means edit that number only. For example, when the question

"Change point #(-1 for all, 0 for done) [-1]?"

appears, an answer of -3 starts with the 3rd point, then continues presenting points for change until the quit sequence ("//") is entered.

Figure 1 Hypermag Structure



2. Step by Step

The following sections contain a step by step explanation of dialog with the Hypermag program. Questions asked by the program and the expected responses are discussed. Each section includes a brief example in which user responses are underlined.

A. Start-up (model input)

Upon start-up a welcoming banner is printed. The banner indicates the version of the program that you are running.

1. Question: "Enter the name of the model:"

   Response: A descriptive name of not more than 40 letters with an optional ".model" suffix. This name is used by the program to name output files associated with the model. For example, if you chose "gravmod" as the name, then the program will store the model in a file called "gravmod.mod".

2. Question: "New, old (hypermag), 2mag, 2dgrav, or saki model (n/o/t/g/s) [n]?"

   Response: Enter either "n", "o", "t", "g", or "s" or enter "h" for a description of each option.

Example:

```
hypermag
*********   WELCOME TO HYPERMAG - version 1   *********
Enter the name of the model:gravmod
New, old (hypermag), 2mag, 2dgrav, or saki model (n/o/t/g/s) [n]?o
Name of model file to read [gravmod.mod]?
Hyper-command [h]?r
Enter the name of the model:gravmod
New, old (hypermag), 2mag, 2dgrav, or saki model (n/o/t/g/s) [n]?s
Saki namelist command file name [none]?
Model filename?gravmod.body
Profile filename?gravmod.profile
Model consists of    5 verticies,  1 bodies, and 13 profile points.
Hyper-command [q]?
Save model (y/n) [y]?n
Thanks for using hypermag.............
```

B. New model input

1. Q: "Gravity or magnetics (g/m) [m]?"

   R: Enter "g" if you are constructing a gravity model, or "m" for a magnetic model.

Note: The next 2 questions are not asked for gravity models.

2. Q: "Azimuth of profile = ?"

   R: The direction of the x axis (degrees).

3. Q: "Inclination and declination of the regional field = ?"

   R: Enter the regional inclination and declination in degrees.

4. Q: "Number of points in the profile = ?"

   R: Enter the number of points where you want the field calculated.

5. Q: "Horizontal (x-axis) units [2]?"

   R: Enter 1, 2, 3, 4, 5, or 6 to select the units for the x-axis. The default (2) is kilometers. See the example below for the other available units.

6. Q: "Vertical (z-axis) units [z]?"

   R: The units to be used on the z (vertical) axis, the choices are the same as for question 5 above.

7. Q: "Are the points of the profile evenly spaced and constant elevation?"

   R: Enter "yes" or "no". If you answer "yes", the system will ask you for the starting (x,z) coordinate and the distance between points. If you answer "no" you will be prompted for every (x,z) pair separately.

8. Q: "Are there observed anomalies?"

   R: Enter "yes" or "no". If you answer "yes", you will be prompted for the observed values for each profile point. If you enter observed values the next question will be asked.

9. Q: "Would you care to subtract a regional trend from these data (y/n)?"

   R: Enter "yes" or "no". If you answer "yes", you will be prompted for the starting value and slope of the linear regional to be removed.

Now you will be asked for information describing the bodies in your model. If you are constructing a gravity model, question 10 will be asked, if you are making a magnetic model, question 11 will be asked.

10. Q: "What is the density contrast for body 1?"

    R: Enter the density contrast in g/cc for this body. Go to question 12.

11. Q: "What is the inclination, declination, and intensity of magnetization for body 1?"

    R: Enter the inclination and declination of the body in degrees and the intensity of magnetization in emu/cc.

12. Q: "How many corners will it have?"

    R: Enter the number of corners for this body (Not to exceed 50). Then you will be queried for the (x,z) coordinate of each corner.

13. Q: "Another body?"

    R: Enter "yes" or "no". If you answer "yes", questions 10-13 will be asked for the next body. Otherwise you are done creating a new model.

Example:

```
hypermag
*********   WELCOME TO HYPERMAG - version 1   *********
Enter the name of the model:gravmod
New, old (hypermag), 2mag, 2dgrav, or saki model (n/o/t/g/s) [n]?n
Gravity or magnetics (g/m) [m]?g
Number of points in the profile = ?13
Unit codes: 1-meters, 2-kilometers, 3-feet, 4-kilofeet,
5-miles, 6-other
Horizontal (x-axis) units [2]?
Vertical (z-axis) units [2]?
Are the points of the profile evenly spaced and constant
elevation ? yes
Spacing between points = ? 1
The x and z coordinates of the first profile point = ? 2,0
Are there observed anomalies ? yes
Type the anomaly values, one at a time...
```

```
Point    1:  x =     2.000  z =     0.000  Anomaly = ? -265
Point    2:  x =     3.000  z =     0.000  Anomaly = ? -267
Point    3:  x =     4.000  z =     0.000  Anomaly = ? -267.5
Point    4:  x =     5.000  z =     0.000  Anomaly = ? -272.5
Point    5:  x =     6.000  z =     0.000  Anomaly = ? -275.7
Point    6:  x =     7.000  z =     0.000  Anomaly = ? -278
Point    7:  x =     8.000  z =     0.000  Anomaly = ? -280.2
Point    8:  x =     9.000  z =     0.000  Anomaly = ? -275
Point    9:  x =    10.000  z =     0.000  Anomaly = ? -271.2
Point   10:  x =    11.000  z =     0.000  Anomaly = ? -269
Point   11:  x =    12.000  z =     0.000  Anomaly = ? -266.7
Point   12:  x =    13.000  z =     0.000  Anomaly = ? -265.7
Point   13:  x =    14.000  z =     0.000  Anomaly = ? -265
Would you care to subtract a regional trend from these
data ? yes
The value of the regional at the first point and the slope
of the regional = ? -265,0
What is the density contrast for body   1 ?-.3
How many corners will it have ? 5
Type the x and z coordinates of each corner...
Corner   1:  x,z = ? 3.5,.1
Corner   2:  x,z = ? 11,.1
Corner   3:  x,z = ? 10,.7
Corner   4:  x,z = ? 9,1.2
Corner   5:  x,z = ? 8,2
Another body ? no
This model is ready
Hyper-command [h]?q
Save model (y/n) [y]?
Name of model file to write to [gravmod.mod]?
Thanks for using hypermag.............
```

C. Hyper-command level

1. Q: "Hyper-command [h]?"

   R: Enter the one-letter abbreviation of one of the hypermag subsystems.

   Or enter "h" to list the abbreviations. If you enter "q" the system will

   ask if you want to save the model before quitting.

Example:

```
hypermag
*********   WELCOME TO HYPERMAG - version 1    *********
Enter the name of the model:gravmod
New, old (hypermag), 2mag, 2dgrav, or saki model (n/o/t/g/s) [n]?o
Name of model file to read [gravmod.mod]?
Hyper-command [h]?h
Enter one of the following:
    b = edit bodies
    c = calculate anomaly
    e = edit the model using the 4112 screen
    f = fast re-calculation and plot.
    m = edit model parameters
    o = output generic model files (for program saki, etc.)
    p = edit profile
    r = read in a new model
    s = see model parameters
    w = write model to a file
    q = quit hypermag
Hyper-command [q]?q
Save model (y/n) [y]?n
Thanks for using hypermag.............
```

D. Edit bodies ("b")

1. Q: "Body command (a/c/d/h/q) [h]?"

   R: Enter the one-letter abbreviation of a body edit subsystem, or enter "h" to list the subsystems. If you answer "a" (add a new body), questions 10-13 from the model input section will be asked. If you answer "c" you will be asked the questions starting with question 2 of this section. If you answer "d" the system will request the number of the body to delete.

2. Q: "Change body # (0 when done) [-1]?"

   R: Enter the number of the body you wish to modify, or enter "0" to exit this subsystem.

3. Q: "Change corners or parameters (c/p/q) [c]?"

   R: Enter "c" or "p". If you enter "p" the intensity, inclination, declination, and density of the body will be presented for change. If you enter "c", question #4 will be asked.

4. Q: "Corner # [1]?"

   R: Enter the number of the corner to be modified, or enter "//" to exit the subsystem.

5. Q: "Add, change, or delete (a/c/d/q) [c]?"

   R: Enter "a", "c", "d", or "q". If you enter "a" you will be queried for the (x,z) coordinates of a corner to add to the body after the current corner. If you enter "c", the x and z coordinates of the current point will be presented for modification. If you enter "d" the current corner will be deleted.

Example:

```
hypermag
*********   WELCOME TO HYPERMAG - version 1   *********
Enter the name of the model:gravmod
New, old (hypermag), 2mag, 2dgrav, or saki model (n/o/t/g/s) [n]?o
Name of model file to read [gravmod.mod]?
Hyper-command [h]?b
```

```
Body command (a/c/d/h/q) [h]?
Body commands:
    a = add new body
    c = change existing body
    d = delete body
    q = quit the body area
Body command (a/c/d/h/q/s) [q]?a
What is the density contrast for body   2 ?-.2
How many corners will it have ? 3
Type the x and z coordinates of each corner...
Corner   1:  x,z = ? 3.5,.1
Corner   2:  x,z = ? 8,2
Corner   3:  x,z = ? 4,.7
Body command (a/c/d/h/q/s) [q]?c
Change body # (0 when done) [-1]?2
Change corners or parameters (c/p/q) [c]?p
Body magnetization [0.0000E+00]?
Inclination [62.000]?
Declination [16.000]?
Density contrast [-0.200]?-.25
Change body # (0 when done) [0]?2
Change corners or parameters (c/p/q) [c]?c
Corner # [1]?3
Corner # 3 x coord=     4.000, z coord=     0.700
Add, change, or delete (a/c/d/q) [c]?a
X coordinate [0.0000]?3
Z coordinate [0.0000]?.15
Corner # [4]?
Corner # 4 x coord=     3.000, z coord=     0.150
Add, change, or delete (a/c/d/q) [c]?c
X coordinate [3.000]?3.6
Z coordinate [0.150]?
Corner # [1]?//
Change body # (0 when done) [0]?
Body command (a/c/d/h/q/s) [q]?d
Delete body # (0 for done) [0]?2
body # 2 is to be deleted.
Are you sure (y/n) [n]?y
Body # 2 deleted.
Body command (a/c/d/h/q/s) [d]?q
Hyper-command [q]?
Save model (y/n) [y]?n
Thanks for using hypermag.............
```

E. Calculate anomaly and output results ("c")

1. Q: "Calculation type (g = grav, m = mag) [m]?"

   R: Enter "g" or "m".

The gravity or magnetic anomaly at each point in the profile is calculated here.

2. Q: "Output type (f/h/o/p/q/t) [f]?"

   R: Enter the one-letter abbreviation for the kind of output desired, or enter "help" to list the output types. See the examples for sample output. See appendix B for complete description of the off-line plot (type "o").

Example:

```
hypermag
*********   WELCOME TO HYPERMAG - version 1   *********
Enter the name of the model:gravmod
New, old (hypermag), 2mag, 2dgrav, or saki model (n/o/t/g/s) [n]?o
Name of model file to read [gravmod.mod]?
Hyper-command [h]?c
Calculation type (g=grav, m=mag) [g]?
Output type (f/h/o/p/q/t) [f]?h
Output types:
    f = print results at a file
    o = off-line plot
    p = printer plot
    t = print at terminal
    q = quit output section
Output type (f/h/o/p/q/t) [q]?f      (** see figure 2)
Output type (f/h/o/p/q/t) [q]?p
```

Plot of total attraction.
Plot symbols: O=Observed, X=Computed, *=Intersection.

```
  gamma   -1.52E+01 -1.29E+01 -1.05E+01 -8.18E+00 -5.85E+00 -3.51E+00 -1.17E+00
          +....+....+....+....+....+....+....+....+....+....+....+....+....+
 2.00E+00 +                                                            X  *
 2.40E+00 +                                                               -
 2.80E+00 +                                                               -
 3.20E+00 +                                                         O  X  -
 3.60E+00 +                                                               -
 4.00E+00 +                                          X       O            <
 4.40E+00 +                                                               -
 4.80E+00 +                                                               -
 5.20E+00 +                          X    O                               -
 5.60E+00 +                                                               -
 6.00E+00 +              X   O                                            <
 6.40E+00 +                                                               -
 6.80E+00 +                                                               -
 7.20E+00 +     X  O                                                      -
 7.60E+00 +                                                               -
 8.00E+00 +    X                                                          <
 8.40E+00 +                                                               -
 8.80E+00 +                                                               -
 9.20E+00 +            X        O                                         -
 9.60E+00 +                                                               -
 1.00E+01 +                           X        O                          <
 1.04E+01 +                                                               -
 1.08E+01 +                                                               -
 1.12E+01 +                                          O   X                -
 1.16E+01 +                                                               -
 1.20E+01 +                                                     O X       <
 1.24E+01 +                                                               -
 1.28E+01 +                                                               -
 1.32E+01 +                                                            *  -
 1.36E+01 +                                                               -
 1.40E+01 +                                                            X *
          +....+....+....+....+....+....+....+....+....+....+....+....+....+
  gamma   -1.52E+01 -1.29E+01 -1.05E+01 -8.18E+00 -5.85E+00 -3.51E+00 -1.17E+00
```

Output type (f/h/o/p/q/t) [q]?t
Results.....

Point	Calc	Obs	Calc-obs	Point	Calc	Obs	Calc-obs
1	-.766	0.000	-.766	2	-1.32	-2.00	.675
3	-4.52	-2.50	-2.02	4	-8.61	-7.50	-1.11
5	-11.7	-10.7	-1.04	6	-13.7	-13.0	-.701
7	-13.9	-15.2	1.25	8	-12.1	-10.0	-2.08
9	-8.43	-6.20	-2.23	10	-3.00	-4.00	.996
11	-1.24	-1.70	.459	12	-.759	-.700	-0.588E-01
13	-.520	0.000	-.520				

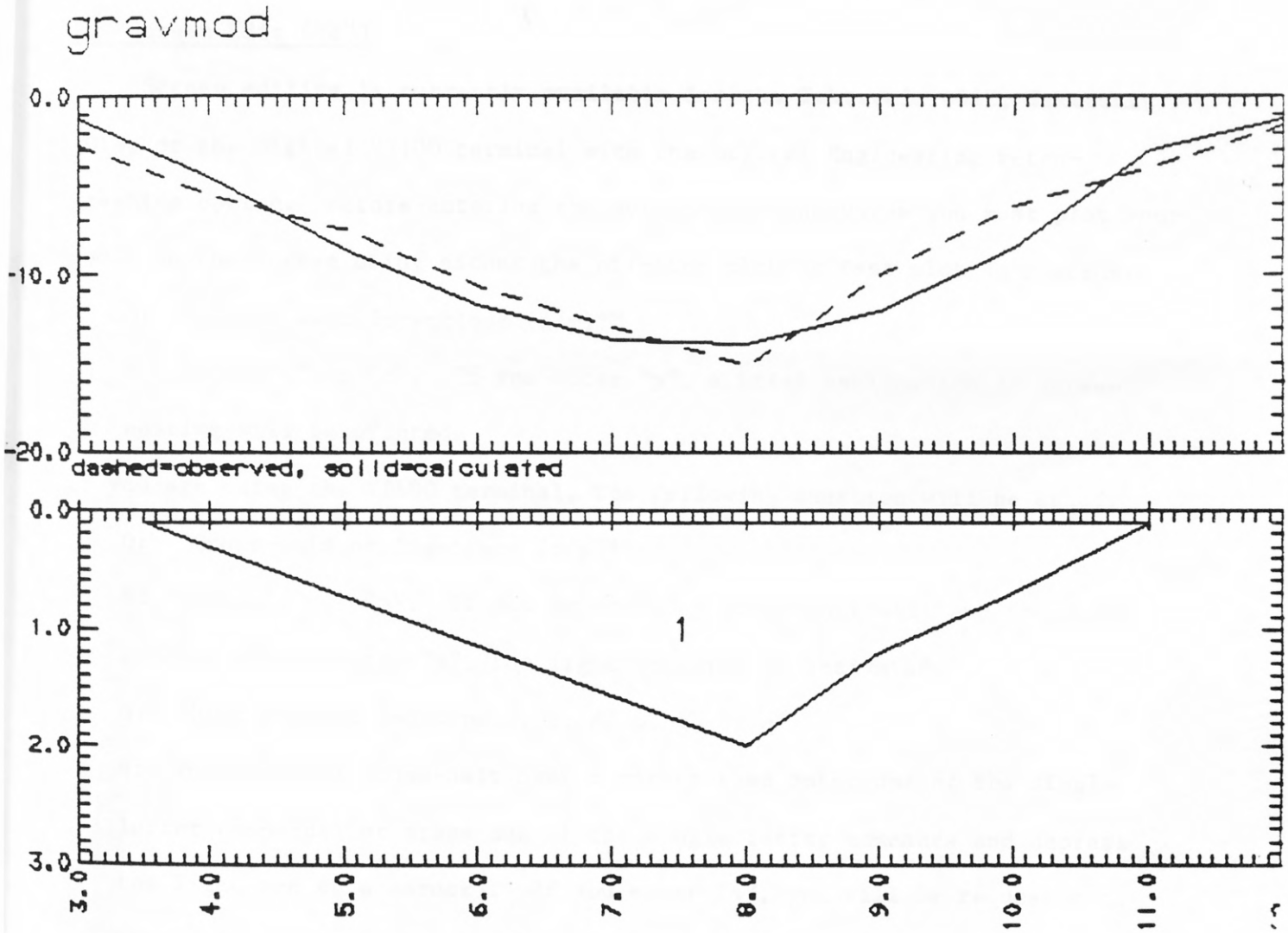
Output type (f/h/o/p/q/t) [q]?o
Plot command (s/c/h/r/w/p/q) [p]?h

```
Enter one of the following
     s= see plot parameters
     c= change plot parameters
     r= read plot parameters from file
     w= write plot parameters to file
     p= plot model
     q= quit plot system
plot command (s/c/h/r/w/p/q) [q]?p      (** see figure 3)
MAPPED - VECTOR

 DEFAULT VALUES OF ALL PARAMETERS WILL BE USED

                    *****PLOT OPTIONS IN EFFECT*****

MODEL  =       8242  ,XMIN   =       0.00  ,XMAX   =     100.00
YMIN   =       0.00  ,YMAX   =      81.90  ,MSGLVL =          1
XSTART =       0.00  ,YSTART =       0.00  ,SCALE  =       1.00
XFACT  =       1.00  ,YFACT  =       1.00  ,UNITS  =       1.00
STRIP  =      40.96  ,STRIP0 =       0.00  ,SPACE  =       6.00
NSCAN  =        382  ,OUT    =      -1.00  ,LYNES  =       5000

     0 VECTORS LOST
    10 ACTIVE LINES USED

Plot command (s/c/h/r/w/p/q) [q]?q
Output type (f/h/o/p/q/t) [q]?
Hyper-command [q]?
Save model (y/n) [y]?n
Thanks for using hypermag.............
```

```
Inclination of field        =    0.00
Declination of field        =    0.00
First value of regional     =   -265.00
Slope of regional           =    0.0000
X conversion to KM          =    1.0000
Z conversion to KM          =    1.0000
Body No.  1
Inclination =     0.00
Declination =     0.00
Intensity   =   0.00000
Density     =  -0.30000
Corner     X          Z

   1      3.500      0.100
   2     11.000      0.100
   3     10.000      0.700
   4      9.000      1.200
   5      8.000      2.000
Profile
Point     X          Z        Observed  Obs-reg
   1      2.000      0.000  -265.000      0.000
   2      3.000      0.000  -267.000     -2.000
   3      4.000      0.000  -267.500     -2.500
   4      5.000      0.000  -272.500     -7.500
   5      6.000      0.000  -275.700    -10.700
   6      7.000      0.000  -278.000    -13.000
   7      8.000      0.000  -280.200    -15.200
   8      9.000      0.000  -275.000    -10.000
   9     10.000      0.000  -271.200     -6.200
  10     11.000      0.000  -269.000     -4.000
  11     12.000      0.000  -266.700     -1.700
  12     13.000      0.000  -265.700     -0.700
  13     14.000      0.000  -265.000      0.000

Results......

Point Calc       Obs        Calc-obs Point Calc       Obs        Calc-obs
    1 -.766      0.000      -.766         2 -1.32      -2.00      .675
    3 -4.52      -2.50      -2.02         4 -8.61      -7.50      -1.11
    5 -11.7      -10.7      -1.04         6 -13.7      -13.0      -.701
    7 -13.9      -15.2      1.25          8 -12.1      -10.0      -2.08
    9 -8.43      -6.20      -2.23        10 -3.00      -4.00      .996
   11 -1.24      -1.70      .459         12 -.759      -.700     -0.588E-01
   13 -.520      0.000      -.520    1          -1.52E+01 -1.37E+01 -1.22E+01 -1.06E+01 -9.12E+00 -7.60E+00 -6.08E+00 -4.56E+00 -3.04E
+00 -1.52E+00  5.22E-08
   X VALUES  +....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+   X VALUES
   2.00E+00  +                                                                                                                +   2.00E+00
   3.00E+00  -                                                                                                                +   3.00E+00
   4.00E+00  -                                                                                                                +   4.00E+00
   5.00E+00  -                                                                                                                +   5.00E+00
   6.00E+00  -                                                                                                                +   6.00E+00
   7.00E+00  +                                                                                                                +   7.00E+00
   8.00E+00  *                                                                                                                +   8.00E+00
   9.00E+00  -                                                                                                                +   9.00E+00
   1.00E+01  -                                                                                                                +   1.00E+01
   1.10E+01  -                                                                                                                +   1.10E+01
   1.20E+01  +                                                                                                                +   1.20E+01
   1.30E+01  -                                                                                                                +   1.30E+01
   1.40E+01  -                                                                                                                +   1.40E+01
   X VALUES  +....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+   X VALUES
          -1.52E+01 -1.37E+01 -1.22E+01 -1.06E+01 -9.12E+00 -7.60E+00 -6.08E+00 -4.56E+00 -3.04E+00 -1.52E+00  5.22E-08
0  YSCALE = 1.5200E-01    XSCALE = 1.0000E+00
```

FIGURE 2 - Example of file ("f") output

FIGURE 3 - Example of off-line plot



F. Screen Edit ("e")

Screen editing is currently available for the Tektronix 4112, Tektronix 4014, or the Digital VT100 terminal with the Digital Engineering retro-graphics option. Before entering the screen edit subsystem you must plot your model on the screen using either the off-line plot or fast plot subsystems.

Q: "Do you want directions (n/y)?"

R: Enter "y" or "n". If you enter "y", a brief explanation of screen editing will be printed.

If you are using the VT100 terminal, the following question will be asked.

Q: "Cross-hair or light pen (c/p)?"

R: Enter "c" or "p". If you enter "c" a cross-hair will appear on the screen, if you enter "p", the light pen will be activated.

Q: "Pick a point and type a, c, d, p, r, or q"

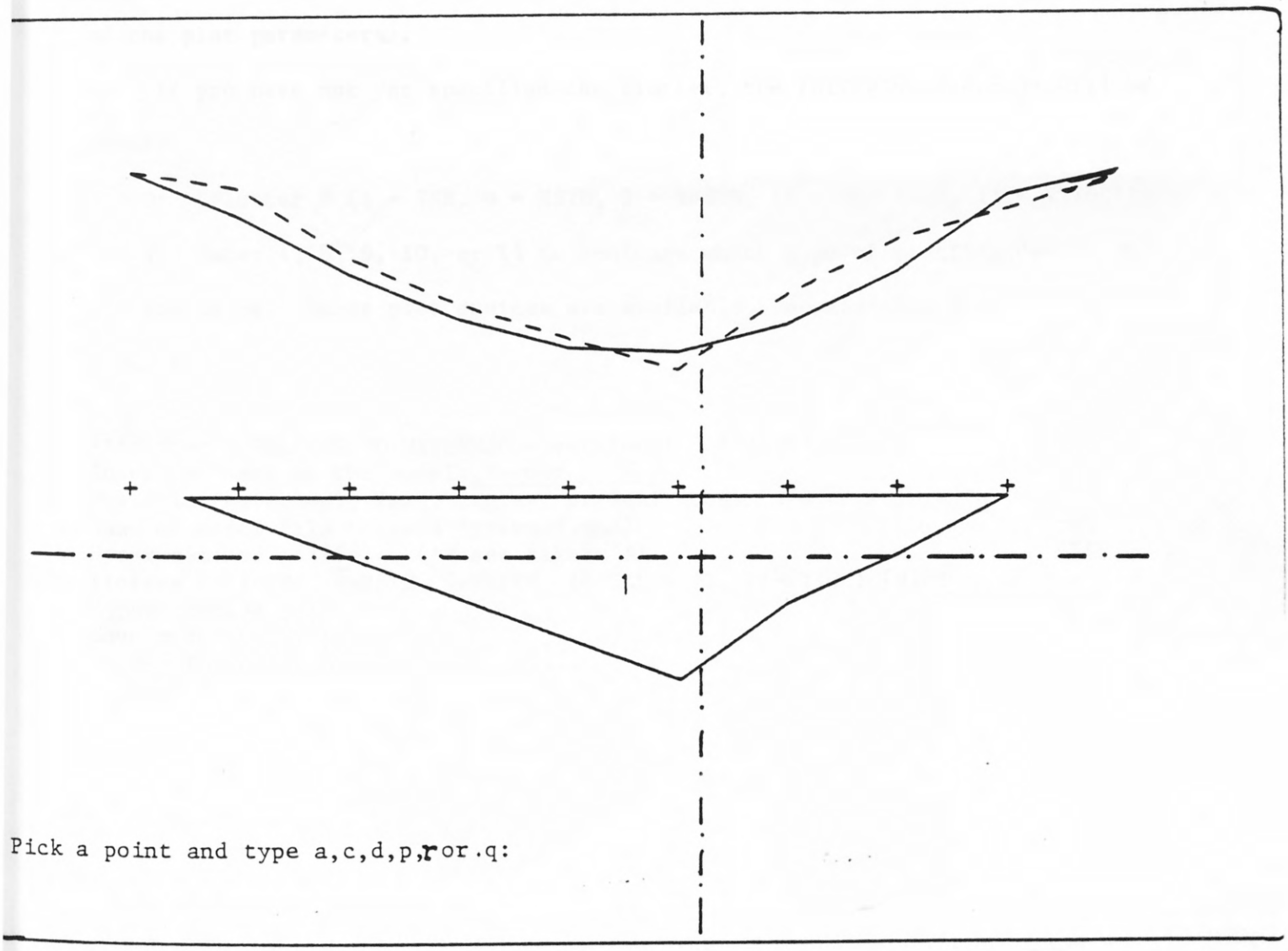
R: Position the cross-hair over a corner then enter one of the single letter commands (or press one of the single letter commands and depress the light pen at a corner). If you enter "a", you will be requested to show, with the crosshair or light pen, the location of the point to add after the corner you have specified. If you enter "c", you will be requested to show the new location of the corner. If you enter "d" the corner will be deleted. If you enter "p", you will be requested to type in the coordinates of the new point location. If you enter "r", the model will be re-calculated and re-drawn. Enter "q" to quit screen editing.

Example:

```
hypermag
*********   WELCOME TO HYPERMAG - version 1   *********
Enter the name of the model:gravmod
New, old (hypermag), 2mag, 2dgrav, or saki model (n/o/t/g/s) [n]?o
Name of model file to read [gravmod.mod]?
Hyper-command [h]?f
Plotter # (1=TKX, 8=HSTN, 9=VRSTK, 10=TKX 4112, 11=VT100) [9]?10
Hyper-command [q]?e
```

```
Do you want directions (n/y) [n]?y
Use the thumb-wheels to position the cross-hair over a
body corner.  Then type:
      p = to pick a point for manual editing
      a = to add a point after the specified point
      c = to change the location of a point
      d = to delete the point
      q = to quit the screen edit mode
      r = re-calc and re-draw of body
If you chose a or c, then you will be asked to pick the new
location with the cross-hair.  If more than one body shares
a corner then a new point will be requested for each match.
Pick a point and type a,c,d,p,r or q
p     (** see figure 4)
Sorry- I can't find any points close to the spot you picked.
Please try again.....
Pick a point and type a,c,d,p,r or q
q
Hyper-command [q]?
Save model (y/n) [y]?n
Thanks for using hypermag.............
```

FIGURE 4 - Screen editing



his figure shows the appearance of the terminal screen during screen editing. he causitive body is labeled with a 1, with the observed and calculated curves bove. The dash-dot lines represent the terminal's cross-hair (which is usually ade up of solid lines). Using the thumb-wheels or other cursor control, position he cross-hair at the corner you wish to edit. Then enter a, c, d, p, r, or q.

G. Fast re-calculation and plot ("f")

This subsystem performs the anomaly calculations and then plots the model using the current plot parameter settings. (See appendix B for a description of the plot parameters).

If you have not yet specified the plotter, the following question will be asked:

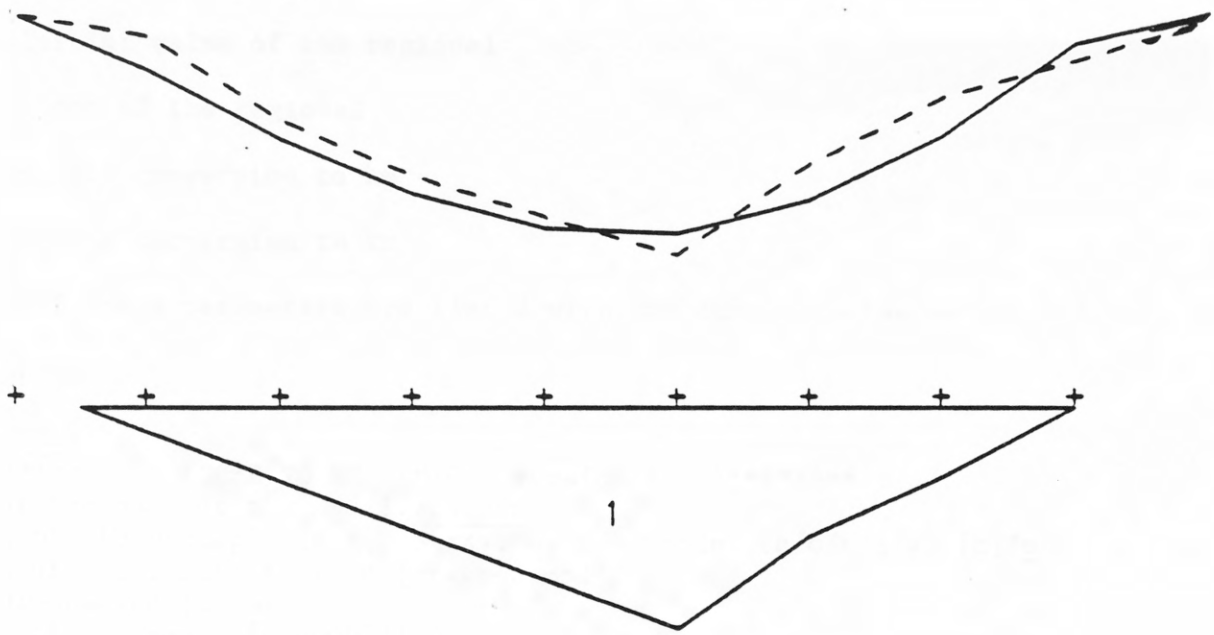
Q: "Plotter # (1 = TKX, 8 = HSTN, 9 = VRSTK, 10 = TKX 4112, 11 = VT100)?"

R: Enter 1, 8, 9, 10, or 11 to indicate which type of plotting device you are using. Other plot devices are available, see appendix B.

Example:

```
hypermag
*********   WELCOME TO HYPERMAG - version 1   *********
Enter the name of the model:gravmod
New, old (hypermag), 2mag, 2dgrav, or saki model (n/o/t/g/s) [n]?o
Name of model file to read [gravmod.mod]?
Hyper-command [h]?f      (** see figure 5)
Plotter # (1=TKX, 8=HSTN, 9=VRSTK, 10=TKX 4112, 11=VT100) [9]?1
Hyper-command [q]?
Save model (y/n) [y]?n
Thanks for using hypermag...........
```

FIGURE 5 - Example of fast plot ("f")



H. Edit model parameters ("m")

In this subsystem, the following parameters are made available for change:

Azimuth of the profile

Inclination of the field

Declination of the field

Initial value of the regional

Slope of the regional

x axis conversion to km

z axis conversion to km

Each of these parameters are listed with the current value as the default.

Example:

```
hypermag
*********   WELCOME TO HYPERMAG - version 1    *********
Enter the name of the model:gravmod
New, old (hypermag), 2mag, 2dgrav, or saki model (n/o/t/g/s) [n]?o
Name of model file to read [gravmod.mod]?
Hyper-command [h]?m
Azimuth (degrees) [0.0000]?180
Field inclination (degrees) [0.0000]?16
Field declination (degrees) [0.0000]?60
Initial regional value [-265.0000]?
Regional slope [0.0000]?
X axis conversion to KM [1.0000]?
Z axis conversion to KM [1.0000]?
Hyper-command [q]?
Save model (y/n) [y]?n
Thanks for using hypermag.............
```

I. Output generic model files ("o")

This subsystem writes out three output files. The first is a fortran namelist file which can be used to control the program saki by Mike Webring (unpublished program). The second is an ascii formatted file containing the description of the bodies in a general format (see appendix D for a complete description of the format). The third file is an ascii formatted file containing (x,z) locations (z positive down) and observed values for the profile. These files are sufficiently general to use as input for other programs you may need to develop. In addition, other programs supporting these formats are being developed at the USGS. To input files of this type to hypermag, use the "s" option of the input_model subsystem.

Example:

```
hypermag
*********   WELCOME TO HYPERMAG - version 1   *********
Enter the name of the model:gravmod
New, old (hypermag), 2mag, 2dgrav, or saki model (n/o/t/g/s) [n]?o
Name of model file to read [gravmod.mod]?
Hyper-command [h]?o
Created a namelist command file for saki called:
     gravmod.sak
Created a file containing body information called:
     gravmod.bod
Created a file containing profile locations and observed values called:
     gravmod.pfl
Output   5 vertices,  1 bodies, & 13 profile points.
Hyper-command [q]?
Save model (y/n) [y]?n
Thanks for using hypermag.............
print gravmod.bod

                    gravmod.bod          11/19/82  0943.7 mst Fri

   3.500  0.100   11.000  0.100   10.000  0.700    9.000  1.200    8.000  2.000
>>>*bodies*<<<
 -0.300E+00  0.000E+00  0.000E+00 <*1*>
   1   2   3   4   5 <*1*>
```

print gravmod.pfl

gravmod.pfl 11/19/82 0943.7 mst Fri

0.20000E+01	0.00000E+00	0.00000E+00
0.30000E+01	0.00000E+00	-0.20000E+01
0.40000E+01	0.00000E+00	-0.25000E+01
0.50000E+01	0.00000E+00	-0.75000E+01
0.60000E+01	0.00000E+00	-0.10700E+02
0.70000E+01	0.00000E+00	-0.13000E+02
0.80000E+01	0.00000E+00	-0.15200E+02
0.90000E+01	0.00000E+00	-0.10000E+02
0.10000E+02	0.00000E+00	-0.62000E+01
0.11000E+02	0.00000E+00	-0.40000E+01
0.12000E+02	0.00000E+00	-0.17000E+01
0.13000E+02	0.00000E+00	-0.70000E+00
0.14000E+02	0.00000E+00	0.00000E+00

J. Edit Profile ("p")

This subsystem controls editing of the profile locations and observed anomalies.

Q: "Profile command (a/c/d/h/o/q)?"

R: Enter the one-letter abbreviation of a profile command, or enter "h" to list a brief explanation of the abbreviations.

If you enter:

"a": the system will request the (x,z) coordinates and observed value for new profile points until you indicate you are done by entering "//".

"c": The system will request the number of the point to change, then present that point for change. If you enter a negative number, the system will begin at that number and continue presenting points until you enter "//".

"d": The system will request the number of the point to delete. It will then print the (x,z) coordinate for the point and ask the question "Delete (y/n)?". If you respond "y", the point will be removed from the profile.

"o": You will be prompted for the observed anomaly at every (x,z) point.

"q": Exit this subsystem.

Example:

```
hypermag
*********   WELCOME TO HYPERMAG - version 1   *********
Enter the name of the model:gravmod
New, old (hypermag), 2mag, 2dgrav, or saki model (n/o/t/g/s) [n]?o
Name of model file to read [gravmod.mod]?
Hyper-command [h]?p
Profile command (a/c/d/h/o/q) [c]?h
Profile commands:
    a = add point to profile
    c = change point on profile
    d = delete point from profile
    o = enter observed anomalies
    q = quit profile area
Profile command (a/c/d/o/q) [q]?a
X coordinate ("//" to finish) [0.0000]?8.5
Z coordinate [0.0000]?
```

```
Observed anomaly [0.0000]?-277
X coordinate ("//" to finish) [8.5000]?//
Profile command (a/c/d/o/q) [q]?c
Change point # (-1 for all, 0 for done) [-1]?-6
Profile point:     6
X coordinate [7.0000]?
Z coordinate [0.0000]?
Observed anomaly [-278.0000]?
Profile point:     7
X coordinate [8.0000]?
Z coordinate [0.0000]?
Observed anomaly [-280.2000]?
Profile point:     8
X coordinate [8.5000]?
Z coordinate [0.0000]?
Observed anomaly [-277.0000]?-277.5
Profile point:     9
X coordinate [9.0000]?//
Change point # (-1 for all, 0 for done) [0]?//
Profile command (a/c/d/o/q) [q]?d
Delete point # ("//" to end) [-1]?8
X=          8.5000 Y=          0.0000 Obs=       -277.5000
Delete (y/n) [n]?y
Delete point # ("//" to end) [-1]?//
Profile command (a/c/d/o/q) [q]?//
Hyper-command [q]?
Save model (y/n) [y]?n
Thanks for using hypermag.............
```

K. Read in a new model ("r")

This area has the same dialog as "Input model", section A of this chapter.

Example:

```
hypermag
*********   WELCOME TO HYPERMAG - version 1   *********
Enter the name of the model:gravmod
New, old (hypermag), 2mag, 2dgrav, or saki model (n/o/t/g/s) [n]?o
Name of model file to read [gravmod.mod]?
Hyper-command [h]?r
Enter the name of the model:mag
New, old (hypermag), 2mag, 2dgrav, or saki model (n/o/t/g/s) [n]?o
Name of model file to read [mag.mod]?mtest.mod
Hyper-command [q]?
Save model (y/n) [y]?n
Thanks for using hypermag.............
```

L. See model parameters ("s")

This subsystem prints a formatted list of all the model parameters, including bodies, profile, and global parameters.

Example:

```
hypermag
*********   WELCOME TO HYPERMAG - version 1   *********
Enter the name of the model:gravmod
New, old (hypermag), 2mag, 2dgrav, or saki model (n/o/t/g/s) [n]?o
Name of model file to read [gravmod.mod]?
Hyper-command [h]?s
Azimuth of profile          =    0.00
No. points in profile       =  13
No. of bodies               =  1
Inclination of field        =    0.00
Declination of field        =    0.00
First value of regional     =   -265.00
Slope of regional           =    0.0000
X conversion to KM          =    1.0000
Z conversion to KM          =    1.0000
Body No.  1
Inclination =    0.00
Declination =    0.00
Intensity   =   0.00000
Density     =  -0.30000
Corner     X          Z
   1     3.500      0.100
   2    11.000      0.100
   3    10.000      0.700
   4     9.000      1.200
   5     8.000      2.000
Profile
Point     X          Z          Observed   Obs-reg
   1     2.000      0.000   -265.000      0.000
   2     3.000      0.000   -267.000     -2.000
   3     4.000      0.000   -267.500     -2.500
   4     5.000      0.000   -272.500     -7.500
   5     6.000      0.000   -275.700    -10.700
   6     7.000      0.000   -278.000    -13.000
   7     8.000      0.000   -280.200    -15.200
   8     9.000      0.000   -275.000    -10.000
   9    10.000      0.000   -271.200     -6.200
  10    11.000      0.000   -269.000     -4.000
  11    12.000      0.000   -266.700     -1.700
  12    13.000      0.000   -265.700     -0.700
  13    14.000      0.000   -265.000      0.000
Hyper-command [q]?
Save model (y/n) [y]?n
Thanks for using hypermag.............
```

M. Write model ("w")

This subsystem controls the output of your model into a binary file which can be read by subsequent runs of this program.

Q: "Name of model file to write to [default name]?"

R: Enter the name you wish the output file to have, or press RETURN if the default name is acceptable.

Example:

```
hypermag
*********   WELCOME TO HYPERMAG - version 1   *********
Enter the name of the model:gravmod
New, old (hypermag), 2mag, 2dgrav, or saki model (n/o/t/g/s) [n]?o
Name of model file to read [gravmod.mod]?
Hyper-command [h]?w
Name of model file to write to [gravmod.mod]?
Hyper-command [q]?
Save model (y/n) [y]?n
Thanks for using hypermag.............
```

3. Errors and Recovery

If Hypermag aborts after you have changed your current model and before it has been written out, run "hyperstart" to return to the hyper-command level.

If you are running a magnetic model, the calculation routine could print this error message:

"Error from ribbon2...i = <profile pt#>, j = <body #>, ier = <1 or 2>

This indicates that a problem was encountered in calculating the magnetic effect of body j at field point i. If ier = 1, that means two of the body corners were closer together than 1e-18 (a very small number). The program will assume those two corners to be a single corner. If ier = 2, the field point i is too close to, or inside, body j. In this case the calculation could produce a wierd result.

In the gravity case, if a profile point is on a body it is moved 0.1 meter away before calculation. If the point is inside the body, the equations still apply, so no error is signaled.

Example:

```
hypermag
*********   WELCOME TO HYPERMAG - version 1   *********
Enter the name of the model:gravmod
New, old (hypermag), 2mag, 2dgrav, or saki model (n/o/t/g/s) [n]?o
Name of model file to read [gravmod.mod]?
Hyper-command [h]?c
Calculation type (g=grav, m=mag) [g]?
Output type (f/h/o/p/q/t) [f]?o
Plot command (s/c/h/r/w/p/q) [p]?r
Plot parameter filename [plotparms]?frog
fortran_io_: Entry not found. Open on file 15.
By >online>Csc>RSaltus>hypermag$hypermag (hypermag¢32154)
hyperstart
Hyper-command [h]?w
Name of model file to write to [gravmod.mod]?
Hyper-command [q]?
Save model (y/n) [y]?n
Thanks for using hypermag.............
```

References

Grant, F. S., and West, G. F., 1965, Interpretation theory in applied geophysics, McGraw-Hill Book Company

Talwani, M. and Heirtzler, J. R., 1964, Computation of magnetic anomalies caused by two-dimensional structures of arbitrary shape, in Computers in the Mineral Industries: Stanford University Publications in the Geological Sciences, v. 9, part 1, p. 464-480.

Talwani, Maanik, Worzek, J. L., and Landisman, Mark, 1959, Rapid gravity computations for two-dimensional bodies with application to the Mendocino submarine fracture zone: Journal of Geophysical Research, v. 64, n. 1, p. 49-59.

Appendix A

Program Listing

Hypermag is written in Ansi77 Fortran as implemented on the Honeywell multics. Below is a description of the common blocks used in the program, followed by a listing of the program.

Common Blocks

Hypermag uses three FORTRAN common blocks to describe the current model. Below is a description of the variables within each block. Following fortran convention, variables begining with the letters i-n are integers unless otherwise noted. Array sizes are shown in parenthesis following array names.

Block 1 - Profile information

nprof = Number of profile points (limited to 1000).

dprof = If >0, the sample interval of profile.

If <0, x and z define the profile.

regfirst = Initial value of regional.

regslope = Slope of regional to be removed from obs.

iobs = 1, if there are observed values.

0, if there are no observed values.

h(1000) = Calculated field.

obs(1000) = Observed field.

x(1000) = X axis locations of profile points.

z(1000) = Z axis locations of profile points (positive down).

robs(1000) = Observed field with regional removed.

Block 2 - Body information

nbody = Number of bodies (limited to 50).

xcorn (50,50) - X coordinates of body corners -

first index = body number

second index = corner number

zcorn (50,50) = Z coordinate of body corners (z positive down).

ncorn (50) = Number of corners.

m(50) [real array] = Intensity of magnetization (emu/cm^3).

mi(50) [real array] = Magnetic inclination (degrees).

md(50) [real array] = Magnetic declination (degrees).

mx(50) [real array] = X component of magnetization.

mz(50) [real array] = Z component of magnetization.

rho (50) = Density (g/cm^3)

Block 3 - Global model parameters

azim = Azimuth of x axis (degrees).

fi = Inclination of regional field.

fd = Declination of regional field.

itype = 0, magnetic model.

1, gravity model.

xconv = X axis conversion to km.

zconv = Z axis conversion to km.

The common blocks "plot" and "title" control the off-line plot.

The common block "hyper" contains one variable.

hyper

iplot = 0, no off-line plot has been done.

1, off-line plot has been run.

```
c
c     HYPER-MAG, magnetic and-or gravity modeling program
c
c  *********************************************
      subroutine hypermag
      common /block3/ azim,fi,fd,itype,xconv,zconv
      common /hyper/ iplot
      common /start/ project
      external asr(descriptors)
      character*80 quest,ans*1,yans*1,gans*1,pfilename*50
      character*40 project,filename*50,model*10,ofilename*50,results*10
      data iprint/6/,icard/5/,model/".mod       "/,results/".res       "/
c
c     MULTICS DEPENDANT SEARCH RULES
c
      call asr(">online>Csc>RSaltus>plot","-af","working_dir")
      call asr(">online>Csc>RSaltus","-af","working_dir")
c
c
      entry hmag
      iplot=0
      pfilename=" "
      write(iprint,110)
 110  format(" *****   WELCOME TO HYPERMAG - version 1.3   *****")
      call input_model(project)
      call concat(ofilename,project,results)
      entry hyperstart
      yans="h"
  15  continue
      if (yans.eq."/") yans="q"
      quest="Hyper-command"
      idumb=iaquest(quest,yans,"(a1)",1)
      if (yans.eq."s") call seemodel(project,6)
      if (yans.eq."c") go to 20
      if (yans.eq."b") call cm_bod(project)
      if (yans.eq."m") call cm_model(project)
      if (yans.eq."p") call cm_prof(project)
      if (yans.eq."w") call write_model(project)
      if (yans.eq."r") call input_model(project)
      if (yans.eq."r") call concat(ofilename,project,results)
      if (yans.eq."q") go to 30
      if (yans.eq."/") go to 30
      if (yans.eq."e") call screen_edit
      if (yans.eq."f") call fast_plot
      if (yans.eq."h") write(iprint,120)
      if (yans.eq."o") call write_saki(project)
      yans="q"
      go to 15
 120  format(" Enter one of the following:",/,
     &"     b = edit bodies",/,
     &"     c = calculate computed anomaly",/,
     &"     e = edit the model using the 4112 screen",/,
     &"     f = fast re-calculation and plot.",/,
     &"     m = edit model parameters",/,
```



```
     &"     o = output generic model files (for program saki, etc.)",/,
     &"     p = edit profile",/,
     &"     r = read in a new model",/,
     &"     s = see model parameters",/,
     &"     w = write model to a file",/,
     &"     q = quit hypermag")
c
c     perform calculations
c
  20  continue
      quest="Calculation type (g=grav, m=mag)"
      gans="m"
      if (itype.eq.1) gans="g"
      idumb=iaquest(quest,gans,"(a1)",1)
      if (gans.eq."g") call tgrav
      if (gans.eq."m") call anomaly
      if (gans.eq."m") itype=0
      if (gans.eq."g") itype=1
      yans="f"
      quest="Output type (f/h/o/p/q/t)"
  25  continue
      idumb=iaquest(quest,yans,"(a1)",1)
      if (yans.eq."t") call model_print(project,ofilename,6)
      if (yans.eq."f") call model_print(project,ofilename,13)
      if (yans.eq."o") call plotquiz(project,pfilename)
      if (yans.eq."p") call model_plot(project,6)
      if (yans.eq."q") go to 15
      if (yans.eq."/") go to 15
      if (yans.eq."h") write(iprint,130)
 130  format(" Output types:",/,
     &"     f = print results at a file",/,
     &"     o = off-line plot",/,
     &"     p = printer plot",/,
     &"     t = print at terminal",/,
     &"     q = quit output section")
      yans="q"
      go to 25
c
c     done
c
  30  continue
      quest="Save model (y/n)"
      ans="y"
      idumb=iaquest(quest,ans,"(a1)",1)
      if (ans.eq."y") call write_model(project)
      write(iprint,150)
 150  format(" Thanks for using hypermag.............")
      end
c
c     CM_MODEL- interactive change of model parameters
c
c  *********************************************
      subroutine cm_model(project)
      common /block1/ nprof,dprof,regfirst,regslope,iobs,h(1000),
```

```
     &obs(1000),x(1000),z(1000),robs(1000)
      common /block3/ azim,fi,fd,itype,xconv,zconv
      character*80 quest,ans*1,project*40,rform*8
      rform="(f15.4)"
      quest="Azimuth (degrees)"
      ival=irquest(quest,azim,rform,0)
      if (ival.eq.-1) go to 99
      quest="Field inclination (degrees)"
      ival=irquest(quest,fi,rform,0)
      if (ival.eq.-1) go to 99
      quest="Field declination (degrees)"
      ival=irquest(quest,fd,rform,0)
      if (ival.eq.-1) go to 99
      quest="Initial regional value"
      ival=irquest(quest,regfirst,rform,0)
      if (ival.eq.-1) go to 99
      quest="Regional slope"
      ival=irquest(quest,regslope,rform,0)
      quest="X axis conversion to KM"
      ival=irquest(quest,xconv,rform,0)
      quest="Z axis conversion to KM"
      ival=irquest(quest,zconv,rform,0)
   99 continue
c  ********************************************
      call sub_regional
      return
      end
c
c     CM_PROF-  interactive change of profile parameters
c
c  ********************************************
      subroutine cm_prof(project)
      common /block1/ nprof,dprof,regfirst,regslope,iobs,h(1000),
     &obs(1000),x(1000),z(1000),robs(1000)
      character*80 quest,ans*1,iform*8,rform*8,eform*8,xquest,zquest
      character*80 oquest
      rform="(f15.4)"
      eform="(e15.4)"
      iform="(i10)"
      xquest="X coordinate"
      zquest="Z coordinate"
      oquest="Observed anomaly"
      ans="c"
   10 continue
      quest="Profile command (a/c/d/h/o/q)"
      idumb=iaquest(quest,ans,"(a1)",1)
      if (ans.eq."o") call add_obs
      if (ans.eq."c") go to 1000
      if (ans.eq."a") go to 2000
      if (ans.eq."d") go to 3000
      if ((ans.eq."q").or.(ans.eq."/")) go to 4000
      if (ans.eq."h") write(6,100)
  100 format(" Profile commands:",/,
     &"      a = add point to profile",/,
```



```
     &"      c = change point on profile",/,
     &"      d = delete point from profile",/,
     &"      o = enter observed anomalies",/,
     &"      q = quit profile area")
   20 ans="q"
      go to 10
c
c     change profile point
c
 1000 continue
      icha=-1
      go to 1010
 1005 icha=0
 1010 quest="Change point # (-1 for all, 0 for done)"
      ival=iiquest(quest,icha,iform,0)
      if ((icha.eq.0).or.(ival.eq.-1)) go to 20
 1015 if (iabs(icha).ge.nprof) icha=nprof
      write(6,105)iabs(icha)
  105 format(" Profile point:",i5)
      ival=irquest(xquest,x(iabs(icha)),rform,0)
      if (ival.eq.-1) go to 1005
      ival=irquest(zquest,z(iabs(icha)),rform,0)
      if (ival.eq.-1) go to 1005
      if (iobs.eq.0) go to 1020
      ival=irquest(oquest,obs(iabs(icha)),rform,0)
 1020 continue
      icha=sign(iabs(icha)+1,icha)
      if (icha.ge.0) go to 1005
      go to 1015
c
c     add profile point
c
 2000 continue
      if (nprof.lt.1000) go to 2005
      write(6,2001)
 2001 format(" Sorry- only 1000 profile points allowed.")
      go to 20
 2005 continue
      xt=0.
      zt=0.
      ot=0.
      quest='X coordinate ("//" to finish)'
 2010 continue
      ival=irquest(quest,xt,rform,0)
      if (ival.eq.-1) go to 20
      ival=irquest(zquest,zt,rform,0)
      if (ival.eq.-1) go to 20
      if (iobs.eq.0) go to 2020
      ival=irquest(oquest,ot,rform,0)
      if (ival.eq.-1) go to 2010
 2020 continue
c
c     insert new point into correct spot
c
```

```
      do 2040 i=nprof,1,-1
      if (x(i).lt.xt) go to 2045
      x(i+1)=x(i)
      z(i+1)=z(i)
      obs(i+1)=obs(i)
2040  continue
      x(1)=xt
      z(1)=zt
      obs(1)=ot
      go to 2050
2045  continue
      x(i+1)=xt
      z(i+1)=zt
      obs(i+1)=ot
2050  continue
      nprof=nprof+1
      go to 2010
c
c     delete point
c
3000  continue
      quest='Delete point # ("//" to end)'
      idel=-1
      ival=iiquest(quest,idel,iform,0)
      if (ival.eq.-1) go to 20
3010  if (iabs(idel).gt.nprof) idel=nprof
      if (iobs.eq.0) write(6,3020)x(iabs(idel)),z(iabs(idel))
3020  format(" X=",f15.4," Y=",f15.4)
      if (iobs.eq.1) write(6,3030)x(iabs(idel)),z(iabs(idel)),
     &obs(iabs(idel))
3030  format(" X=",f15.4," Y=",f15.4," Obs=",f15.4)
      ans="n"
      quest="Delete (y/n)"
      idumb=iaquest(quest,ans,"(a1)",1)
      if (ans.ne."y") go to 3060
      if (iabs(idel).ge.nprof) go to 3050
      do 3040 i=iabs(idel),nprof-1
      x(i)=x(i+1)
      z(i)=z(i+1)
      obs(i)=obs(i+1)
3040  continue
3050  nprof=nprof-1
3060  idel=sign(iabs(idel+1),idel)
      if (idel.ge.0) go to 3000
      go to 3010
 4000 continue
      call sub_regional
      return
      end
c
c     CM_BOD- interactive body modification routine
c
c  ******************************************
      subroutine cm_bod(project)
```



```
      common /block2/ nbody,xcorn(50,50),zcorn(50,50),ncorn(50),
     &m(50),mi(50),md(50),mx(50),mz(50),rho(50)
      real m,mi,md,mx,mz
      common /block3/ azim,fi,fd,itype,xconv,zconv
      character*40 project,quest*80,ans*1,iform*10,rform*10,ans2*1
      iform="(i10)"
      rform="(f15.3)"
      ans="h"
 200  quest="Body command (a/c/d/h/q)"
      idumb=iaquest(quest,ans,"(a1)",1)
      if (ans.eq."h") write(6,210)
 210  format(" Body commands:",/,
     &"      a = add new body",/,
     &"      c = change existing body",/,
     &"      d = delete body",/,
     &"      q = quit the body area")
      if (ans.eq."a") call add_body
      if (ans.eq."c") go to 1000
      if (ans.eq."d") go to 2000
      if ((ans.eq."q").or.(ans.eq."/")) return
 300  ans="q"
      go to 200
c
c     change body
c
1000  continue
      icha=-1
      go to 1010
1005  icha=0
1010  quest="Change body # (0 when done)"
      ival=iiquest(quest,icha,iform,0)
      if ((icha.eq.0).or.(ival.eq.-1)) go to 300
      if (iabs(icha).ge.nbody) icha=nbody
      ibod=iabs(icha)
      ans="c"
1020  quest="Change corners or parameters (c/p/q)"
      icorn=0
      idumb=iaquest(quest,ans,"(a1)",1)
      if ((ans.eq."q").or.(ans.eq."/")) go to 1005
      if (ans.eq."p") go to 1500
      if (ans.eq."c") go to 1800
1025  ans="q"
      go to 1020
c
c     change parms
c
1500  continue
      quest="Body magnetization"
      ival=irquest(quest,m(ibod),"(e10.4)",0)
      if (ival.eq.-1) go to 1025
      quest="Inclination"
      ival=irquest(quest,mi(ibod),"(f10.3)",0)
      if (ival.eq.-1) go to 1025
      quest="Declination"
```

```
      ival=irquest(quest,md(ibod),"(f10.3)",0)
      if (ival.eq.-1) go to 1025
c
c     grav
c
1600  continue
      quest="Density contrast"
      ival=irquest(quest,rho(ibod),rform,0)
      go to 1005
c
c     change corners
c
1800  continue
      icorn=icorn+1
      quest="Corner #"
      if (icorn.gt.ncorn(ibod)) icorn=1
      ival=iiquest(quest,icorn,"(i10)",0)
      icorn=iabs(icorn)
      if ((ival.eq.-1).or.(icorn.eq.0)) go to 1005
      if (icorn.gt.ncorn(ibod)) icorn=1
      write(6,1810)icorn,xcorn(ibod,icorn),zcorn(ibod,icorn)
1810  format(" Corner #",i2," x coord=",f10.3,", z coord=",f10.3)
      ans="c"
      quest="Add, change, or delete (a/c/d/q)"
      idumb=iaquest(quest,ans,"(a1)",1)
      if ((ans.eq."q").or.(ans.eq."/")) go to 1800
      if (ans.eq."c") call corn_cha(ibod,icorn)
      if (ans.eq."d") call corn_del(ibod,icorn)
      if (ans.eq."a") call corn_add(ibod,icorn)
      go to 1800
c      Delete body
2000  continue
      idel=0
      if (nbody.le.1) go to 200
      quest="Delete body # (0 for done)"
      ival=iiquest(quest,idel,iform,0)
      if ((idel.eq.0).or.(ival.eq.-1)) go to 200
      idel=iabs(idel)
      if (idel.ge.nbody) idel=nbody
      ans2="n"
      write(6,2001)idel
2001  format(" body #",i2," is to be deleted.")
      quest="Are you sure (y/n)"
      idumb=iaquest(quest,ans2,"(a1)",1)
      if (ans2.ne."y") go to 200
      if (idel.eq.nbody) go to 2200
      do 2100 i=idel,nbody-1
      do 2050 j=1,ncorn(i+1)
      xcorn(i,j)=xcorn(i+1,j)
      zcorn(i,j)=zcorn(i+1,j)
2050  continue
      ncorn(i)=ncorn(i+1)
2100  continue
2200  nbody=nbody-1
```



```
      write(6,2201)idel
2201  format(" Body #",i2," deleted.")
      go to 200
      end
c
c     BUILD MODEL- Initial entry of model
c
c  ******************************************
      subroutine build_model(project)
      common /block1/ nprof,dprof,regfirst,regslope,iobs,h(1000),
     &obs(1000),x(1000),z(1000),robs(1000)
      common /block2/ nbody,xcorn(50,50),zcorn(50,50),ncorn(50),
     &m(50),mi(50),md(50),mx(50),mz(50),rho(50)
      real m,mi,md,mx,mz
      common /block3/ azim,fi,fd,itype,xconv,zconv
      character*32 new_file,old_file,project*40,ans*1,gans*1,quest*80
      gans="m"
      itype=0
      quest="Gravity or magnetics (g/m)"
      ival=iaquest(quest,gans,"(a1)",1)
      if (gans.ne."m") itype=1
      azim=0
      fi=0
      fd=0
      if (itype.ne.0) go to 77
      write(6,100)
  100 format(" Azimuth of the profile = ?",$)
      read(5,*)azim
      write(6,102)
  102 format(" Inclination and declination of the regional field = ? "$)
      read(5,*)fi,fd
   77 write(6,200)
  200 format(" Number of points in the profile = ?",$)
      read(5,*)nprof
      call enter_units
    3 ans="n"
      quest="Are the points of the profile evenly spaced and
     & elevation (y/n)"
      idum=iaquest(quest,ans,"(a1)",1)
      if(ans.eq."n")go to 1
      if(ans.eq."y")go to 2
      go to 3
    2 write(6,104)
  104 format(" Spacing between points = ? "$)
      read(5,*)dprof
      write(6,108)
  108 format(" The x and z coordinates of the first profile point = ?"$)
      read(5,*)x(1),z(1)
      do 5 i=2,nprof
      x(i)=x(1)+(i-1)*dprof
    5 z(i)=z(1)
      go to 4
    1 write(6,106)
  106 format(" Enter the x and z coordinates of the profile points,"
```

```
     &          " one at a time..."//)
      do 6 i=1,nprof
      write(6,107)i
  107 format(" Point",i5,":  x,z = ? "$)
    6 read(5,*)x(i),z(i)
    4 continue
      regfirst=0.
      regslope=0.
    9 ans="n"
      quest="Do you wish to enter observed anomalies (y/n)"
      idum=iaquest(quest,ans,"(a1)",1)
      if(ans.eq."y")go to 7
      if(ans.eq."n")go to 8
      go to 9
    8 iobs=0
      do 10 i=1,nprof
   10 obs(i)=0.
      go to 11
    7 iobs=1
      write(6,109)
  109 format(" Type the anomaly values, one at a time..."//)
      do 12 i=1,nprof
      write(6,110)i,x(i),z(i)
  110 format(" Point",i5,":  x =",f10.3,"  z =",f10.3,"  Anomaly = ? "$)
   12 read(5,*)obs(i)
   14 ans="n"
      quest=
     &"Do you wish to subtract a regional trend from these data (y/n)"
      idum=iaquest(quest,ans,"(a1)",1)
      if(ans.eq."n")go to 11
      if(ans.eq."y")go to 13
      go to 14
   13 write(6,116)
  116 format(" The value of the regional at the first point and the"
     &,/," slope of the regional = ?"$)
      read(5,*)regfirst,regslope
   11 continue
      nbody=0
 15   call add_body
   18 ans="n"
      quest="Another body (y/n)"
      idum=iaquest(quest,ans,"(a1)",1)
      if(ans.eq."y")go to 15
      if(ans.eq."n")go to 17
      go to 18
 17   write(6,118)
  118 format(" This model is ready"//)
      call sub_regional
      return
      end
c
c     ADD BODY - asks information about a new body
c
c  ********************************************
```



```
      subroutine add_body
      common /block2/ nbody,xcorn(50,50),zcorn(50,50),ncorn(50),
     &m(50),mi(50),md(50),mx(50),mz(50),rho(50)
      real m,mi,md,mx,mz
      common /block3/ azim,fi,fd,itype,xconv,zconv
   15 nbody=nbody+1
      if (itype.ne.0) go to 77
      write(6,111)nbody
  111 format(" What is the inclination, declination, and intensity"
     &" of ",/," magnetization for body ",i3," ?",$)
      read(5,*)mi(nbody),md(nbody),m(nbody)
      go to 78
   77 write(6,115)nbody
  115 format(" What is the density contrast for body ",i3," ?",$)
      read(5,*)rho(nbody)
   78 continue
    9 write(6,112)
  112 format(" How many corners will it have ? "$)
      read(5,*)ncorn(nbody)
      if (ncorn(nbody).le.50) go to 10
      write(6,116)
  116 format(" Sorry, but you are limited to 50 corners per body.")
      go to 9
   10 write(6,113)
  113 format(" Type the x and z coordinates of each corner..."//)
      do 16 i=1,ncorn(nbody)
      write(6,114)i
  114 format(" Corner ",i3,":  x,z = ? "$)
   16 read(5,*)xcorn(nbody,i),zcorn(nbody,i)
      return
      end
c
c     WRITE MODEL - stores model in a file
c
c  ********************************************
      subroutine write_model(project)
      common /block1/ nprof,dprof,regfirst,regslope,iobs,h(1000),
     &obs(1000),x(1000),z(1000),robs(1000)
      common /block2/ nbody,xcorn(50,50),zcorn(50,50),ncorn(50),
     &m(50),mi(50),md(50),mx(50),mz(50),rho(50)
      real m,mi,md,mx,mz
      common /block3/ azim,fi,fd,itype,xconv,zconv
      character*40 project,quest*80,filename*50
      call concat(filename,project,".mod       ")
      quest="Name of model file to write to"
      idumb=iaquest(quest,filename,"(a40)",40)
      open(11,file=filename,form="unformatted",mode="out")
      write(11)azim,nprof,fi,fd,dprof,regfirst,regslope,nbody,iobs,
     &xconv,zconv,itype
      write(11)(x(i),z(i),obs(i),i=1,nprof)
      do 1 j=1,nbody
      write(11)mi(j),md(j),m(j),ncorn(j),rho(j)
    1 write(11)(xcorn(j,i),zcorn(j,i),i=1,ncorn(j))
      close(11)
```

```
      return
      end
c
c     READ MODEL - reads old model from a file
c
c  *********************************************
      subroutine read_model(filename)
      common /hyper/iplot
      common /block1/ nprof,dprof,regfirst,regslope,iobs,h(1000),
     &obs(1000),x(1000),z(1000),robs(1000)
      common /block2/ nbody,xcorn(50,50),zcorn(50,50),ncorn(50),
     &m(50),mi(50),md(50),mx(50),mz(50),rho(50)
      real m,mi,md,mx,mz
      common /block3/ azim,fi,fd,itype,xconv,zconv
      character*40 filename
      open(10,file=filename,form="unformatted",mode="in")
      read(10)azim,nprof,fi,fd,dprof,regfirst,regslope,nbody,iobs,
     &xconv,zconv,itype
      read(10)(x(i),z(i),obs(i),i=1,nprof)
      do 1 j=1,nbody
      read(10)mi(j),md(j),m(j),ncorn(j),rho(j)
    1 read(10)(xcorn(j,i),zcorn(j,i),i=1,ncorn(j))
      close(10)
      call sub_regional
      iplot=0
      return
      end
c
c     RIBBON2  -  performs mag calculations-  by Richard J. Blakely
c
c  *********************************************
      subroutine ribbon2(x0,z0,xc,zc,nc,jx,jz,fx,fz,ier)
      real jx,jz
      dimension xc(50),zc(50),theta(100),r(100),sx(100),sz(100),qs(100)
      pi=3.1415927
      solid=0.
      fx=0.
      fz=0.
      ier=0
      xc(nc+1)=xc(1)
      zc(nc+1)=zc(1)
c
c  --  for each face, calculate arrays sx, sz, qs, r, and theta
c
      do 10 i=1,nc
      sx(i)=xc(i+1)-xc(i)
      sz(i)=zc(i+1)-zc(i)
      s=sqrt(sx(i)**2+sz(i)**2)
c
c  --  if two corners are too close, return
c
      if (s.gt.1.e-18)go to 20
      ier=1
      return
```



```
   20 continue
      sx(i)=sx(i)/s
      sz(i)=sz(i)/s
      qs(i)=jx*sz(i)-jz*sx(i)
      rx=xc(i)-x0
      rz=zc(i)-z0
c
c  --  if field point is too near a corner, signal error
c
      if (abs(rx).gt.1.e-18.or.abs(rz).gt.1.e-18)go to 30
      ier=2
      rx=1.e-18
      rz=1.e-18
   30 continue
      r(i)=sqrt(rx**2+rz**2)
   10 theta(i)=atan2(rz,rx)
      r(nc+1)=r(1)
      theta(nc+1)=theta(1)
c
c  --  for each face, combine all parameters into fx and fz
c
      do 40 i=1,nc
      angle=theta(i)-theta(i+1)
      if (angle.gt.pi)angle=angle-2.*pi
      if (angle.lt.-pi)angle=angle+2.*pi
      if (abs(angle).gt.(.995*pi))ier=2
c
c  --  if field point is too close to side, signal error
c
      fs=2.*qs(i)*(alog(r(i))-alog(r(i+1)))
      fn=2.*qs(i)*angle
      fx=fx+fn*sz(i)+fs*sx(i)
      fz=fz-fn*sx(i)+fs*sz(i)
   40 solid=solid+angle
c
c  --  if field point is inside body, signal error
c
      if(solid.lt.(-pi/2.))ier=2
      return
      end
c
c     SEE MODEL  - prints a formatted list of model parms
c
c  *********************************************
      subroutine seemodel(project,num)
      common /block1/ nprof,dprof,regfirst,regslope,iobs,h(1000),
     &obs(1000),x(1000),z(1000),robs(1000)
      common /block2/ nbody,xcorn(50,50),zcorn(50,50),ncorn(50),
     &m(50),mi(50),md(50),mx(50),mz(50),rho(50)
      real m,mi,md,mx,mz
      common /block3/ azim,fi,fd,itype,xconv,zconv
      character*40 project,ans*1
      write(num,100)azim,nprof,nbody,fi,fd,regfirst,regslope,xconv,zconv
  100 format(" Azimuth of profile          =",f8.2,/,
```

```
     &        " No. points in profile       =",i4,/,
     &        " No. of bodies               =",i3,/,
     &        " Inclination of field        =",f8.2,/,
     &        " Declination of field        =",f8.2,/,
     &        " First value of regional     =",f10.2,/,
     &        " Slope of regional           =",f10.4,/,
     &        " X conversion to KM          =",f10.4,/,
     &        " Z conversion to KM          =",f10.4)
      do 10 j=1,nbody
      write(num,110)j,mi(j),md(j),m(j),rho(j)
 110  format(" Body No.",i3,/,
     &" Inclination =",f8.2,/,
     &" Declination =",f8.2,/,
     &" Intensity   =",f10.5,/,
     &" Density cont=",f10.5,/,
     &(" Corner",4x,"X",9x,"Z",8x),/)
      do 5 i=1,ncorn(j)
   5  write(num,120)i,xcorn(j,i),zcorn(j,i)
 120  format(i5,2f10.3,5x,i5,2f10.3)
  10  continue
      write(num,130)
 130  format(" Profile",/,
     &(" Point",4x,"X",9x,"Z",8x,"Observed",2x,"Obs-reg"))
      do 20 i=1,nprof
      write(num,140)i,x(i),z(i),obs(i),robs(i)
 140  format(x,i4,4f10.3)
  20  continue
      end
c
c     ANOMALY - sets up ribbon call
c
c  ********************************************
      subroutine anomaly
      common /block1/ nprof,dprof,regfirst,regslope,iobs,h(1000),
     &obs(1000),x(1000),z(1000),robs(1000)
      common /block2/ nbody,xcorn(50,50),zcorn(50,50),ncorn(50),
     &m(50),mi(50),md(50),mx(50),mz(50),rho(50)
      real m,mi,md,mx,mz
      common /block3/ azim,fi,fd,itype,xconv,zconv
      character*1 ans
      dimension xc(50),zc(50),xt(1000),zt(1000)
      conv=3.1415927/180.
      fx=cos(fi*conv)*cos((fd-azim)*conv)
      fz=sin(fi*conv)
      do 3 i=1,nprof
      xt(i)=x(i)*xconv
      zt(i)=z(i)*zconv
3     h(i)=0
      do 20 j=1,nbody
      mx(j)=cos(mi(j)*conv)*cos((md(j)-azim)*conv)
      mz(j)=sin(mi(j)*conv)
      do 10 k=1,ncorn(j)
      xc(k)=xcorn(j,k)*xconv
 10   zc(k)=zcorn(j,k)*zconv
```



```
      do 20 i=1,nprof
      call ribbon2(xt(i),zt(i),xc,zc,ncorn(j),mx(j),mz(j),hx,hz,ier)
      if (ier.eq.0) go to 20
      write(6,100)i,j,ier
 100  format(" Error from ribbon2... i=",i4," j=",i4," ier=",i2)
  20  h(i)=(hx*fx+hz*fz)*m(j)*1.e5+h(i)
      return
      end
c
c   PLOT QUIZ  -   Top level of interactive model plotting routines
c
c  ********************************************
      subroutine plotquiz(project,filename)
      common /hyper/ iplot
      character*40 project,yans*1,filename*50,quest*80
      if (filename.eq." ") call concat(filename,
     &project,".ppm      ")
      call defaultplotparms(project)
  10  continue
 130  format(" Enter one of the following",/,
     &"      s= see plot parameters",/,
     &"      c= change plot parameters",/,
     &"      r= read plot parameters from file",/,
     &"      w= write plot parameters to file",/,
     &"      p= plot model",/,
     &"      q= quit plot system")
      yans="p"
  15  continue
      quest="Plot command (s/c/h/r/w/p/q)"
      idumb=iaquest(quest,yans,"(a1)",1)
      if (yans.eq."h") write(6,130)
      if (yans.eq."s") call seeplotparm
      if (yans.eq."c") call changeparm
      if (yans.eq."r") call readplotparm(filename)
      if (yans.eq."w") call writeplotparm(filename)
      if (yans.eq."p") call plotmodel
      if (yans.eq."p") iplot=1
      if (yans.eq."q") return
      yans="q"
      go to 15
      end
c
c
c       READ PLOT PARM
c
c
c  ********************************************
      subroutine readplotparm(filename)
      character*50 filename,quest*80
      quest="Plot parameter filename"
      idumb=iaquest(quest,filename,"(a50)",50)
      call parmio(filename,0)
      return
      end
```

```
c
c
c       WRITEPLOTPARM
c
c
c  *******************************************
      subroutine writeplotparm(filename)
      character*50 filename,quest*80
      quest="Filename for parameter storage"
      idumb=iaquest(quest,filename,"(a50)",50)
      call parmio(filename,1)
      return
      end
c
c
c       BODY PLOT
c
c  *******************************************
      subroutine bodyplot
      common /block1/ nprof,dprof,regfirst,regslope,iobs,h(1000),
     &obs(1000),x(1000),z(1000),robs(1000)
      common /block2/ nbody,xcorn(50,50),zcorn(50,50),ncorn(50),
     &m(50),mi(50),md(50),mx(50),mz(50),rho(50)
      real m,mi,md,mx,mz
      common /plot/ bmargin,rlmargin,cmargin,xdel,zdel,gdel,
     &ixip,izip,igip,csize,ssize,tsize,xsize,ysize,iplotr,xbody,
     &zbody,ganom,xmin,xmax,zmin,zmax,gmin,gmax,iabody,ixunit,
     &nxchar,nzchar,ngchar,ntlchar,nt2char,vex,scale,gscale
     &,ixnum,ignum,itek,ifast,ixrev,iyrev,rnsize,tmargin,rmargin
      common /title/ xtitle,ztitle,gtitle,titlel,title2,fmt,gfmt,pfile
      character*80 xtitle,ztitle,gtitle,titlel,title2,fmt*8,gfmt*8,
     &pfile*32
      character*4 bnum,cnum,esc*30
      do 500 i=1,nbody
c     plot body
c
      cscale=xbody/(xmax-xmin)
      zscale=zbody/(zmax-zmin)
      icon=0
      ipn=0
      do 15 j=1,ncorn(i)
      call line(xcorn(i,j),zcorn(i,j),1,icon,ipn)
      icon=1
   15 continue
      call line(xcorn(i,1),zcorn(i,1),1,icon,ipn)
      if (iabody.eq.0) go to 500
c
c     annotate body
c
      itry=1
      write(bnum,"(i4)")i
      lnum=ideblank(bnum,4)
c
c     locate center of body
```



```
      bzmax=zcorn(i,1)
      bzmin=zcorn(i,1)
      bxmax=xcorn(i,1)
      bxmin=xcorn(i,1)
      do 20 k=1,ncorn(i)
      bzmax=amax1(bzmax,zcorn(i,k))
      bzmin=amin1(bzmin,zcorn(i,k))
      bxmax=amax1(bxmax,xcorn(i,k))
      bxmin=amin1(bxmin,xcorn(i,k))
   20 continue
      xc=(bxmax-bxmin)/2+bxmin
      zc=(bzmax-bzmin)/2.+bzmin
   30 continue
      itest=incpt(xc,zc,xcorn,zcorn,ncorn(i),i,x1,x2)
      if (itest.ge.0) go to 50
c
c     come up with another xc,zc
c
      itry=itry+1
      if (itry.ge.10) go to 500
      xc=(bxmax-bxmin)*itry/10.+bxmin
      zc=(bzmax-bzmin)*itry/10.+bzmin
      go to 30
   50 continue
      if ((x1.le.xmin).and.(x2.gt.xmin)) x1=xmin
      if ((x2.ge.xmax).and.(x1.lt.xmax)) x2=xmax
      xc=(x2-x1)/2.+x1
      if (mod(iabody,10).gt.1) go to 100
      blsize=csize
      if ((x2-x1)*cscale.lt.lnum*blsize)
     &blsize=(x2-x1)*cscale/float(lnum)
      if (blsize.lt..04) blsize=.04
      xoff=-(lnum-1)*blsize/2.
      call char(xc,zc,bnum,lnum,2,blsize,0.,xoff,0.)
      go to 490
  100 continue
  490 continue
      if (iabody.lt.10) go to 500
c
c     label corners
c
      do 550 k=1,ncorn(i)
      write(cnum,"(i4)")k
      lcnum=ideblank(cnum,4)
c
c     decide size of label
c
      temp=amax1(abs(xcorn(i,k)-xc),abs(zcorn(i,k)-zc))
      clsize=amin1(rnsize,temp)
c
c     calculate xoff & yoff for corner label
c
      dist=((xcorn(i,k)-xc)**2.+(zcorn(i,k)-zc)**2.)**.5
```

```
      if (dist.eq.0) go to 550
      xoff=(clsize*(xc-xcorn(i,k)))/dist
      yoff=(clsize*(zcorn(i,k)-zc))/dist
c
c     print corner label
c
      call char(xcorn(i,k),zcorn(i,k),cnum,lcnum,2,clsize,0.,xoff,yoff)
 550  continue
 500  continue
 600  continue
      return
      end
c
c     PLOT MODEL
c
c  *********************************************
      subroutine plotmodel
      external char(descriptors),xaxis(descriptors),yaxis(descriptors)
      common /block1/ nprof,dprof,regfirst,regslope,iobs,h(1000),
     &obs(1000),x(1000),z(1000),robs(1000)
      common /block2/ nbody,xcorn(50,50),zcorn(50,50),ncorn(50),
     &m(50),mi(50),md(50),mx(50),mz(50),rho(50)
      real m,mi,md,mx,mz
      common /plot/ bmargin,rlmargin,cmargin,xdel,zdel,gdel,
     &ixip,izip,igip,csize,ssize,tsize,xsize,ysize,iplotr,xbody,
     &zbody,ganom,xmin,xmax,zmin,zmax,gmin,gmax,iabody,ixunit,
     &nxchar,nzchar,ngchar,ntlchar,nt2char,vex,scalel,gscalel
     &,ixnum,ignum,itek,ifast,ixrev,iyrev,rnsize,tmargin,rmargin
      common /title/ xtitle,ztitle,gtitle,titlel,title2,fmt,gfmt,pfile
      character*80 xtitle,ztitle,gtitle,titlel,title2,fmt*8,gfmt*8,
     &pfile*32
      common /plot2/ xp(4),yp(4),dxp(2),dyp(2),xbd,ybd,ip
      character*30 esc
      dimension isl(5),ypa(4),dypa(2)
      isl(1)=0
      if ((iplotr.ne.1).or.(itek.ne.1)) go to 1
c
c     disable dialog area
c
      esc="KA0"
      call send_esc(esc)
c
c     delete view
c
      esc="RK0"
      call send_esc(esc)
c
c     issue page command
c
      call page
c
c
c
c      esc="KN1"
```



```
c      call send_esc(esc)
c
c     4 line dialog area
c
      esc="LL4"
      call send_esc(esc)
      esc="KA1"
      call send_esc(esc)
  1   continue
      if ((iplotr.eq.1).and.(itek.eq.0)) write(6,2001)
 2001 format(" !wor 30 !gra 1,35 !shrink")
      if ((iplotr.ne.1).or.(itek.ne.1)) go to 20
c
c     delete segment 99
c
      esc="SKF3"
      call send_esc(esc)
c
c     begin segment 99
c
      esc="SOF3"
      call send_esc(esc)
 20   continue
      call pltset(iplotr,xbd,ybd,isl)
      iscale=4
      xscale=1.
      yscale=1.
c
c     if xbd or ybd is smaller than plot size, then re-scale to fit
c
      if (xbd.lt.xsize) xscale=xbd/xsize
      if (ybd.lt.ysize) yscale=ybd/ysize
c
c     set up body plotting area
c
      xp(1)=xbody*xscale
      yp(1)=zbody*yscale
      xp(2)=0
      yp(2)=0
      xp(3)=rlmargin*xscale
      yp(3)=bmargin*yscale
      xp(4)=xsize*xscale
      yp(4)=ysize*yscale
      if (ixrev.eq.0) then
      dxp(1)=xmin
      dxp(2)=xmax
      else
      dxp(1)=xmax
      dxp(2)=xmin
      end if
      if (iyrev.eq.0) then
      dyp(1)=zmax
      dyp(2)=zmin
      else
```

```
      dyp(1)=zmin
      dyp(2)=zmax
      end if
      if (zbody.le.0) go to 100
      call scale(dxp,dyp,xp,yp,4,ier)
      iscale=3
      if (ier.ne.0) go to 1313
      call bodyplot
      call char(x,z,"+",nprof,1,ssize,0.,0.,0.)
      if (ifast.ne.0) go to 50
      call neatl
      call xaxis(dxp,dyp,xp,xdel,ixip,csize*xscale,fmt,ixnum)
      call yaxis(dyp,dxp,yp,zdel,izip,csize*yscale,fmt,ixnum)
      if (nxchar.gt.0)
     &call char(xp(3),0.,xtitle,nxchar,3,csize*yscale,0.,0.,
     &+csize*yscale)
      if (nzchar.gt.0)
     &call char(0.,yp(3),ztitle,nzchar,3,csize*xscale,1.5706,0.,
     &-csize*xscale)
   50 continue
c
c     set up anomaly area
c
 100  continue
      if (ganom.le.0) go to 200
      ypa(1)=ganom*yscale
      ypa(2)=0
      ypa(3)=(bmargin+zbody+cmargin)*yscale
      ypa(4)=yp(4)
      dypa(1)=gmin
      dypa(2)=gmax
      call scale(dxp,dypa,xp,ypa,iscale,ier)
      if (ier.ne.0) go to 1313
      call line(x,h,nprof,0,0)
      call line(x,robs,nprof,0,1)
      if (ifast.ne.0) go to 150
      call neatl
      call xaxis(dxp,dypa,xp,xdel,ixip,csize*xscale,fmt,0)
      call yaxis(dypa,dxp,ypa,gdel,igip,csize*yscale,gfmt,ignum)
      if (ngchar.gt.0)
     &call char(0.,ypa(3),gtitle,ngchar,3,csize*xscale,1.5706,0.,
     &-csize*xscale)
      call char(dxp,dypa,"dashed=observed, solid=calculated",33,2,
     &csize*yscale,0.,0.,-csize*yscale)
 150  continue
c
c     Main titles
c
 200  continue
      if (ntlchar.gt.0)
     &call char(dxp,dypa(2),titlel,ntlchar,2,tsize*yscale,0.,0.,
     &2.5*tsize*yscale)
      if (nt2char.gt.0)
     &call char(dxp,dypa(2),title2,nt2char,2,tsize*yscale,0.,0.,
```



```
     &1.*tsize*yscale)
      call endpt(0)
      if ((iplotr.ne.1).or.(itek.ne.1)) go to 300
      esc="SC"
      call send_esc(esc)
 300  continue
      return
c
c     Oops, bad scale call
c
 1313 continue
      write(6,2000)
 2000 format(" Bad call to scale- check the plot boundaries.")
      call endpt(0)
      return
      end
c
c        DEFAULT PLOT PARMS
c
c
c  ******************************************
      subroutine defaultplotparms(project)
      character*40 project
      common /hyper/ iplot
      common /block1/ nprof,dprof,regfirst,regslope,iobs,h(1000),
     &obs(1000),x(1000),z(1000),robs(1000)
      common /block2/ nbody,xcorn(50,50),zcorn(50,50),ncorn(50),
     &m(50),mi(50),md(50),mx(50),mz(50),rho(50)
      real m,mi,md,mx,mz
      common /block3/ azin,fi,fd,itype,xconv,zconv
      common /plot/ bmargin,rlmargin,cmargin,xdel,zdel,gdel,
     &ixip,izip,igip,csize,ssize,tsize,xsize,ysize,iplotr,xbody,
     &zbody,ganom,xmin,xmax,zmin,zmax,gmin,gmax,iabody,ixunit,
     &nxchar,nzchar,ngchar,ntlchar,nt2char,vex,scale,gscale
     &,ixnum,ignum,itek,ifast,ixrev,iyrev,rnsize,tmargin,rmargin
      common /title/ xtitle,ztitle,gtitle,titlel,title2,fmt,gfmt,pfile
      character*80 xtitle,ztitle,gtitle,titlel,title2,fmt*8,gfmt*8,
     &pfile*32

c
c     default page size= 8.5 x 11 inches
c
      xsize=11.0
      ysize=8.5
c
c     default plotter=9 for versatec
c
      if (iplot.eq.0) iplotr=9
c
c     subdivide page into two plotting areas each 3x9.5 inches
c
      zbody=3.
      xbody=9.5
      ganom=3.
```

```
c
c     set bottom, left, and center margins
c
      bmargin=1.0
      rlmargin=1.0
      cmargin=.5
      tmargin=1.
      rmargin=.5
c
c     character size, symbol size, and title character size
c
      csize=.125
      ssize=.125
      tsize=.25
      rnsize=.125
c
c     ixunit=0 for feet, iabody=1 for body number annotation
c
      ixunit=0
      iabody=1
c
c     formats and string length for axis labels
c
      fmt="(f8.1)"
      gfmt="(f8.1)"
      ixnum=8
      ignum=8
      ifast=0
c
c     scan data to determine maxima and minima
c
      zmin=zcorn(1,1)
      zmax=zcorn(1,1)
      xmin=xcorn(1,1)
      xmax=xcorn(1,1)
      do 10 i=1,nbody
      do 10 j=1,ncorn(i)
      xmin=aminl(xmin,xcorn(i,j))
      xmax=amaxl(xmax,xcorn(i,j))
      zmin=aminl(zmin,zcorn(i,j),z(i))
      zmax=amaxl(zmax,zcorn(i,j),z(i))
   10 continue
      gmin=h(1)
      gmax=h(1)
      do 20 i=1,nprof
      gmin=aminl(gmin,h(i),robs(i))
      gmax=amaxl(gmax,h(i),robs(i))
   20 continue
c
c     round max and min to powers of ten and set label factors.
c
      call round(xmin,xmax,xdel,ixip,ixpower)
      call round(zmin,zmax,zdel,izip,izpower)
      call round(gmin,gmax,gdel,igip,igpower)
```



```
c
c     calculate effective scale and vertical exageration
c
      scale=(xmax-xmin)/xbody
      gscale=(gmax-gmin)/ganom
      vex=scale/((zmax-zmin)/zbody)
c
c     construct default plot file name
c
      ilen=itlen(project,30)
      pfile=project(1:ilen)//".plt"
c
c     make up default titles
c
      title1=project
      title2=" "
      xtitle=" "
      ztitle="Depth"
      gtitle="Anomaly"
      nxchar=0
      ngchar=7
      nzchar=5
      ntlchar=ilen
      nt2char=0
      return
      end
c
c        CHANGE PARMS
c
c ********************************************
      subroutine changeparm
      common /plot/ bmargin,rlmargin,cmargin,xdel,zdel,gdel,
     &ixip,izip,igip,csize,ssize,tsize,xsize,ysize,iplotr,xbody,
     &zbody,ganom,xmin,xmax,zmin,zmax,gmin,gmax,iabody,ixunit,
     &nxchar,nzchar,ngchar,ntlchar,nt2char,vex,scale,gscale
     &,ixnum,ignum,itek,ifast,ixrev,iyrev,rnsize,tmargin,rmargin
      common /title/ xtitle,ztitle,gtitle,title1,title2,fmt,gfmt,pfile
      character*80 xtitle,ztitle,gtitle,title1,title2,fmt*8,gfmt*8,
     &pfile*32
      character*8 iform,rform,quest*80,aform
      dimension imod(50)
      iform="(i10)"
      rform="(f15.4)"
      aform="(a80)"
      ichange=-1
      do 1 i=1,50
    1 imod(i)=0
      go to 3
    2 ichange=0
    3 quest="Change parameter # (-1 for all, 0 for done)"
      ival=iiquest(quest,ichange,iform,0)
      goto (6,10,20,30,40,50,60,70)iabs(ichange)+1
      goto (80,90,100,110,120,130,140,150)iabs(ichange)-7
      goto (160,170,180,190,200,210,220,230)iabs(ichange)-15
```

```
      goto (240,250,260,270,280,290,300,310)iabs(ichange)-23
      goto (320,330,340,350,360,370,380,390,400)iabs(ichange)-31
      goto (410,420,430,440,450)iabs(ichange)-40
      goto 2
c
c   control comes here for return
c
    6 continue
      call chekpp(imod)
      return
c
c     begin of parm change code
c
   10 quest="1. Total plot width (inches)"
      ival=irquest(quest,xsize,rform,0)
      go to (2, 15) ival+2
      imod(1)=1
   15 if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
c
   20 quest="2. Total plot height"
      ival=irquest(quest,ysize,rform,0)
      go to (2, 25) ival+2
      imod(2)=1
   25 if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
c
   30 quest=
     &"3. Plotter # (1=TKX, 8=HSTN, 9=VRSTK, 10=TKX 4112, 11=VT100)"
      if (itek.eq.1) iplotr=10
      if (itek.eq.2) iplotr=11
      ival=iiquest(quest,iplotr,iform,0)
      itek=0
      if (iplotr.eq.10) itek=1
      if (iplotr.eq.11) itek=2
      if (iplotr.ge.10) iplotr=1
      if ((ichange.ge.0).or.(ival.eq.-1)) go to 2
c
   40 quest="4. Height of body plot area"
      ival=irquest(quest,zbody,rform,0)
      go to (2, 45) ival+2
      imod(4)=1
   45 if ((ichange.ge.0).or.(ival.eq.-1)) go to 2
c
   50 quest="5. Width of body plot area"
      ival=irquest(quest,xbody,rform,0)
      go to (2, 55) ival+2
      imod(5)=1
   55 if ((ichange.ge.0).or.(ival.eq.-1)) go to 2
c
   60 quest="6. Height of anomaly area"
      ival=irquest(quest,ganom,rform,0)
      go to (2, 65) ival+2
      imod(6)=1
   65 if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
c
```



```
   70 quest="7. Left limit of profile (data units)"
      ival=irquest(quest,xmin,rform,0)
      go to (2, 75) ival+2
      imod(7)=1
   75 if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
c
   80 quest="8. Right limit of profile"
      ival=irquest(quest,xmax,rform,0)
      go to (2, 85) ival+2
      imod(8)=1
   85 if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
c
   90 quest="9. Top limit of body plot area"
      ival=irquest(quest,zmax,rform,0)
      go to (2, 95) ival+2
      imod(9)=1
   95 if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
c
  100 quest="10. Lower limit of body plot area"
      ival=irquest(quest,zmin,rform,0)
      go to (2, 105) ival+2
      imod(10)=1
  105 if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
c
  110 quest="11. Lower limit of anomaly plot area"
      ival=irquest(quest,gmin,rform,0)
      go to (2, 115) ival+2
      imod(11)=1
  115 if ((ichange.ge.0).or.(ival.eq.-1)) go to 2
c
  120 quest="12. Upper limit of anomaly plot area"
      ival=irquest(quest,gmax,rform,0)
      go to (2, 125) ival+2
      imod(12)=1
  125 if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
c
  130 quest="13. Bottom margin"
      ival=irquest(quest,bmargin,rform,0)
      go to (2, 135) ival+2
      imod(13)=1
  135 if ((ichange.ge.0).or.(ival.eq.-1)) go to 2
c
  140 quest="14. Left margin"
      ival=irquest(quest,rlmargin,rform,0)
      go to (2, 145) ival+2
      imod(14)=1
  145 if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
c
  150 quest="15. Center margin"
      ival=irquest(quest,cmargin,rform,0)
      go to (2, 155) ival+2
      imod(15)=1
  155 if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
c
```

```
160 quest="16. Tick mark interval on the X axis"
    ival=irquest(quest,xdel,rform,0)
    if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
c
170 quest="17. Tick mark interval on the Z axis"
    ival=irquest(quest,zdel,rform,0)
    if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
c
180 quest="18. Tick mark interval on the Anomaly axis"
    ival=irquest(quest,gdel,rform,0)
    if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
c
190 quest="19. Tick marks between each label, X axis"
    ival=iiquest(quest,ixip,iform,0)
    if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
c
200 quest="20. Tick marks between each label, Z axis"
    ival=iiquest(quest,izip,iform,0)
    if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
c
210 quest="21. Tick marks between each label, Anomaly axis"
    ival=iiquest(quest,igip,iform,0)
    if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
c
220 quest="22. Height of characters in plot annotation"
    ival=irquest(quest,csize,rform,0)
    if ((ichange.ge.0).or.(ival.eq.-1)) go to 2
c
230 quest="23. Height of symbols on profile plot"
    ival=irquest(quest,ssize,rform,0)
    if ((ichange.ge.0).or.(ival.eq.-1)) go to 2
c
240 quest="24. X axis title"
    ival=iaquest(quest,xtitle,aform,80)
    nxchar=itlen(xtitle,80)
    if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
c
250 quest="25. Z axis title"
    ival=iaquest(quest,ztitle,aform,80)
    nzchar=itlen(ztitle,80)
    if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
c
260 quest="26. Anomaly axis title"
    ival=iaquest(quest,gtitle,aform,80)
    ngchar=itlen(gtitle,80)
    if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
c
270 quest="27. Main title"
    ival=iaquest(quest,titlel,aform,80)
    ntlchar=itlen(titlel,80)
    if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
c
280 quest="28. Second title"
    ival=iaquest(quest,title2,aform,80)
```



```
    nt2char=itlen(title2,80)
    if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
c
290 quest="29. Character height for main and second title"
    ival=irquest(quest,tsize,rform,0)
    go to (2, 295) ival+2
    imod(29)=1
295 if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
c
300 quest="30. Vertical exaggeration"
    ival=irquest(quest,vex,rform,0)
    go to (2, 305) ival+2
    imod(30)=1
305 if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
310 quest="31. Body plot scale (in units/inch)"
    ival=irquest(quest,scale,rform,0)
    go to (2, 315) ival+2
    imod(31)=1
315 if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
c
320 quest="32. Anomaly scale (gammas/inch)"
    ival=irquest(quest,gscale,rform,0)
    go to (2, 325) ival+2
    imod(32)=1
325 if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
c
330 quest="33. Plot file name"
    ival=iaquest(quest,pfile,aform,30)
    if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
c

340 quest="34. Body units (0=none, 1=m, 2=km, 3=ft, 4=kf)"
    ival=iiquest(quest,ixunit,iform,0)
    if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
c
350 quest="35. Body annotation (0=none, 1=#, 2=mag, 3=grav)"
    ival=iiquest(quest,iabody,iform,0)
    if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
c
360 quest="36. Fortran format for body axis labels"
    ival=iaquest(quest,fmt,aform,8)
    if (ichange.eq.36) go to 380
    if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
c
370 quest="37. Fortran format for anomaly axis labels"
    ival=iaquest(quest,gfmt,aform,8)
    if (ichange.eq.37) go to 390
    if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
c
380 quest="38. # of characters in body axis labels"
    ival=iiquest(quest,ixnum,iform,0)
    if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
c
390 quest="39. # of characters in anomaly axis labels"
```

```
      ival=iiquest(quest,ignum,iform,0)
      if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
  400 quest="40. Fast plot (0=no, 1=yes)"
      ival=iiquest(quest,ifast,iform,0)
      if ((ichange.ge.0).or.(ival.eq.-1)) goto 2
c
  410 quest="41. Reverse X axis (0=no, 1=yes)"
      ival=iiquest(quest,ixrev,iform,0)
      if ((ichange.ge.0).or.(ival.eq.-1)) go to 2
c
  420 quest="42. Reverse Z axis (0=no, 1=yes)"
      ival=iiquest(quest,iyrev,iform,0)
      if ((ichange.ge.0).or.(ival.eq.-1)) go to 2
c
  430 quest="43. Size of body corner numbers"
      ival=irquest(quest,rnsize,rform,0)
      if ((ichange.ge.0).or.(ival.eq.-1)) go to 2
c
  440 quest="44. Top margin"
      ival=irquest(quest,tmargin,rform,0)
      go to (2,445) ival+2
      imod(44)=1
  445 if ((ichange.ge.0).or.(ival.eq.-1)) go to 2
c
  450 quest="45. Right margin"
      ival=irquest(quest,rmargin,rform,0)
      go to (2,455) ival+2
      imod(45)=1
  455 if ((ichange.ge.0).or.(ival.eq.-1)) go to 2
c
      go to 6
      end
c
c    MODEL PLOT   -  controls printer plots
c
c  *********************************************
      subroutine model_plot(project,num)
      character*40 project,quest*80,ans*1,sym*1
      common /block1/ nprof,dprof,regfirst,regslope,iobs,h(1000),
     &obs(1000),x(1000),z(1000),robs(1000)
      common /block2/ nbody,xcorn(50,50),zcorn(50,50),ncorn(50),
     &m(50),mi(50),md(50),mx(50),mz(50),rho(50)
      real m,mi,md,mx,mz
      common /block3/ azim,fi,fd,itype,xconv,zconv
      dimension p(1000,3),sym(3),xzu(2)
      data sym(1)/"*"/,sym(2)/"."/,sym(3)/"+"/,xzu(1)/"    "/,
     &xzu(2)/"    "/
      xmax=x(1)
      xmin=x(1)
      ymax=h(1)
      ymin=h(1)
      zmax=z(1)
      zmin=z(1)
      do 10 i=1,nprof
```



```
      xmax=amax1(xmax,x(i))
      xmin=amin1(xmin,x(i))
      ymax=amax1(ymax,h(i),robs(i))
      ymin=amin1(ymin,h(i),robs(i))
      zmax=amax1(zmax,z(i))
      zmin=amin1(zmin,z(i))
      p(i,1)=robs(i)
      p(i,2)=h(i)
      p(i,3)=0.
   10 continue
      if (num.ne.6)
     & call prlplt(num,xmax,xmin,ymax,ymin,1000,3,3,nprof,nprof,sym,p,x)
      if (num.eq.6)
     &call prlpt(x,p,xmax,xmin,ymax,ymin,0,0,3,1000,31,nprof,sym,80,xzu)
      if (num.eq.6) write(6,110)
  110 format(/)
      return
      end
c
c     Model Print - creates print-out of results
c
c  *********************************************
      subroutine model_print(project,ofile,num)
      character*40 project,ofile*50
      common /block1/ nprof,dprof,regfirst,regslope,iobs,h(1000),
     &obs(1000),x(1000),z(1000),robs(1000)
      common /block2/ nbody,xcorn(50,50),zcorn(50,50),ncorn(50),
     &m(50),mi(50),md(50),mx(50),mz(50),rho(50)
      real m,mi,md,mx,mz
      common /block3/ azim,fi,fd,itype,xconv,zconv
      if (num.eq.6) go to 10
      open(num,file=ofile,form="formatted",mode="out")
      call seemodel(project,num)
   10 continue
      write(num,130)
  130 format(//," Results.....",/)
      if (iobs.eq.0) go to 20
      write(num,135)
  135 format(2(" Point"," Calc",6x,"Obs",7x,x,"Calc-obs"))
      do 15 i=1,nprof
      write(num,140)i,h(i),robs(i),h(i)-robs(i)
  140 format(" ",i5,3g10.3,$)
      i=i+1
      if (i.gt.nprof) go to 15
      write(num,150)i,h(i),robs(i),h(i)-robs(i)
  150 format(" ",i5,3g10.3)
   15 continue
      write(6,155)
  155 format(/)
      if (num.ne.6) call model_plot(project,num)
      if (num.ne.6) close(num)
      return
   20 continue
      write(num,160)
```

```
 160  format(4(" Point",6x,"Calc"))
      do 30 i=1,nprof
      write(num,170)i,h(i)
 170  format(" ",i5,g10.3,$)
      if (mod(i,4).eq.0) write(num,180)
 180  format(" ")
  30  continue
      if (num.ne.6) call model_plot(project,num)
      if (num.ne.6) close(num)
      return
      end
c
c     ADD OBS  -  for adding observed values to data
c
c  *********************************************
      subroutine add_obs
      common/block1/nprof,dprof,regfirst,regslope,iobs,
     &h(1000),obs(1000),x(1000),z(1000),robs(1000)
      character*80 quest
      quest="Observed value"
      do 10 i=1,nprof
      write(6,110)i,x(i),z(i)
 110  format(" ",i5," x=",f15.4," y=",f15.4)
      ival=irquest(quest,obs(i),"(f15.4)",0)
      if (ival.eq.-1) go to 99
  10  continue
  99  iobs=1
      return
      end
c
c       PRLPLT - printer plot of results
c
c  *********************************************
      subroutine prlplt(lun,xmax,xmin,ymax,ymin,iydm1,iydm2,ny,lines,
     &last,sym,y,x)
      dimension zy(11),y(iydm1,iydm2),x(iydm1)
      character*1 sym(iydm2),blank,cr,pl,graph(101)
      data blank/" "/,cr/"-"/,pl/"+"/
      xscale=(xmax-xmin)/(lines-1.)
      yscale=(ymax-ymin)/100.
      do 10 k=1,11
10    zy(k)=10.*(k-1)*yscale+ymin
      write(lun,15) (zy(k),k=1,11)
15    format(1h1,8x,1p11e10.2)
      write(lun,20)
20    format(1h ,3x,"X VALUES",2x,20("+....."),"+",3x,"X VALUES")
      index=1
30    if(x(index).ge.xmin) go to 40
      index=index+1
      if(index.le.last) go to 30
      write(lun,35)
35    format(1h0,"NO X IS GREATER THAN XMIN")
      return
40    x1=xmin
```



```
      do 80 i=1,lines
      if(mod(i,5).eq.1) go to 45
      graph(1)=cr
      graph(101)=cr
      go to 50
45    graph(1)=pl
      graph(101)=pl
50    do 55 j=2,100
55    graph(j)=blank
      x2=((i-.5)*(xmax-xmin))/(lines-1.)+xmin
57    if(index.gt.last) go to 70
      if(x(index).ge.x2) go to 70
      if(x(index).lt.x1) go to 65
      do 60 j=1,ny
      if(y(index,j).lt.ymin.or.y(index,j).gt.ymax) go to 60
      iy=(y(index,j)-ymin)/yscale+1.5
      graph(iy)=sym(j)
60    continue
      index=index+1
      go to 70
65    index=index+1
      go to 57
70    xes=(i-1)*xscale+xmin
      write(lun,75) xes,(graph(j),j=1,101),xes
75    format(1h ,2x,1pe9.2,2x,101a1,1pe11.2)
      x1=x2
80    continue
      write(lun,20)
      write(lun,85) (zy(k),k=1,11)
85    format(1h ,8x,1p11e10.2)
      write(lun,90) yscale,xscale
90    format(1h0,3x,"YSCALE =",1pe11.4,4x,"XSCALE =",1pe11.4)
      return
      end
c
c      PRLPT  -  printer plot stolen from Ponce
c
c  *********************************************
      subroutine prlpt(x,y,xmax,xmin,ymax,ymin,tmax,tmin,ny,iydm,
     &      lines,last,sym1,iw,xzunit)
c
c     long printer plot, one x, several y values
c     arguments
c   modified to label grav,mag and to allow variable plot width
c   7/8/80 modified for use with Talwanii program
c     x     the array of x coordinates
c     y     the array of y coordinates
c     xmax  the largest value of x
c     xmin  the smallest value of x
c     ymax  the largest value of y
c     ymin  the smallest value of y
c     tmax  the max of auxiliary variable t (for axis label)
c     tmin  the min of auxiliary variable t
c     ny    the number of y variables
```

```
c     iydm  the actual dimension of the first subscript of y
c     lines the number of lines to be used for the x axis
c     last  the number of data points
c     sym   the symbols to plot, one for each y
c     iw    the width of plot (eg 80 for teletype, 126 for printer)
c
      dimension x(last),y(iydm,ny),syml(ny),zy(11),tzy(11),graph(112)
      dimension tick(22),ticky(22),xzunit(2)
      dimension sym(3)
      data io/6/
      data sym(1)/"O"/,sym(2)/"X"/,sym(3)/"R"/
      data blank/" "/,cr/"-"/,pl/"<"/
      data (tick(i),i=1,22)/22*"...."/,(ticky(i),i=1,22)/22*"+"/
c
c      write(6,1313)ny,lines,last,iw
1313  format(" number of y=",i3," lines=",i3," last=",i3," iw=",i3)
c      do 2 i=1,last
c  2     write(6,1314)x(i),y(i,1),y(i,2),y(i,3)
1314  format(" ",4e10.2)
      write(6,1315)
1315  format(" Plot of total attraction.")
      write(6,1316)
1316  format(" Plot symbols: O=Observed, X=Computed, *=Intersection.")
      xscale=(xmax-xmin)/(lines-1.)
      ir=min0(iw-15,111)
      icol=ir/10+1
      icol2=icol*2-1
      irpl=ir+1
      tyscal=(tmax-tmin)/ir
      yscale=(ymax-ymin)/ir
      do 10 k=1,icol
   10 zy(k)=10.*(k-1)*yscale+ymin
      write(io,15) (zy(k),k=1,icol)
   15 format(/,"  anomaly ",1p12e10.2)
      do 11 k=1,icol
11    tzy(k)=10.*(k-1)*tyscal+tmin
c      write(io,16) (tzy(k),k=1,icol)
16    format("    mag    ",1p12e10.2,/)
169   write(io,20) xzunit(1),xzunit(2),ticky(1),
     & (tick(k),ticky(k),k=1,icol2)
   20 format("    ",2a4,3x,a1,22(a4,a1))
      index=1
   30 if(x(index).ge.xmin) go to 40
      index=index+1
      if(index.le.last) go to 30
      write(io,35)
   35 format(1h0,"no x is greater than xmin")
      return
   40 x1=xmin
      do 80 i=1,lines
      if(mod(i,5).eq.1) go to 45
      graph(1)=cr
      graph(irpl)=cr
      go to 50
```



```
   45 graph(1)=pl
      graph(irpl)=pl
   50 do 55 j=2,ir
   55 graph(j)=blank
      x2=((i-.5)*(xmax-xmin))/(lines-1.)+xmin
   57 if(index.gt.last) go to 70
      if(x(index).ge.x2) go to 70
      if(x(index).lt.x1) go to 65
      do 60 j=1,ny
      if(y(index,j).lt.ymin.or.y(index,j).gt.ymax) go to 60
      iy=(y(index,j)-ymin)/yscale+1.5
      if (j.eq.3) go to 60
      if (graph(iy).ne.blank) graph(iy)="*"
      if (graph(iy).eq.blank) graph(iy)=sym(j)
   60 continue
      index=index+1
      go to 70
   65 index=index+1
      go to 57
   70 xes=(i-1)*xscale+xmin
      write(io,75) xes,(graph(j),j=2,irpl)
   75 format(1h ,2x,1pe9.2,2x,"+",101a1)
      x1=x2
   80 continue
      write(io,20) xzunit(1),xzunit(2),ticky(1),
     & (tick(k),ticky(k),k=1,icol2)
      write(io,85) (zy(k),k=1,icol)
   85 format(/," anomaly ",1p12e10.2)
c      write(io,86) (tzy(k),k=1,icol)
86    format("    mag    ",1p12e10.2)
      return
      end
c
c     TGRAV- Equations from Grant & West, 1965, Interpretation Theory
c            in Applied Geophysics, page 289.
c
c  ******************************************
      subroutine tgrav
      common /block1/ nprof,dprof,regfirst,regslope,iobs,h(1000),
     &obs(1000),x(1000),z(1000),robs(1000)
      common /block2/ nbody,xcorn(50,50),zcorn(50,50),ncorn(50),
     &m(50),mi(50),md(50),mx(50),mz(50),rho(50)
      common /block3/ azim,fi,fd,itype,xconv,zconv
c
      do 30 i=1,nprof
      sumi=0.0
c
      do 20 j=1,nbody
      if (rho(j).eq.0.0) go to 20
      nvert=ncorn(j)
      sumj=0.0
c
      do 10 k=1,nvert
      k2=k+1
```

```
      if (k.ge.nvert) k2=1

      a1=(xcorn(j,k2)-xcorn(j,k))*xconv
      a2=(zcorn(j,k2)-zcorn(j,k))*zconv
      if (a2.eq.0) a2=.00001
      a=a1/a2

c
      xk=(xcorn(j,k)-x(i))*xconv
      zk=(zcorn(j,k)-z(i))*zconv
      if (zk.eq.0) zk=.00001
c
      xkp1=(xcorn(j,k2)-x(i))*xconv
      zkp1=(zcorn(j,k2)-z(i))*zconv
      if (zkp1.eq.0) zkp1=.00001
c
      b1=xk*zkp1-xkp1*zk
      b=b1/a2
c
      c=b/(1+a**2)
      tk=atan2(xk,zk)
      tkp1=atan2(xkp1,zkp1)
c
      d1=xkp1**2+zkp1**2
      d2=xk**2+zk**2
c
      sumj=sumj+c*(.5*alog(d1/d2)+a*(tkp1-tk))
c      write(37,100)i,j,k,tk,zk,xk,tkp1,zkp1,xkp1,sumj
  100 format(x,3i3,7f9.4)
   10 continue
      sumi=sumi+13.34*rho(j)*sumj
   20 continue
      h(i)=sumi
   30 continue
      return
      end
c
c     ENTER_UNITS querys user for measurement units
c
c ********************************************
      subroutine enter_units
      common /block3/ azim,fi,fd,itype,xconv,zconv
      dimension conv(6)
      character*80 quest,ans*1
      conv(1)=0.001
      conv(2)=1.0
      conv(3)=0.3048e-3
      conv(4)=0.3048
      conv(5)=1.609
      xconv=1.0
      zconv=1.0
      ins=2
      write(6,120)
  120 format(" Unit codes: 1-meters, 2-kilometers, 3-feet, 4-kilofeet,"
```



```
     &,/," 5-miles, 6-other")
      quest="Horizontal (x-axis) units"
      ival=iiquest(quest,ins,"(i3)",0)
      if (ins.le.5) xconv=conv(ins)
      if (ins.le.5) go to 10
      quest="Real number for conversion to kilometers"
      ival=irquest(quest,xconv,"(e15.5)",0)
   10 continue
      quest="Vertical (z-axis) units"
      ival=iiquest(quest,ins,"(i3)",0)
      if (ins.le.5) zconv=conv(ins)
      if (ins.le.5) go to 20
      quest="Number for conversion to kilometers"
      ival=irquest(quest,zconv,"(e15.5)",0)
   20 continue
      end
c
c     READ_2MAG_MODEL   -  reads input file for Blakely's 2mag
c
c ********************************************
      subroutine read_2mag_model
c
      common /hyper/ iplot
      common /block1/ nprof,dprof,regfirst,regslope,iobs,h(1000),
     &obs(1000),x(1000),z(1000),robs(1000)
      common /block2/ nbody,xcorn(50,50),zcorn(50,50),ncorn(50),
     &m(50),mi(50),md(50),mx(50),mz(50),rho(50)
      real m,mi,md,mx,mz
      common /block3/ azim,fi,fd,itype,xconv,zconv
      character*40 filename,ofilename,head*4
      write(6,100)
  100 format(" 2mag input file name?",$)
      read(5,*)ofilename
      open(13,file=ofilename,mode="in",form="formatted")
      istart=0
      iobs=0
      nbody=0
   10 read(13,120,end=800)head
  120 format(a4)
      if (head.eq."STAR") go to 200
      if (head.eq."VSL ") go to 300
      if (head.eq."VSH ") go to 400
      if (head.eq."PROF") go to 500
      if (head.eq."PLOT") go to 600
      if (head.eq."RANG") go to 600
      if (head.eq."SCAL") go to 600
      if (head.eq."BODY") go to 700
      write(6,122)head
  122 format(" Couldn't figure out the ",a4," card- so it was skipped.")
      go to 10
c
c     START card
c
  200 continue
```

```
      istart=istart+1
      if (istart.ge.2) go to 800
      read(13,*)nprof,xstart,xdelt,regfirst,regslope,fi,fd,azim,
     &zstart,zdelt
      if (nprof.gt.1000)nprof=1000
      x(1)=xstart
      z(1)=zstart
      do 20 i=2,nprof
      z(i)=z(i-1)+zdelt
      x(i)=x(i-1)+xdelt
   20 continue
      go to 10
c
c     VSL card
c
  300 continue
      call array_read(x,nprof)
      go to 10
c
c     VSH card
c
  400 continue
      call array_read(z,nprof)
      go to 10
c
c     PROF card
c
  500 continue
      iobs=1
      call array_read(obs,nprof)
      go to 10
c
c     PLOT, RANGE, and SCALE cards- skip
c
  600 continue
      read(13,125)
  125 format()
      go to 10
c
c     BODY card
c
  700 continue
      nbody=nbody+1
      read(13,*)ncorn(nbody),mi(nbody),md(nbody),m(nbody)
      read(13,*)(xcorn(nbody,i),zcorn(nbody,i),i=1,ncorn(nbody))
      go to 10
c
c     All done
c
  800 continue
      dprof=0
      itype=0
      xconv=1
      zconv=1
```



```
      iplot=0
      close(13)
      return
      end
c
c      READ 2DGRAV MODEL - inputs command file for 2dgrav3 or 4
c
c  ******************************************
      subroutine read_2dgrav_model
      common /hyper/ iplot
      common /block1/ nprof,dprof,regfirst,regslope,iobs,h(1000),
     &obs(1000),x(1000),z(1000),robs(1000)
      common /block2/ nbody,xcorn(50,50),zcorn(50,50),ncorn(50),
     &m(50),mi(50),md(50),mx(50),mz(50),rho(50)
      real m,mi,md,mx,mz
      common /block3/ azim,fi,fd,itype,xconv,zconv
      character*50 ofilename,sam*30,hunit*4,salt*8,zunit*4
c
      write(6,100)
  100 format(" 2dgrav input file name?",$)
      read(5,*)ofilename
      open(13,file=ofilename,mode="in",form="formatted")
   10 read(13,120,end=999)sam
  120 format(a30)
      write(6,130)sam
  130 format(" Reading profile:",a30)
c
c     read units, points in profile, number of bodies, and constant
c
      read(13,*)hunit,zunit,nprof,nbody,con
c
c     read profile
c
      do 20 i=1,nprof
   20 read(13,*)x(i),z(i),obs(i)
c
c     set x and z conversions to km
c
      xconv=1.
      zconv=1.
c
      if ((hunit.eq."FEET").or.(hunit.eq."feet")) xconv=1/3280.84
      if ((zunit.eq."FEET").or.(zunit.eq."feet")) zconv=1/3280.84
c
      if ((hunit.eq."KILF").or.(hunit.eq."kilf")) xconv=1/3.28084
      if ((zunit.eq."KILF").or.(zunit.eq."kilf")) zconv=1/3.28084
c
      if ((hunit.eq."MILE").or.(hunit.eq."mile")) xconv=1.6093
      if ((zunit.eq."MILE").or.(zunit.eq."mile")) zconv=1.6093
c
      if ((hunit.eq."METR").or.(hunit.eq."metr")) xconv=1/1000.
      if ((zunit.eq."METR").or.(zunit.eq."metr")) zconv=1/1000.
c
c     read bodies
```

```
c
      do 30 i=1,nbody
      read(13,*)rho(i),ncorn(i),salt
      do 40 j=1,ncorn(i)
   40 read(13,*)xcorn(i,j),zcorn(i,j)
   30 continue
c
c     all done
c
  999 continue
      regfirst=-con
      regslope=0
      azim=0
      fi=0
      fd=0
      itype=1
      iplot=0
      call sub_regional
      close(13)
      return
      end
c
c     ARRAY READ - variable format read of formatted real arrays
c
c  ********************************************
      subroutine array_read(arr,len)
      dimension arr(len)
      read(13,*)(arr(i),i=1,len)
      return
      end
c
c      MATCH_PT   - matches a an X,Y pair within 5 screen points
c
c      when first called, ibod & icorn are 0,0- then subsequent calls
c      search for matches from ibod, icorn on... finally returns 0,0
c      when no more matches are found.
c
c  ********************************************
      subroutine match_pt(x,y,ibod,icorn)
      common /block2/ nbody,xcorn(50,50),zcorn(50,50),ncorn(50),
     &m(50),mi(50),md(50),mx(50),mz(50),rho(50)
      common /plot2/xp(4),yp(4),dxp(2),dyp(2),xbd,ybd,ip
c
      ibst=1
      if (ibod.ne.0) ibst=ibod
      icst=icorn+1
c
c     xtol=5 screen units (here converted to body units)
c
      xtol=((dxp(2)-dxp(1))/xp(1))*(20*(xp(4)/4096))
      ytol=((dyp(2)-dyp(1))/yp(1))*(20*(yp(4)/3127))
      xtol=abs(xtol)
      ytol=abs(ytol)
```



```
c      write(6,1314)xtol,ytol
 1314 format(" xtol=",f10.4," ytol=",f10.4)
c
c     go through bodies in acsending order looking for match within tol
c
      do 30 i=ibst,nbody
      do 20 j=icst,ncorn(i)
      if ((abs(xcorn(i,j)-x).le.xtol).and.((abs(zcorn(i,j)-y).le.ytol)))
     & go to 99
   20 continue
      icst=1
   30 continue
c
c     no matches
c
      ibod=0
      icorn=0
      return
c
c     found a match
c
   99 continue
      ibod=i
      icorn=j
      return
      end
c
c     GET_GIN_PT - gets a point from the Tektronix 4112 terminal
c
c  ********************************************
      subroutine get_gin_pt(a,x,y)
      integer hix,hiy,extra,xext,yext
      character*30 esc,gin*20,a*1
c
c     set the EOM character for the terminal
c
      esc="NC=="
      call send_esc(esc)
c
c     set Graphic Input (GIN) mode for one event
c
      esc="IE01"
      call send_esc(esc)
c
c     read in the gin point
c
      read(5,100)gin
  100 format(a20)
c      write(6,110) gin
  110 format(x,a20)
c
c     character typed
c
      a=gin(1:1)
```

```
      if (ichar(a).gt.20) go to 13
      gin(1:19)=gin(2:20)
      a=gin(1:1)
   13 continue
c
c     decode x,y points (refer to 4112 manual for explanation)
c
      hiy=mod(ichar(gin(2:2)),32)
      extra=mod(ichar(gin(3:3)),16)
      loy=mod(ichar(gin(4:4)),32)
      hix=mod(ichar(gin(5:5)),32)
      lox=mod(ichar(gin(6:6)),32)
      xext=mod(extra,4)
      yext=extra/4
c
c     assemble x and y
c
      x=hix*128+lox*4+xext
      y=hiy*128+loy*4+yext
c
c      write(6,1313)x,y
 1313 format(" In terminal coords x=",f5.0," y=",f5.0)
      return
      end
c
c     GET_BODY_PT - gets a body point from the 4112 terminal
c
c  ********************************************
      subroutine get_body_pt(a,x,y)
      common /plot2/xp(4),yp(4),dxp(2),dyp(2),xbd,ybd,ip
      character*1 a
c
      call get_gin_pt(a,xg,yg)
c
c     convert x,y to inches
c
      xi=xg*xbd/4096
      yi=yg*ybd/3127
c
c     convert to body units
c
      xb=xi-xp(3)
      yb=yi-yp(3)
      x=xb*(dxp(2)-dxp(1))/xp(1)+dxp(1)
      y=yb*(dyp(2)-dyp(1))/yp(1)+dyp(1)
c
c      write(6,1313)x,y
 1313 format(" In body coords x=",f10.3," y=",f10.3)
      return
      end
c
c     SUB_REGIONAL - subtract regional from observed
c
c  ********************************************
```



```
      subroutine sub_regional
      common /block1/ nprof,dprof,regfirst,regslope,iobs,h(1000),
     &obs(1000),x(1000),z(1000),robs(1000)
      do 10 i=1,nprof
   10 robs(i)=obs(i)-(regfirst+regslope*(x(i)-x(1)))
      return
      end
c
c      CORN_CHA
c
c     change corner
c  ********************************************
      subroutine corn_cha(ibod,icorn)
      common /block2/ nbody,xcorn(50,50),zcorn(50,50),ncorn(50),
     &m(50),mi(50),md(50),mx(50),mz(50),rho(50)
      real m,mi,md,mx,mz
      character*80 quest,rform*8
      rform="(f15.3)"
 1850 quest="X coordinate"
      ival=irquest(quest,xcorn(ibod,icorn),rform,0)
      if (ival.eq.-1) return
      quest="Z coordinate"
      ival=irquest(quest,zcorn(ibod,icorn),rform,0)
      return
      end
c
c      CORN_DEL
c
c
c     delete corner
c
c  ********************************************
      subroutine corn_del(ibod,icorn)
      common /block2/ nbody,xcorn(50,50),zcorn(50,50),ncorn(50),
     &m(50),mi(50),md(50),mx(50),mz(50),rho(50)
 1900 continue
      write(6,1910)icorn,ibod
 1910 format(" Corner #",i2," deleted from body:",i2)
      if (icorn.eq.ncorn(ibod)) go to 1930
      do 1920 i=icorn,ncorn(ibod)-1
      xcorn(ibod,i)=xcorn(ibod,i+1)
      zcorn(ibod,i)=zcorn(ibod,i+1)
 1920 continue
 1930 continue
      ncorn(ibod)=ncorn(ibod)-1
      return
      end
c
c      CORN_ADD
c
c
c     add corner
c
c  ********************************************
```

```
      subroutine corn_add(ibod,icorn)
      common /block2/ nbody,xcorn(50,50),zcorn(50,50),ncorn(50),
     &m(50),mi(50),md(50),mx(50),mz(50),rho(50)
      character*80 quest
      character*10 rform
      rform="(f15.4)"
1950  xt=0.
      zt=0.
      if (ncorn(ibod).ge.50) return
      quest="X coordinate"
      ival=irquest(quest,xt,rform,0)
      if (ival.eq.-1) return
      quest="Z coordinate"
      ival=irquest(quest,zt,rform,0)
      if (ival.eq.-1) return
      call new_pt(ibod,icorn,xt,zt)
      return
      end
c
c     NEW_PT  -  adds a new point to a body
c
c ********************************************
      subroutine new_pt(ibod,icorn,xt,zt)
      common /block2/ nbody,xcorn(50,50),zcorn(50,50),ncorn(50),
     &m(50),mi(50),md(50),mx(50),mz(50),rho(50)
      if (icorn.eq.ncorn(ibod)) go to 1970
      do 1960 i=ncorn(ibod),icorn+1,-1
      xcorn(ibod,i+1)=xcorn(ibod,i)
      zcorn(ibod,i+1)=zcorn(ibod,i)
1960  continue
1970  continue
      xcorn(ibod,icorn+1)=xt
      zcorn(ibod,icorn+1)=zt
      ncorn(ibod)=ncorn(ibod)+1
      return
      end
c
c     SCREEN_EDIT   - allows cross-hair editing of model bodies
c
c ********************************************
      subroutine screen_edit
      common /plot/ bmargin,rlmargin,cmargin,xdel,zdel,gdel,
     &ixip,izip,igip,csize,ssize,tsize,xsize,ysize,iplotr,xbody,
     &zbody,ganom,xmin,xmax,zmin,zmax,gmin,gmax,iabody,ixunit,
     &nxchar,nzchar,ngchar,ntlchar,nt2char,vex,scale,gscale
     &,ixnum,ignum,itek,ifast,ixrev,iyrev,rnsize,tmargin,rmargin
      common /hyper/ iplot
      common /block2/ nbody,xcorn(50,50),zcorn(50,50),ncorn(50),
     &m(50),mi(50),md(50),mx(50),mz(50),rho(50)
      real m,mi,md,mx,mz
      character*80 quest
      character*1 ans,a,ans2,a2,vtm
      if (iplot.ne.0) go to 5
      write(6,101)
```



```
101   format(" Please use fast plot, or the off-line option to plot",
     &" the model first.")
      return
 5    continue
      if ((itek.eq.1).or.(itek.eq.2).or.(iplotr.eq.4)) go to 8
      write(6,108)
108   format(" Cross-hair editing only works with the Tektronix 4112 or",
     &" 4014 or DEC VT100 terminal.")
      return
 8    continue
      ans="n"
      quest="Do you want directions (n/y)"
      idumb=iaquest(quest,ans,"(a1)",1)
      if (ans.ne."y") go to 10
      write(6,110)
110   format(" Use the thumb-wheels to position the cross-hair over a",/,
     &" body corner.  Then type:",/,
     &"        p = to pick a point for manual editing",/,
     &"        a = to add a point after the specified point",/,
     &"        c = to change the location of a point",/,
     &"        d = to delete the point",/,
     &"        q = to quit the screen edit mode",/,
     &"        r = re-calc and re-draw of body",/,
     &" If you chose a or c, then you will be asked to pick the new",/,
     &" location with the cross-hair.  If more than one body shares",/,
     &" a corner then a new point will be requested for each match.")
 10   continue
      if (itek.ne.2) go to 11
      vtm="c"
      quest="Cross-hair or light pen (c/p)"
      idum=iaquest(quest,vtm,"(a1)",1)
      if (idum.eq.-1) return
      mode=0
      if (vtm.ne."c") mode=1
 11   continue
      write(6,115)
115   format(" Pick a point and type a,c,d,p,r or q")
      if (itek.eq.1) call get_body_pt(a,x,y)
      if (itek.eq.2) call get_vt100_pt(a,x,y,mode)
      if (iplotr.eq.4) call get_4014_pt(a,x,y)
      if ((iplotr.eq.1).and.(itek.eq.0)) call get_4014_pt(a,x,y)
      imatch=0
      ibod=0
      icorn=0
      if ((a.eq."q").or.(a.eq."/")) go to 9999
      if (a.eq."r") call quick_draw
      if (a.eq."r") go to 11
 20   continue
      if (itek.eq.1) call match_pt(x,y,ibod,icorn)
      if (itek.eq.2) call match_vt100_pt(x,y,ibod,icorn)
      if (iplotr.eq.4) call match_4014_pt(x,y,ibod,icorn)
      if ((iplotr.eq.1).and.(itek.eq.0)) call match_4014_pt(x,y,ibod,icorn)
      if ((ibod.eq.0).and.(imatch.eq.0)) go to 40
      imatch=imatch+1
```

```
      if (ibod.eq.0) go to 11
      if (a.eq."p") go to 1000
      if (a.eq."a") go to 2000
      if ((a.eq."c").or.(a.eq."m")) go to 3000
      if (a.eq."d") call corn_del(ibod,icorn)
      go to 20
c
c     no match
c
   40 continue
      write(6,120)
  120 format(" I can't find any points close to the spot you picked.",/,
     &" Please try again.....")
      go to 11
c
c     hand picked for hand edit
c
 1000 continue
      write(6,1010)ibod,icorn,xcorn(ibod,icorn),zcorn(ibod,icorn)
 1010 format(" Body #",i2," Corner #",i2," x=",f10.3," z=",f10.3)
      ans2="c"
      quest="Add, change, or delete (a/c/d)"
      idumb=iaquest(quest,ans2,"(a1)",1)
      if (ans2.eq."c") call corn_cha(ibod,icorn)
      if (ans2.eq."d") call corn_del(ibod,icorn)
      if (ans2.eq."a") call corn_add(ibod,icorn)
      go to 20
c
c     add - using GIN
c
 2000 continue
      if (ncorn(ibod).lt.50) go to 2005
      write(6,2001)ibod,ncorn(ibod)
 2001 format(" Sorry- body #",i2," already has",i3," corners.")
      go to 20
 2005 continue
      write(6,2010)ibod,icorn
 2010 format(" Give point to add after Body #",i2," Corner #",i2)
      if (itek.eq.1) call get_body_pt(a2,x2,y2)
      if (itek.eq.2) call get_vt100_pt(a2,x2,y2,mode)
      if (iplotr.eq.4014) call get_4014_pt(a2,x2,y2)
      if ((iplotr.eq.1).and.(itek.eq.0)) call get_4014_pt(a2,x2,y2)
      if (a2.ne."/") call new_pt(ibod,icorn,x2,y2)
      go to 20
c
c     change
c
 3000 continue
      write(6,3010)ibod,icorn,xcorn(ibod,icorn),zcorn(ibod,icorn)
 3010 format(" Body #",i2," corner #",i2," x=",f10.3," z=",f10.3,/,
     &" select new position.")
      if (itek.eq.1) call get_body_pt(a2,x2,y2)
      if (iplotr.eq.4) call get_4014_pt(a2,x2,y2)
      if ((iplotr.eq.1).and.(itek.eq.0)) call get_4014_pt(a2,x2,y2)
```



```
      if (itek.eq.2) call get_vt100_pt(a2,x2,y2,mode)
      if (a2.eq."/") go to 20
      xcorn(ibod,icorn)=x2
      zcorn(ibod,icorn)=y2
      go to 20
c
c     all done having fun
c
 9999 continue
      return
      end
c
c     QUICK_DRAW - quick re-calc and re-draw
c
c  ********************************************
      subroutine quick_draw
      common /block3/ azim,fi,fd,itype,xconv,zconv

      ifast=1
      if (itype.eq.0) call anomaly
      if (itype.eq.1) call tgrav
      call plotmodel
      return
      end
c
c -- rounding subroutine
c
c  ********************************************
      subroutine round(amin,amax,del,ip,ibase)
      range=amax-amin
      smax=.99999999
      smin=0.0
      if (amax.lt.0.0) smax=0.0
      if (amin.lt.0.0) smin=.99999999
      base=alog10(range)
      ibase=int(base)
      del=10.**ibase
      amax=float(int((amax/10.**ibase)+smax))*del
      amin=float(int((amin/10.**ibase)-smin))*del
      ip=10
      del=del/10.
      return
      end
c
c       PARMIO
c
c
c  ********************************************
      subroutine parmio(filename,mode)
c
c     modes:   1=write, 0=read.
c
      common /plot/ bmargin,rlmargin,cmargin,xdel,zdel,gdel,
     &ixip,izip,igip,csize,ssize,tsize,xsize,ysize,iplotr,xbody,
```

```
     &zbody,ganom,xmin,xmax,zmin,zmax,gmin,gmax,iabody,ixunit,
     &nxchar,nzchar,ngchar,ntlchar,nt2char,vex,scale,gscale
     &,ixnum,ignum,itek,ifast,ixrev,iyrev,rnsize,tmargin,rmargin
      common /title/ xtitle,ztitle,gtitle,titlel,title2,fmt,gfmt,pfile
      character*80 xtitle,ztitle,gtitle,titlel,title2,fmt*8,gfmt*8,
     &pfile*32
      character*50 filename
      if (mode.ne.0) go to 1000
      open(15,file=filename,form="unformatted",mode="in")
      read(15)bmargin,rlmargin,cmargin,xdel,zdel,gdel,ixip,izip,igip,
     &csize,ssize,tsize,xsize,ysize,iplotr,xbody,zbody,ganom,xmin,xmax,
     &zmin,zmax,gmin,gmax,iabody,ixunit,nxchar,nzchar,ngchar,ntlchar,
     &nt2char,vex,scale,gscale,ixnum,ignum,xtitle,ztitle,gtitle,titlel,
     &title2,fmt,gfmt,pfile
      close(15)
      return
1000  continue
      open(15,file=filename,form="unformatted",mode="out")
      write(15)bmargin,rlmargin,cmargin,xdel,zdel,gdel,ixip,izip,igip,
     &csize,ssize,tsize,xsize,ysize,iplotr,xbody,zbody,ganom,xmin,xmax,
     &zmin,zmax,gmin,gmax,iabody,ixunit,nxchar,nzchar,ngchar,ntlchar,
     &nt2char,vex,scale,gscale,ixnum,ignum,xtitle,ztitle,gtitle,titlel,
     &title2,fmt,gfmt,pfile,itek,ifast,ixrev,iyrev,rnsize,tmargin
     &,rmargin
      close(15)
      return
      end
c
c     CONCAT - puts suffix on file names
c  ********************************************
      subroutine concat(str1,str2,str3)
      character str1*(*),str2*(*),str3*(*)
      len=index(str2," ")
      str1=str2(1:len-1)//str3
c      write(6,100)str1
100   format(" ",a50)
      end
c
c      INCPT   - finds intercepts of the horizontal line thru a body
c
c    x,y= point to find nearest intercepts to
c    xbod,ybod= two-dimensional arrays with corners of body
c    npt= number of points in body
c    nbd= body number (first subscript of xbod, ybod)
c    x1,x2= return for closest intercepts to x,y
c
c    RETURNS:  0- x,y inside
c              1- x,y outside
c             -1- x,y on parallel edge of body
c             -2- no intercept or just one intercept
c
c  ********************************************
      integer function incpt(x,y,xbod,ybod,npt,nbd,x1,x2)
      dimension xbod(50,50),ybod(50,50),xint(100)
```



```
c
c     statement function to find intersection of line with horizontal
c
      xsect(a1,b1,a2,b2,b)=(b-b1)*((a2-a1)/(b2-b1))+a1
c
c     int is number of intersections, ipar # of parallel lines found
c
      int=1
      ipar=0
      do 50 i=1,npt-1
      xp1=xbod(nbd,i)
      yp1=ybod(nbd,i)
      xp2=xbod(nbd,i+1)
      yp2=ybod(nbd,i+1)
      call insect(x,y,xp1,yp1,xp2,yp2,xint,int,ipar)
   50 continue
      xp1=xbod(nbd,npt)
      yp1=ybod(nbd,npt)
      xp2=xbod(nbd,1)
      yp2=ybod(nbd,1)
      call insect(x,y,xp1,yp1,xp2,yp2,xint,int,ipar)
c
c     examine int and ipar and report results
c
      if (ipar.eq.0) go to 60
c
c     on parallel edge
c
      x1=xint(1)
      x2=xint(2)
      incpt=-1
      return
   60 continue
      if (int.gt.1) go to 70
c
c     no intercept
c
      x1=x
      x2=x
      incpt=-2
      return
   70 continue
      if (int.gt.2) go to 80
c
c     only one intercept
c
      x1=xint(1)
      x2=xint(2)
      incpt=-2
      return
   80 continue
c
c     two or more intercepts
c
```

```
 85   isort=0
c
c     bubble sort of intercepts
c
      do 100 j=1,int-2
      if (xint(j).le.xint(j+1)) go to 100
      xtemp=xint(j+1)
      xint(j+1)=xint(j)
      xint(j)=xtemp
 100  continue
      if (isort.ne.0) go to 85
c
c     count # of intersections less than x
c
c
c    check for duplicate intersections
c
      l=1
 190  continue
      if (l.ge.int) go to 195
      if (abs(xint(l+1)-xint(l)).gt.1e-5) go to 192
      if (l+1.eq.int) go to 193
      do 191 ll=l+1,int-1
      xint(ll)=xint(ll+1)
 191  continue
 193  int=int-1
 192  l=l+1
      go to 190
 195  continue
      if (int.le.1) go to 70
      do 200 k=1,int-1
      if (xint(k).ge.x) go to 210
 200  continue
 210  continue
      if (k.eq.1) go to 400
      if (mod(k-1,2).eq.0) go to 350
c
c     odd number of intersections to the left- inside
c
      x1=xint(k-1)
      x2=xint(k)
      incpt=0
      return
 350  continue
c
c     even number to the left, outside
c
      x1=xint(k-2)
      x2=xint(k-1)
      incpt=1
      return
 400  continue
c
c     no intersections to the left, outside
```



```
c
      x1=xint(k)
      x2=xint(k+1)
      incpt=1
      return
      end
c
c     INSECT  finds intersection with individual side
c
c  ********************************************
      subroutine insect(x,y,a1,b1,a2,b2,xint,int,ipar)
      dimension xint(100)
c
c     statement function to find intersection
c
      xsect(x1,y1,x2,y2,yloc)=(yloc-y1)*((x2-x1)/(y2-y1))+x1
c
c     check if y is between b1 and b2
c
      if ((b1.le.y).and.(b2.ge.y)) go to 100
      if ((b1.ge.y).and.(b2.le.y)) go to 100
      return
 100  continue
      if (b2-b1.eq.0.) go to 105
      xint(int)=xsect(a1,b1,a2,b2,y)
      int=int+1
      return
 105  continue
c
c     parallel
c
      xint(1)=a1
      xint(2)=a2
      ipar=ipar+1
      return
      end
c
c     FAST_PLOT  -  fast re-calc and plot
c
c  ********************************************
      subroutine fast_plot
      common /hyper/ iplot
      common /block3/ azim,fi,fd,itype,xconv,zconv
      common /plot/ bmargin,rlmargin,cmargin,xdel,zdel,gdel,
     &ixip,izip,igip,csize,ssize,tsize,xsize,ysize,iplotr,xbody,
     &zbody,ganom,xmin,xmax,zmin,zmax,gmin,gmax,iabody,ixunit,
     &nxchar,nzchar,ngchar,nt1char,nt2char,vex,scale,gscale
     &,ixnum,ignum,itek,ifast,ixrev,iyrev,rnsize,tmargin,rmargin
      character*40 project,quest*80,iform*8

      iform="(i10)"
      if (itype.eq.0) call anomaly
      if (itype.eq.1) call tgrav
```

```
      if (iplot.ne.0) go to 10
      call defaultplotparms(project)
      quest=
     &"Plotter # (1=TKX, 8=HSTN, 9=VRSTK, 10=TKX 4112, 11=VT100)"
      ival=iiquest(quest,iplotr,iform,0)
      itek=0
      if (iplotr.eq.10) itek=1
      if (iplotr.eq.11) itek=2
      if (iplotr.ge.10) iplotr=1
      ifast=1
   10 continue
      call plotmodel
      iplot=1
      return
      end
c
c         SEE PLOT PARMS
c
c  ********************************************
      subroutine seeplotparm
      common /block1/ nprof,dprof,regfirst,regslope,iobs,h(1000),
     &obs(1000),x(1000),z(1000),robs(1000)
      common /block2/ nbody,xcorn(50,50),zcorn(50,50),ncorn(50),m(50),
     &mi(50),md(50),mx(50),mz(50),rho(50)
      common /block3/ azim,fi,fd,itype,xconv,zconv
      real mx,mz,mi,md,m
      common /plot/ bmargin,rlmargin,cmargin,xdel,zdel,gdel,
     &ixip,izip,igip,csize,ssize,tsize,xsize,ysize,iplotr,xbody,
     &zbody,ganom,xmin,xmax,zmin,zmax,gmin,gmax,iabody,ixunit,
     &nxchar,nzchar,ngchar,ntlchar,nt2char,vex,scale,gscale,
     &ixnum,ignum,itek,ifast,ixrev,iyrev,rnsize,tmargin,rmargin
      common /title/ xtitle,ztitle,gtitle,titlel,title2,fmt,gfmt,pfile
      character*80 xtitle,ztitle,gtitle,titlel,title2,fmt*8,gfmt*8,
     &pfile*32
      character*32 desc
      write(6,100)
  100 format(" Current Plot Parameter Settings")
      write(6,110)
  110 format(" Parm # Description                          Value")
      write(6,120)1 ,"Width of plot (inches)        ",xsize
  120 format(2x,i4,2x,a30,x,f15.4)
      write(6,120)2 ,"Height of plot                ",ysize
      write(6,130)3 ,"Plotter (1=TKRX, 8=HN, 9=VSTK)",iplotr
  130 format(2x,i4,2x,a30,x,i10)
      write(6,120)4 ,"Height of body plot area      ",zbody
      write(6,120)5 ,"Width of plot areas           ",xbody
      write(6,120)6 ,"Height of anomaly plot area   ",ganom
      write(6,120)7 ,"Left edge of prof (data units)",xmin
      write(6,120)8 ,"Right limit of profile        ",xmax
      write(6,120)9 ,"Top limit of body plot        ",zmax
      write(6,120)10,"Bottom limit of body plot     ",zmin
      write(6,120)11,"Lower anomaly limit (gammas)  ",gmin
      write(6,120)12,"Upper anomaly limit           ",gmax
      write(6,120)13,"Bottom plot margin (inches)   ",bmargin
```



```
      write(6,120)14,"Left plot margin              ",rlmargin
      write(6,120)15,"Center plot margin            ",cmargin
      write(6,120)16,"X axis tick mark interval     ",xdel
      write(6,120)17,"Z axis tick interval          ",zdel
      write(6,120)18,"Anomaly axis tick interval    ",gdel
      write(6,130)19,"X axis ticks between labels   ",ixip
      write(6,130)20,"Z axis ticks between labels   ",izip
      write(6,130)21,"G axis ticks between labels   ",igip
      write(6,120)22,"Character height              ",csize
      write(6,120)23,"Plot symbol size              ",ssize
      write(6,140)24,"X axis title                  ",xtitle
  140 format(2x,i4,2x,a30,/,10x,a70)
      write(6,140)25,"Z axis title                  ",ztitle
      write(6,140)26,"Anomaly area title            ",gtitle
      write(6,140)27,"Main title                    ",titlel
      write(6,140)28,"Second title                  ",title2
      write(6,120)29,"Height of main titles         ",tsize
      write(6,120)30,"Vertical exaggeration         ",vex
      write(6,120)31,"Body area scale (units/inch)  ",scale
      write(6,120)32,"Anom area scale (units/inch)  ",gscale
      write(6,140)33,"Plot file name                ",pfile
      write(6,130)34,"Unit (0=ft,1=m,2=km,3=mi,4=in)",ixunit
      write(6,130)35,"Label code (0=no,1=#,2=m,3=g) ",iabody
      write(6,145)36,"Format for X axis labels      ",fmt
  145 format(2x,i4,2x,a30,x,a8)
      write(6,145)37,"Format for Z axis labels      ",gfmt
      write(6,130)38,"# of chars in X axis las      ",ixnum
      write(6,130)39,"# of chars in Anom labels     ",ignum
      write(6,130)40,"Fast plot (0=no, 1=yes)       ",ifast
      write(6,130)41,"Reverse X axis (0=no, 1=yes)  ",ixrev
      write(6,130)42,"Reverse Z axis (0=no, 1=yes)  ",iyrev
      write(6,120)43,"Size of corner labels         ",rnsize
      write(6,120)44,"Top margin                    ",tmargin
      write(6,120)45,"Right margin                  ",rmargin
      return
      end
c
c       GET_MODEL   gets another model
c
c  ********************************************
      subroutine get_model(filename)
      character*40 filename,quest*80
      quest="Name of model file to read"
      idumb=iaquest(quest,filename,"(a40)",40)
      if (idumb.eq.-1) return
      call read_model(filename)
      return
      end
c
c     INPUT_MODEL - Get a model for hypermag to work on
c
c  ********************************************
      subroutine input_model(project)
```

```
      character project*40, ans*1, quest*80, filename*50
      write(6,110)
  110 format(" Enter the name of the model:",$)
      read(5,*)project
      ilen=itlen(project,40)
      if (ilen.gt.6) then
        ind=index(project,".mod")
        if (ind.ge.2) project=project(1:ind-1)
        end if
      call concat(filename,project,".mod      ")
  10  continue
      ans="n"
      quest=
     &"New, old (hypermag), 2mag, 2dgrav, or saki model (n/o/t/g/s)"
      idumb=iaquest(quest,ans,"(a1)",1)
      if (ans.eq."h") write(6,115)
  115 format(" Enter one of the following:",/,
     &" n = New model, you will be queried for its parameters.",/,
     &" o = Old model. A model created by a previous run of Hypermag",/,
     &"          will be read in.",/,
     &" t = Twomag (2mag) model.  An input file for program 2mag will",/,
     &"          be read in.",/,
     &" g = 2dgrav model.  An input file for program 2dgrav3 will",/,
     &"          be read in.",/,
     &" s = Saki model. Namelist input file (optional), body file, and",/,
     &"          profile file as used by program saki will be read in.")
      if (ans.eq."n") call build_model(project)
      if (ans.eq."o") call get_model(filename)
      if (ans.eq."t") call read_2mag_model
      if (ans.eq."g") call read_2dgrav_model
      if (ans.eq."s") call read_saki(project)
      if ((ans.eq."n").or.(ans.eq."o").or.(ans.eq."t").or.(ans.eq."g")
     & .or.(ans.eq."s")) return
c
c     didn't give a good answer
c
      if (ans.ne."h") write(6,117)
  117 format(" Please answer n,o,t,g, or s")
      go to 10
      end
c
c     WRITE_SAKI - writes namelist, body, and profile files for saki
c
c  ********************************************
      subroutine write_saki(project)
      common /hyper/ iplot
      common /block1/ nprof,dprof,regfirst,regslope,iobs,h(1000),
     &obs(1000),x(1000),z(1000),robs(1000)
      common /block2/ nbody,xcorn(50,50),zcorn(50,50),ncorn(50),
     &m(50),mi(50),md(50),mx(50),mz(50),rho(50)
      real m,mi,md,mx,mz
      common /block3/ azim,fi,fd,itype,xconv,zconv
      common /sakitemp/ width(2,100)
      character*50 ofile,bfile,gfile,project*40,bstr*4,wstr*8
```



```
      dimension xb(1000),zb(1000),icrn(50,50),temp(2000)
c
      call concat(ofile,project,".sak      ")
      call concat(bfile,project,".bod      ")
      call concat(gfile,project,".pfl   ")
c
      open(10,file=ofile, form="formatted", mode="out")
      write(10,100)bfile,gfile
  100 format("&parms",/,'bfile="',a56,'"',/,'gfile="',a56,'"')
      mag=0
      if (itype.eq.0) mag=1
      write(10,120)mag,fi,fd,azim
  120 format("magswt=",i1,", einc=",e15.5,", edec=",e15.5,", azmuth=",
     &e15.5)
      write(10,140)
  140 format("&")
      close(10)
      write(6,150)ofile
  150 format(" Created a namelist command file for saki called:",/,a56)
c
c     create body file
c
      open(10,file=bfile,form="formatted",mode="out")
      xb(1)=xcorn(1,1)
      zb(1)=zcorn(1,1)
      nvert=1
      do 40 i=1,nbody
      do 30 j=1,ncorn(i)
      do 20 k=1,nvert
   20 if ((xb(k).eq.xcorn(i,j)).and.(zb(k).eq.zcorn(i,j))) go to 25
      nvert=nvert+1
      k=nvert
      xb(nvert)=xcorn(i,j)
      zb(nvert)=zcorn(i,j)
   25 continue
      icrn(i,j)=k
   30 continue
   40 continue
c
c     output verticies
c
      ic=0
      do 50 i=1,nvert
      ic=ic+1
      temp(ic)=xb(i)*xconv
      ic=ic+1
      temp(ic)=zb(i)*zconv
   50 continue
      write(10,155) (temp(i),i=1,2*nvert)
  155 format(200(f8.3,f7.3,4(1x,f8.3,f7.3),/))
      write(10,157)
  157 format(">>>*bodies*<<<")
c
c     write out bodies
```

```
c
      do 80 i=1,nbody
      write(bstr,"(i4)")i
      do 73 i2=1,4
   73 if (bstr(i2:i2).ne." ") go to 74
   74 wstr="<*"//bstr(i2:4)//"*>"
  160 format(a12)
      if (itype.eq.0)
     &write(10,170)rho(i),width(1,i),width(2,i),m(i),mi(i),md(i),wstr
  170 format(6e11.3,1x,a8)
      if (itype.eq.1) write(10,171)rho(i),width(1,i),width(2,i),wstr
  171 format(3e11.3,1x,a8)
      write(10,180) (icrn(i,j),j=1,ncorn(i))
  180 format(100i4,$)
      write(10,182)wstr
  182 format(1x,a8)
   80 continue
      close(10)
      write(6,200)bfile
  200 format(" Created a file containing body info. called:",/,a56)
c
c      write out profile
c
      open(10,file=gfile,form="formatted",mode="out")
      do 90 i=1,nprof
      write(10,205)x(i)*xconv,z(i)*zconv,robs(i)
  205 format(3e15.5)
   90 continue
      close(10)
      write(6,210)gfile
  210 format(" Created a file containing profile locations and observed"
     &," values called:",/,a56)
      write(6,220)nvert,nbody,nprof
  220 format(" Output ",i3," vertices,",i3," bodies, &",i3,
     &" profile points.")
      return
      end
c
c      MATCH_VT100_PT   - matches a an X,Y pair within 5 screen points
c
c      when first called, ibod & icorn are 0,0- then subsequent calls
c      search for matches from ibod, icorn on... finally returns 0,0
c      when no more matches are found.
c
c ********************************************
      subroutine match_vt100_pt(x,y,ibod,icorn)
      common /block2/ nbody,xcorn(50,50),zcorn(50,50),ncorn(50),
     &m(50),mi(50),md(50),mx(50),mz(50),rho(50)
      common /plot2/xp(4),yp(4),dxp(2),dyp(2),xbd,ybd,ip
c
      ibst=1
      if (ibod.ne.0) ibst=ibod
      icst=icorn+1
c
```



```
c     xtol=2 screen units (here converted to body units)
c
      xtol=((dxp(2)-dxp(1))/xp(1))*(20*(xp(4)/1023))
      ytol=((dyp(2)-dyp(1))/yp(1))*(10*(yp(4)/779))
      xtol=abs(xtol)
      ytol=abs(ytol)
c      write(6,1314)xtol,ytol
 1314 format(" xtol=",f10.4," ytol=",f10.4)
c
c     go through bodies in acsending order looking for match within tol
c
      do 30 i=ibst,nbody
      do 20 j=icst,ncorn(i)
      if ((abs(xcorn(i,j)-x).le.xtol).and.((abs(zcorn(i,j)-y).le.ytol)))
     & go to 99
   20 continue
      icst=1
   30 continue
c
c     no matches
c
      ibod=0
      icorn=0
      return
c
c     found a match
c
   99 continue
      ibod=i
      icorn=j
      return
      end
c
c     GET_VT100_GIN   get a graphic input point from the vt100 term
c
c ********************************************
      subroutine get_vt100_gin(a,x,y,mode)
      integer hix,hiy
      character esc*30,gin*20,a*1
c
c     mode= 0 for crosshaair, 1 for light pen
c
      esc='"4g'
      if (mode.ne.0) esc='"5g'
      call send_esc(esc)
c
c     get response
c
      read(5,100)gin
  100 format(a20)
c
c     get character
c
      a=gin(1:1)
```

```
c       write(6,1313)a
 1313 format(" ",a1)
c
c     decode x and y
c
      hix=mod(ichar(gin(2:2)),32)
      lox=mod(ichar(gin(3:3)),32)
      hiy=mod(ichar(gin(4:4)),32)
      loy=mod(ichar(gin(5:5)),32)
c
c     assemble x and y
c
      x=hix*32+lox
      y=hiy*32+loy
c
c     write(6,120)x,y
 120  format(" In terminal coords, x=",f5.0," y=",f5.0)
      return
      end
c
c     GET_VT100_PT    gets point in body coords from VT100
c
c  ********************************************
      subroutine get_vt100_pt(a,x,y,mode)
      common /plot2/ xp(4),yp(4),dxp(2),dyp(2),xbd,ybd,ip
      character*1 a
c
    5 continue
      call get_vt100_gin(a,xg,yg,mode)
c
c     check range of answer
c
      if ((a.eq."q").or.(a.eq."r").or.(a.eq."/")) go to 10
      if ((xg.lt.1023).and.(yg.lt.779)) go to 10
      write(6,110)
  110 format(" Please enter that point again-",/,
     &" The light pen returned a wierd number.")
      go to 5
   10 continue
c     convert x and y to inches
c
      xi=xg*xbd/1023
      yi=yg*ybd/779
c
c     convert to body units
c
      xb=xi-xp(3)
      yb=yi-yp(3)
      x=xb*(dxp(2)-dxp(1))/xp(1)+dxp(1)
      y=yb*(dyp(2)-dyp(1))/yp(1)+dyp(1)
c
c     write(6,120)x,y
 120  format(" In body coords, x=",f10.3," y=",f10.3)
      return
```



```
      end
c
c     PAGE- sends page command to 4112
c
c  ********************************************
      subroutine page
      character*10 esc
      esc=char(12)
      call send_esc(esc)
      return
      end
c
c     SEND_ESC - sends an "escape" sequence.
c
      subroutine send_esc(stuff)
      character*30 stuff
      external stty(descriptors)
      call stty("-modes","rawo")
      write(6,100)stuff
 100  format(x,"¬033",a5,$)
      call stty("-modes"," rawo")
      return
      end
c
c     CHEKPP - check plot parms for consistency
c  ********************************************
      subroutine chekpp(imod)
      common /plot/ bmargin,rlmargin,cmargin,xdel,zdel,gdel,
     &ixip,izip,igip,csize,ssize,tsize,xsize,ysize,iplotr,xbody,
     &zbody,ganom,xmin,xmax,zmin,zmax,gmin,gmax,iabody,ixunit,
     &nxchar,nzchar,ngchar,ntlchar,nt2char,vex,scale,gscale
     &,ixnum,ignum,itek,ifast,ixrev,iyrev,rnsize,tmargin,rmargin
      logical rclos
      dimension imod(50)
c
c     scale must equal (xmax-xmin)/xbody
c
      if (xbody.ne.0) then
      if (.not.rclos(scale,(xmax-xmin)/xbody)) then
        if (imod(5).ne.0) then
          scale=(xmax-xmin)/xbody
          else if (imod(31).ne.0) then
          xbody=(xmax-xmin)/scale
          else
          scale=(xmax-xmin)/xbody
          end if
        end if
      end if
c
c      gscale must equal (gmax-gmin)/ganom
c
      if (ganom.ne.0) then
      if (.not.rclos(gscale,(gmax-gmin)/ganom)) then
        if (imod(6).ne.0) then
```

```
          gscale=(gmax-gmin)/ganom
          else if (imod(32).ne.0) then
          ganom=(gmax-gmin)/gscale
          else
          gscale=(gmax-gmin)/ganom
          end if
        end if
      end if
c
c      vex must equal scale/((zmax-zmin)/zbody)
c
      if (zbody.ne.0) then
      if (.not.rclos(vex,scale/((zmax-zmin)
     &/zbody))) then
        if (imod(4).ne.0) then
          vex=scale/((zmax-zmin)/zbody)
          else if (imod(30).ne.0) then
          zbody=(zmax-zmin)*vex/scale
          else
          vex=scale/((zmax-zmin)/zbody)
          end if
        end if
      end if
c
c      ysize must equal zbody+gamon+cmargin+bmargin+tmargin
c
      if (.not.rclos(ysize,(zbody+ganom+cmargin+bmargin+tmargin))) then
        if (imod(2).ne.0) then
          temp=zbody+ganom+cmargin+bmargin+tmargin
          temp=ysize/temp
          zbody=zbody*temp
          ganom=ganom*temp
          cmargin=cmargin*temp
          bmargin=bmargin*temp
          tmargin=tmargin*temp
          else
          ysize=zbody+ganom+cmargin+bmargin+tmargin
          end if
        end if
c
c      xsize must equal rlmargin+xbody+rmargin
c
      if (.not.rclos(xsize,(rlmargin+xbody+rmargin))) then
        if (imod(1).ne.0) then
          temp=xsize/(rlmargin+xbody+rmargin)
          rlmargin=rlmargin*temp
          xbody=xbody*temp
          rmargin=rmargin*temp
          else
          xsize=rlmargin+xbody+rmargin
          end if
        end if
c
c      tmargin must exceed 2*tsize
```



```
c
      if (tmargin.lt.tsize*2.) tsize=tmargin/2.
c
      end
c
c     RCLOS - decides if two single precision reals are close
c  ******************************************
      logical function rclos(a,b)
      rmax=amax1(abs(a),abs(b))
      rmax=amax1(rmax,.1)
      rclos=(abs(a-b).le.(1.0e-6*rmax))
      return
      end
c
c    IDEBLANK- deblanks and returns length
c  ******************************************
      integer function ideblank(string,ilen)
      character*256 string*(*),temp
      j=1
      temp=" "
      itemp=ilen
      if (ilen.le.0) itemp=len(string)
      do 10 i=1,itemp
      if (string(i:i).eq." ") go to 10
      temp(j:j)=string(i:i)
      j=j+1
   10 continue
      string=temp
      ideblank=j-1
      return
      end
c
c     ITLEN    - Gives length of string without trailing blanks
c  ******************************************
      integer function itlen(string,ilen)
      character string*(*)
      i=ilen
      if (ilen.le.0) i=len(string)
  10  continue
      if (string(i:i).ne." ") go to 15
      i=i-1
      if (i.le.0) go to 15
      go to 10
  15  continue
      itlen=i
      return
      end
c
c    IAQUEST   -asks question with character answer
c
c       returns:  -1 if "//" was given as re (user wants out)
c                  0 if no response (user took default)
c                  1 if user responded (returns response in aval)
c  ******************************************
```

```
      integer function iaquest(quest,aval,form,mode)
      character quest*(*),form*(*),aval*(*)
      character*100 str,form2*10,ans*80,astr*30
      iaquest=0
      iqlen=itlen(quest,0)
      irlen=itlen(aval,mode)
      if (irlen.eq.0) irlen=1
      str=quest(1:iqlen)//" ["//aval(1:irlen)//"]?"
      islen=iqlen+irlen+4
      write(form2,105)islen
105   format("(x,a",i3,",$)")
  13  write(6,form2)str
      read(5,form)ans
      ialen=31
      ialen=itlen(ans,80)
      if (ialen.ne.0) then
        if (ans.eq."//") then
          iaquest=-1
          else
          iaquest=1
          aval(1:mode)=ans(1:mode)
          end if
        end if
      return
      end
c
c    IRQUEST   - asks question with real answer
c  ********************************************
      integer function irquest(quest,rval,form,mode)
      character quest*(*),form*(*),rstr2*30
      character*100 str,form2*10,ans*30,astr*30
      irquest=0
      iqlen=itlen(quest,0)
      write(rstr2,form)rval
      irlen=ideblank(rstr2,30)
      str=quest(1:iqlen)//" ["//rstr2(1:irlen)//"]?"
      islen=iqlen+irlen+4
      write(form2,105)islen
105   format("(x,a",i3,",$)")
  13  write(6,form2)str
      read(5,110)ans
110   format(a30)
      ialen=itlen(ans,30)
      if (ialen.ne.0) then
        if (ans.eq."//") then
          irquest=-1
          else
          irquest=1
          read(ans,120,err=25)rval
 120format(f20.5)
          end if
        end if
      return
c     couldn't decode
```



```
  25  continue
      write(6,130)
 130  format(" Please answer again, I'm expecting a number.")
      go to 13
      end
c
c    IIQUEST
c  ********************************************
      integer function iiquest(quest,ival,form,mode)
      character quest*(*),form*(*),rstr2*30
      character*100 str,form2*10,ans*30,astr*30
      iiquest=0
      iqlen=itlen(quest,0)
      write(rstr2,form)ival
      irlen=ideblank(rstr2,30)
      str=quest(1:iqlen)//" ["//rstr2(1:irlen)//"]?"
      islen=iqlen+irlen+4
      write(form2,105)islen
105   format("(x,a",i3,",$)")
   13 write(6,form2)str
      read(5,110)ans
110   format(a30)
      ialen=itlen(ans,30)
      if (ialen.ne.0) then
        if (ans.eq."//") then
          iiquest=-1
          else
          iiquest=1
          read(ans,120,err=25)ival
 120 format(i20)
          end if
        end if
      return
c     couldn't decode
  25  continue
      write(6,130)
 130  format(" Please answer again, I expect a number.")
      go to 13
      end
c
c     READ_SAKI - reads in saki command, body, & profile files
c  ********************************************
      subroutine read_saki(project)
      common /hyper/ iplot
      common /block1/ nprof,dprof,regfirst,regslope,iobs,h(1000),
     &obs(1000),x(1000),z(1000),robs(1000)
      common /block2/ nbody,xcorn(50,50),zcorn(50,50),ncorn(50),
     &m(50),mi(50),md(50),mx(50),mz(50),rho(50)
      real m,mi,md,mx,mz
      common /block3/ azim,fi,fd,itype,xconv,zconv
      common /sakitemp/ width(2,100)
      dimension xb(1000), zb(1000), icrn(50,50)
      character*20 fmtx,titlx,fmtg,titlg
      character*56 ifile,quest*80,bfile,gfile,mfile,ifmtg,ifmtm
```

```
      namelist /parms/ bfile,gfile,mfile,ifmtg,ifmtm,iave,iparsh,
     & iplotr,xscale,adelx,lintx,ncharx,fmtx,sizex,titlx,
     & xbmin,xbmax,ybmin,ybmax,
     & gscale,adelg,lintg,ncharg,fmtg,sizeg,titlg,
     & scalem,adelm,lintm,ncharm,fmtm,sizem,titlm,
     & efield,einc,edec,azmuth,plenth,magswt
      azmuth=0
      gfile=" "
      einc=0
      edec=0
      magswt=0
      ifile="none"
      quest="Saki namelist command file name"
      bfile=" "
      fmtg="(v)"
      ifmtg="(v)"
      ival=iaquest(quest,ifile,"(a50)",50)
      if (ival.lt.0) return
      if ((ival.eq.0).or.(ifile.eq."none")) go to 15
c
c     read saki namelist file
c
      open(9,file=ifile,mode="in",form="formatted")
      read(9,parms)
      close(9)
c
15 continue
      if (bfile.ne." ") go to 30
      write(6,110)
 110  format(" Model filename?",$)
      read(5,*)bfile
  30  continue
      open(10,file=bfile,mode="in",form="formatted")
c
c     read in generic model file
c
      nmax=50
      nvert=1000
      read(10,*,err=40) (xb(i),zb(i),i=1,nvert)
  40  nvert=i-1
c
c     read in bodies
c
      nbody=0
  50  continue
      nbody=nbody+1
      read(10,*,err=55,end=80)rho(nbody),width(1,nbody),width(2,nbody),
     & m(nbody),m1(nbody),md(nbody)
  55  read(10,*,err=60) (icrn(nbody,j),j=1,nmax)
  60  ncorn(nbody)=j-1
c
c     construct body from look-up list
c
      do 70 i=1,ncorn(nbody)
```



```
      xcorn(nbody,i)=xb(icrn(nbody,i))
  70  zcorn(nbody,i)=zb(icrn(nbody,i))
      go to 50
c
c     got all the bodies
c
  80  continue
      nbody=nbody-1
      close(10)
c
c     read in observed profile
c
      nprof=0
      if (gfile.ne." ") go to 83
      write(6,115)
 115  format(" Profile filename?",$)
      read(5,*)gfile
  83  open(10,file=gfile,form="formatted",mode="in")
      iobs=1
  85  nprof=nprof+1
      read(10,ifmtg,end=90) x(nprof),z(nprof),obs(nprof)
      go to 85
c
c     got the whole profile
c
90  continue
      nprof=nprof-1
      close(10)
      write(6,120)nvert,nbody,nprof
 120  format(" Model consists of ",i4," verticies,",i3," bodies, and",
     & i3," profile points.")
c
c     set up misc parameters
c
      iplot=0
      regfirst=0
      regslope=0
      azim=azmuth
      fi=einc
      fd=edec
      itype=1
      if (magswt.ne.0) itype=0
      call sub_regional
      xconv=1
      zconv=1
      return
      end
c
c      MATCH_4014_PT   - matches a an X,Y pair within 5 screen points
c
c      when first called, ibod & icorn are 0,0- then subsequent calls
c      search for matches from ibod, icorn on... finally returns 0,0
c      when no more matches are found.
c
```

```
c  ******************************************
      subroutine match_4014_pt(x,y,ibod,icorn)
      common /block2/ nbody,xcorn(50,50),zcorn(50,50),ncorn(50),
     &m(50),mi(50),md(50),mx(50),mz(50),rho(50)
      common /plot2/xp(4),yp(4),dxp(2),dyp(2),xbd,ybd,ip
c
      ibst=1
      if (ibod.ne.0) ibst=ibod
      icst=icorn+1
c
c     xtol=2 screen units (here converted to body units)
c
      xtol=((dxp(2)-dxp(1))/xp(1))*(20*(xp(4)/1023))
      ytol=((dyp(2)-dyp(1))/yp(1))*(10*(yp(4)/767))
      xtol=abs(xtol)
      ytol=abs(ytol)
c      write(6,1314)xtol,ytol
 1314 format(" xtol=",f10.4," ytol=",f10.4)
c
c     go through bodies in acsending order looking for match within tol
c
      do 30 i=ibst,nbody
      do 20 j=icst,ncorn(i)
      if ((abs(xcorn(i,j)-x).le.xtol).and.((abs(zcorn(i,j)-y).le.ytol)))
     & go to 99
   20 continue
      icst=1
   30 continue
c
c     no matches
c
      ibod=0
      icorn=0
      return
c
c     found a match
c
   99 continue
      ibod=i
      icorn=j
      return
      end
c
c     GET_4014_GIN   get a graphic input point from the 4014 term
c
c  ********************************************
      subroutine get_4014_gin(a,x,y)
      integer hix,hiy
      character esc*30,gin*20,a*1
c
c
      esc=char(26)
      call send_esc(esc)
c
c     get response
c
      read(5,100)gin
  100 format(a20)
c
c     get character
c
      a=gin(1:1)
c      write(6,1313)a
 1313 format(" ",a1)
c
c     decode x and y
c
      hix=mod(ichar(gin(2:2)),32)
      lox=mod(ichar(gin(3:3)),32)
      hiy=mod(ichar(gin(4:4)),32)
      loy=mod(ichar(gin(5:5)),32)
c
c     assemble x and y
c
      x=hix*32+lox
      y=hiy*32+loy
c
c     write(6,120)x,y
  120 format(" In terminal coords, x=",f5.0," y=",f5.0)
      return
      end
c
c     GET_4014_PT    gets point in body coords from 4014
c
c  ********************************************
      subroutine get_4014_pt(a,x,y)
      common /plot2/ xp(4),yp(4),dxp(2),dyp(2),xbd,ybd,ip
      character*1 a
c
    5 continue
      call get_4014_gin(a,xg,yg)
c
c     check range of answer
c
      if ((a.eq."q").or.(a.eq."r").or.(a.eq."/")) go to 10
      if ((xg.lt.1023).and.(yg.lt.767)) go to 10
      write(6,110)
  110 format(" Please enter that point again-",/,
     &" The light pen returned a wierd number.")
      go to 5
   10 continue
c     convert x and y to inches
c
      xi=xg*xbd/1023
      yi=yg*ybd/767
c
c     convert to body units
c
```

```
      xb=xi-xp(3)
      yb=yi-yp(3)
      x=xb*(dxp(2)-dxp(1))/xp(1)+dxp(1)
      y=yb*(dyp(2)-dyp(1))/yp(1)+dyp(1)
c
c     write(6,120)x,y
 120  format(" In body coords, x=",f10.3," y=",f10.3)
      return
      end
```

A35

Appendix B

Off-line plot subsystem

An interactive plotting routine has been developed to plot bodies and calculated and observed anomaly curves for gravity and magnetic modeling programs. This routine has been grafted into hypermag.

The routine allows extensive control of the plot and provides intelligent defaults for the 45 plot parameters available. The plots can be produced at the Houston or Versatec plotters, as well as the TEKTRONIX 4025 terminal or any other terminal supported by Ron Wahl's plot package. Once plot parameters have been tailored to your specifications, they may be stored in a parameter file to simplify subsequent plots of the same model. Following is a detailed description of the 45 parameters, instructions on program use, and examples.

Plot Parameters:

1. Width of plot. This is the total width of the plotting area in inches. It is equal to the width of the left margin plus the width of the body plot plus the width of the right margin. If this is changed it can cause a proportional change in the left margin, body plot width, and right margin.

2. Height of plot. This is the total height of the plotting area in inches. It is equal to the height of the bottom margin plus height of body plot plus center margin plus height of anomaly area plus the top margin. If this parameter is changed, it can have a proportional effect on all these constituent parameters.

3. Plotter. This is an integer 0-11 indicating the plot device to use where:

0 - Calcomp 7900	6 - Vector output
1 - Tektronix 4010 (4025)	7 - not used
2 - not used	8 - Houston
3 - HP7203	9 - Versatec
4 - Tektronix 4014	10 - Tektronix 4112
5 - not used	11 - Digital VT100

4. Height of the body plot area. This parameter is the height in inches of the body area of the plot. If this is changed, it can change the vertical exaggeration. If it is 0, then no body plot area will appear.

5. Width of plot area. This is the width in inches of both the body and anomaly areas. If this is changed it can change the body area scale and vertical exaggeration.

6. Height of anomaly area. This is the height in inches of the anomaly plot. Changing this can affect the anomaly axis scale.

7. Left edge of profile. This is the left boundary (in data units) of the plots. Changing this can change the scale or width of the plot.

8. Right limit of profile. Right boundary (in data units) of the plots. Change can affect plot width or scale.

9. Top limit of body plot. Upper limit (lowest value, Z positive down) of the body plot. Change can affect the height or vertical exaggeration of the plot.

10. Bottom limit of body plot. Lower limit (highest value, Z positive down) of the body plot. Change can affect the height or vertical exaggeration of the plot.

11. Lower anomaly limit. This is the bottom boundary of the anomaly plot. Change can affect the height or scale of the plot.

12. Upper anomaly boundary. This is the top boundary of the anomaly plot. Change can affect the height or scale of the plot.

13. Bottom plot margin. The distance between the lower edge of the body plot and the bottom of the page. Change will affect the height of the plot.

14. Left plot margin. The distance between the left edge of the plot areas and the left edge of the page. Change will affect the width of the plot.

15. Center plot margin. The distance between the bottom of the anomaly area and the top of the body area. Change will affect the height of the plot.

16. X axis tick mark interval. The data unit interval between tick marks on the horizontal body axis.

17. Z axis tick interval. The data unit interval between tick marks on the vertical body axis.

18. Anomaly axis tick interval. The data unit interval between tick marks on the anomaly axis.

19. X axis ticks between labels. The number of tick marks between labeled tick marks on the horizontal body axis.

20. Z axis ticks between labels. The number of tick marks between labeled tick marks on the vertical body axis.

21. G axis ticks between labels. The number of tick marks between labeled tick marks on the anomaly axis.

22. Character height. The height in inches of all letters and numbers on the plot except the main and second titles.

23. Plot symbol size. The size in inches of the plus signs marking the profile on the body plot.

24. X axis title. Up to 80 characters which will be printed at the bottom of the page.

25. Z axis title. Up to 80 characters which will be printed at the lower left edge of the plot.

26. Anomaly axis title. Up to 80 characters which will be printed at the middle left edge of the plot.

27. Main title. Up to 80 characters plotted at the top of the page.

28. Second title. Up to 80 characters plotted under the main title.

29. Height of main titles. The character height of the main and second titles.

30. Vertical exaggeration. The difference in scale between the vertical and horizontal axis of the body plot. Change can affect the height of the plot.

31. Body area scale. The scale in units/inch of the horizontal body axis. Change can affect the width of the plot.

32. Anomaly area scale. The scale in units/inch of the vertical anomaly axis. Change can affect the height of the plot.

33. Plot file name. The name given to the plot file (for parm #3 = 8 or 9 only). Not currently implemented.

34. Unit. The units used in the body area. Not currently implemented.

35. Label code. 0 for no body labels, 1 for body numbers inside the bodies. Add 10 for body corner numbers.

36. Format for X axis labels. A fortran "f" (or "e") format for plotting of both body axis labels. For example, (f8.1).

37. Format for Y axis labels. A fortran "f" (or "e") format for plotting of the anom axis.

38. Number of characters in X axis labels. The number of characters written by the fortran format of parm 36.

39. Number of characters in Anom labels. The number of characters written by the fortran format of parm 37.

40. Fast plot (0=no, 1=yes). Indicates whether axis are to be plotted (1) or just body and anomaly (0).

41. Reverse X axis (0=no, 1=yes). The direction of the X axis. If 0 then minimum is on the left. If 1 then minimum is on the right.

42. Reverse Z axis (0=no, 1=yes). The direction of the Z axis. If 0 then minimum is top, if 1 then minimum is bottom.

43. Size of body corner numbers. The character size of the numbers used to label individual body corners (plotted if parameter #34 equals 10 or 11).

44. Top margin. Size of the top margin. Change can affect plot height.

45. Right margin. Size of the right margin. Change can affect plot width.

Instructions

These instructions describe the program flow of the model plot system only. Since this system will be run within a modeling program (hypermag, for example), it may be necessary to refer to the documentation for the modeling program too.

This program offers default answers in many places. When a default answer is available, this answer is displayed in brackets. If no brackets are printed, you must answer the question. If you wish to take the default,

simply press RETURN, otherwise enter your own answer. All data is read with a fortran "a" format, so quotes need not be entered around titles containing blanks.

When the program requests:

"Plot command (s/c/h/r/w/p/q) [h]?"

Enter:

"s" to see settings

"c" to change settings

"r" to read settings from a file

"w" to write settings to a file

"p" to plot the model

"q" to quit the plot system

Each of these options is discussed below.

SEE SETTINGS

If you entered "s", the system will print a list of the current settings (see example 1).

CHANGE SETTINGS

If you entered "c", the system will allow you to selectively change and/or examine the plot parameter settings.

The system prints the question:

"Change parameter # (-1 for all, 0 for done) [-1]?"

If you enter a positive number from 1-45 the system will ask for the new value of that parameter, offering the old value as the default. If a negative number is entered, the system will begin with that parameter and request every subsequent parameter until parameter 45 is reached or you enter "//". Entry of "//" causes the system to stop sequential requests and ask the "Change parameter #?" question again. Entry of 0 or "//" will terminate the change procedure.

READ SETTINGS

If you entered "r", the system will request the name of a file. This should be a file you created with the model plot system using the "write" option. Reading this file will reset all the plot parameters to the values in the file.

WRITE SETTINGS

If you entered "w", the system will request the name of a file to receive the plot parameter values for storage. They could then be read in by the read option on a subsequent run of the program.

PLOT MODEL

If you entered "p", the system will produce a plot controlled by the current plot parameters. If the plotter # is 8, or 9, a plot file will be created. If the plotter # is 1 (or 11) and you are at a Tektronix-compatible device, the plot will appear on the screen

QUIT

If you entered "q", the model plot system will terminate, returning control to the main modeling program.

Examples

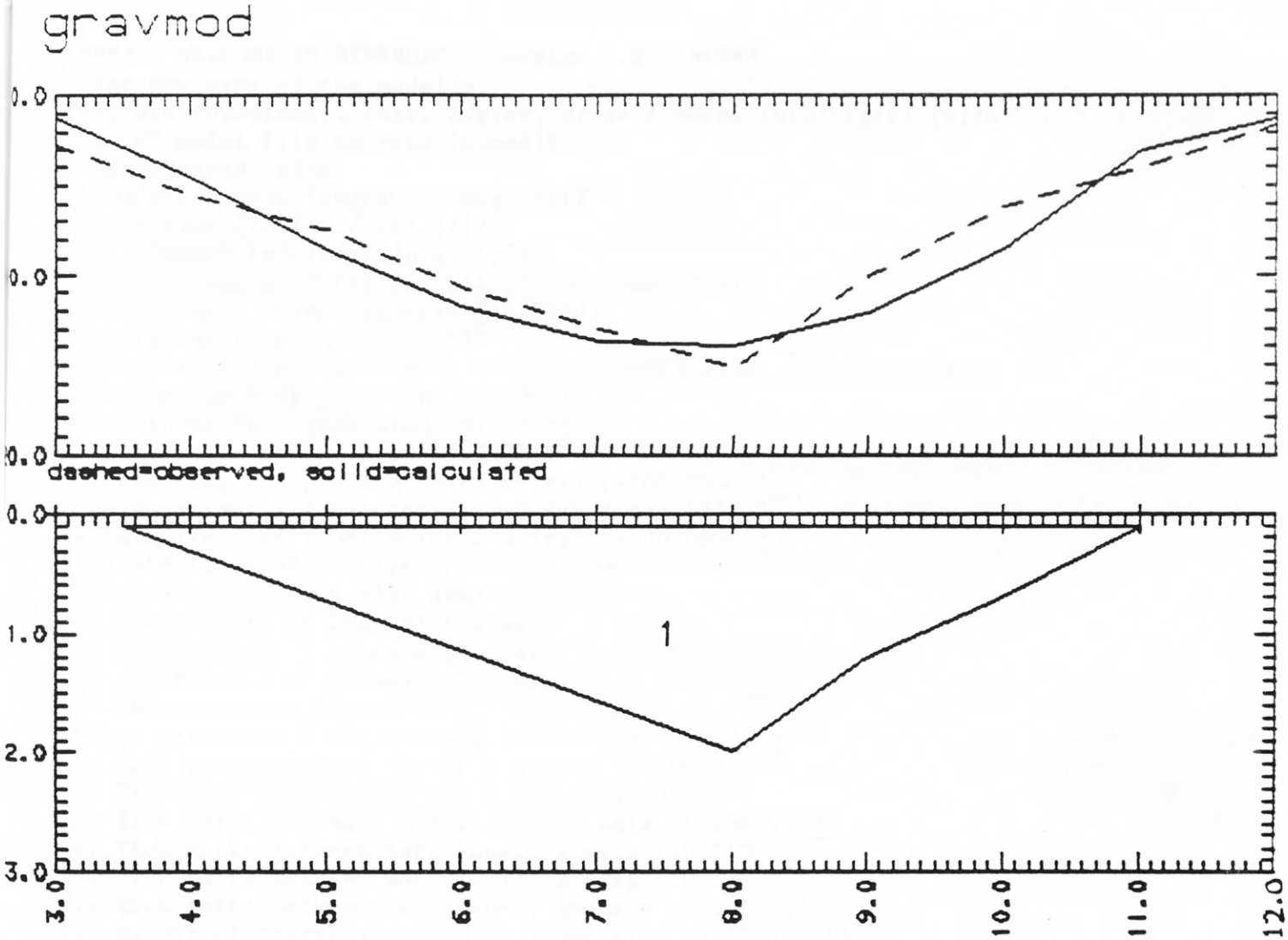
Following are two examples of plots and the dialog that produced them. Both were produced in program "hypermag".

```
*****   WELCOME TO HYPERMAG - version 1.2   *****
Enter the name of the model:gravmod
New, old (hypermag), 2mag, 2dgrav, or saki model (n/o/t/g/s) [n]?o
Name of model file to read [gravmod.mod]?
Hyper-command [h]?c
Calculation type (g=grav, m=mag) [g]?
Output type (f/h/o/p/q/t) [f]?o
Plot command (s/c/h/r/w/p/q) [p]?h
Enter one of the following
     s= see plot parameters
     c= change plot parameters
     r= read plot parameters from file
     w= write plot parameters to file
     p= plot model
     q= quit plot system
Plot command (s/c/h/r/w/p/q) [q]?c
Change parameter # (-1 for all, 0 for done) [-1]?
1. Total plot width (inches) [11.0000]?
2. Total plot height [8.5000]?
3. Plotter # (1=TKX, 8=HSTN, 9=VRSTK, 10=TKX 4112, 11=VT100) [9]?8
4. Height of body plot area [3.0000]?
5. Width of body plot area [9.5000]?
6. Height of anomaly area [3.0000]?
7. Left limit of profile (data units) [3.0000]?
8. Right limit of profile [12.0000]?
9. Top limit of body plot area [3.0000]?
10. Lower limit of body plot area [0.0000]?
11. Lower limit of anomaly plot area [-20.0000]?
12. Upper limit of anomaly plot area [0.0000]?//
Change parameter # (-1 for all, 0 for done) [0]?-8
8. Right limit of profile [12.0000]?
9. Top limit of body plot area [3.0000]?
10. Lower limit of body plot area [0.0000]?
11. Lower limit of anomaly plot area [-20.0000]?
12. Upper limit of anomaly plot area [0.0000]?
13. Bottom margin [1.0000]?
14. Left margin [1.0000]?
15. Center margin [0.5000]?
16. Tick mark interval on the X axis [0.1000]?
17. Tick mark interval on the Z axis [0.1000]?
18. Tick mark interval on the Anomaly axis [1.0000]?
19. Tick marks between each label, X axis [10]?
20. Tick marks between each label, Z axis [10]?
21. Tick marks between each label, Anomaly axis [10]?
22. Height of characters in plot annotation [0.1250]?
23. Height of symbols on profile plot [0.1250]?
24. X axis title [ ]?
25. Z axis title [Depth]?
26. Anomaly axis title [Anomaly]?
27. Main title [gravmod]?
28. Second title [ ]?
29. Character height for main and second title [0.2500]?
30. Vertical exaggeration [0.9474]?
31. Body plot scale (in units/inch) [0.9474]?
```

```
32. Anomaly scale (gammas/inch) [6.6667]?
33. Plot file name [gravmod.plt]?
34. Body units (0=none, 1=m, 2=km, 3=ft, 4=kf) [0]?
35. Body annotation (0=none, 1=#, 2=mag, 3=grav) [1]?
36. Fortran format for body axis labels [(f8.1)]?
37. Fortran format for anomaly axis labels [(f8.1)]?
38. # of characters in body axis labels [8]?
39. # of characters in anomaly axis labels [8]?
40. Fast plot (0=no, 1=yes) [0]?
41. Reverse X axis (0=no, 1=yes) [0]?
42. Reverse Z axis (0=no, 1=yes) [0]?
43. Size of body corner numbers [0.1250]?
44. Top margin [1.0000]?
45. Right margin [0.5000]?
Plot command (s/c/h/r/w/p/q) [q]?s
Current Plot Parameter Settings
Parm # Description                          Value
    1  Width of plot (inches)                11.0000
    2  Height of plot                         8.5000
    3  Plotter (1=TKRX, 8=HN, 9=VSTK)         8
    4  Height of body plot area               3.0000
    5  Width of plot areas                    9.5000
    6  Height of anomaly plot area            3.0000
    7  Left edge of prof (data units)         3.0000
    8  Right limit of profile                12.0000
    9  Top limit of body plot                 3.0000
   10  Bottom limit of body plot              0.0000
   11  Lower anomaly limit (gammas)         -20.0000
   12  Upper anomaly limit                    0.0000
   13  Bottom plot margin (inches)            1.0000
   14  Left plot margin                       1.0000
   15  Center plot margin                     0.5000
   16  X axis tick mark interval              0.1000
   17  Z axis tick interval                   0.1000
   18  Anomaly axis tick interval             1.0000
   19  X axis ticks between labels           10
   20  Z axis ticks between labels           10
   21  G axis ticks between labels           10
   22  Character height                       0.1250
   23  Plot symbol size                       0.1250
   24  X axis title
   25  Z axis title
         Depth
   26  Anomaly area title
         Anomaly
   27  Main title
         gravmod
   28  Second title
   29  Height of main titles                  0.2500
   30  Vertical exaggeration                  0.9474
   31  Body area scale (units/inch)           0.9474
   32  Anom area scale (units/inch)           6.6667
   33  Plot file name
                                              gravmod.plt
```

```
 34  Unit (0=ft,1=m,2=km,3=mi,4=in)          0
 35  Label code (0=no,1=#,2=m,3=g)           1
 36  Format for X axis labels       (f8.1)
 37  Format for Z axis labels       (f8.1)
 38  # of chars in X axis las                8
 39  # of chars in Anom labels               8
 40  Fast plot (0=no, 1=yes)                 0
 41  Reverse X axis (0=no, 1=yes)            0
 42  Reverse Z axis (0=no, 1=yes)            0
 43  Size of corner labels                   0.1250
 44  Top margin                              1.0000
 45  Right margin                            0.5000
Plot command (s/c/h/r/w/p/q) [q]?p     *see page B10a for this plot
Plot command (s/c/h/r/w/p/q) [q]?
Output type (f/h/o/p/q/t) [q]?
Hyper-command [q]?
Save model (y/n) [y]?
Name of model file to write to [gravmod.mod]?
Thanks for using hypermag.............
```

Example of off-line plot

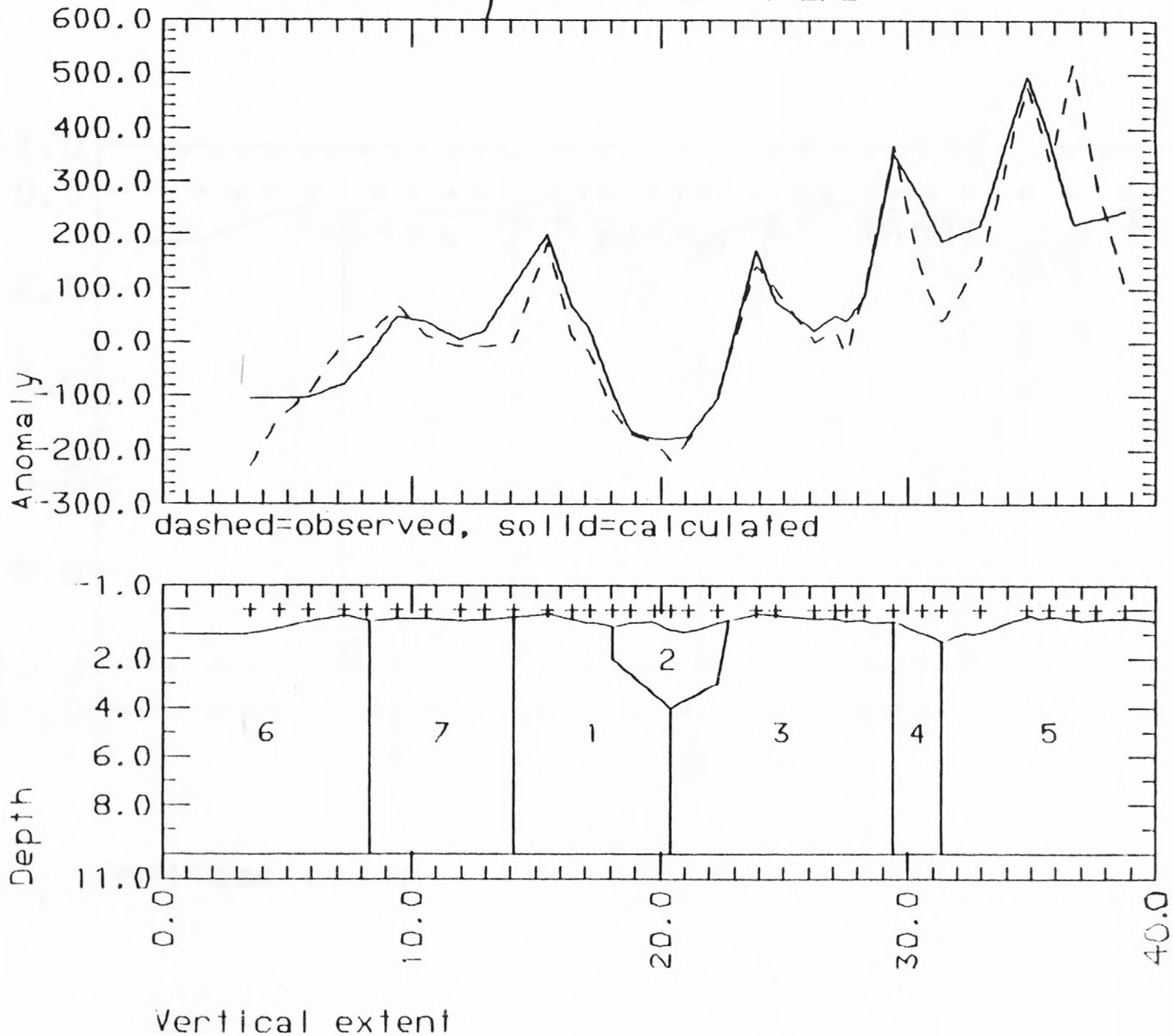


B.10a

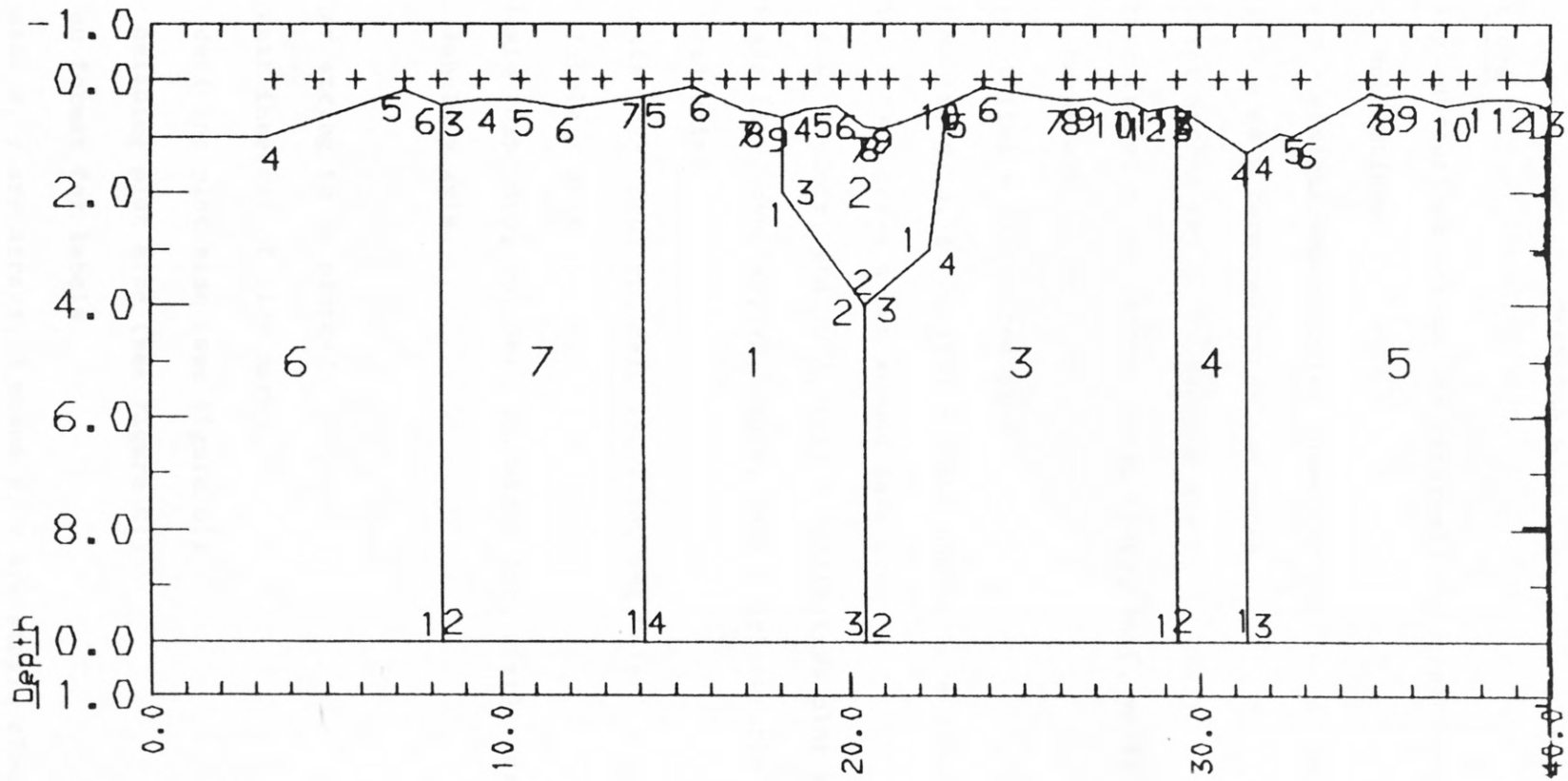
```
*****   WELCOME TO HYPERMAG - version 1.2   *****
Enter the name of the model:m
New, old (hypermag), 2mag, 2dgrav, or saki model (n/o/t/g/s) [n]?o
Name of model file to read [m.mod]?
Hyper-command [h]?c
Calculation type (g=grav, m=mag) [m]?
Output type (f/h/o/p/q/t) [f]?o
Plot command (s/c/h/r/w/p/q) [p]?c
Change parameter # (-1 for all, 0 for done) [-1]?
1. Total plot width (inches) [11.0000]?
2. Total plot height [8.5000]?
3. Plotter # (1=TKX, 8=HSTN, 9=VRSTK, 10=TKX 4112, 11=VT100) [9]?8
4. Height of body plot area [3.0000]?
5. Width of body plot area [9.5000]?
6. Height of anomaly area [3.0000]?
7. Left limit of profile (data units) [-100.0000]?//
Change parameter # (-1 for all, 0 for done) [0]?-7
7. Left limit of profile (data units) [-100.0000]?0
8. Right limit of profile [200.0000]?40
9. Top limit of body plot area [20.0000]?11
10. Lower limit of body plot area [0.0000]?-1
11. Lower limit of anomaly plot area [-300.0000]?-300
12. Upper limit of anomaly plot area [600.0000]?600
13. Bottom margin [1.0000]?//
Change parameter # (-1 for all, 0 for done) [0]?-16
16. Tick mark interval on the X axis [10.0000]?1
17. Tick mark interval on the Z axis [1.0000]?1
18. Tick mark interval on the Anomaly axis [10.0000]?20
19. Tick marks between each label, X axis [10]?10
20. Tick marks between each label, Z axis [10]?2
21. Tick marks between each label, Anomaly axis [10]?5
22. Height of characters in plot annotation [0.1250]?.125
23. Height of symbols on profile plot [0.1250]?.125
24. X axis title [ ]?Vertical extent
25. Z axis title [Depth]?
26. Anomaly axis title [Anomaly]?
27. Main title [m]?Will's Trout Model
28. Second title [ ]?Stolen by R Saltus
29. Character height for main and second title [0.2500]?.25
30. Vertical exaggeration [4.7368]?//
Change parameter # (-1 for all, 0 for done) [0]?0
Plot command (s/c/h/r/w/p/q) [q]?p     *see page B12a for this plot
Plot command (s/c/h/r/w/p/q) [q]?w
Filename for parameter storage [m.ppm]?
Plot command (s/c/h/r/w/p/q) [q]?c
Change parameter # (-1 for all, 0 for done) [-1]?6
6. Height of anomaly area [3.0000]?0
Change parameter # (-1 for all, 0 for done) [0]?35
35. Body annotation (0=none, 1=#, 2=mag, 3=grav) [1]?11
Change parameter # (-1 for all, 0 for done) [0]?43
43. Size of body corner numbers [0.1250]?.1
Change parameter # (-1 for all, 0 for done) [0]?0
Plot command (s/c/h/r/w/p/q) [q]?p     *see page B12b for this plot
Plot command (s/c/h/r/w/p/q) [q]?q
```

```
Output type (f/h/o/p/q/t) [q]?
Hyper-command [q]?
Save model (y/n) [y]?
Name of model file to write to [m.mod]?
Thanks for using hypermag.............
```

Will's trout model
Stolen by R Saltus



B12a



Vertical extent

B12.6

Appendix C

External Subroutines

The following subroutines are called externally by hypermag:

Multics subroutines

asr - assigns search rules (used for Multics dynamic linking)

stty - sets terminal types and modes

Basic Plot System (by G. I. Evenden and R. R. Wahl)

char (x, y, a, nc, icode, size, theta, xoff, yoff) - writes characters on the plot

endpt (ie) - terminates plot

line (x, y, n, icon, ipn) - draws one or more lines

neatl () - draws a box around data area

pltset (iplotr, xbd, ybd, isl) - initializes plot system

scale (dxp, dyp, xp, yp, nopts, ier) - defines the plot area and units

xaxis (dxp, dyp, xp, del, ip, size, fmt, nfmt) - draws and labels x axis

yaxis (dxp, dyp, xp, del, ip, size, fmt, nfmt) - draws and labels y axis

where:

a = character string to be plotted

del = data unit interval of tick marks

dxp = array defining plot size (see Figure C1)

dyp = array defining plot size (see Figure C1)

fmt = Fortran format for labels

icode = 1 means x, y are arrays, 3 means x, y are single elements

icon = 0 start new line, 1 continue old line

ie = 10 element array containing diagnostics

ier = set to 0 to indicate no problems

ip = interval between tick mark lables

ipn = line type, 0=solid, 1-7 dashed lines

iplotr = plotter number

isl = 5 element array, isl(1)=1 inches, 2 cm

n = dimension of x, y arrays in line call

nc = if icode is 0 or 1, dimension of x and y arrays, otherwise length of a

nfmt = length of format

nopts = length of xp and yp arrays

size = character size

theta = rotation of char in radians

x = x location (array of dimension n)

y = y location (array of dimension n)

xbd = returned maximum x size for selected plotter

ybd = returned maximum y size for selected plotter

xoff = x offset for center of label

yoff = y offset for center of label

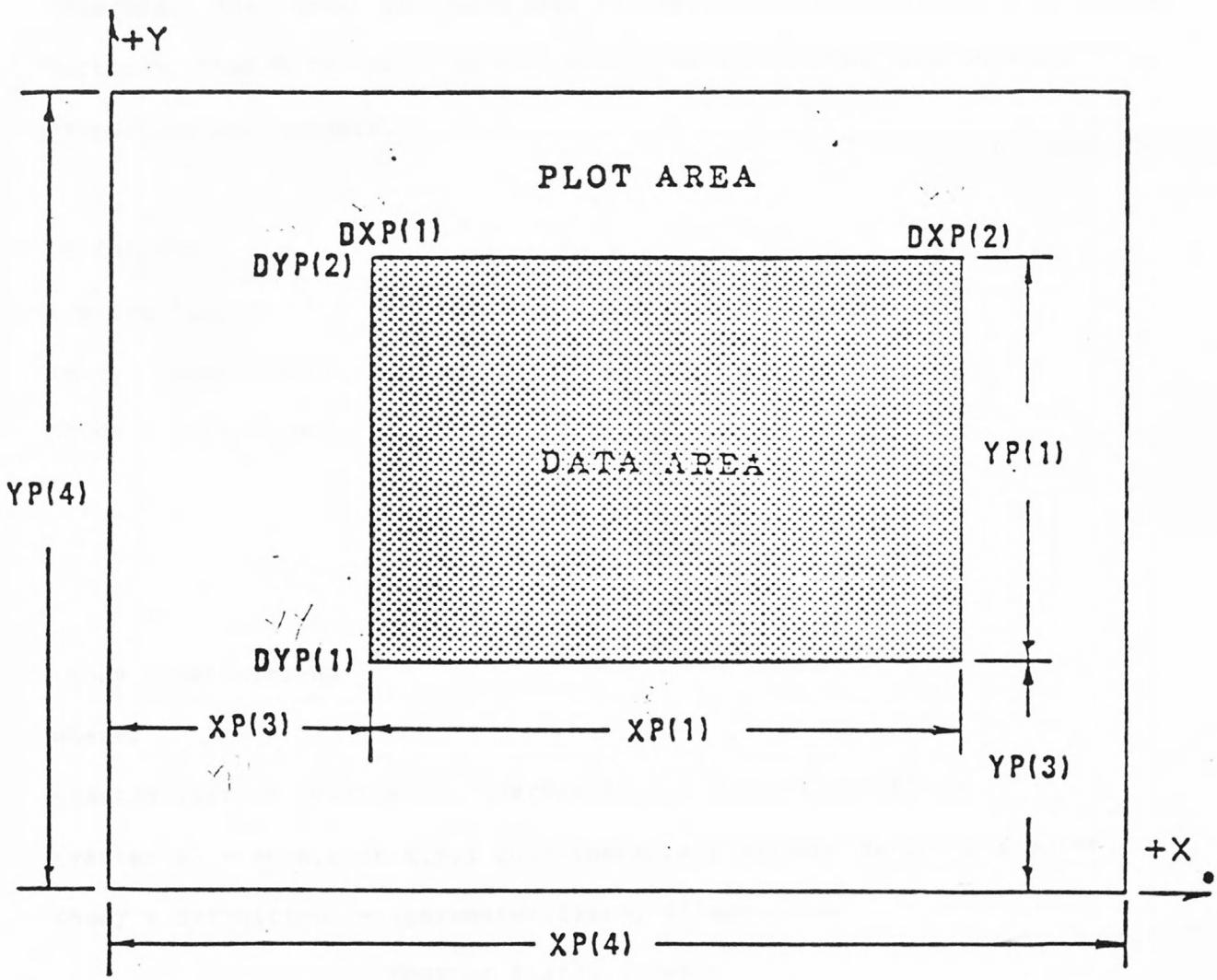
xp = array defining plot size (see Figure C1)

yp = array defining plot size (see Figure C1)

Figure C1. DXP,DYP,XP,YP arrays and their relation to the plot area.

DXP and DYP are in data units.

XP and YP are in plot units (inches).



Appendix D

10/29/82 Bodyfile, Generic Bodyfile

An ascii file format has been defined to facilitate the input of 2 and 3 dimension polygonal bodies to various modelling programs including saki and hypermag. The format is free-field (variable format) consisting of a list of vertices, then a two part entry for each polygon giving its physical properties and corners.

Definition:

<vertex list>

<body 1 definition>

<body 2 definition>

.

.

.

<body n definition>

where:

<vertex list> = <vertex 1>, <vertex 2>,... <vertex n> <flag>

<vertex x> = An x,y or x,y,z coordinate pair/triplet defining a point.

<body x definition> = <parameter list>, <flag>

<corner list>, <flag>

<parameter list> = A list of one to ten numbers in variable format.

<corner list> = A list of integers in variable format, each referring to a vertex in the vertex list.

<flag> = A non-numeric character string. (Indicates the end of a number list.)

Notes:

The parameter line of the body definition may contain from one to ten values.The meaning of these numbers may vary from program to program, but the following are standard definitions for the first six values:

parm1 = density (gravity) or suseptibility (magnetics)

parm2 = top of the polygon (3d models) or near side (2 1/2d models)

parm3 = bottom of the polygon, or far side

parm4 = intensity of magnetization

parm5 = magnetic inclination

parm6 = magnetic declination

The corners should be listed in clockwise order.

Programs saki and hypermag assume the vertex coordinates are in kilometers.

Example:

```
  12.000 -1.460   14.000 -1.560   18.500 -1.475   34.500 -1.900   55.000 -0.825
  50.000 -0.500   47.000  0.100   45.000  0.200   40.000  4.300   37.006  1.745
  36.000  1.750   32.000  4.000   31.000  4.000   28.000  3.500   27.000  3.200
  21.500  1.400   20.000  1.500   16.897  0.629   14.700  0.800   14.000  0.700
   0.000 -0.600   11.273 -0.189    9.211 -0.189    7.358 -0.589   10.000 -1.250
  11.000 -1.365   11.273 -0.335   10.242 -0.436
>>>*bodies*<<<
  0.300E+00  0.000E+00  0.000E+00 <*1*>
   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20 <*1*>
  0.100E+00  0.000E+00  0.000E+00 <*2*>
  21   1  20  22  23  24 <*2*>
  0.400E+00  0.000E+00  0.000E+00 <*3*>
  25  26  27  28 <*3*>
```

Explanation of example:

The above example is of a 3 body model consisting of a total of 28 vertices.The 28 vertices are listed first as x,y pairs. The character string ">>>*bodies*<<<" flags the end of the vertex list. Body one has a density of 0.3 with parameters 2 and 3 both set to zero. The end of the parameter list is flagged with "<*1*>". Body one has 20 corners and they fall on the first 20 vertices from the vertex list. The corner list is flagged at the end with "<*1*>". Body 2 has density of 0.1 and 6 corners falling on vertices 21, 1, 20, 22, 23, and 24. Body 3 has 4 corners falling on vertices 25, 26, 27, and 28. Body 3 has a density of 0.4.