GEOPLOT: User guide and programmer's guide

S D Buxton & T P LeBas

Report No 279 1990
GEOPLOT: User guide and programmer's guide

S D Buxton & T P LeBas

1990
TITLE

GEOPLOT: User guide and programmer's guide.

REFERENCE

Institute of Oceanographic Sciences Deacon Laboratory, Report, No. 279, 89pp.

ABSTRACT

The program GEOPLOT can plot ship's tracks and their associated geophysical data in several ways. This report is intended as a complete guide for both users and programmers. It gives full instructions and example plots for the novice user. For the programmer it includes a structural description of the program and all its subroutines, together with details of the library subroutines used.

KEYWORDS

COMPUTER GRAPHICS
COMPUTER PROGRAMMES
DATA PROCESSING
GEOPHYSICAL DATA
INSTITUTE OF OCEANOGRAPHIC SCIENCES - cruises

ISSUING ORGANISATION

Institute of Oceanographic Sciences
Deacon Laboratory
Wormley, Godalming
Surrey GU8 SUB. UK.
Director: Colin Summerhayes DSc

Telephone Wormley (0428) 684141
Telex 858833 OCEANS C
Facsimile (0428) 683066

Copies of this report are available from: The Library, PRICE £23.00
# CONTENTS

**INTRODUCTION**  
The data  
Program features

**USER GUIDE**  
Running the program  
Hardware and software requirements  
Starting GEOPLOT  
Using the program  
Plotting the ship's track only  
Plotting data annotation along track  
Plotting data profiles along the track  
Plotting the ship's track only when the data is valid  
Plotting the ship's track with time  
Plotting the ship's track with cruise names  
Plotting data profiles and time along track  
Choosing devices  
Choosing a projection  
Choosing an area  
Choosing a scale  
Completing the input  
Changing default parameters  
The plots

**PROGRAMMER'S GUIDE**  
Language details  
Files  
Program structure  
Routines  
Program TRACKS  
Subroutine MAGMES  
Subroutine GETINT  
Subroutine GETLOG
1 INTRODUCTION

1.1 The Data

Cruises by the Institute of Oceanographic Sciences Deacon Laboratory and other organisations collect geophysical data such as bathymetry, magnetics and gravity. During the cruises the data is recorded on tape in Merge-Merge format, together with navigational fixes and time. The completed tapes are loaded on the Natural Environment Research Council's IBM 4381 computer at Wormley, Surrey, and the data is placed in the ORACLE database system. It may be retrieved from the database by specifying either the name of the cruise or a geographical area [Drake & Miles, 1985; LeBas & Evans, 1988].

Data Description:

Cruise name: consists of an abbreviation of the ship's name and a reference number, eg DIS161 is Discovery cruise number 161. This generally has a maximum of 8 alphanumeric characters.

Date: day, month and year, e.g. 010289 represents 1st February 1989.

Time: hour and minute, GMT, e.g. 1620 represents 4.20 pm.

Latitude and longitude: in decimal degrees (to 4 or 5 decimal places). Negative values correspond to the Southern Hemisphere and west of the Greenwich Meridian, respectively.

Water depth: in metres, measured using an echo sounder, uncorrected for variation in transmission speed through the water column. Assumed velocity = 1500 m/s.

Water depth: in metres, corrected using Carter's tables [Carter, 1980].
Total magnetic field: in nanoteslas, acquired using a proton precession magnetometer towed behind the ship.

Residual magnetic anomaly: in nanoteslas, calculated by taking the background magnetic field for that area and time (given by the International Geomagnetic Reference Field [IAGA 1986]) from the total magnetic field.

Free air anomaly: in milligals, calculated from the 1967 geodetic reference system [UGGI, 1971] and readings from the onboard gravimeter.

1.2 Program Features

GEOPILOT is a program for plotting cruise data on a chart. A simple map of the ship's track may be produced, or the data (as described in the previous section) may be plotted on the track. Options include:

Plotting bathymetry, magnetic or gravity data as a series of number annotations.
Plotting profiles of bathymetry, magnetic and gravity data along the ship's track.
Plotting the sections of the track where certain chosen data is available.
Plotting dates and times on the track.
Plotting cruise names on the track (especially useful when the data has been obtained by an area search and several cruise tracks cross each other).

The program will accept data in Merge-Merge format, or from the ORACLE database, or in the standard MGD-77 format [USDC, 1981]. It is also possible to enter a user-defined format.

The form of the chart can be controlled by specifying different map projections, adding a grid of latitude and longitude, and by changing the colours used within the plot. Plots may be displayed on graphics terminals (including PCs emulating graphics terminals) or sent to pen plotters, both colour and black & white.
2 USER GUIDE

The purpose of this section is to explain how to use GEOPlot to plot cruise data. Details of the programming language, program structure, etc. can be found in the programmer's guide which follows.

It gives a step-by-step description of how to run the program, how to operate it, and how to produce plots on paper.

2.1 Running the Program

2.1.1 Hardware and software requirements

GEOPlot runs on an IBM 4381 mainframe with the CMS operating system together with a GRAFIX/GKS graphics system. Output devices supported are IBM 3192 and Tektronix 4205/4211 graphics terminals, Calcomp 1075 colour plotter, Versatec 80 black and white plotter and IBM PS/2 in Tektronix 4205 emulation mode. GEOPlot itself requires two files: GEOPlot EXEC and TRACKS MODULE.

2.1.2 Starting GEOPlot

There are two ways of starting the program. One is simply to type 'GEOPlot'. If this is done the screen will prompt you for the name of the file containing the data to be plotted. The name can then be typed at the keyboard. If the file mode is omitted it is assumed to be 'A'; if the file type is omitted it is assumed to be 'IN'.

E.g. CRUISE DATA G will cause the computer to search for the file CRUISE DATA on your G disk.

CRUISE DATA will cause the computer to search your A disk.

CRUISE will cause the computer to search for the file CRUISE IN on your A disk.

Alternatively the program can be started by entering 'GEOPlot' followed on the same line by the file name, e.g. 'GEOPlot CRUISE DATA', which will use CRUISE DATA on your A disk as its input data file.
There is a short delay while the graphics are initialised and the program is loaded. When this is completed the screen clears and the program title and a menu of options are displayed.

2.2 Using the Program

When the program has finished loading an initial menu of options is displayed:

1. Plot ship's track only
2. Plot data annotation along track
3. Plot data profile along track
4. Plot ship's track only when data valid
5. Plot ship's track with time
6. Plot ship's track with cruise names
7. Plot data profiles and time along track

Type the number of the option required and press ENTER.

2.2.1 Plotting the ship's track only

The ship's track is plotted on a map annotated with latitude and longitude. Any coastlines within the area covered by the map are also plotted.

2.2.2 Plotting data annotation along track

The program offers a choice of three different variables for annotation:

1. Depth
2. Residual Magnetic Anomaly
3. Free Air Anomaly

If you choose depth you are asked:

Corrected depth? (Y/N)
Answering 'N' to this question will give the original uncorrected depth data. Answering 'Y' will give the depth corrected for variation in water transmission speed through the water column [Carter, 1980]. If any corrected depths are missing from the data file it is possible to plot uncorrected depths to fill the gap. This is done by replying 'Y' to the question which follows:

Use uncorrected depth if corrected is unavailable? (Y/N)

If you choose residual magnetic anomaly, you are given the dates of the latest DGRF (Definitive Geomagnetic Reference Field) and the IGRF (International Geomagnetic Reference Field). A new IGRF is defined every five years. The old one is corrected and becomes a DGRF [IAGA, 1986]. The residual magnetic anomaly contained in the data file will have been calculated from the total field using the IGRF current at the time of data acquisition. Thus it is often possible to get a more accurate residual anomaly by recalculating it using an improved reference field. This is done by replying 'Y' to the question:

Recalculate Residual Magnetic Anomaly? (Y/N)

The variable selected is plotted as a series of numbers along the course of the ship's track. If uncorrected depths are being used to fill gaps in the corrected depths, the uncorrected depths are plotted in a different colour, so that they can be identified. To avoid overcrowding the plot, numbers are only plotted when they do not overlap with the previous value plotted.
2.2.3 Plotting data profiles along the track

A data profile is a line plotted on top of the ship's track which shows how a value varies along its course. The value is represented by the displacement of the profile from the line representing the ship's track.

Profiles may be plotted of the same three variables described above:

1. Depth
2. Residual Magnetic Anomaly
3. Free Air Anomaly

Since it is possible to plot more than one profile on the same track you can choose more than one variable. The variables are selected by typing the appropriate number and pressing ENTER. An asterisk appears next to the number in the menu to show it has been selected. If you choose a variable by mistake you can cancel it by entering its number again. When you are ready to proceed press ENTER without typing a number.

Next you will be asked for the bearing for the data profiles. This is an angle between 0° and 360° measured clockwise from north. The bearing is the direction in which the profile is to be offset from the ship's track. It should usually be approximately perpendicular to the direction of the track, if possible.

The program then asks for two values for each variable to be profiled. These are the 'Scaling factor for data', a value to convert from the variable's units to sheet millimetres, and an 'Origin to be subtracted' from the scaled value, also in millimetres. Default values for the scale and origin are supplied, which should suit most plots. These are available by pressing ENTER when the program requests these values. The default values are:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Scale</th>
<th>Offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth</td>
<td>5 mm / 1000 m</td>
<td>0 mm</td>
</tr>
<tr>
<td>Residual Magnetic Anomaly</td>
<td>10 mm / 100 nanoteslas</td>
<td>0 mm</td>
</tr>
<tr>
<td>Free Air Anomaly</td>
<td>20 mm / 100 milligals</td>
<td>0 mm</td>
</tr>
</tbody>
</table>
As with annotation plots and depending on the variables selected, you may be asked the following questions:

Corrected depth? (Y/N)
Use corrected depth if corrected is unavailable? (Y/N)
Recalculate residual magnetic anomaly? (Y/N)

These questions are explained in section 2.2.2.

2.2.4 Plotting the ship's track only when the data is valid

This is essentially the same as the first option ("Plot ship's track only") except that if a specified variable is absent from the data file the track will not be plotted.

The variable is selected from the three described above:

Depth
Residual Magnetic Anomaly
Free Air Anomaly

If you select depth you are given the option of corrected or uncorrected depth, as in section 2.2.2.

2.2.5 Plotting the ships track with time

With this option the track is annotated with the day and/or time that the ship was sailing along each part of the track. You will be asked for a tick interval in minutes. It must either be an exact division of an hour (e.g. 30 minutes) or an exact multiple (e.g. 60 minutes, 120 minutes, 180 minutes etc.). This controls how often the time is marked on the track.

The information plotted depends on the tick interval. If it is 1440 minutes (one day) or greater, the Julian day number is plotted. The Julian day number is a number between 1 (1st January) and 365 (31st December) in a non-leap year, or 1 and 366 in a leap year.
If the tick interval is less than 1440 minutes, the time in hours and minutes is plotted, e.g. 2145. This is except for every 6 hours, when the Julian day number and hour are plotted, e.g. noon on the 16th October 1990 is equivalent to 1200, day 289, and would be plotted as 289/12.

2.2.6 Plotting the ship's track with cruise names

With this option the beginning of each section of the ship's track (i.e. every uninterrupted line) is annotated with the cruise name and Julian day. This information is also plotted at the start of each new day. This is useful if the data file contains the results from more than one cruise.

2.2.7 Plotting data profiles and time along track

This is a combination of options to give data profiles annotated with the time (as described in sections 2.2.3 and 2.2.5).
2.2.8 Choosing devices

After setting up the type of plot required, the next stage is to choose an output device. The program allows four different devices:

1 Graphics Terminal (IBM 3179 or Tektronix 4211 directly connected)
2 Calcomp (Calcomp 1075 colour plotter)
3 Versatec (Versatec 80 B+W plotter)
4 IBM PS/2 (Emulating a Tektronix 4205)

It is possible to select more than one device. Type the number of each device required and press ENTER. Asterisks will appear next to the numbers to show which devices have been selected. To cancel a selection re-enter the appropriate number. When you are ready to proceed press ENTER without typing a number.

2.2.9 Choosing a projection

Once the devices have been selected, the next stage is to choose a map projection. This is selected from a list of five:

1 Ellipsoidal Mercator
2 Universal Transverse Mercator
3 Gnomonic Transverse
4 Lambert Cylindrical
5 Orthographic Transverse

Type the number of the projection and press ENTER.

The irregular projections (2, 3 and 5) require you to enter a central meridian. This is the line of longitude which will be plotted as a straight vertical line and which should pass through the area covered by the map. All longitudes are entered as a decimal number of degrees between -180 and 180. Positive numbers are used for degrees east and negative for degrees west.
For all projections it is necessary to enter a standard latitude. This is the latitude at which the scale is correct (see choosing an area, section 2.2.10). All latitudes are entered as a decimal number of degrees between -90 and 90. Positive numbers are used for degrees north and negative for degrees south.

2.2.10 Choosing an area

When you have set up a projection the program will require you to choose the area to be covered by the map. This is done by asking for a series of longitude and latitude limits, in the order:

Longitude of left edge
Longitude of right edge
Latitude of lower edge
Latitude of upper edge

Latitudes and longitudes are entered in degrees, with north or east values being positive and south or west values being negative. Note that if the value entered for the right edge is less than that entered for the left edge, the program will assume that the map is to cross the 180° meridian.

2.2.11 Choosing a scale

Once the area has been entered the program calculates a range of scales which will produce a map of a sensible size. The calculations are based on the Ellipsoidal Mercator Projection but provide a rough guide for all projections. An example display is given below.

<table>
<thead>
<tr>
<th>Suggested scales:</th>
<th>Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approximate size</td>
<td></td>
</tr>
<tr>
<td>100 mm N-S, 100 mm E-W</td>
<td>1140000</td>
</tr>
<tr>
<td>200 mm N-S, 200 mm E-W</td>
<td>570000</td>
</tr>
<tr>
<td>400 mm N-S, 390 mm E-W</td>
<td>290000</td>
</tr>
<tr>
<td>800 mm N-S, 780 mm E-W</td>
<td>140000</td>
</tr>
</tbody>
</table>

After displaying the suggested scales the program asks:
Scale (e.g. 0.0000001 or 10000000)

If the entered scale is less than 1 it is treated as a fraction (e.g. 0.0000001 = 1 : 10,000,000) otherwise it is treated as the \('x\) in the ratio '1:x' (e.g. 10000000 = 1 : 10,000,000). You may also enter the scale in scientific notation (e.g. 10E6 or 1E-7).

2.2.12 Completing the input

The final question asked is:

Title (up to 45 characters)

The purpose of the title is to identify the map. It is plotted in the box containing a description of the map (see section 2.3). If the title has more than 45 characters it will be truncated.

When you have entered the title a summary appears on the screen giving title, grid limits, scale, projection and output devices. At this stage you are asked if you want to re-enter the parameters. Replying 'Y' will take you back to the start of the program. For each question the program asks you can either enter a new response, or press ENTER to keep the old response. The old one is displayed on the screen so that you can check it. When you reach the question 'Re-enter these parameters?' again press 'N' to continue or 'Y' to make further changes.
2.2.13 Changing default parameters

At this point the program is ready to begin the plot. However you may want to change some of the default values for the plot and data file. These values are displayed at the top of the screen:

<table>
<thead>
<tr>
<th>Format</th>
<th>Oracles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid</td>
<td>On</td>
</tr>
<tr>
<td>Coastline</td>
<td>Black</td>
</tr>
<tr>
<td>Track</td>
<td>Black</td>
</tr>
<tr>
<td>Annotation</td>
<td>Black</td>
</tr>
<tr>
<td>Bathymetry</td>
<td>Green</td>
</tr>
<tr>
<td>Magnetics</td>
<td>Red</td>
</tr>
<tr>
<td>Gravity</td>
<td>Blue</td>
</tr>
</tbody>
</table>

Below that is a menu:

Please select an option:
1. Choose different format
2. Turn grid off
3. Change default colours
4. Start plot

Entering 1 allows you to change the format that the program will use to read the data file. The preprogrammed formats are:

1. ORACLES (ORACLE relational database format)
   Fixed length lines, 80 characters per line, two-line header

<table>
<thead>
<tr>
<th>Cruise name</th>
<th>A8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day</td>
<td>1X, 12</td>
</tr>
<tr>
<td>Month</td>
<td>1X, 12</td>
</tr>
<tr>
<td>Year</td>
<td>1X, 12</td>
</tr>
<tr>
<td>Hour</td>
<td>1X, 12</td>
</tr>
<tr>
<td>Minute</td>
<td>12</td>
</tr>
</tbody>
</table>
2. Merge-Merge (DXFMT)

Fixed length lines, 80 characters per line, no header

Cruise name       A8
Year              5X, I2
Month             I2
Day               I2
Hour              1X, I2
Minute            I2
Latitude          1X, F8.4
Longitude         F9.4
Uncorrected Depth 10X, F5.0
Corrected Depth   F5.0
Total Mag. Anomaly 3X, F5.0
Resid. Mag. Anomaly F5.0
Free Air Anomaly   F5.0

3. MGD-77 (Marine Geophysical Data Exchange Format) [USDC, 1981]

Fixed length lines, 120 characters per line, 24-line header

Cruise name       1X, A8
Year              5X, I2
Month             I2
Day               I2
Hour              I2
Minute            I2
<table>
<thead>
<tr>
<th></th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latitude</td>
<td>(3X, F8.0) (and divide by 100000)</td>
</tr>
<tr>
<td>Longitude</td>
<td>(F9.0) (and divide by 100000)</td>
</tr>
<tr>
<td>2-way Travel Time</td>
<td>(1X, F6.0) (and divide by 10000)</td>
</tr>
<tr>
<td>Corrected Depth</td>
<td>(F6.0) (and divide by 10)</td>
</tr>
<tr>
<td>Total Mag. Anomaly</td>
<td>(3X, F6.0) (and divide by 10)</td>
</tr>
<tr>
<td>Resid. Mag. Anomaly</td>
<td>(6X, F6.0) (and divide by 10)</td>
</tr>
<tr>
<td>Free Air Anomaly</td>
<td>(25X, F5.0) (and divide by 10)</td>
</tr>
</tbody>
</table>

Uncorrected depth = 2-way travel time * speed of sound / 2
(The program assumes the speed of sound is 1500 m/s)

Alternatively, you can type 4 for a user-defined format. The program will assume that it has the same fields as in the ORACLES format in the same order. It is entered using the same syntax as a FORTRAN format statement, including the brackets, e.g. \(A8, 1X, 3I2, 1X, 2I2, 2X, F8.4, 1X, 2I2, 2X, F8.4, 1X, F9.4, 2X, 5\) (\(F5.0, 1X\)). Note that the cruise name is a character variable, the day, month, year, hour and minute are integers, and the other variables are real. The program will also ask for the number of lines in the header.

The default is that grid lines are drawn on the map. The lines plotted are chosen by the software at an appropriate spacing which will not obscure the data. To prevent the grid from being drawn enter 2 from the 'Please select an option' menu.

Entering 3 allows you to change the colours used for the plot. Normally the plot will be in black and white, except for the data profiles which will be in colour. For the coastline, track, annotation, bathymetry, magnetics and gravity you have a choice of 4 colours:

1. Black
2. Red
3. Green
4. Blue

If the graphics output device you are using cannot produce colour, the plot will default to black and white.
Entering 4 from the 'Please select an option' menu will start the plot.

2.3 The Plots

The program actually plots two pictures. The first is a box containing the title, a description of the map, and the time and date when the program was run. The second is the map itself. If you are using a graphics terminal you will need to press ENTER after each plot. If you are using either of the Calcomp or Versatec plotters the pictures go to a file which can be plotted later.

When the plot has finished you are asked 'Run program again for same data?'. Entering 'Y' restarts the program, 'N' finishes. After entering 'N' you can send the plot file to the plotter by typing 'N PLOT' and entering the data requested, specific to the plotter required.

Example Plots

The scales given below are for the original plots, which are reproduced here smaller than their actual size. The standard latitude for all the plots is 0°.

Plot 1: Box containing description of data profile plot (plot 8). The scaling factor is 5, i.e. 5 mm per hundred nanoteslas, and the offset is 0 mm.

Plot 2: *Discovery* cruise 161, Mercator projection, 1 : 15 million, plot of ship's track only.

Plot 3: *Farnella* cruise 2/90, Mercator projection, 1 : 1 million, data annotation (residual magnetic anomaly).

Plot 4: *Charles Darwin* cruise 20/87, Mercator projection, 1 : 1 million, data profiles (corrected depth and free air anomaly). Plotted without grid.

Plot 5: *Discovery* cruise 161, Mercator projection, 1 : 15 million, plot where data is valid (residual magnetic anomaly). This plot can be compared with plot 2 to show the sections of the cruise where magnetic data was collected.

Plot 6: *Discovery* cruise 189, Mercator projection, 1 : 15 million, annotated with time every 1440 minutes to give day numbers.

Plot 7: Section of the Madeira Rise showing all IOS cruises, Mercator projection, 1 : 750,000, plot of ships' tracks with cruise names, without grid.

Plot 8: *Farnella* cruise 2/90, Mercator projection, 1 : 1 million, data profile (residual magnetic anomaly) with time annotation. This plot shows the same area as plot 3.
Plot 1: Box containing description of data profile plot (plot 8). The scaling factor is 5, i.e. 5 mm per hundred nanoteslas, and the offset is 0 mm.
Plot 2: 
*Discovery* cruise 161, Mercator projection, 1:15 million, plot of ship's track only.
Plot 3: *Famella* cruise 2/90, Mercator projection, 1:1 million, data annotation (residual magnetic anomaly).
Plot 4: *Charles Darwin* cruise 20/87, Mercator projection, 1:1 million, data profiles (corrected depth and free air anomaly). Plotted without grid.
Plot 5: *Discovery* cruise 161, Mercator projection, 1:15 million, plot where data is valid (residual magnetic anomaly). This plot can be compared with plot 2 to show the sections of the cruise where magnetic data was collected.
Plot 6. *Discovery* cruise 189. Mercator projection, 1 : 15 million, annotated with time every 1440 minutes to give day numbers.
Plot 7: Section of the Madeira Rise showing all IOS cruises. Mercator projection, 1: 750,000, plot of ships' tracks with cruise names, without grid.
Plot 8. *Farnella* cruise 2/80, Mercator projection, 1:1 million, data profile (residual magnetic anomaly) with time annotation. This plot shows the same area as plot 3.
3 PROGRAMMER'S GUIDE

The purpose of this section is to aid understanding of the program so that it may be modified. It describes the language used, file names, program structure, the routines and the macro used to run it.

3.1 Language Details

GEOPLT runs on an IBM mainframe using the CMS operating system. It is written in standard FORTRAN 77 [Metcalf 1985], with the exception that it makes use of the END DO statement as implemented by most compilers. The macro language used in the control program is IBM REXX [Jones, 1987].

3.2 Files

GEOPLT EXEC G  macro to run program
TRACKS MODULE G  executable code. Generated from TRACKS TRAK by typing 'F77C TRACKS TRAK', 'LOAD TRACKS' then 'GENMOD TRACKS'.
TRACKS TRAK G  FORTRAN source code

3.3 Program Structure

Figure 3.1 shows the structure of the TRACKS TRAK source code. Each procedure is shown above the subprocedures that it may involve. For instance TRAKDP involves drawing the track and then drawing the profiles. FORTRAN subroutines are shown in capital letters.

C1 to C11 are conditions which control how often a procedure is called, or whether it is called at all (see key below diagram).

C1: Until user answers 'N' to 'Re-enter parameters'
C2: Until user starts plot
C3: Plot of ship's track only required
C4: Data annotation required
C5: Plot of track with profile required
C6: Plot when variable is present required
C7: Plot with time annotation required
C8: Plot with cruise names
C9: Time not required on data profiles
C10: Time required on data profiles
C11: For each profile
Figure 3.1: Structure diagram of TRACKS TRAK

Key:

A consists of B and C

D chooses E if C1 is true, F if C2 is true

G consists of H repeated. C3 controls the number of repetitions
3.4 Routines

This section describes the input, output and action of each routine in the program.

Key to variable types:

[I] = integer
[R] = real
[C] = character * (*), i.e. unknown length
[Ch] = character * n, i.e. fixed length
[L] = logical
[F] = function

3.4.1 Program TRACKS

Gets the input parameters from the keyboard and calls the plot routines.

Action: Set up initial values
        Repeat
        Display title
        Call MENU to get plot option
        If option = 2 or 4
          Call MENU to get variable
          Get data associated with variable
          If option = 5 or 7
            Get time interval
          If option = 3 or 7
            Call CLIST to get variables
            Get data associated with variables
          If corrected depth required
            Ask if uncorrected depth should be used when corrected depth
            is unavailable
            Call CLIST to get devices
            Call MENU to get projection
            If irregular projection
              Get central meridian
            Get standard latitude
            Get map limits
            Display list of suggested scales
            Get scale
            Get title
            Display summary
            Ask if user wants to re-enter parameters
        Until user replies 'N'
Repeat
Display default values
Call MENU to get option
If option = 1
   Call MENU to get format
   If user defined format
      Get format details
   Else if option = 2
      toggle grid
   Else if option = 3
      call MENU to get new colours
      set colour for uncorrected depth
Set up format
Until option = 4
Open path to devices
Call DESC to plot description
Call DRAW to plot map
Terminate program

3.4.2 Subroutine MAGMES
Displays information about the geomagnetic reference fields available.

Called by: PROGRAM TRACKS

Action: Call MAGSYN to get date of latest DGRF
          Display date, followed by date + 5 (= date of IGRF)

3.4.3 Subroutine GETINT (PROMPT, NUMBER)
Prompts the user for an integer.

Called by: PROGRAM TRACKS

In: PROMPT [C]: character string used to prompt user
    NUMBER [I]: previous value of the number (-999 if no previous value)
Out: NUMBER [I]: new value of the number

Action: Display prompt
        Loop
        Get input
        If input present
           NUMBER = input
           Exit loop
        Else if previous value of NUMBER present
           Exit loop
        End loop
3.4.4 Subroutine GETLOG (PROMPT, LVALUE, SET)

Prompts the user for a YES/NO answer, using the reply to set a logical variable.

Called by: PROGRAM TRACKS

In:  
PROMPT [C]: character string used to prompt user  
LVALUE [L]: previous value of logical variable  
SET [L]: true if LVALUE has been previously set

Out:  
LVALUE [L]: new value of logical variable

Action:  
If SET  
   If LVALUE  
      Set previous reply to 'Y'  
   Else  
      Set previous reply to 'N'  
   Else  
      Set previous reply to 'null' character ('$')  
Repeat  
   Call GETSTR to get new reply  
   Until reply = 'Y' or 'N'  
   Set LVALUE to (reply = 'Y')

3.4.5 Subroutine GETREL (PROMPT, LENGTH, MIN, MAX, NUMBER)

Prompts the user for a real number.

Called by: PROGRAM TRACKS

In:  
PROMPT [C]: character string used to prompt user  
LENGTH [I]: number of characters used for prompt  
MIN [I]: minimum value of number  
MAX [I]: maximum value of number  
(NUMIN and MAX are both 0 if there are no limits)  
NUMBER [R]: previous value of the number (-999. if no previous value)

Out:  
NUMBER [R]: new value of number

Action:  
Display prompt, with range if specified  
Loop  
   Get input  
   If input present  
      NUMBER = input  
      If within limits (or no limits)  
         Exit loop  
   Else if previous value of NUMBER present  
      Exit loop  
End loop
3.4.6 Subroutine GETSTR (PROMPT, STRING)

Prompts the user for a string.

Called by: PROGRAM TRACKS, SUBROUTINE GETLOG

In:   
PROMPT [C]: character string used to prompt user
STRING [C]: previous value of the string ('$' if no previous value)

Out:  
STRING [C]: new value of string

Action:  
Display prompt
Loop
   Get input
      If input present
         STRING = input
      Exit loop
      Else if previous value of STRING present
         Exit loop
End loop

3.4.7 Subroutine MESSGE (MESS, LENGTH, NOLINS)

Displays a message on the screen.

Called by: PROGRAM TRACKS, SUBROUTINE MAGMES

In:  
MESS [C]: array containing lines of message
LENGTH [I]: maximum line length
NOLINS [I]: number of lines in message

Action:  
Display message
3.4.8 Subroutine **MENU** (**TITLE, LENGTH, NOLINS, CHOICE, SELECT**)  

Prompts the user to choose from a menu.

Called by: PROGRAM TRACKS

In:
- **TITLE [C]**: header for menu
- **LENGTH [I]**: maximum length of elements of CHOICE
- **NOLINS [I]**: number of lines of choices
- **CHOICE [C]**: array containing choices, one per element
- **SELECT [I]**: number of previous selection (0 if no previous selection)

Out:
- **SELECT [I]**: number of new selection

Action:
Display menu
Loop
  Get input
  If input present
    SELECT = input
  If SELECT is valid choice number
    Exit loop
End loop

3.4.9 Subroutine **CLIST** (**TITLE, LENGTH, NOLINS, CHOICE, SELECT**)  

Prompts the user to choose one or more items from a checklist.

Called by: PROGRAM TRACKS

In:
- **TITLE [C]**: header for checklist
- **LENGTH [I]**: maximum length of elements of CHOICE
- **NOLINS [I]**: number of lines of choices
- **CHOICE [C]**: array containing choices, one per element
- **SELECT [L]**: array containing previous selections

Out:
- **SELECT [L]**: array containing new selections

Action:
Count number of items selected
Loop
  Display checklist, marking those items selected
  SEL = 0
  Get input
  If input present
    SEL = input
  If SEL = 0 and number of items selected > 0
    Exit loop
  If SEL is valid selection
    If already selected
      SELECT (SEL) = false
      decrement number of items selected
    Else
      SELECT (SEL) = true
      increment number of items selected
  End loop
3.4.10 Subroutine DESCR (TITLE, SCALE, SLAT, PLOT, VARN, DPVAR, PANGLE, PSacle, PSTART, COLOUR, UNCCOL, FILLIN, RECALC)

Draws a box containing a description of the plot.

Called by: PROGRAM TRACKS

In:

TITLE [C45]: title of plot
SCALE [R]: scale of map
SLAT [R]: standard latitude
PLOT [I]: number of plot option
VARN [I]: number of variable selected
DPVAR [I]: array containing numbers of variables selected for profiles
PANGLE [R]: angle for profiles
PSacle [R]: array containing scale factors for profiles
PSTART [R]: array containing offsets for profiles
COLOUR [I]: array containing pen numbers for profiles
UNCCOL [I]: pen number to be used for uncorrected depth, if FILLIN is true
FILLIN [L]: true if uncorrected depth is being used to fill gaps in corrected
RECALC [L]: true if residual magnetic anomalies are to be recalculated

Action:

Draw box
Write box header
Clear text array
Write descriptions to array
Plot title
For each array element
  If data profile variable name
    change pen to correct colour
  If 'Uncorrected depth' line in corrected depth annotation
    change pen to UNCCOL
Plot element
Change to black pen
Plot IOS logo
3.4.11 Subroutine DRAW (MAP2)

Sets up the map and calls the routines to plot the track.

Called by: PROGRAM TRACKS

In:

MAP2 [F]: map projection function
XLEFT [R]: longitude of left map limit
XRIGHT [R]: longitude of right map limit
YBASE [R]: latitude of lower map limit
YTOP [R]: latitude of upper map limit
PLOT [I]: number of plot option
SCALE [R]: scale of map
SLAT [R]: standard latitude
SLONG [R]: central meridian
VVARNO [I]: number of variable selected
DPVAR [I]: array containing numbers of variables selected for profiles
PANGLE [R]: angle for profiles
PSCALE [R]: array containing scale factors for profiles
PSTART [R]: array containing offsets for profiles
INTVAL [I]: time interval between tick marks
ODR [L]: true if grid is required
LNDCOL [I]: pen number used to plot land
TRKCOL [I]: pen number used to plot the ship's track
ANNCOL [I]: pen number used to plot annotation
UNNCOL [I]: pen number used to plot uncorrected depth where this is plotted in place of corrected
COLOUR [I]: array containing pen numbers for profiles
NUMTIS [I]: array containing number of units used when calculating scale factors
FILLIN [L]: true if uncorrected depth is being used to fill gaps in corrected
RECALC [L]: true if residual magnetic anomalies are to be recalculated
PROJ [I]: projection number
NHLINS [I]: number of lines in file header

Action:
Set up plot
Plot and annotate axes
Set pen colours
Read file header
Case PLOT of 1: call TRAK
  2: call DATANN
  3: call TRAKDP, passing 0 as time interval
  4: call TRAKDV
  5: call TRAKTA
  6: call TRAKCD
  7: call TRAKDP, passing INTVAL as time interval

Plot land outline
3.4.12 Subroutine TRAK (MAP2)

Draws the ship's track.

Called by: SUBROUTINEs DRAW, TRAKDP

In:
- MAP2 [F]: map projection
- FMTA [C]: input file format
- FMTNO [I]: format number
- XLEFT [R]: longitude of left map limit
- XRIGHT [R]: longitude of right map limit
- YBASE [R]: latitude of lower map limit
- YTOP [R]: latitude of upper map limit

Action:
- repeat
  - read line from file
  - if longitude < XLEFT
    - longitude = longitude + 360
  - if latitude and longitude are on map and less than 20 minutes since last plot
    - plot point
  - else
    - signal break
- until end of file

3.4.13 Subroutine DATANN (MAP2, VARNO, UNCCOL, ANNCOL, FILLIN, RECALC)

Plots a variable along the ship's track.

Called by: SUBROUTINE DRAW

In:
- MAP2 [F]: map projection function
- VARNO [I]: number of variable to be plotted
- ANNCOL [I]: pen number used to plot annotation
- UNCCOL [I]: pen number used to plot uncorrected depth where this is plotted in place of corrected
- FILLIN [L]: true if uncorrected depth is being used to fill gaps in corrected
- RECALC [L]: true if residual magnetic anomalies are to be recalculated
- FMTA [C60]: input file format
- FMTNO [I]: format number
- XLEFT [R]: longitude of left map limit
- XRIGHT [R]: longitude of right map limit
- YBASE [R]: latitude of lower map limit
- YTOP [R]: latitude of upper map limit

Action:
- repeat
  - read line from file
  - if longitude < XLEFT
    - longitude = longitude + 360
  - if MGD-77 format
    - convert to ORACLES format
  - if RECALC and variable is residual magnetic anomaly
    - recalculate residual magnetic anomaly
  - if latitude and longitude are on map and less than 20 minutes since last plot
    - if variable present
plot variable
else if FILLIN and variable is corrected depth and uncorrected is available
    change pen to UNCCOL
    plot uncorrected depth
    change pen to ANNCOL
until end of file

3.4.14 Subroutine TRAKDP (MAP2, VARNO, COLOUR, FILLIN, RECALC, OFFSET, ANGLE, SCALE, NUNITS, INTVAL, NHLINS)

Plots data profiles along the ship’s track.

Called by: DRAW

In:

MAP2 [F]: map projection function
VARNO [I]: array containing numbers of variables selected for profiles
COLOUR [I]: array containing pen numbers for profiles
FILLIN [L]: true if uncorrected depth is being used to fill gaps in corrected
RECALC [L]: true if residual magnetic anomalies are to be recalculated
OFFSET [R]: array containing offsets for profiles
ANGLE [R]: angle for profiles
SCALE [R]: array containing scale factors for profiles
NUNITS [I]: array containing number of units used when calculating scale factors
INTVAL [I]: time interval between tick marks
NHLINS [I]: number of lines in file header

Action:

If INTVAL > 0
    Call TRAKTA to plot ship’s track
Else
    Call TRAK to plot ship’s track
For each variable to be plotted
    Select pen
    Call PROFIL
3.4.15 Subroutine PROFIL (MAP2, VARN0, FILLIN, RECALC, OFFSET, ANGLE, SCALE, NHLINS)

Plots one data profile.

Called by: TRAKDP

In:
- MAP2 [F]: map projection function
- VARN0 [I]: number of variable selected for profile
- FILLIN [L]: true if uncorrected depth is being used to fill gaps in corrected
- RECALC [L]: true if residual magnetic anomalies are to be recalculated
- OFFSET [R]: offsets for profile
- ANGLE [R]: angle for profile
- SCALE [R]: scale factors for profile
- NHLINS [I]: number of lines in file header
- FMTA [C60]: input file format
- FMTNO [I]: format number
- XLEFT [R]: longitude of left map limit
- XRIGHT [R]: longitude of right map limit
- YBASE [R]: latitude of lower map limit
- YTOP [R]: latitude of upper map limit

Action:
- rewind file and skip header
- set up profile mode
- repeat
- read line from file
- if longitude < XLEFT
-   longitude = longitude + 360
- if RECALC and variable is residual magnetic anomaly
-   recalculate residual magnetic anomaly
- if FILLIN and no corrected depth and uncorrected available
-   corrected depth = uncorrected
- if MGD-77 format
-   convert to ORACLES format
- if latitude and longitude are on map and variable present
-   if less than 20 minutes since last plot
-     plot profile point
- else
-   signal break
- else
-   signal missing point
- until end of file
3.4.16 Subroutine TRAKDV (MAP2, VARNO)

Plots the ship's track when a variable is present.

Called by: SUBROUTINE DRAW

In:
- MAP2 [F]: map projection function
- VARNO [I]: variable whose presence is to be checked
- FMTA [C80]: input file format
- FMTNO [I]: format number
- XLEFT [R]: longitude of left map limit
- XRIGHT [R]: longitude of right map limit
- YBASE [R]: latitude of lower map limit
- YTOP [R]: latitude of upper map limit

Action:
- repeat
  - read line from file
  - if longitude < XLEFT
    - longitude = longitude + 360
  - if MGD-77 format
    - convert to ORACLES format
  - if latitude and longitude are on map and variable present
    - if less than 20 minutes since last plot
      - plot point
    - else
      - signal break
  - else
    - signal missing point
- until end of file

3.4.17 Subroutine TRAKTA (MAP2, INTVAL)

Plots the ship's track annotated with the time

Called by: SUBROUTINES DRAW, TRAKDP

In:
- MAP2 [F]: map projection function
- INTVAL [I]: time interval between tick marks in minutes
- FMTA [C80]: input file format
- FMTNO [I]: format number
- XLEFT [R]: longitude of left map limit
- XRIGHT [R]: longitude of right map limit
- YBASE [R]: latitude of lower map limit
- YTOP [R]: latitude of upper map limit
Action: repeat
read line from file
if longitude < XLEFT
  longitude = longitude + 360
if latitude and longitude are on map and less than 20 minutes since last plot
  if interval has passed
    if interval is at least a day
      plot point and annotate with day number
    else if six-hour mark
      plot point and annotate with day number and hour
    else
      plot point and annotate with hour and minute
  else
    plot point
else
  signal break
until end of file

3.4.18 Subroutine TRAKCD (MAP2)

Plots the ship's track and annotates it with the day number and cruise name.

Called by: SUBROUTINE DRAW

In:
  MAP2 [F]: map projection
  FMTA [C80]: input file format
  FMTNO [I]: format number
  XLEFT [R]: longitude of left map limit
  XRIGHT [R]: longitude of right map limit
  YBASE [R]: latitude of lower map limit
  YTOP [R]: latitude of upper map limit

Action: WANT = false
repeat
read line from file
if longitude < XLEFT
  longitude = longitude + 360
if latitude and longitude are on map and less than 20 minutes since last plot
  if midnight has passed
    WANT = true
  if WANT and the ship is moving
    calculate the ship's bearing
    annotate with day and cruise number in the direction of the bearing
    WANT = false
  else
    plot point
else
  signal break
WANT = true
until end of file
3.4.19 Logical Function ONMAP (X, Y)

Tests if the coordinates are within the map limits.

Called by: SUBROUTINES TRAK, DATANN, PROFIL, TRAKDV, TRAKTA, TRAKCD

In:
- X [R]: longitude to be tested
- Y [R]: latitude to be tested
- XLEFT [R]: longitude of left map limit
- XRIGHT [R]: longitude of right map limit
- YBASE [R]: latitude of lower map limit
- YTOP [R]: latitude of upper map limit

Out:
- ONMAP [L]: true if the coordinates are within the map limits

Action:
\[ \text{ONMAP} = (X \geq XLEFT \text{ and } X \leq XRIGHT \text{ and } Y \geq YBASE \text{ and } Y \leq YTOP) \]

3.4.20 Subroutine MGDZOR (VAR)

Converts MGD-77 format variables to ORACLES format.

Called by: SUBROUTINES DATANN, PROFIL, TRAKDV

In:
- VAR [R]: array containing MGD-77 variables

Out:
- VAR [R]: array containing ORACLES variables

Action:
- if VAR (1) [two way travel time] > 99
  - VAR (1) = 0
- if VAR (2) [corrected depth], VAR (3) [total magnetic anomaly] or VAR (4) [residual magnetic anomaly] > 99999
  - VAR (2, 3, or 4) = 0
- if VAR (5) [free air anomaly] > 9999
  - VAR (5) = 0
- multiply VAR (1) by 750 to get uncorrected depth

3.4.21 Integer Function DAYNO (DAY, MONTH, YEAR)

Converts a date from day/month/year format to the Julian day.

Called by: SUBROUTINES TRAK, DATANN, PROFIL, TRAKDV, TRAKTA, TRAKCD

In:
- DAY [I]: day of the month
- MONTH [I]: month number
- YEAR [I]: year

Out:
- DAYNO [I]: Julian day

Action:
- \[ \text{DAYNO} = \text{number of days from start of year to end of previous month} + \text{DAY} \]
- if \text{YEAR} \text{ is a leap year and } \text{MONTH} > 2
  - \[ \text{DAYNO} = \text{DAYNO} + 1 \]
3.4.22 Integer Function FULLYR (YEAR)

Converts a 2-digit year to a 4-digit year.

Called by: SUBROUTINES DATANN, PROFIL

In:
  YEAR [I]:  2-digit year

Out:
  FULLYR [I]:  4-digit year

Action:
  if YEAR > 30
    FULLYR = 1900 + YEAR
  else
    FULLYR = 2000 + YEAR

3.4.23 Subroutine MAGSYN (DATE, ITYPE, ALT, COLAT, ELONG, X, Y, Z, F)

Calculates the expected total magnetic field.

Called by: SUBROUTINES MAGMES, DATANN, PROFIL

In:
  DATE [R]:  date in years and fractions of a year, or 0 to return date of latest
             DGRF
  ITYPE [I]:  1 if geodetic coordinates are being used, 2 if geocentric
  ALT [R]:  height above sea level in km (ITYPE = 1), or distance from
             centre of earth in km (ITYPE = 2)
  COLAT [R]:  colatitude in degrees
  ELONG [R]:  east longitude in degrees

Out:
  DATE [R]:  Date of latest DGRF (only returned if a DATE
             of 0, is passed to the subroutine)
  X [R]:  north component of magnetic field (nanoteslas)
  Y [R]:  east component of magnetic field (nanoteslas)
  Z [R]:  vertical component of magnetic field (nanoteslas)
  F [R]:  total magnetic field (nanoteslas)

Action:
  if date is out of bounds
    if date is 0
      return date of latest DGRF
    else
      abort
  set initial values
  if ITYPE = 1
    convert to geocentric coordinates
    compute Schmidt quasi-normal coefficients
    synthesize X, Y and Z
    if ITYPE = 1
      convert back to geodetic coordinates
    calculate F
3.5 Listing of PROGRAM TRACKS TRAK

PROGRAM TRACKS

C Plots ships' tracks with optional annotation

C Written May 1990 by Stephen Buxton, Institute of Oceanographic
C Sciences, Wormley, Surrey.

C Based on an earlier program COMTRACK, written by Tim Le Bas, IOS.

EXTERNAL EMERG, EUTM, SGNOMT, SLEAC, SORTHT

CHARACTER * 1 NULL
CHARACTER * 9 UNITS (3)
CHARACTER * 10 COLNAM (4)
CHARACTER * 20 FMTNAM (4), DEVNAM (4)
CHARACTER * 30 OPTNAM (4)
CHARACTER * 40 PRONAM (5), VARNAM (3), DEFNAM (8), MSIZE (6)
CHARACTER * 43 SCALEQ
CHARACTER * 45 TITLE
CHARACTER * 50 PLTNAM (7)
CHARACTER * 70 VALNAM (11)
CHARACTER * 80 IN, OUT, FMTA

INTEGER PLOT, PROJ, INTVAL, VAR, VARNO, DPVAR (3)
INTEGER FMTNO, DEVNO, OPTION, NUNITS (3), NHLINS
INTEGER LNDCOL, TRKCOL, ANNCOL, UNCCOL, COLOUR (3)
INTEGER BLACK, RED, GREEN, BLUE
REAL XLEFT, XRIGHT, XR, YBASE, YTOP, SCALE, SLONG, SLAT
REAL PANGLE, PSSCALE (3), PSTART (3), SM, SN, C1, C2
LOGICAL VARSEL (3), USEDEV (4), GRID, FILLIN, RECALC, REDO
LOGICAL CORDEP, SET

COMMON /FMT/ FMTA, FMTNO
COMMON /LIMITS/ XLEFT, XRIGHT, YBASE, YTOP
COMMON /PARAMS/ PLOT, SCALE, SLAT, SLONG, VARNO, DPVAR, PANGE
& PSSCALE, PSTART, INTVAL, GRID, LNDCOL, TRKCOL, ANNCOL, UNCCOL
& COLOUR, NUNITS, NHLINS, FILLIN, RECALC, PROJ
COMMON /CLEAR/ OUT
PARAMETER (BLACK = 1, RED = 2, GREEN = 3, BLUE = 4)
PARAMETER (NULL = '')
DATA PRONAM /Ellipsoidal Mercator
& 'Universal Transverse Mercator
& 'Gnomonic Transverse
& 'Lambert Cylindrical
& 'Orthographic Transverse
& 'DATA DEVNAM /Graphics Terminal
& 'Calcomp
& 'Versatec
& 'IBM PS/2
& 'DATA VARNAM /Depth
& 'Residual Magnetic Anomaly
& 'Free Air Anomaly
& 'DATA OPTNAM /Choose different format
& 'Turn grid off
& 'Change default colours
& 'Start plot
& 'DATA FMTNAM /Oracles
& 'Merge-Merge
& 'MCD-77
& 'User-Defined
& 'DATA COLNAM /Black
& 'Red
& 'Green
& 'Blue
& 'DATA PLTNAM /Plot ship's track only
& 'Plot data annotation along track
& 'Plot data profile along track
& 'Plot ship's track only when data valid
& 'Plot ship's track with time
& 'Plot ship's track with cruise names
& 'Plot data profile and time along track
C Set up initial values

PLOT = 0
VAR = 0
PROJ = 0
PANGLE = -999.
SLONG = -999.
SLAT = -999.
XLEFT = -999.
XR0 = -999.
YBASE = -999.
YTOP = -999.
SCALE = -999.
INTVAL = -999
NHLENS = -999
TITLE (1 : 1) = NULL
FMTA (1 : 1) = NULL
FSTART (1) = 0.
FSTART (2) = 0.
FSTART (3) = 0.
PSCALE (1) = 5.
PSCALE (2) = 10.
PSCALE (3) = 20.
NUNIT (1) = 1000.
NUNIT (2) = 100.
NUNIT (3) = 100.
UNITS (1) = 'm'
UNITS (2) = 'nanoTesla'
UNITS (3) = 'milligals'
SET = .FALSE.
CORDEP = .FALSE.
DO I = 1, 3
    VARSEL (I) = .FALSE.
END DO
DO I = 1, 4
    USEDEV (I) = .FALSE.
END DO

IN = 'EXEC CLRSCRN'
OUT = '
CALL TOKEN (IN, OUT, NTOK, ISTAT)

C CALL CMSCMD (OUT, ISTAT) will now clear the screen ***
1 CALL CMSCMD (OUT, ISTAT)
WRITE (6, '(3TX, A6)') 'TRACKS'
WRITE (6, '(3TX, A6/)') '-------'
WRITE (6, '(2SX, A30/)') 'Program to plot ships' tracks'
CALL MENU ('Please choose an option', 50, 7, PLTNAM, PLOT)
IF (PLOT_EQ. 2 OR PLOT_EQ. 4) THEN
C      Plot variable / track when variable present
CALL CMSCMD (OUT, ISTAT)
    CALL MENU ('Which variable?', 40, 3, VARNAM, VAR)
    CALL CMSCMD (OUT, ISTAT)
    IF (VAR_EQ. 1) THEN
C      Depth, so ask if corrected
    CALL GETLOG ('Corrected depth (Y/N)', CORDEP, SET)
    IF (CORDEP) THEN
        VARNO = 2
    ELSE
        VARNO = 1
    END IF
    ELSE IF (VAR_EQ. 2) THEN
C      Magnetics, so ask if recalculated
    CALL MAGMES
    CALL GETLOG
    & ('Recalculate residual magnetic anomaly? (Y/N)',
    & RECALC, SET)
    VARNO = 4
    ELSE IF (VAR_EQ. 3) THEN
C      Gravity
    VARNO = 5
    END IF
    END IF

IF (PLOT_EQ. 5 OR PLOT_EQ. 7) THEN
C      Plot with time
    CALL CMSCMD (OUT, ISTAT)
    CALL GETINT ('Tick interval in minutes (must be exact divisio
&' n/multiple of an hour)', INTVAL)
    IF (MOD (60, INTVAL) .NE. 0 .AND. MOD (INTVAL, 60) .NE.
& 0) INTVAL = -.999
    IF (MOD (60, INTVAL) .NE. 0 .AND. MOD (INTVAL, 60) .NE.
& 0) GO TO 30
    END IF
IF (PLOT .EQ. 3 .OR. PLOT .EQ. 7) THEN
C Data profile
CALL CLIST ('Which variables?', 40, 3, VARNAM, VARSEL)
CALL CMSCMD (OUT, ISTAT)
CALL GETREL ('Bearing for data profile ', 25, 0, 360, PANGLE)
DO I = 1, 3
   IF (VARSEL (I)) THEN
      CALL CMSCMD (OUT, ISTAT)
      WRITE (6, * ) VARNAM (I)
      WRITE (SCALEQ, ('"Scaling factor for data (mm)/",
         14, 1X, A, "")') NUNITS (I), UNITS (I)
      CALL GETREL (SCALEQ, 43, 0, 0, PSCALE (I))
      CALL GETREL
      & ('Origin to be subtracted (mm)
         43, 0, 0, PSTART (I))
   END IF
END DO
DPVAR (1) = 0
DPVAR (2) = 0
DPVAR (3) = 0
CALL CMSCMD (OUT, ISTAT)
IF (VARSEL (1)) THEN
C Depth, so ask if corrected
20 CALL GETLOG ('Corrected depth? (Y/N)', CORDEP, SET)
   IF (CORDEP) THEN
      DPVAR (1) = 2
   ELSE
      DPVAR (1) = 1
   END IF
END IF
   IF (VARSEL (2)) THEN
C Magnetics, so ask if recalculated
      CALL MAGMES
      CALL GETLOG
      & ('Recalculate residual magnetic anomaly? (Y/N)',
         & RECALC, SET)
      DPVAR (2) = 4
   END IF
   IF (VARSEL (3)) THEN
C Gravity
      DPVAR (3) = 5
   END IF
END IF

IF ((PLOT .EQ. 2 .OR. PLOT .EQ. 3 .OR. PLOT .EQ. 7) .AND. CORDEP) THEN
CALL GETLOG ('Use uncorrected depth if corrected is unavailable
   &le? (Y/N)', FILLIN, SET)
END IF

CALL CLIST ('Which devices?', 20, 4, DEVNAM, USEDEV)

C Get projection and map limits
CALL CMSCMD (OUT, ISTAT)
CALL MENU ('Which projection?', 40, 5, PRONAM, PROJ)
CALL CMSCMD (OUT, ISTAT)
C Ask for Central Meridian if transverse projection
  IF (PROJ EQ. 2 .OR. PROJ EQ. 3 .OR. PROJ EQ. 5) CALL GETREL
  & ("Central Meridian (positive = east) ", 36, -180, 180, SLONG)
  CALL GETREL ("Standard Latitude (positive = north)", 36,
  & -90, 90, SLAT)
  CALL CMSCMD (OUT, ISTAT)
  CALL GETREL ("Longitude of left edge  (positive = east)",
  & 42, -180, 180, XLEFT)
  CALL GETREL ("Longitude of right edge (positive = east)",
  & 42, -180, 180, XR0)
  CALL CMSCMD (OUT, ISTAT)
  CALL GETREL ("Latitude of lower edge (positive = north)",
  & 41, -90, 90, YBASE)
  CALL GETREL ("Latitude of upper edge (positive = north)",
  & 41, -90, 90, YTOP)
C Right limit must be greater than left limit
  IF (XR0 LT. XLEFT) THEN
    XRIGHT = XR0 + 360.
  ELSE
    XRIGHT = XR0
  END IF
C Suggest scales and get the actual scale from the user

C1 = 111317099.7 * COS (SLAT * 0.017453292)
C2 = 57.2957751 * (ALOG (TAN (0.785398163 + YTOP * 0.008726646))
  & - ALOG (TAN (0.785398163 + YBASE * 0.008726646)))
MSIZE (1) = 'Suggested scales:'
MSIZE (2) = 'Approximate size Scale'
DO I = 0, 3
  SN = (2 ** I) * 100
  SM = SN / C2
  WRITE (MSIZE (3 + I), '(3, " mm N-S.", 14, " mm E-W.",
  & 110, "0000""") NINT (SN), 10 * NINT (SM / 10. * (XRIGHT
  & - XLEFT)), NINT (C1 / (SM * 10000)))
END DO
CALL CMSCMD (OUT, ISTAT)
CALL MESSAGE (MSIZE, 40, 6)
CALL GETREL ("Scale (eg. 0.0000001 or 10000000)", 35, 0, 0,
  & SCALE)
IF (SCALE LT. 1) SCALE = 1. / SCALE
CALL CMSCMD (OUT, ISTAT)
CALL GETSTR ('Title (up to 45 characters)', TITLE)
CALL CMSCMD (OUT, ISTAT)
WRITE (VALNAM (1), '("Title", T21, A)') TITLE
WRITE (VALNAM (2), '("Left grid limit", T21, F10.2)') XLEFT
WRITE (VALNAM (3), '("Right grid limit", T21, F10.2)') XR0
WRITE (VALNAM (4), '("Lower grid limit", T21, F10.2)') YBASE
WRITE (VALNAM (5), '("Upper grid limit", T21, F10.2)') YT0P
WRITE (VALNAM (6), '("Scale", T19, F10.0)') SCALE
WRITE (VALNAM (7), '("Projection", T21, A)') PRONAM (PROJ)
DEVNO = 0
DO I= 1, 4
   IF (USEDEV (I)) THEN
      DEVNO = DEVNO + 1
      WRITE (VALNAM (7 + DEVNO), '("Device", I2, T21, A)') &
      DEVNO, DEVNAM (I)
   END IF
END DO
CALL MESSAGE (VALNAM, 70, 7 + DEVNO)
SET = TRUE
CALL GETLOG ('Re-enter these parameters? (Y/N)', REDO, FALSE.)
IF (REDO) GO TO 1

C Set up default values

FMTNO = 1
GRID = TRUE
LNDCOL = BLACK
TRKCOL = BLACK
ANNCOL = BLACK
COLOUR (1) = GREEN
COLOUR (2) = RED
COLOUR (3) = BLUE
50 WRITE (DEFNAM (1), '("Format", T12, A)') FMTNAM (FMTNO)
   IF (GRID) THEN
      WRITE (DEFNAM (2), '("Grid", T12, A)') 'On'
   ELSE
      WRITE (DEFNAM (2), '("Grid", T12, A)') 'Off'
   END IF
WRITE (DEFNAM (3), '("Coastline", T12, A)') COLNAM (LNDCOL)
WRITE (DEFNAM (4), '("Track", T12, A)') COLNAM (TRKCOL)
WRITE (DEFNAM (5), '("Annotation", T12, A)') COLNAM (ANNCOL)
WRITE (DEFNAM (6), '("Bathymetry", T12, A)') COLNAM (COLOUR (1))
WRITE (DEFNAM (7), '("Magnetics", T12, A)') COLNAM (COLOUR (2))
WRITE (DEFNAM (8), '("Gravity", T12, A)') COLNAM (COLOUR (3))
CALL CMSCMD (OUT, ISTAT)
CALL MESSAGE (DEFNAM, 40, 8)
OPTION = 0
CALL MENU ('Please select an option', 30, 4, OPTNAM, OPTION)
IF (OPTION .EQ. 1) THEN
  C Choose a different format
  CALL CMSCMD (OUT, ISTAT)
  CALL MENU ('Which format', 20, 4, FMTNAM, FMTNO)
  IF (FMTNO .EQ. 4) THEN
    CALL GETSTR ('Enter format (Fortran syntax)', FMTA)
    CALL GETINT ('Number of lines in header', NHLINS)
  END IF
ELSE IF (OPTION .EQ. 2) THEN
  C Toggle grid
  IF (GRID) THEN
    GRID = .FALSE.
    OPTNAM (2) (11 : 13) = 'on'
  ELSE
    GRID = .TRUE.
    OPTNAM (2) (11 : 13) = 'off'
  END IF
ELSE IF (OPTION .EQ. 3) THEN
  C Change default colours
  CALL CMSCMD (OUT, ISTAT)
  CALL MENU ('Coastline', 10, 4, COLNAM, LNDCOL)
  CALL CMSCMD (OUT, ISTAT)
  CALL MENU ('Track', 10, 4, COLNAM, TRKCOL)
  CALL CMSCMD (OUT, ISTAT)
  CALL MENU ('Annotation', 10, 4, COLNAM, ANNCOL)
  CALL CMSCMD (OUT, ISTAT)
  CALL MENU ('Bathymetry', 10, 4, COLNAM, COLOUR (1))
  CALL CMSCMD (OUT, ISTAT)
  CALL MENU ('Magnetics', 10, 4, COLNAM, COLOUR (2))
  CALL CMSCMD (OUT, ISTAT)
  CALL MENU ('Gravity', 10, 4, COLNAM, COLOUR (3))
END IF
IF (OPTION .NE. 4) GO TO 50
C Set up colour for uncorrected depth annotation
IF (ANNCOL .EQ. RED) THEN
  UNCCOL = GREEN
ELSE
  UNCCOL = RED
END IF
IF (FMTNO .EQ. 1) THEN
C Oracles format
   FMTA =
 & ' (A8, 1X, 3 (12, 1X), 212, 2X, 2 (F9.4, 2X), 5(F5.0, 2X))'
   NHLINS = 2
ELSE IF (FMTNO .EQ. 2) THEN
C Merge-Merge format
   FMTA =
 & ' (A8, 5X, 312, 1X, 212, 1X, F8.4, F9.4, 10X, 2F5.0, 3X, 3F5.0)'
   NHLINS = 0
ELSE IF (FMTNO .EQ. 3) THEN
C MGDT77 format
   FMTA = ' (1X,A8,5X,312,3X,F8.5,F9.5,1X,F6.1,3X,F6.1,6X,F6
 & 1.25X,F5.1)'
   NHLINS = 24
END IF

C Open path to devices
IF (USEDEV (1)) CALL DEVICE (3179)
IF (USEDEV (2)) CALL DEVICE (1075)
IF (USEDEV (3)) CALL DEVICE (80)
IF (USEDEV (4)) CALL DEVICE (4205)

CALL PAGBEG
CALL DESCR (TITLE, SCALE, SLAT, PLOT, VARNOP, DPPN, PANGLE,
 & PSSCALE, PSTART, UNCCOL, COLOUR, FILEN, RECALC)
IF (PROJ .EQ. 1) THEN
   CALL DRAW (EMERG)
ELSE IF (PROJ .EQ. 2) THEN
   CALL DRAW (EUTM)
ELSE IF (PROJ .EQ. 3) THEN
   CALL DRAW (SNUMT)
ELSE IF (PROJ .EQ. 4) THEN
   CALL DRAW (SLEAC)
ELSE IF (PROJ .EQ. 5) THEN
   CALL DRAW (SORTHT)
END IF
CALL GCLOSE
CALL NCLOSE (2, ISTAT)
END

C-------------------------------
SUBROUTINE MAGMES
C Calls MAGSYN to get the date of the latest DGRF, then displays the
C date, together with the date of the latest IGRF.

REAL DGRF
CHARACTER * 20 MESS (2)

DGRF = 0
CALL MAGSYN (DGRF, 0, 0., 0., 0., X, Y, Z, F)
WRITE (MESS (1), '("Latest DGRF is ", 4I4)') NINT (DGRF)
WRITE (MESS (2), '("Latest IGRF is ", 4I4)') NINT (DGRF) + 5
CALL MESSGE (MESS, 20, 2)
SUBROUTINE GETINT (PROMPT, NUMBER)

C Prints a prompt and gets an integer from the user. If the NUMBER
C parameter is -999 then a value must be input, otherwise the previous
C value is available by pressing Enter.

CHARACTER (*) PROMPT, LINE * 80
INTEGER NUMBER, INPUT
DATA LINE / __________________________
&________________________/'

WRITE (6, '(2X, A)') LINE (LEN (PROMPT))
WRITE (6, ('"","", A, "","")') PROMPT
WRITE (6, ('"","", A, "","")') LINE (LEN (PROMPT))
IF (NUMBER NE. -999) WRITE (6, ('" Or press enter for ", 18)'
& NUMBER

10 REWIND 5
   READ (5, '(BN, 18) END = 100) INPUT
   NUMBER = INPUT
   RETURN

C No number has been input, so check for previous value
100 IF (NUMBER NE. -999) RETURN
C No previous value, so request input again
   WRITE (6, *) "PLEASE TRY AGAIN *
   GO TO 10
   END

SUBROUTINE GETLOG (PROMPT, LVALUE, SET)

C sets a logical value by prompting the user for a 'Y' or 'N' answer.
C SET is true if the value has been set previously.

CHARACTER * 1 REPLY, NULL
CHARACTER * (*) PROMPT
LOGICAL LVALUE, SET
PARAMETER (NULL = "]")
IF (SET) THEN
  IF (LVALUE) THEN
    REPLY = 'Y'
  ELSE
    REPLY = 'N'
  END IF
ELSE
  REPLY = NULL
END IF

10 CALL GETSTR (PROMPT, REPLY)
  IF (REPLY .NE. 'Y' .AND. REPLY .NE. 'N') GO TO 10

  LVALUE = REPLY .EQ. 'Y'
END

SUBROUTINE GETREL (PROMPT, LENGTH, MIN, MAX, NUMBER)
C Prints a prompt of the length given and gets a real number from the
C user, within the range given by MIN & MAX. If MIN & MAX are both zero
C any value is accepted. A default can be passed in using NUMBER, if
C this is not required NUMBER should be -999.

CHARACTER (*) PROMPT, LINE * 50, SPACE * 50
INTEGER LENGTH, MIN, MAX
REAL NUMBER, INPUT
LOGICAL DEFAULT
DATA LINE /'__________________________/'
DATA SPACE /'__________________________/'
WRITE (6, '(2X, A)') LINE (LENGTH)
WRITE (6, ' (" ", A, "]")') PROMPT
IF (MIN .NE. 0 .OR. MAX .NE. 0) THEN
  Print range
    WRITE (6, ' (" ]Between", I7, ", and", I7, A, "]")')
  & MIN, MAX, SPACE (LENGTH - 25)
END IF
WRITE (6, ' (" ", A, "]")') LINE (LENGTH)
DEFAULT = ABS (NUMBER + 999) .GT. 0.01
IF (DEFAULT) WRITE (6, ' (" Or press enter for ".F12.3")') NUMBER

10 REWIND 5
READ (S, '(BN, F10.0)', END = 100) INPUT
NUMBER = INPUT
IF (MIN .EQ. 0 .AND. MAX .EQ. 0) RETURN
IF (NUMBER .GE. REAL (MIN) .AND. NUMBER .LE. REAL (MAX)) RETURN
WRITE (6, '* * OUT OF RANGE *'
GO TO 10
C No number has been input, so check for default
100 IF (DEFAULT) RETURN
C No default, so request input again
   WRITE (6, *) 'PLEASE TRY AGAIN *'
   GO TO 10
END

SUBROUTINE GETSTR (PROMPT, STRING)

C Prints a prompt and gets a string from the user. If the STRING
C parameter is NULL then a value must be input, otherwise the previous
C value is available by pressing Enter.

   CHARACTER * (*) PROMPT, STRING, LINE * 80, INPUT * 80, NULL * 1
   PARAMETER (NULL = 'S')
   DATA LINE / (NULL = 'S') /
   & __________________________ /
   WRITE (6, '(2X, A)') LINE (LEN (PROMPT))
   WRITE (6, (' ', A, '')) PROMPT
   WRITE (6, (' ', A, '')) LINE (LEN (PROMPT))
   IF (STRING (1 : 1) NE NULL) WRITE (6, &
   (' ', Or press enter for ', A)) STRING

10 REWIND 5
   READ (5, '(A)', END = 100) INPUT
   STRING = INPUT
   RETURN

C No string has been input, so check for previous value
100 IF (STRING (1 : 1) NE NULL) RETURN
C No previous value, so request input again
   WRITE (6, *) 'PLEASE TRY AGAIN *'
   GO TO 10
END

SUBROUTINE MESSAGE (MESS, LENGTH, NOLINS)

C Prints a message stored in the array MESS, with maximum line length
C and no. of lines as given

   CHARACTER * (*) MESS (*), LINE * 80, SPACE * 80
   INTEGER LENGTH
   DATA LINE /.________________________ /
   & DATA SPACE / ____________________ /
   & ________________________ /
   C One has to be added to LENGTH to avoid a null string
   WRITE (6, '(2X, A)') LINE (LENGTH + 1)
   DO I = 1, NOLINS
      WRITE (6, (' ', A, A, 'A')) MESS (I), SPACE (LENGTH
   & - LEN (MESS (I)) + 1)
END DO
WRITE (6, ('" \[", A, "]")') LINE (LENGTH + 1)
RETURN
END

SUBROUTINE MENU (TITLE, LENGTH, NOLINS, CHOICE, SELECT)
C Displays a menu with NOLINS selections. TITLE is displayed at the top
C and the choices are in array CHOICE. The user's selection is returned
C in SELECT. If SELECT > 0 is passed to the subroutine, pressing enter
C will keep the current value of SELECT.

CHARACTER (*) TITLE, CHOICE (NOLINS). LINE * 80. SPACE * 80
INTEGER LENGTH, NOLINS, SELECT, INPUT
DATA LINE / ---------------------------------------
& DATA SPACE /-------------------------------------
&   
WRITE (6, ('2X, A)') LINE (LENGTH + 4)
WRITE (6, ('", A, "]")') TITLE. SPACE (LENGTH + 4 -
& LEN (TITLE))
WRITE (6, ('", A, "]")') LINE (LENGTH + 4)

DO I = 1, NOLINS
  WRITE (6, ('", I2, "]", A, "]")') I, CHOICE (I)
END DO
WRITE (6, ('", A, "]")') LINE (LENGTH)
IF (SELECT .GT. 0) WRITE (6, ('" Or press enter for ", A)')
& CHOICE (SELECT)

10 REWIND 5
READ (5, *, END = 100) INPUT
SELECT = INPUT
100 IF (SELECT .GE. 1 .AND. SELECT .LE. NOLINS) RETURN
WRITE (6, *) * PLEASE CHOOSE AGAIN *
GO TO 10
END
SUBROUTINE CLIST (TITLE, LENGTH, NOLINS, CHOICE, SELECT)

C Displays a checklist from which the user can make one or more choices.
C TITLE is displayed at the top, LENGTH is the line length, NOLINS is
C the number of lines of choices, CHOICE is an array containing the
C choices and SELECT is an array containing the user's selections. The
C subroutine terminates when the user selects 0.

CHARACTER * 80 OUT, LINE, SPACE
CHARACTER * (*) TITLE, CHOICE (NOLINS)
INTEGER LENGTH, NOLINS, NOSELS, SEL, INPUT
LOGICAL SELECT (NOLINS)
DATA LINE / ' ' /
& DATA SPACE /
& ' ' / COMMON / CLEAR/ OUT

NOSELS = 0
DO I = 1, NOLINS
  IF (SELECT (I)) NOSELS = NOSELS + 1
END DO
10 CALL CMSCMD (OUT, ISTAT)
  WRITE (6, '(2X, A)') LINE (LENGTH + 5)
  WRITE (6, ('*', ' ', A, ' ', ' ') ) TITLE, SPACE (LENGTH + 5 -
& LEN (TITLE))
  WRITE (6, (' ', A, ' ', ' ') ) 'Enter to finish', SPACE
& (LENGTH - 10)
  WRITE (6, ('*', ' ', A, ' ', ' ') ) LINE (LENGTH + 5)

DO I = 1, NOLINS
  MARK choice if already selected
  IF (SELECT (I)) THEN
    WRITE (6, ('*', ' ', I2, ' ', ' ') ) I, CHOICE
& (I)
  ELSE
    WRITE (6, (' ', ' ', I2, ' ', ' ') ) I, CHOICE
& (I)
  END IF
END DO
WRITE (6, (' ', ' ', ' ') ) LINE (LENGTH)
C Read and examine selection

    SEL = 0
    REWIND 5
    READ (5, *, END = 100) INPUT
    SEL = INPUT
100 IF (SEL .EQ. 0 AND. NOSELS .GT. 0) RETURN
    IF (SEL .GE. 1 AND. SEL .LE. NOLINS) THEN
      IF (SELECT (SEL)) THEN
        SELECT (SEL) = .FALSE.
        NOSELS = NOSELS - 1
      ELSE
        SELECT (SEL) = .TRUE.
        NOSELS = NOSELS + 1
      END IF
    END IF
    GO TO 10
END

C---------------------------------------------------------------------
C
SUBROUTINE DESCR (TITLE, SCALE, SLAT, PLOT, VARNO, DPVAR, PANGLE,
& PSSCALE, PSTART, UNCCOL, COLOUR, FILLIN, RECALC)
C
C Draws a box containing a description of the plot
C
    CHARACTER * 45 TITLE, LINE (11)
    CHARACTER * 1 PIC (36)
    CHARACTER * 8 DATE
    CHARACTER * 30 VNAM (5)
    CHARACTER * 16 DT

    INTEGER PLOT, VARNO, DPVAR (3), NEXT, HOUR, MINUTE, COLOUR (3)
    INTEGER KEYCOL (3), VAR, UNCCOL
    REAL SCALE, SLAT, PANGLE, PSSCALE (3), PSTART (3), TIME
    LOGICAL FILLIN, RECALC

DATA VNAME /UNCORRECTED DEPTH
& 'CORRECTED DEPTH',
& 'NOT USED',
& 'RESIDUAL MAGNETIC ANOMALY',
& 'FREE AIR ANOMALY' /
C Draw box

CALL GRAFIX (210., 110., 0.)
CALL DEFLA2 (0., 210., 0., 110.)
CALL RECLA2 (10., 10., 190., 90., 1, 1)
CALL LINL2 (10., 20., 190., 20., 2)
CALL LINL2 (190., 70., 10., 70., 2)
CALL LINL2 (10., 72., 190., 72., 2)

C Write box header

CALL CHSETS (2)
CALL CHHTS (4)
CALL TEXTB2 (90., 63., 'INSTITUTE OF OCEANOGRAPHIC SCIENCES', 1,
& 35., 0., 2, 4, 0, 2)
CALL CHHTS (2)
CALL TEXTB2 (90., 79.,
& 'DEACON LABORATORY, BROOK ROAD, WORMLEY, SURREY GU8 5UB, ENGLAND',
& 1, 63., 0., 2, 4, 0, 2)
CALL TEXTB2 (90., 74., 'TEL. WORMLEY (0428) 684141', 1, 26., 0., 2,
& 4, 0, 1)

C Clear text array

DO I = 1, 11
   LINE (I) = ''
END DO

C Write plot descriptions to array

WRITE (LINE (1), '("SCALE 1 TO ", -6PF5.2, ",M")') SCALE
IF (SLAT .GT. 0) THEN
   WRITE (LINE (2), '("STANDARD LATITUDE", F5.1,
& " DEG. N")') SLAT
ELSE
   WRITE (LINE (2), '("STANDARD LATITUDE", F5.1,
& " DEG. S")') -SLAT
END IF

IF (PLOT_EQ.1 OR PLOT_EQ.8) THEN
   WRITE (LINE (4), *) 'PLOT OF SHIP'S TRACK'
ELSE IF (PLOT_EQ.2 THEN
   WRITE (LINE (4), *) 'ANNOTATION ALONG SHIP'S TRACK'
   WRITE (LINE (5), *) VNAME (VARN)
   IF (VARN_EQ.2 AND. FILLIN) WRITE (LINE (6), *)
   & 'UNCORRECTED DEPTH'
   IF (VARN_EQ.4 AND. RECALC) WRITE (LINE (6), *)
   & '(RECALCULATED)'
ELSE IF (PLOT_EQ. 3 OR PLOT_EQ. 7) THEN
    WRITE (LINE (4), *) 'PROFILES PLOTTED ALONG TRACK'
    WRITE (LINE (5), 'AT A BEARING OF ', F8.4) PANGLE
    VAR = 0
    DO I = 1, 3
        IF! (DPVAR (I) GT 0) THEN
            WRITE (LINE (6 + VAR * 2), *) VNAME (DPVAR (I))
            WRITE (LINE (7 + VAR * 2), 'SCALING FACTOR', F8.2, 'OFFSET', F10.4) PSCLAE (I), PSTART (I)
            VAR = VAR + 1
            KEYCOL (VAR) = COLOUR (I)
        END IF
    END DO
ELSE IF (PLOT_EQ. 4) THEN
    WRITE (LINE (4), *) 'PLOT WHERE VARIABLE IS PRESENT'
    WRITE (LINE (5), *) VNAME (VARNO)
ELSE IF (PLOT_EQ. 5) THEN
    WRITE (LINE (4), *)
    & 'PLOT OF SHIP S TRACK WITH TIME ANNOTATION'
END IF

C Set date and time of plot.
CALL DATIM (DATE, TIME)
C Time is real number in hours, so convert to hours and minutes
HOUR = INT (TIME)
MINUTE = INT ((TIME - HOUR) * 60)
WRITE (DT, 'I2u":"I2.2, 3X, A8') HOUR, MINUTE, DATE

C Plot descriptions
    CALL CHHITS (2.5)
    CALL TEXLA2 (15, 64, TITLE, 1, 45, 2)
    CALL CHHITS (2.)
    Y = 57.
    DO I = 1, 11
        C Change pens for data profile variable names...
        IF (PLOT_EQ. 3 OR PLOT_EQ. 7 AND (I EQ. 6 OR I
            & EQ. 8 OR I EQ. 10)) CALL TXPENS (KEYCOL (I - 4) / 2))
        C ...and for uncorrected depth on corrected depth plot
        IF (PLOT_EQ. 2 AND I EQ. 6 AND FILLIN) CALL TXPENS
            & (UNCCOL)
            CALL TEXLA2 (15, Y, LINE (I), 1, 45, 2)
            CALL TXPENS (1)
            Y = Y - 3.5
    END DO
    CALL TEXLA2 (15, 13, DT, 1, 16, 2)
CALL TXUPRS (0)
AA = 155.
BB = 45.
RAD = 20.
THET = 153.0
DO I = 1, 36
   THETH = THET - 90.
   RTHET = (THET / 180.) * 3.14
   XP = RAD * COS (RTHET) + AA
   YP = RAD * SIN (RTHET) + BB
   CALL TEXLB2 (XP, YP, PIC (0), 1, 1, THETHW, 2, 3, 0, 2)
   THET = THET - 8.5
   IF (THET LE 0.) THET = 360.
END DO
CALL CCHITS (2.7)
CALL TEXLB2 (155., 47., 'INSTITUTE OF', 1, 12, 0., 3, 4, 0, 2)
CALL TEXLB2 (155., 42., 'OCEANOGRAPHIC', 1, 13, 0., 3, 4, 0, 2)
CALL TEXLB2 (155., 37., 'SCIENCES', 1, 8, 0., 3, 4, 0, 2)

CALL PAGEND
RETURN
END

SUBROUTINE DRAW (MAP2)
C Sets up the chart and calls the routines to plot the track

EXTERNAL MAP2

REAL EQSC, XLEFT, XRIGHT, YBASE, YTOP, SCALE, SLAT, SLONG
REAL PANGLE, PSSCALE (3), PSTART (3)

INTEGER PLOT, VARN0, DPVAR (3), INTVAL, LNDCOL, TRKCOL, ANNCOL
INTEGER UNCCOL, COLOUR (3), NUNITS (3), PROJ, NHLINS
LOGICAL GRID, FILLIN, RECALC

COMMON /LIMITS/ XLEFT, XRIGHT, YBASE, YTOP,
   COMMON /PARAMS/ PLOT, SCALE, SLAT, SLONG, VARN0, DPVAR, PANGLE,
   & PSSCALE, PSTART, INTVAL, GRID, LNDCOL, TRKCOL, ANNCOL, UNCCOL,
   & COLOUR, NUNITS, NHLINS, FILLIN, RECALC, PROJ
PARAMETER (BLACK = 1)
C Reserve space for annotation
   CALL FMTANN (1, 12, 12)
C Find scale at equator
   CALL MAPSCL (MAP2, 1 / SCALE, SLAT, EQSC)
C Set central latitude and longitude
   CALL MAPCEN (SLONG, SLAT)
   IF (PROJ_EQ. 1 OR PROJ_EQ. 4 OR PROJ_EQ. 5) THEN
       C Rectangular projection
       CALL FMTCHR (MAP2, XLEFT, XRIGHT, YBASE, YTOP, EQSC)
       ELSE
           C Irregular projection
           CALL FMTCHI (MAP2, XLEFT, XRIGHT, YBASE, YTOP, EQSC)
       END IF
C Annotate axes

   CALL GRASET
   CALL LINPENS (BLACK)
   CALL TXPENS (BLACK)
   CALL MKPENS (BLACK)
   CALL ANNGEO (2, 1)
   IF (GRID) THEN
       CALL GRISEL (3, 3, 1)
   ELSE
       CALL GRISEL (3, 3, 0)
   END IF
   CALL AXIFA2 (MAP2, 1)
   CALL AXIFA2 (MAP2, 2)
   CALL PL2STD
C Plot track

    CALL ERRSET (255, 50, -1)
    CALL TXPENS (ANNCOL)
    CALL LNPENS (TRKCOL)
    CALL MKPENS (ANNCOL)
C   Turn off marks
    CALL MARSEL (0)
C   Skip header information
    DO I = 1, NHLINS
       READ (2, '()')
    END DO
    IF (PLOT .EQ. 1) THEN
       CALL TRAK (MAP2)
    ELSE IF (PLOT .EQ. 2) THEN
       CALL DATANN (MAP2, VARNO, UNCCOL, ANNCOL, FILLIN, RECALC)
    ELSE IF (PLOT .EQ. 3) THEN
       CALL TRAKDP (MAP2, DPVAR, COLOUR, FILLIN, RECALC, PSTART,
                   &    PANGLE, PSCALE, NUNITS, 0, NHLINS)
    ELSE IF (PLOT .EQ. 4) THEN
       CALL TRAKDV (MAP2, VARNO)
    ELSE IF (PLOT .EQ. 5) THEN
       CALL TRAKTA (MAP2, INTVAL)
    ELSE IF (PLOT .EQ. 6) THEN
       CALL TRAKCD (MAP2)
    ELSE IF (PLOT .EQ. 7) THEN
       CALL TRAKDP (MAP2, DPVAR, COLOUR, FILLIN, RECALC, PSTART,
                   &    PANGLE, PSCALE, NUNITS, INTVAL, NHLINS)
    END IF

C Plot land outline. Terminate

    CALL LNFENS (LNDCOL)
    CALL WORLDM (MAP2)
    CALL FACPND
    RETURN
    END

C-----------------------------------------------------------

SUBROUTINE TRAK (MAP2)
C Plots track of ship

EXTERNAL MAP2

CHARACTER * 80 FMTA
CHARACTER * 8 CNAME
INTEGER DAY, MONTH, YEAR, HOUR, MINUTE, FMTNO
INTEGER TIME, TIME0, DAYNO
REAL LAT, LONG, VAR (5)
LOGICAL ONMAP
COMMON /FMT/FMTA, FMTNO
COMMON /LIMITS/XLEFT, XRIGHT, YBASE, YTOP

C Adjacent points are joined up
CALL LINSEL (1)
CALL POIBEG

10 IF (FMTNO, EQ, 1) OR (FMTNO, EQ, 4) THEN
   READ (2, FMTA, END = 100) CNAME, DAY, MONTH, YEAR, HOUR,
   &       MINUTE, LAT, LONG, (VAR (I), I = 1, 5)
ELSE
   READ (2, FMTA, END = 100) CNAME, YEAR, MONTH, DAY, HOUR,
   &       MINUTE, LAT, LONG, (VAR (I), I = 1, 5)
END IF
C Ensure that lines which cross 180 degrees are plotted correctly
IF (LONG LT XLEFT) LONG = LONG + 360.
IF (ONMAP (LONG, LAT)) THEN
C Check for gap in data
   TIME0 = TIME
   TIME = DAYNO (DAY, MONTH, YEAR) * 1440 + HOUR * 60 + MINUTE
   IF (TIME - TIME0 LE 20 AND TIME - TIME0 GE 0) THEN
      CALL POIFA2 (MAP2, LONG, LAT)
   ELSE
      CALL POIBRK
   END IF
ELSE
   CALL POIBRK
END IF
GO TO 10

100 CALL POIEND
RETURN
END

C---------------------------------------------------------------------

SUBROUTINE DATANN (MAP2, VARNO, UNCCOL, ANNCOL, FILLIN, RECALC)

C Plots the given variable along the track. If FILLIN is true, gaps in
C the corrected depth are filled by uncorrected depth. If RECALC is
C true, the residual magnetic anomaly is recalculated using the latest
C version of the IGRF.
EXTERNAL MAP2

CHARACTER * 80 FMTA
CHARACTER * 8 CNAME
CHARACTER * 6 NUMBER
INTEGER DAY, MONTH, YEAR, HOUR, MINUTE, VARNNO, FMTNO
INTEGER FULLYR, DAYNO, UNCCOL, ANNCOL
REAL LAT, LONG, VAR (5), TOTMAG
LOGICAL ONMAP, FILLN, RECALC
COMMON /FMT/, FMTA, FMTNO
COMMON /LIMITS/, XLEFT, XRIGHT, YBASE, YTOP

C  Don't mark points
CALL MARSEL (0)

C  Numbers are centred on ship's track
CALL PANLIS (2, 3)

C  Extended ASCII set
CALL CHSETS (1)

C  Include spaces in strings
CALL TXSPAS (0)

C  Set character height
CALL CHHITS (1, 2)
CALL POIBEG
10 IF (FMTNO .EQ. 1 .OR. FMTNO .EQ. 4) THEN
   READ (2, FMTA, END = 100) CNAME, DAY, MONTH, YEAR, HOUR, &
   MINUTE, LAT, LONG, (VAR (I), I = 1, 5)
   ELSE
   READ (2, FMTA, END = 100) CNAME, YEAR, MONTH, DAY, HOUR, &
   MINUTE, LAT, LONG, (VAR (I), I = 1, 5)
   END IF
C Ensure that lines which cross 180 degrees are plotted correctly
IF (LONG .LT. XLEFT) LONG = LONG + 360.
C Convert MGD-77 format to Oracles
IF (FMTNO .EQ. 3) CALL MGDS2OR (VAR)
IF (RECALC .AND. VARNO .EQ. 4) THEN
   Recalculate RMA
   CALL MAGSYN (FULLYR (YEAR) + DAYNO (DAY, MONTH, YEAR) / &
   360.0, 1, 0., 90. - LAT, LONG, X, Y, Z, TOTMAG)
   IF (VAR (3) .GT. 0) VAR (4) = VAR (3) - TOTMAG
END IF
IF (ONMAP (LONG, LAT)) THEN
   IF (VAR (VARNO) .NE. 0) THEN
      Plot specified value
      WRITE (NUMBER, ' (15)') NINT (VAR (VARNO))
      CALL POIFC2 (MAP2, LONG, LAT, NUMBER, 5)
   ELSE IF (FILLIN AND VARNO .EQ. 2 AND VAR (1) .NE. 0) THEN
      Plot uncorrected depth in place of corrected
      CALL TXFENS (UNCCOL)
      WRITE (NUMBER, ' (15)') NINT (VAR (1))
      CALL POIFC2 (MAP2, LONG, LAT, NUMBER, 5)
      CALL TXFENS (ANNCOL)
   END IF
END IF
GO TO 100
100 CALL PANOFF
CALL POEND
RETURN
END

SUBROUTINE TRAKDP (MAP2, VARNO, COLOUR, FILLIN, RECALC, OFFSET, &
ANGLE, SCALE, NUNITS, INTVAL, NHLINS)
C Plot data profiles of the given variables along ship's track, using
C the colours given. If INTVAL > 0, the ship's track is annotated with
C the time every INTVAL minutes.
EXTERNAL MAP2

INTEGER VARNO (3), COLOUR (3), NUNITS (3), INTVAL, NHLINS
REAL OFFSET (3), ANGLE, SCALE (3)
LOGICAL FILLIN, RECALC
C Draw ship's track
IF (INTVAL .GT. 0) THEN
   CALL TRAKTA (MAP2, INTVAL)
ELSE

CALL TRAK (MAP2)
END IF

CALL POIBEG
DO I = 1, 3
  IF (VARNO (I) .GT. 0) THEN
    C Select pen
    CALL LNPENS (COLOUR (I))
    C Plot profile
    CALL PROFI (MAP2, VARNO (I), FILLIN, RECALC,
    & OFFSET (I), ANGLE, SCALE (I) / NUNITS (I), NHLINS)
  END IF
END DO
CALL POIEND
RETURN
END

C--------------------------------------------

SUBROUTINE PROFI (MAP2, VARNO, FILLIN, RECALC, OFFSET, ANGLE,
& SCALE, NHLINS)

C Plot data profile of the given variable. If FILLIN is true gaps in
C the corrected depth are filled by uncorrected depth. If RECALC is
C true, the residual magnetic anomaly is recalculated using the latest
C version of the IGRF.

EXTERNAL MAP2

CHARACTER * 80 FMTA
CHARACTER * 8 CNAME
INTEGER DAY, MONTH, YEAR, HOUR, MINUTE, VARNO, FMTNO
INTEGER TIME, TIMEU, DAYNO, FULLYR, NHLINS
REAL LAT, LONG, OFFSET, ANGLE, SCALE, VAR (9), TOTMAG
LOGICAL FILLIN, RECALC, ONMAP
COMMON /FMT/ FMTA, FMTNO
COMMON /LIMTST/ XLEFT, XRIGHT, YBASE, YTOP

C Go back to start of file and skip header information

REWIND 2.
DO I = 1, NHLINS
  READ (2, '(I)')
END DO
C Set up profile mode

    CALL PL2PRF (.OFFSET, .90 - ANGLE, SCALE)

C Points within 3 readings are joined up

    CALL LINSEL (3)

10 IF (FMTNO .EQ. 1 .OR. FMTNO .EQ. 4) THEN
    READ (2, FMTA, END = 100) CNAME, DAY, MONTH, YEAR, HOUR,
    & MINUTE, LAT, LONG, (VAR (1), I = 1, 5)
ELSE
    READ (2, FMTA, END = 100) CNAME, YEAR, MONTH, DAY, HOUR,
    & MINUTE, LAT, LONG, (VAR (1), I = 1, 5)
END IF

C Ensure that lines which cross 180 degrees are plotted correctly
IF (LONG .LT. XLEFT) LONG = LONG + 360.

C Convert MGD-77 format to Oracles
IF (FMTNO .EQ. 3) CALL MGD2OR (VAR)

C Check if uncorrected depth should be plotted instead of corrected
IF (FILLIN .AND. VAR (2) .EQ. 0 .AND. VAR (1) .NE. 0) VAR (2) =
  & VAR (1)
IF (RECALC .AND. VARNO .EQ. 4) THEN

C Recalculate RMA
    CALL MARGSYN (FULLYR (YEAR) + DAYNO (DAY, MONTH, YEAR) /
    & 365.0, 1, 0.90 - LAT, LONG, X, Y, Z, TOTMAG)
    IF (VAR (3) .GT. 0) VAR (4) = VAR (3) - TOTMAG
END IF

C Check for gap in data
TIME0 = TIME
TIME = DAYNO (DAY, MONTH, YEAR) * 1440 + HOUR * 60 + MINUTE
IF (TIME - TIME0 .LE. 20 AND TIME - TIME0 .GE. 0) THEN
    CALL POIFR2 (MAP2, LONG, LAT, VAR (VARNO))
ELSE
    CALL POIBRK
END IF
ELSE

C Missing point
    CALL POIMIS
END IF
GO TO 10

100 RETURN

END

---------------------------------------------------------------------

SUBROUTINE TRAKDV (MAP2, VARNO)

C Plots track of ship when the given variable is valid

EXTERNAL MAP2

    CHARACTER * 80 FMTA
    CHARACTER * 8 CNAME
    INTEGER DAY, MONTH, YEAR, HOUR, MINUTE, VARNO, FMTNO
INTEGER TIME, TIME0, DAYNO
REAL LAT, LONG, VAR (5)
LOGICAL ONMAP
COMMON /FMT2, FMTA, FMTNO
COMMON /LIMITS/ XLEFT, XRIGHT, YBASE, YTOP

C Points within 3 readings are joined up
CALL LINSEL (3)
CALL POIBEG

10 IF (FMTNO EQ. 1 OR FMTNO EQ. 4) THEN
    READ (2, FMTA, END = 100) CNAME, DAY, MONTH, YEAR, HOUR,
    & MINUTE, LAT, LONG, (VAR (I), I = 1, 5)
ELSE
    READ (2, FMTA, END = 100) CNAME, YEAR, MONTH, DAY, HOUR,
    & MINUTE, LAT, LONG, (VAR (I), I = 1, 5)
END IF
C Ensure that lines which cross 180 degrees are plotted correctly
IF (LONG .LT. XLEFT) LONG = LONG + 360.
C Convert MGD-77 format to Oracles
IF (FMTNO EQ. 3) CALL MGD2OR (VAR)
IF (ONMAP (LONG, LAT) AND. VAR (VARN0) .NE. 0) THEN
C Check for gap in data
    TIME0 = TIME
    TIME = DAYNO (DAY, MONTH, YEAR) * 1440 + HOUR * 60 + MINUTE
    IF (TIME - TIME0 .LE. 20 AND. TIME - TIME0 GE. 0) THEN
        CALL POIF2A2 (MAP2, LONG, LAT)
    ELSE:
        CALL POIBRK
    END IF
ELSE
    CALL POIBRK
END IF
C Missing point
CALL POIMIS
END IF
GO TO 10

100 CALL POIEND
RETURN
END

C-----------------------------------------------

SUBROUTINE TRAKTA (MAP2, INTVAL)
C Plots the ship's track, annotated with the time at the given interval
C (measured in minutes)

EXTERNAL MAP2

CHARACTER * 80 FMTA
CHARACTER * 8 ANNOT, CNAME
INTEGER DAY, MONTH, YEAR, HOUR, MINUTE, FMTNO, INTVAL
INTEGER DAYNO, TIME, TIME0
REAL LAT, LONG, VAR (5)
LOGICAL ONMAP
COMMON /FMT2, FMTA, FMTNO
COMMON /LIMITS/ XLEFT, XRIGHT, YBASE, YTOP
C   Extended ASCII set
    CALL CHSETS (1)
C   Include spaces in strings
    CALL TXSPAS (0)
C   Set character height
    CALL CHHTS (2.0)
C   Adjacent points are joined up
    CALL LINSEL (1)
    CALL POIBEG
10 IF (FMTNO .EQ. 1 OR FMTNO .EQ. 4) THEN
   READ (2, FMTA, END = 100) CNAME, DAY, MONTH, YEAR, HOUR,
   & MINUTE, LAT, LONG, (VAR (I), I = 1, 5)
   ELSE
   READ (2, FMTA, END = 100) CNAME, YEAR, MONTH, DAY, HOUR,
   & MINUTE, LAT, LONG, (VAR (I), I = 1, 5)
   END IF
C Ensure that lines which cross 180 degrees are plotted correctly
IF (LONG .LT. XLEFT) LONG = LONG + 360.
IF (CNDMPL (LONG, LAT)) THEN
   TIME0 = TIME
   TIME = DAYNO (DAY, MONTH, YEAR) * 1440 + HOUR * 60 + MINUTE
C Check for gap in data
   IF (TIME - TIME0 .LE. 20 .AND. TIME - TIME0 .GT. 0) THEN
      IF (TIME / INTERVAL .NE. TIME0 / INTERVAL) THEN
C Interval has passed
      IF (INTERVAL .GE. 1440) THEN
         Interval is at least a day
         WRITE (ANNOT, '(1X, I3.3, 1X)') DAYNO (DAY,
         & MONTH, YEAR)
         CALL MARLIN
         CALL PNLIS (3, 3)
         CALL POIFC2 (MAP2, LONG, LAT, ANNOT, 5)
         CALL PANOFF
         CALL MARSEL (0)
      ELSE IF (MOD (HOUR, 6) .EQ. 0 .AND. MINUTE .LT
      & INTERVAL) THEN
C Six-hour mark
         WRITE (ANNOT, '(1X, I3.3, "/", I2.2, 1X)')
         & DAYNO (DAY, MONTH, YEAR), HOUR
         CALL MARLIN
         CALL PNLIS (3, 3)
         CALL POIFC2 (MAP2, LONG, LAT, ANNOT, 8)
         CALL PANOFF
         CALL MARSEL (0)
      ELSE
C Ordinary time mark
         WRITE (ANNOT, '(3X, 2I2.2, 1X)') HOUR, MINUTE
         CALL MARLIN
         CALL PNLIS (3, 3)
         CALL POIFC2 (MAP2, LONG, LAT, ANNOT, 8)
         CALL MARSEL (0)
         CALL PANOFF
      END IF
      ELSE
C No time mark
         CALL POIFA2 (MAP2, LONG, LAT)
      END IF
   ELSE
      CALL POIFRK
   END IF
ELSE
   CALL POIFRK
END IF
GO TO 10
100 CALL PANOFF
CALL POIEND
RETURN
END

C-----------------------------------------------

SUBROUTINE TRAKCD (MAP2)
C Plots track of ship and annotates with day and cruise name

EXTERNAL MAP2

CHARACTER * 80 FMTA
CHARACTER * 8 CNAME
CHARACTER * 12 ANNOT
INTEGER DAY, MONTH, YEAR, HOUR, MINUTE, FMTNO, DAYNO, TIME, TIME0
REAL LAT, LONG, LAT0, LONG0, BEARING, VAR (5)
LOGICAL ONMAP, WANT
COMMON /FMT/ FMTA, FMTNO
COMMON /LIMITS/ XLEFT, XRIGHT, YBASE, YTOP
PARAMETER (PI = 3.1416)

C Statement function to covert radians to degrees

        DEG (RAD) = RAD / PI * 180.

C Extended ASCII set
CALL CHSETS (1)
C Set character height
CALL CHHTS (2)
C Adjacent points are joined up
CALL LINSEL (1)
C Don't mark points
CALL MARSEL (0)
CALL POIBEG
WANT = .FALSE.
10 LAT0 = LAT
    LONG0 = LONG
    IF (FMNNO .EQ. 1 .OR. FMNNO .EQ. 4) THEN
        READ (2, FMTA, END = 100) CNAME, DAY, MONTH, YEAR, HOUR,
            & MINUTE, LAT, LONG, (VAR(I), I = 1, 5)
    ELSE
        READ (2, FMTA, END = 100) CNAME, YEAR, MONTH, DAY, HOUR,
            & MINUTE, LAT, LONG, (VAR(I), I = 1, 5)
    END IF

C Ensure that lines which cross 180 degrees are plotted correctly
C IF (LONG .LT. XLEFT) LONG = LONG + 360.
C IF (ONMAP (LONG, LAT)) THEN
C     Check for gap in data
C     TIME0 = TIME
C     TIME = DAYNO (DAY, MONTH, YEAR) * 1440 + HOUR * 60 + MINUTE
C     IF (TIME - TIME0 .LE. 20 .AND. TIME - TIME0 .GE. 0) THEN
C         If new day then want to annotate plot
C         IF (TIME / 1440 .NE. TIME0 / 1440) WANT = .TRUE.
C         IF (WANT .AND. (LAT0 .NE. LAT OR LONG0 .NE. LONG)) THEN
C             Bearing is calculable, so annotate parallel to
C             ship's track
C             BEARING = DEG (ATAN2 (LAT - LAT0, LONG - LONG0))
C             WRITE (ANNOT, '(13, ",", A8)') DAYNO (DAY, MONTH,
C                 & YEAR), CNAME
C             CALL TEXFB2 (MAP2, LONG, LAT, ANNOT, 1, 12, BEARING,
C                 & 1, 1, 1, 1)
C             WANT = .FALSE.
C         ELSE
C             CALL POIFA2 (MAP2, LONG, LAT)
C         END IF
C ELSE
C     CALL POIBRK
C     WANT = .TRUE.
C END IF
    GO TO 10

100 CALL POEND
    RETURN
    END

C---------------------------------------------------------------------------

LOGICAL FUNCTION ONMAP (X, Y)
C Returns true if X and Y are within the limits given

REAL X, Y, XLEFT, XRIGHT, YBASE, YTOP
COMMON/LIMITS/ XLEFT, XRIGHT, YBASE, YTOP

ONMAP = X .GE. XLEFT .AND. X .LE. XRIGHT .AND. Y .GE. YBASE .AND.
& Y .LE. YTOP

END

SUBROUTINE MGD2OR (VAR)
C
C Converts MGD-77 format variables to Oracles format.

REAL VAR (5)
C
C Convert unused records from all 9's to 0.

IF (VAR (1) .GT. 99.) VAR (1) = 0.
DO I = 2, 4
  IF (VAR (I) .GT. 99999.) VAR (I) = 0.
END DO
IF (VAR (5) .GT. 99999.) VAR (5) = 0.

C Convert travel time to uncorrected depth
VAR (1) = VAR (1) * 750.

END

INTEGER FUNCTION DAYNO (DAY, MONTH, YEAR)
C
C Converts a date from day/month/year format to the day number (eg
C 1/1/90 = 1, 31/12/92 = 366)

INTEGER DAY, MONTH, YEAR, NDAYS (12)
DATA NDAYS /0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334/

DAYNO = NDAYS (MONTH) + DAY
C
C Check for leap year
IF (MOD (YEAR, 4) .EQ. 0 .AND. MONTH .GT. 2) DAYNO = DAYNO + 1

END
INTEGER FUNCTION FULLYR (YEAR)
C Converts a two-digit year to a four-digit year. Works for dates in the
C range 1931 to 2030.

   INTEGER YEAR
   IF (YEAR GT 30) THEN
      FULLYR = 1900 + YEAR
   ELSE
      FULLYR = 2000 + YEAR
   END IF
END

SUBROUTINE MAGSYN(DATE,ITYPE,ALT,COLAT,ELONG,X,Y,Z,F)
C
C THIS ROUTINE COMPUTES VALUES OF X,Y,Z AND F FOR A GIVEN
C POSITION AND DATE(BETWEEN 1960.0 AND 1990.0) FROM THE
C SPHERICAL HARMONIC COEFFICIENTS OF THE 1965 INTERNATIONAL
C GEOMAGNETIC REFERENCE FIELD.
C
C IF THE DATE SUPPLIED IS 0, THE ROUTINE RETURNS THE YEAR OF THE LATEST
C DGRF, IN THE DATE PARAMETER.
C
C DATA DATE=REQUIRED DATE IN YEARS AND DECIMALS OF A YEAR, A.D.
C ITYPE=1 IF GEODETIC COORDINATES ARE BEING USED
C ITYPE=2 IF GEOCENTRIC COORDINATES ARE BEING USED
C ALT=HEIGHT ABOVE MEAN SEA LEVEL IN KM (ITYPE=1)
C ALT=RADIAL DISTANCE FROM CENTRE OF EARTH IN KM (ITYPE=2)
C COLAT=COLATTITUDE IN DEGREES (0.0 TO 180.0)
C ELONG=EAST LONGITUDE IN DEGREES (0.0 TO 360.0)
C
C OUTPUT X=NORTH COMPONENT OF MAGNETIC FORCE IN NT
C      Y=EAST
C      Z=VERTICAL
C      F=TOTAL MAGNETIC FORCE IN NT
C N.B. THE COORDINATE SYSTEM FOR X,Y, AND Z IS THE SAME AS THAT
C SPECIFIED BY ITYPE.
C
C DIMENSION GH(720),G1(120),G2(120),G3(120),G4(120),G5(120),
C &
C   G6(120),P(66),Q(66),CL(10),SL(10)
C   EQUIVALENCE(G1,GH(-1)),(G2,GH(121)),(G3,GH(241)),(G4,GH(361)),
C   &
C   (G5,GH(481)),(G6,GH(601))
C
C IGRF 1985 COMPRISUES MAIN-FIELD COEFFICIENTS FOR 1965, 70, 75, 80, 85,
C THE 1965 COEFFICIENTS ARE IN G1, 70 IN G2, 75 IN G3, 80 IN G4,
C 85 IN G5.
C
C THE S.H.COEFFICIENTS ARE SUPPLIED BY THE FOLLOWING DATA STATEMENTS.
C FOR UPDATE OF THE PROGRAM EVERY 5 YEARS WITH NEW VALUES OF THE GEO-
C MAGNETIC REFERENCE FIELD 6 ALTERATIONS OR ADDITIONS MUST BE MADE.
C 1. INCREASE THE DIMENSION STATEMENT ON GH( ) BY 120 AND ADD
C   ANOTHER G#(120)
2. ADD ANOTHER VARIABLE ON THE EQUIVALENCE STATEMENT TO MATCH
3. CHANGE THE LAST IGRF TO DGRF
4. CHANGE THE LAST PGRF TO IGRF
5. ADD THE LATEST PGRF
6. CHANGE LDYEAR IN THE INITIAL SETUP TO THE DATE OF THE LATEST DGRF

C 1965 DGRF
DATA G1/-30334.,-2119.,5776.,-1662.,2997.,-2016.,1594.,114.,
   1297.,-2038.,-404.,1292.,240.,856.,-165.,957.,
   804.,148.,479.,-269.,-390.,13.,252.,-269.,
   -219.,358.,19.,254.,128.,-31.,-128.,-157.,
   -97.,-62.,81.,45.,61.,-11.,8.,100.,
   -228.,68.,4.,-32.,1.,-8.,-111.,-7.,
   75.,-57.,-61.,1.,-27.,13.,-2.,-26.,
   6.,-6.,26.,13.,-23.,1.,-12.,13.,
   5.,7.,-4.,-12.,-14.,9.,0.,-16.,
   8.,4.,-1.,24.,11.,-3.,4.,-17.,
   8.,10.,-22.,2.,15.,-13.,7.,10.,
   -4.,1.,-5.,-1.,10.,5.,10.,1.,
   -4.,-2.,1.,-2.,-3.,2.,2.,1.,
   -5.,2.,-2.,6.,4.,-4.,4.,0.,
   0.,-2.,2.,3.,2.,0.,0.,-6./

C 1970 DGRF
DATA G2/-30220.,-2068.,5737.,-1781.,3000.,-2047.,1611.,25.,
   1287.,-2091.,-366.,1278.,251.,838.,-196.,952.,
   800.,167.,461.,-266.,-395.,26.,234.,-279.,
   -216.,359.,26.,262.,139.,-42.,-139.,-160.,
   -91.,-56.,83.,43.,64.,-12.,15.,100.,
   -212.,72.,2.,-37.,3.,-6.,-112.,1.,
   72.,-57.,-70.,1.,-27.,14.,-4.,-22.,
   8.,-2.,23.,13.,-23.,-2.,-11.,14.,
   6.,7.,-2.,-15.,-13.,6.,-3.,-17.,
   5.,6.,0.,21.,11.,-6.,3.,-16.,
   8.,10.,-21.,2.,16.,-12.,6.,10.,
   -4.,-1.,-5.,0.,10.,3.,11.,1.,
   -2.,-1.,1.,-3.,-3.,1.,2.,1.,
   -5.,3.,-1.,4.,6.,-4.,4.,0.,
   1.,-1.,0.,3.,3.,1.,-1.,-4./
C 1975 DGRF
DATA G3/ -30100.-2013, 5675.-1902, 3010.-2067, 1632.-68,
& 1276.-2144, -333, 1260, 262, 830, -223, 946,
& 791, 191, 438, -265, -405, 39, 216, -288,
& -218, 356, 31, 264, 148, -59, -152, -159,
& -83, -49, 88, 45, 66, -13, 28, 99,
& -198, 75, 1, -41, 6, -4, -111, 11,
& 71, -56, -77, 1, -26, 16, -5, -14,
& 10, 0, 22, 12, -23, -5, -12, 14,
& 6, 6, -1, -16, -12, 4, -8, -19,
& 4, 6, 0, 18, 10, -10, 1, -17,
& 7, 10, -21, 2, 16, -12, 7, 10,
& -4, -1, -5, -1, 10, 4, 11, 1,
& 3, -2, 1, -3, -3, 1, 2, 1,
& -5, 3, -2, 4, 5, -4, 4, -1,
& 1, -1, 0, 3, 3, 1, -1, -5/

C 1980 DGRF
DATA G4/ -29992.-1956, 5604.-1997, 3027.-2129, 1663, -200,
& 1281, -2180, -336, 1251, 271, 833, -252, 938,
& 782, 212, 398, -257, -419, 53, 199, -297,
& -218, 357, 46, 261, 150, -74, -151, -162,
& -28, -48, 92, 48, 66, -15, 42, 93,
& -192, 71, 4, -43, 14, -2, -108, 17,
& 72, -59, -82, 2, -27, 21, -5, -12,
& 16, 1, 18, 11, -23, -2, -10, 18,
& 6, 7, 0, -18, -11, 4, -7, -22,
& 4, 9, 3, 16, 6, -13, -1, -15,
& 5, 10, -21, 1, 16, -12, 9, 9,
& -5, -3, -6, -1, 9, 7, 10, 2,
& -6, -5, 2, -4, -4, 1, 2, 0,
& -5, 3, -2, 6, 5, -4, 3, 0,
& 1, -1, 2, 4, 3, 0, 0, -6/

C 1985 IGRF
DATA G5/ -29877.-1903, 5497.-2073, 3045.-2191, 1691, -309,
& 1300, -2208, -312, 1244, 284, 835, -296, 937,
& 780, 233, 363, -250, -426, 68, 169, -298,
& -215, 356, 47, 253, 148, -94, -155, -161,
& -75, -48, 95, 52, 65, -16, 50, 90,
& -186, 69, 4, -50, 17, -4, -102, 20,
& 75, -61, -82, 2, -26, 24, -1, -6,
& 23, 4, 17, 9, -21, 0, -6, 21,
& 6, 7, 0, -21, -11, 5, -9, -25,
& 2, 11, 4, 12, 4, -16, -6, -10,
& 5, 10, -21, 1, 16, -12, 9, 9,
& -5, -3, -6, -1, 9, 7, 10, 2,
& -6, -5, 2, -4, -4, 1, 2, 0,
& -5, 3, -2, 6, 5, -4, 3, 0,
& 1, -1, 2, 4, 3, 0, 0, -6/
C 1990 Predicted model

DATA G6/
&-29761.,-1853.,5374.5,-2141.5,3062.,-2248.5,1726.4,-10.,
& 1325.5,-2231.,-285.5,1241.5,295.5,835.5,-350.,937.5,
& 77.,252.,324.,-239.,-433.,80.5,135.,-293.5,
& -208.5,356.5,47.5,245.5,147.5,-110.5,-155.5,-160.5,
& -72.,-48.5,95.,59.,63.5,-18.,58.5,84.5,
& -183.,65.,4.,-61.5,21.5,-6.5,-96.,19.5,
& 76.,-64.,-81.,-0.5,-21.,28.,4.5,-1.,
& 32.5,6.,18.5,6.5,-20.,-0.5,-1.5,24.5,
& 6.,7.5,1.5,-26.,-9.,5.5,-10.5,-29.,
& 0.5,12.,4.5,8.,1.5,-16.5,-10.,-3.5,
& 5.,10.,-21.,1.,16.,-12.,9.,9.,
& -5.,-3.,-6.,-1.,9.,7.,10.,2.,
& -6.,-5.,2.,-4.,-4.,1.,2.,0.,
& -5.,3.,-2.,6.,5.,-4.,3.,0.,
& 1.,-1.,2.,4.,3.,0.,0.,-6/.
C
C SET INITIAL VALUES
C
X=0.0
Y=0.0
Z=0.0
LDYEAR=1980
IF(DATE.LE.1960.0.OR.DATE.GE.LDYEAR+10.0) GOTO 9
T=0.2*(DATE-1965.0)
LL=T
ONE=LL
T=T-ONE
LL=120*LL
TC=1.0-T
R=ALT
ONE=COLAT*0.0174533
CT=COS(ONE)
ST=SIN(ONE)
ONE=ELONG*0.0174533
CL(1)=COS(ONE)
SL(1)=SIN(ONE)
CD=1.0
SD=0.0
L=1
M=1
N=0
IF(ITYPE.EQ.2) GOTO 1
C
CONVERSION FROM GEODETIC TO GEOCENTRIC COORDINATES

A2=40680925.
B2=40408585.
ONE=A2*ST*ST
TWO=B2*CT*CT
THREE=ONE+TWO
RHO=SQRT(THREE)
R=SQRT((ALT+(ALT+2.0*RHO)+(A2*ONE+B2*TWO)/THREE)
CD=(ALT+RHO)/R
SD=(A2-B2)/RHO*CT*ST/R
ONE=CT
CT=CT*CD-ST*SD
ST=ST*CD+ONE*SD

1 RATIO=6371.2/R
RR=RATIO*RATIO

COMPUTATION OF SCHMIDT QUASI-NORMAL COEFFICIENTS P AND X(=Q)

P(1)=1.0
P(3)=ST
Q(1)=0.0
Q(3)=CT
DO 8 K=2,66
IF(N.GE.M) GOTO 2
M=0
N=N+1
RR=RR*RATIO
FN=N
GN=N-1
2 FM=M
IF(M.LE.N) GOTO 3
IF(K.EQ.3) GOTO 4
ONE=SQRT(1.0-0.5/FM)
J=K-N-1
P(K)=ONE*ST*P(J)
Q(K)=ONE*(ST*Q(J)+CT*P(J))
CL(M)=CL(M-1)*CL(1)-SL(M-1)*SL(1)
SL(M)=SL(M-1)*CL(1)+CL(M-1)*SL(1)
GOTO 4

3 GM=M*M
ONE=SQRT(FN*FN-GM)
TWO=SQRT(GN*GN-GM)*ONE
THREE=(FN+GN)*ONE
I=K-N
J=I-N+1
P(K)=THREE*CT*P(I)-TWO*P(J)
Q(K)=THREE*(CT*Q(I)-ST*P(I))-TWO*Q(J)
C SYNTHESIS OF X, Y AND Z IN GEOCENTRIC COORDINATES
C
4 LM=LL+L
   ONE=(TC*GH(LM)+T*GH(LM+120))*RR
   IF(ML.EQ.0) GOTO 7
   TWO=(TC*GH(LM+1)+T*GH(LM+121))*RR
   THREE=ONE*CL(M)+TWO*SL(M)
   X=X+THREE*Q(K)
   Z=Z-(FN+1.0)*THREE*P(K)
   IF(ST.EQ.0.0) GOTO 5
   Y=Y+(ONE*SL(M)-TWO*CL(M))*FM*P(K)/ST
   GOTO 6
5  Y=Y+(ONE*SL(M)-TWO*CL(M))*Q(K)*CT
6  L=L+2
   GOTO 8
C
7  X=X+ONE*Q(K)
   Z=Z-(FN+1.0)*ONE*P(K)
   L=L+1
   M=M+1
C
C CONVERSION TO COORDINATE SYSTEM SPECIFIED BY ITYPE
C
   ONE=X
   X=X*CD+Z*SD
   Z=Z*CD-ONE*SD
   F=SQRT(X*X+Y*Y+Z*Z)
   RETURN
C
C ERROR RETURN IF DATE OUT OF BOUNDS, PASSING DATE OF DGRF IF DATE IS 0
C
9  IF (DATE.EQ.0) DATE=LDYEAR
   F=99999.0
   RETURN
C
C
C LIBRARY FUNCTIONS USED BY THIS ROUTINE ARE: SIN,COS, SQRT
C
C SAMPLE RESULTS-
C MAGSYN(1962.0,1., 0.0,0.300,2520..-98.56107..56164..)
C MAGSYN(1969.3,1., 0.0,57.0,195.0,25036.,5682.30100.,39561..)
C MAGSYN(1973.0,2,637.8,245.0,10.0,22108.,-798.40372.,46036.)
C MAGSYN(1977.7,2,9000.0,31.0,359.0,5832.,-1152.,17832.,18797.)
C MAGSYN(1980.5,1., 0.0,30.0,195.0,24923.,5606.,29984.,39390.)
C MAGSYN(1982.0,1., 0.0,40.0,5.0,18997..-1124.,43236.,47608.)
C MAGSYN(1986.5,1., 0.0,55.0,2.0,27850..-1269.,31708.,42221..)
END
3.6 Listing of GEOPlot EXEC

This is the macro which runs the TRACKS MODULE. It is written in REXX, the CMS operating system's macro language.

/* GEOPlot EXEC: macro to run TRACKS MODULE */
'CLRSCHR'
say'
say' GEOPlot Package'
say' ----------------------'
/* Look for arguments */
arg FILNA FILNB FILNC.
/* Test for filename */
if FILNA <> '' then signal GO
/* No filename, so prompt the user */
say'
say' [Before running this program you should have produced an input]
say' ] file using ORACLES
say' ]

GO:
/* Use default filetype and filemode if not specified */
if FILNB='' then FILNB='IN'
if FILNC='' then FILNC='A'
/* Check that file exists, by turning screen output off and attempting*/
/* to list file */
'SET CMSTYPE HT'
'LIST' FILNA FILNB FILNC
TC = RC
'SET CMSTYPE RT'
if TC <> 0 then do
    say '
    say ' Input File: FILNA FILNB FILNC 'does not exist'
    say '
    exit
end
/* Set up GRAFIX library */
'SETUP GRAFIX'

/* Ensure that input file has 120 character records */
/* Set current status values of messages */
cmstype_status=qgetval('CMSTYPE')
emsg_status=qgetval('EMSG')
imsg_status=qgetval('IMSG')
/* Place selected files on the program stack */
'LISTFILES' FILNA FILNB FILNC 'DATE STACK'
/* While there is still something on the stack */
do while queued() > 0
    pull fn fm format .
/* Test for variable length and FIX */
    if format = 'V' then 'FIX' fn fm 'LRECL 120'
end
/* Set messages to original status */
'SET EMSG' emsg_status
'SET IMSG' imsg_status
'SET CMSTYPE' cmstype_status

FLICT Give file logical device number 2 */
'FILEDEF 2 DISK' FILNA FILNB FILNC '(RECFM F LRECL 120'

Y:
  say 'Program loading and executing'
  say ']
/* Run module */
'TRACKS'
'CLRSCRN'
  say ']
  say 'Do you want to run the program again for the same data? (Y or N)]'
  say ']
/* Rerun module if 'Y' input */
parse pull ANSA
if ABBREV(ANSA,'Y') then signal Y
if ABBREV(ANSA,'y') then signal Y
exit
4 REFERENCES

Taunton: Hydrographic Department, Ministry of Defence.

DRAKE, S. & MILES P.R. 1985. IOS geophysical data held on magnetic tape - summary. Bathymetry,
magnetic and gravity data collected during marine geology and geophysics cruises 1973-1984

IAGA DIVISION 1, WORKING GROUP 1 1986. International Geomagnetic Reference Field Revision
1985.

Swindon: NERC Computer Services.

Institute of Oceanographic Sciences Deacon Laboratory Report, No. 254, 83pp.

METCALF, M.; 1985, Effective FORTRAN 77.

Swindon: NERC Computer Services.

IAG Publication Special No. 3.

(Bathymetry, Magnetics and Gravity), KGRD No. 10 (Revised).

5 ACKNOWLEDGEMENTS

We would like to thank NERC Computer Services for their assistance in the writing of this
program. Thanks must also go to several members of the Geology and Geophysics group for their
suggestions and improvements during the writing of this program and documentation.
APPENDIX A: GRAFIX ROUTINES USED BY PROGRAM

This appendix describes the GRAFIX routines used by GEOPLTO, as an aid to understanding
the program [Read & Neale, 1987].

A.1 Beginning and ending the plot

DEVICE (NDEV)  initialises GRAFIX and selects device number NDEV.
GCLOSE          ends the plot.

A.2 Beginning and ending a page

PAGBEG          begins a new page
PAGEND          closes the current page

A.3 Defining the data area

GRAFIX (XLENG, YLENG, ZLENG) initialises a data area of XLENG by YLENG mm (ZLENG is ignored).
GRASET          initialises a data area covering the whole page.
DEFLA2 (XPRI, XSEC, YPRI, YSEC) defines a linear coordinate system (L-space) with XPRI at
               the left edge of the data area, XSEC at the right edge, YPRI at the lower edge and
               YSEC at the upper edge.
EMERC           Mercator projection for ellipsoidal Earth.
EUTM            Universal transverse Mercator projection for ellipsoidal Earth.
SGNOMT          Gnomonic transverse projection for spheroidal Earth.
SLEAC           Lambert equal-area cylindrical projection for spherical Earth.
SORTHT          Orthographic transverse projection for spherical Earth.
MAPCEN (CENLON, CENLAT) specifies the CENTrl LONgitude and CENTrl LATitude for map
               plotting.
MAPSCL (MAP2, SCALE, SLAT, STDSCL) converts the SCALE of a map at the latitude SLAT to
               the value at the equator, STDSCL MAP2 is the projection.
FMTCHR (MAP2, STLON, FILON, STLAT, FILAT, SCALE) defines a coordinate system (F-space)
               based on rectangular projection MAP2 with the STARTing LONgitude, Final
               LONgitude, STARTing LATitude, Final LATitude and SCALE given.
FMTCHI (MAP2, STLON, FILON, STLAT, FILAT, SCALE). Same as FMTCHR but for irregular
               projections.
FMTANN (KTOP, NLEFT, NRIGHT) reserves space for annotation. KTOP = 1 means allow space at
               the top of the data area, KTOP = 0 means don't allow space at the top. NLEFT is
               the number of characters for which space is to be allowed on the left of the data
               area, NRIGHT is the number of characters on the right.

A.4 Drawing axes

ANNAGEO (NFIELD, NDEC) sets axis annotation to latitude and longitude. NFIELD = 1 for degrees,
               2 for degrees and minutes, 3 for degrees, minutes and seconds. NDEC is the
               number of decimal places.
GRSEL (LGRI1, LGRI2, LGRID). LGRI is the level of annotation on the primary axes, LGRI2 is the
               level of annotation on the secondary axes. They take the values 0 to suppress
               the axis, 1 to draw the axes only, 2 to draw the axes and annotate only where
               there is space, 3 to draw the axes and annotate without checking if there is
               space. LGRID = 0 for no grid, 1 for a partial grid using the annotation marks,
               2 for a full grid using the annotation marks and tick marks.
AXIIFA2 (MAP2, NAXIS) draws the specified axis in the F-space defined by the projection MAP2.
               NAXIS = 1 for the X-axis, 2 for the Y-axis.
A.5 Plotting points

LINLA2 (X1, Y1, X2, Y2, NSTYLE) draws a straight line between (X1, Y1) and (X2, Y2) in L-space. NSTYLE is the line style.
RECLIA2 (XL1, YL1, XL2, YL2, NLSTYL, NASTYL) draws a rectangle with opposite corners at (XL1, YL1) and (XL2, YL2) in L-space. NLSTYL is the line style, NASTYL is the area-fill style.
PL2STD selects standard plotting mode.
PL2PRF (OFFSET, ANGLE, SCALE) selects profile plotting mode, where the X value is offset by (AUXVAL * SCALE + OFFSET) * COS (ANGLE) mm and the Y value is offset by (AUXVAL * SCALE + OFFSET) * SIN (ANGLE) mm.
PANLIS (NHOFF, NVOFF) selects point annotation mode, with annotation perpendicular to the line through the data points. NHOFF = 1 for left justified, 2 for centred, 3 for right justified. NVOFF = 1 for aligned at top of characters, 2 for top of capitals, 3 for halfway point, 4 for base.
PANOFF terminates point annotation.
LINSFL (LINE) selects the minimum number of missing points which will cause the line joining the points to be broken. If LINE = 0, no line will be drawn through the points at all.
MARSEL (MARK) selects the mark used to mark points (MARK = 0 gives no marks).
MARLIN selects a line to mark points, plotted perpendicular to the line through the data points.
POIBEG initialises low-level plotting of a set of points (i.e. one at a time).
POIFN terminates low-level plotting.
POIFA2 (MAP2, X, Y, VALUE) plots the point (X, Y) in the F-space defined by the projection MAP2.
POIFC2 (MAP2, X, Y, TEXT, NCHAR) plots the point (X, Y) in the F-space defined by the projection MAP2 and annotates with TEXT. NCHAR is the number of characters.
POIFR2 (MAP2, X, Y, VALUE) plots the point (X, Y) in the F-space defined by the projection MAP2, offset by the values given in PL2PRF (profile plotting).
POIFBK signals a discontinuity in the points being plotted.
POIMIS signals a missing point.

A.6 Plotting text

CHSETS (NSET) selects character set number NSET.
CHHTS (HEIGHT) sets character height in mm.
TXSPAS (NSPA). If NSPA = 1, preceding and trailing blanks will be deleted from strings plotted. If NSPA = 0, the string is plotted as it is.
TXUPRS (IONOFF). If IONOFF = 0, text will be plotted at the angle specified. If IONOFF is 1, the text will be turned the right way up before plotting.
TEXLA2 (XL, YL, STRING, NBEG, NEND, NSTYLE) plots characters NBEG through to NEND of the given string horizontally in L-space, starting at the coordinates (XL, YL) and using text style NSTYLE.
TEXLB2 (XL, YL, STRING, NBEG, NEND, ANGLE, NHOFF, NVOFF, NBLNK, NSTYLE) plots characters NBEG through to NEND of the given string at the given angle in L-space, starting at the coordinates (XL, YL) and using text style NSTYLE. NFOFF = 1 for left justified, 2 for centred and 3 for right justified. NVOFF = 1 for text aligned at the top of the characters, 2 for the top of capitals, 3 for the halfway point and 4 for the base. NBLNK = 1 to remove leading and trailing blanks, 0 to output the whole string.
TEXFB2 (XL, YL, STRING, NBEG, NEND, ANGLE, NHOFF, NVOFF, NBLNK, NSTYLE) is the F-space equivalent of TEXLB2.
A.7 Changing pens

TXPENS (NPEN) changes the pen used for text plotting.

LNPENS (NPEN) changes the pen used for line plotting.

MKPENS (NPEN) changes the pen used for mark plotting.

A.8 Plotting land outlines

WORLDM (MAP2) plots land outlines in medium resolution in the F-space defined by projection

MAP2.

APPENDIX B: OTHER ROUTINES USED BY PROGRAM

This appendix describes the NERC and other non-GRAFIX routines used by GEOPLT, as an
aid to understanding the program.

TOKEN (INSTR, OUTSTR, NTOKEN, ISTAT) tokenises INSTR and puts the result in OUTSTR. The
number of tokens is returned in NTOKEN and the error code in ISTAT.

CMSCMD (CMDSR, ISTAT) executes a CMS command. CMDSTR must be the result of tokenising
the string 'EXEC command-name'. The error code is returned in ISTAT.

NCLOSE (K, ISTAT) closes stream number K and is used instead of the CLOSE statement. The error
code is returned in ISTAT.

DATIM (DATE, TIME) returns the current date and time. DATE is an 8-character string in the form
'12/06/90', TIME is a real number measured in hours.

ERRSET (NERR, N, NACT) changes the behaviour of the program when it encounters an error.

APPENDIX C: REXX COMMANDS AND MACROS

This appendix describes the REXX commands and the presupplied macros used in GEOPLT
EXEC, as well as NPLOT, the macro for sending a file to a plotter [Jones, 1987].

C.1 CMS COMMANDS

CLRSCRN: clears the terminal screen.

SET CMSTYPE HT: turns screen messages off

SET CMSTYPE RT: turns screen messages on again.

LISTF filename filetype filemode: lists the filename(s) specified

FILEDEF m DISK filename filetype filemode (RECFM F LRECL n): sets logical device
number m to the filename specified, with a fixed length of n characters per
line.

filename: runs module 'filename MODULE'

C.2 REXX COMMANDS

SAY 'message': displays a message on the screen

ARG argument1 argument2...: puts the arguments after the macro name into the variables
'argument1', 'argument2' etc.

IF condition THEN action: performs the action if the condition is true

SIGNAL label: transfers control to the line following 'label'.

PULL variable1 variable2...: sets the variables 'variable1', 'variable2' etc from what is typed at
the keyboard.

variable 1 = variable 2: assigns the contents of 'variable2' to 'variable1'.

RC: variable containing the return code from a command.
DO ... commands ... END: marks the beginning and end of a set of commands, eg after a THEN. 
EXIT: end execution of the macro.

ABBREV (string1, string2) is a function which returns true if string1 starts with string2.

C.3 MACROS

SETUP package: sets up the library required for 'package'.

FIX filename filetype filemode (LRECL n: converts the variable line length file specified to a fixed line length n.

NPLOT: prompts the user for information and then sends the file requested to the plotter requested.