Reference guide for "contours": a contour plotting program for the Hewlett Packard HP 9836 / Guide de référence pour des "contours" : un programme qui trace des contours pour le Hewlett Packard HP 9836

McCrea, B.

Laboratory Memorandum (National Research Council Canada. Division of Mechanical Engineering. Engine Laboratory); no. LM-ENG-003, 1986-07

CONTROLLED
UNCLASSIFIED

CONTRÔLÉ
NON CLASSIFIÉ

Laboratory
Memorandum

Mémoire de
laboratoire

1986/07

LM-ENG-003

REFERENCE GUIDE FOR "CONTOURS": A CONTOUR PLOTTING PROGRAM FOR THE HEWLETT PACKARD HP 9836

B. McCrea

Division of
Mechanical Engineering

Division de
génie mécanique

National Research Council Canada
Conseil national de recherches Canada

Canada

DIVISION OF MECHANICAL ENGINEERING PUBLICATIONS

DM **(Division of Mechanical Engineering Report)**
Scientific and technical information considered important, complete and a lasting contribution to existing knowledge.

TR **(Technical Report)**
Information less broad in scope but a substantial contribution to existing knowledge.

CTR **(Controlled/Classified Technical Report)**
A Technical Report with controlled distribution for national security, proprietary or other reasons.

LM **(Laboratory Memorandum)**
Preliminary or exploratory information with controlled distribution.

CAT **(Calibration Analysis and Test Report)**
Information on minor laboratory projects or services.

PUBLICATIONS DE LA DIVISION DE GÉNIE MÉCANIQUE

DM **(Rapport de la Division de génie mécanique)**
Informations scientifiques et techniques jugées importantes, complètes et susceptibles de contribuer de façon durable à l'avancement des connaissances courantes.

TR **(Rapport technique)**
Informations de moindre importance, mais pouvant contribuer substantiellement à l'avancement des connaissances actuelles.

CTR **(Rapport technique à diffusion contrôlée/classifiée)**
Rapport technique à diffusion contrôlée pour des raisons de sécurité nationale, de propriété intellectuelle et autres.

LM **(Mémoire de laboratoire)**
Informations préliminaires ou de nature exploratoire à diffusion contrôlée.

CAT **(Rapport d'étalonnage d'analyse et d'essai)**
Informations sur de petits projets ou des services de laboratoire.

REFERENCE GUIDE FOR "CONTOURS": A CONTOUR PLOTTING PROGRAM FOR THE HEWLETT PACKARD HP 9836

GUIDE DE RÉFÉRENCE POUR DES "CONTOURS": UN PROGRAMME QUI TRACE DES CONTOURS POUR LE HEWLETT PACKARD HP 9836

B. McCrea

Laboratory Memorandum — Mémoire de laboratoire

1986/07 — LM-ENG-003

Engine Laboratory
Laboratoire des Engins

Copies of this report may be obtained on request from:

Engine Laboratory, Bldg. M-7
National Research Council
Montreal Road
Ottawa, Ontario
K1A OR6

SUMMARY

This report provides a user's description of a programme for plotting of contour lines from data sets ranging in size from a 3 X 3 array of 9 points to a maximum or 2000 points for use with an HP 9836 P.C. with plotting capability. The contours may be plotted in either cartesian or polar co-ordinates.

The basic 13 steps of operating procedure of this method are stated on page 4 of this report. Using no more than this information a user with no prior knowledge of computers can proceed by responding merely to the choices displayed on the computer screen.

The purpose of this report is to furnish a somewhat more detailed explanation with typical video display illustrations in case errors are made by the operator while interacting with the machine, and to suggest possible corrective actions.

For the more experienced programmer a complete listing of the computer code has been included in the text of this report.

TABLE OF CONTENTS

NOMENCLATURE

CONTOUR POINT: Interpolated point which lies on a contour line.

CONTOUR LINE: Line having a uniform height formed by the joining of contour points.

CROSS PLOT: A two dimensional curve passing through the set of data along a grid line.

CUBIC SPLINE: A curve which passes through all the data points and is defined by cubic equations within each grid interval. The cubic spline used in this program has a second derivative boundary condition set to zero, called a natural spline.

DAMPING FACTOR: Positive numerical value controlling the shape of the fitted rational spline. A small value produces a smooth curve with large extraneous peaks and valleys while a very large value produces an angular curve with less pronounced peaks and valleys.

GRID, GRID LAYOUT: Set of data specifying the number of grid lines and their co-ordinates.

GRID CO-ORDINATE, LOCATION, POSITION : Numerical value assigned by the user to describe the arrangement of grid lines within the grid. These co-ordinates have the value of the running dimension to the the grid lines from a common datum.

GRID INTERVAL: Space between any two consecutive parallel grid lines.

GRID LINE: Line joining a set of longitudinal or transverse grid points.

GRID POINT: Point at which a data item is specified.

LONGITUDINAL: Pertaining to the x-wise direction on cartesian contour maps. In polar plots, the longitudinal direction pertains to the outward radial direction.

NORMALIZING FACTOR: Divisor of the entire data set enabling non-dimensional analysis of the data.

RATIONAL SPLINE: A curve which passes through all the data points and is defined by a set of rational equations within each grid interval. The rational spline used in this program has first derivative boundary conditions.

TRANSVERSE: Pertains to the x-wise direction on cartesian contour maps In polar plots, the transverse direction pertains to the circumferential direction with increasing in value in the counter-clockwise direction.

VERTICAL INTERVAL: Numerical value of the height between consecutive contour lines.

N.B. Figure 1 displays schematically some of the above terms.

CONTOUR PLOTTING PROGRAM

1.0 PROGRAM DESCRIPTION

This program will accept data sets ranging in size from a 3x3 grid up to a 2000 point array and will plot report-quality contour maps and three-dimensional views of the data. The program is intended primarily for relatively small data sets due to the elaborate curve-fitting scheme required to create smooth contour lines, since large data sets may impose on memory and computing time limitations. Within the program, there exists several user-friendly menus from which one may select any of the listed items. The interaction between these menus and the user to obtain the desired outputs is shown in figure 2.

1.1 PROGRAM'S ALGORITHM

To produce a contour map, the program deals with one contour level at a time. Initially, it fits a piecewise rational spline along each of the transverse grid lines to the given data. Heights on all of these cross plots are read at a particular value of the transverse co-ordinate. By fitting a rational spline to these heights, a longitudinal cross plot is obtained. In this manner, many longitudinal cross plots can be generated by slowly increasing the value of the transverse co-ordinate. Along these longitudinal cross plots, the locations of the contour points for a particular contour level are computed and these points are then ordered into temporary line segments (all traversing from left to right, counter-clockwise for polar plots). These segments are then selectively joined together into full contour lines, which are then further smoothed by a parametric cubic spline passed in the horizontal plane of the contour map and then plotted. The above search for contour points is repeated for the next contour level and until all the chosen levels are plotted. The three dimensional plot is merely the drawing of the described dense set of longitudinal cross plots and of the front and rear transverse cross plots. A mean height of the data set is computed by dividing the total volume defined by the data set by its base area. The total volume is obtained by summing the area under each of the longitudinal cross plots (which are obtained by means of an exact integral of the rational spline equations) using the trapezoidal rule approximation.

1.2 AVAILABLE OUTPUTS

This program will accept data in three basic shapes: cartesian, polar annulus, or polar sector (fig.3). Data in any of these grid shapes may be plotted as a contour map on an HP plotter or on the 9836's video display. Also, one may wish to plot the data in a three dimensional representation, but this capability exists only for data with cartesian grids and is available only as a screen plot. However, any screen plots may be dumped onto an HP2934A printer (or other suitable printer) using the 'DUMP GRAPHICS' key on the 9836.

For contour plots, the user has several options from which to choose. One may wish to have printouts of the data set and of the list of the contour points. One may choose to label the data point locations using a '+', using the actual data values at the points, or not to specify the data point locations at all. The grid may also be drawn with a frame along its bounds. For cartesian grids, it is also possible to label the grid axes with the grid co-ordinates. A plot specifier denoting the data file, the date, the time, the damping and normalizing factors may also be printed at the bottom left corner of the grid. The above options are available for both screen and plotter contour maps. Screen plots are drawn as large as possible while maintaining the scale between the specified longitudinal and transverse (radial and circumferential for polar plots) grid co-ordinates, but, plotter contour maps may be drawn to a scale other than the one specified and may also be drawn to any size and location on the plotter's surface. Furthermore, the plotter contour maps may use different colour pens and have any desired title/label in any location and with any character size the user desires. Additional rectangular frames outside the grid area are also available to be drawn on the plotter. Various plots and options are shown in figures 4-14 and 16-22.

The integrated average height over the plotted surface is also computed and printed on the screen. The damping factor controlling the shape of the rational spline cross plots can also be changed, making the plots smoother but with large extraneous peaks and valleys or more anugular with less pronounced peaks and valleys. By using different damping factors, the average

height will of course change. For polar plots, the computed mean height is related to the scale of the plot and thus, if a polar plot has been distorted from its specified grid co-ordinates, the mean height will be slightly different.

1.3 APPLICATIONS

The contour maps produced by this program present tabular data in a manner enabling the user to analyze quickly and accurately the phenomenon causing the particular distributions denoted in the contour map. Data taken from events similar in nature but different in magnitude may be more easily compared by the plotting of normalized data (non-dimensionalized). Also, the computed integrated average height of the data set gives a single numerical value to the data set, which is not only area-weighted, but which is also weighted in relation to the particular distribution of contour lines within the data set. For example, this average height may be used as a measuring instrument to determine the net flow through a duct given a set of anemometer readings placed in a cross section of the duct. Hence, the integrated height may be used to compare data sets representing the same measured quantity but which have different contour map distributions. However, the grid is not restricted to representing an actual area (as above), but may merely be two independent quantities for which a third quantity has been determined, and thus, a contour map is an excellent method of displaying the dependent relation of a given variable on two other quantities.

Although, the three dimensional plot is quantitavely less informative than a contour map, it may nonetheless be favored over a contour map since it emphasizes the highlights in the overall distribution of the data set. In any case, a contour map, a three dimensional representaton, or the integated average height may be used as an on-line analysis of the acquired data during experimental test measurements.

2.0 OPERATING PROCEDURES

1- Insert BASIC 2.0 operating disc in the right disc drive (#0)

2- Turn on the 9836 (on switch is at the lower front right corner of the keyboard, below and to the right of the EXECUTE key)
Wait for prompt: BASIC Ready 2.0

3- Remove this disc and insert the BASIC 2.1 disc in drive #0

4- Type : LOAD BIN "AP2_1" & press EXECUTE
Wait for prompt: BASIC Ext. AP2.1

5- Type : LOAD BIN "GRAPH2_1" & press EXECUTE
Wait for prompt: BASIC Ext. GRAPH2.1

7- Now remove this disc & insert the CONTOUR PLOTTING disc in drive #0

8- Type : LOAD "CONTOURS" & press EXECUTE

9- Either keep the CONTOUR PLOTTING disc in drive #0 or replace it by a disc containing your personal 'GRIDS' file.

10- Insert a data disc in drive #1

11- Turn on plotter (address 705) & printer (address 701).

12- Now press RUN to invoke a program run.

13- If an error occurs, keeping the user from continuing with the program run, one should press RESET and then press RUN to invoke a new run.

The program offers several options presented in menus consisting of either special function keys (SFK) or of a textual list of options. To select the appropriate option simply press the corresponding SFK (k0-k9 at the upper left of the keyboard) or type the appropriate response (numeric and/or alphabatic) and then this response is entered by pressing the CONTINUE key. The CONTINUE key must be pressed to invoke action within the program once it has requested a response from the user unless it wishes a SFK to be pressed, in which case, program run will resume immediately. For information concerning specific options within the program, please refer to the appropriate sections within this manual.

3.0 DESCRIPTION OF AVAILABLE OPTIONS AND INPUTS

The order in which the input and option items are described follow the natural flow through the program from one step to another to obtain the various outputs. Figures 23 to 51 show the actual screen displays obtained within a program run presenting all the important inputs, options and informational displays available to the user and these figures also follow the natural flow through a program run (fig.2).

3.1 TIME AND DATE ENTRY

Since the time and date is printed with some of the outputs, it is necessary that the computer be given the current time and date since the computer's real time clock is turned off whenever power to the HP9836 is broken. Thus, the first time the 'CONTOURS' program is run after the HP9836 has been switched on again, the computer will no longer have the correct time and date. Once these inputs have been made (fig.24), the program offers the choice of three special function keys. One key enables the user to correct his time input, another to change the date input, while the third key exits the user from the TIME/DATE verification routine so that he can continue with the rest of the program. If a program run has been initiated with the computer having in its memory an appropriate time and date, the computer will display the current time and date and will offer the user the choice of the above described SFK's, enabling him to pass quickly to the rest of the program.

3.2 MAIN KEY MENU

This is the main menu for the program and offers the user seven basic options (fig.26). After the task specified by the chosen key has been performed, the user is returned back to this key selection.

k0 - DATA INPUT

This key invokes access to the data management routine where data can be entered, edited, retrieved from or stored onto a floppy disc.

k1 - USE SAME DATA

This key is similar to k0 since they both access the data management routine. However, use of this key enables the user to retain the data previously used for a plot and, hence, does not ask the user for more data. Rather, the routine proceeds directly to the DATA EDITING MENU. If this is the first key to be struck after a new program run has been invoked then, use of this key is exactly the same as k0.

k2 - PLOT ON SCREEN

Use of this key selects the drawing on the video display terminal of a contour map of the data in memory. A plot on the screen will take nearly as much time as one on the plotter, but, since only some of the PLOT MENU #1 options are available for a screen display, a screen plot is generally quicker to invoke than a hardcopy plot.

k3 - PLOT ON PAPER

Use of this key directs the contour plot to be drawn on the HP plotter. For hardcopy plots, all the options listed in both PLOT MENU #1 and PLOT MENU #2 are available for plotter drawings.

k4 - THREE-D PLOT

Selection of this key offers the user a three dimensional view of the data drawn on the video display terminal available only for cartesian grids. No options are available to alter the drawn 3-D image.

k5 - AUTOMATIC MODE

This key allows the plots of several data sets to be plotted onto a suitable HP printer (e.g. HP2934) automatically. Once this option is selected, the user can then select any or all of the data sets on one disc to be plotted. All data sets will be plotted in the same representation, (i.e. contour map or 3-D view). Selection of the representation and the desired data files is made within the AUTOMATIC MODE MENU. Furthermore, if a contour map is selected, the user will be offered the options given in a reduced version of PLOT MENU #1. This automatic process retrieves the data from disc in the specified order, produces the desired plot on the screen, dumps the screen raster display onto a new page on the printer and continues in this fashion until all the data sets have been plotted.

K6 - EXIT PROGRAM

This key stops the program run, prints a display to the fact that the program run has been terminated, and resets the mass storage device to be the right hand disc drive (#0). If one quits a program run without pressing this key, in general, the mass storage device will be set to be the left hand disc drive (#1).

3.3 DATA MANAGEMENT

If this routine has been accessed by 'k0 - DATA INPUT' then the user is prompted to give a suitable response to the manner in which data is to be entered into the computer (fig.27). This allows data to be retrieved from a data file on a floppy disc or to be manually entered at the keyboard.

3.3.1 MANUAL DATA ENTRY

3.3.1.1 GRID MENU

When manually entering several data sets, often the parameters defining the grid are the same. Thus, a file, called 'GRIDS', on the 'CONTOUR PLOTTING' disc aids in the process of keyboard data entry. This data file contains the grid specifications (i.e. grid size, type, name and dimensions) for the grids previously stored. Hence, the GRID MENU (fig.28) is basically a list of the grids which may be retrieved to speed the manual data entry process. To select the appropriate existing grid, the number associated with that item in the list is entered. One may also edit the 'GRIDS' file by purging any of the existing grids. Once item 'B' from the GRID MENU is selected, the computer prompts the user to enter the desired item to be purged from the 'GRIDS' file and once purging of the selected grid has been completed, the updated version of the GRID MENU is offered to the user. When an item is purged, the last grid in the list will take its place in the GRID MENU. Fifty grids is the maximum number of grids allowed in the 'GRIDS' file. If several users are using the same program disc, they may initially wish to make copies of the 'GRIDS' file so that each may have their own personal set of available grids in the future. Likewise, one may wish to have several 'GRIDS' files but they must be stored on separate discs since

the program requires access to a file having 'GRIDS' as its file name and each file on a floppy disc must have a distinct name.

3.3.1.2 ENTERING A NEW GRID

To enter a new grid, item 'A' in the GRID MENU must be selected. The user will then be prompted to select a grid shape and enter the size and co-ordinates of the grid. The grid shape must be selected (fig.29) and can be one three basic shapes (fig.3): CARTESIAN, POLAR ANNULUS, or POLAR SECTOR (less than 360 degrees).

Once a shape has been selected, the user will be prompted to enter the number of transverse (circumferential for a polar plot) co-ordinates and then the number of longitudinal (radial for a polar plot) co-ordinates defining the size of the grid. Next, the grid co-ordinates are entered and which have the values of the running dimension to the grid lines from a common datum. The transverse/circumferential co-ordinates are dealt with first and, of course, these co-ordinates must increase monotonically in value. The program will prompt the user until a complete set of monotonically increasing grid co-ordinates has been entered. The same requirements and procedures must be met in entering the co-ordinates for the longitudinal/radial grid positions. The above grid entries are shown in figure 30. For polar grids, the circumferential grid co-ordinates (angular positions) have a dimension of degrees. A angle of zero defines the position of a radial line moving rightward (parallel to the 'x' axis) from the center of a circular arc to the arc's radius. From this datum of angle zero, the angular co-ordinates are defined in a counter-clockwise manner with increasing angular value.

These grid co-ordinates define the scale of one base axes to the other. Since all screen plots, draw the grid to the specified scale, it is wise to enter the longitudinal and transverse grid co-ordinates with values with the same order of magnitude for cartesian grids. Also, for three dimensional plots, the base, and thus, the entire plot will be drawn to the specified grid scale. For polar grid plots, it is unwise to have the ratio of the co-ordinates of the inner over the outer radius close to a value of 1 since this will define a very thin annulus in which the contour map is plotted

making the distribution difficult to visualize.

Once the grid has been entered, the user will be prompted whether he wishes to store the grid (fig.31). To store, the user simply types 'YES' and presses CONTINUE, whereas a negative response may be achieved by simply pressing CONTINUE. To complete the process of storing the grid, the user is required to enter a grid name whose length must not exceed 40 characters. This name is used to identify the specific grid within the GRID MENU. If the new name is identical to the name of an existing grid, then the program will prompt the user by telling him that the name already exists. At this point, the user has the option of updating the grid specifications under the current grid name (i.e. deleting the existing grid with the conflicting grid name and storing the newly entered grid under the same name). The user also has the option of aborting storage of the grid or of entering a new name which will again be checked for a conflict with existing grids.

3.3.1.3 ENTERING A DATA SET

To facilitate the task of entering data, a tabular representation of the data is displayed on the screen (fig.32). Since a large data set may not be suitably displayed on the screen, the data set is broken into several blocks, each of which is at most 8x8 in size. These blocks are displayed as part of the data set by a graphical representation in the upper right corner of the screen showing a rectangle representing the data set divided into several blocks and in which the current block has been shaded in.

However, before data entry can commence, a special function key must be pressed (fig.32). The user can proceed with the data entry (k2), return to the GRID MENU (k3), or proceed to the DATA EDITING MENU (k4). The purpose of keys 3&4 is to allow the user to view the existing grids within the GRID MENU without needing to enter all the required data.

If 'k2' is pressed, then the data items are to be entered, but this process may be interrupted at any time by typing 'QUIT' rather than entering a data value.

To enter a data set correctly, the data values must be entered at the appropriate locations within the grid. Thus, a rectangular box representing

the location of the requested data item moves from one location to the next until all data have been entered. This box commences at the upper left of the data block (fig.33). It then moves down the column as the data items are entered. After the data item for the bottom position within the column is entered, a beep is sounded to alert the user of the fact that the box has travelled to the top of the next column. This beep is important for large data sets, since not an entire column of data is entered consecutively but rather only the portion spanning the current data block. Thus, the beep alows the user to enter data rapidly without needing to monitor the box position within the data block. Thus, to fill a data block, the data items are entered filling the columns from top to bottom, moving from left to right (fig.34). This procedure is also the mechanism for choosing the data blocks to fill the data set (fig.35&36). The order in which the data items are entered is also displayed in table 5.

3.3.2 DATA EDITING AND MANIPULATION

3.3.2.1 DATA EDITING MENU

Once the data has been manually entered, the user is prompted by the DATA EDITING MENU (fig.37) offering the six options described below. This menu is also immediately selected if the key 'k1 - USE SAME DATA' from the MAIN KEY MENU was pressed (providing data had been previouly loaded into memory) or if data has been retrieved from a disc file.

1- VERIFY/CHANGE DATA

The same tabular method of displaying the data as it was entered allows the user to verify the data set. However, some additional features allow the user to quickly scan and/or edit the data (fig.38). These features are part of the DATA EDITING KEY MENU and will be described in that section.

2- STORE DATA

It is good practice to store data on a floppy disc before attempting to produce a plot. If for some reason an error is generated and locks the user out of continuing on with the current program run, the user's only way of proceeding with the program is to invoke another program run by pressing the 'RUN' key, which will of course reset all program variables, and hence,

delete the data set. To store a data set the user is asked to enter a suitable file name (fig.39). A file name can be from one to ten characters in length and be composed of any of the 26 alphabetic letters, 10 numeric digits and the underscore '_' symbol. Blanks are disregarded in the creation of a file name and any other symbol produces an illegal file name. Any arrangement of the 37 valid file name symbols is legal.

Once a suitable file name has been entered, the data will be stored under that name only if no other file on the current disc has the same name. If there is a conflicting file name, the program informs the user of this fact and offers him the possibility of updating the specified file with the new data, aborting data storage, or of creating a new file name. Once a file has been created, the only method of erasing the file from the floppy disc (other than updating it with a new data set) is to purge the file by typing 'PURGE "file name"' outside the context of a program run.

3- NORMALIZE DATA

After selecting this option, the user is prompted to enter the desired factor by which the data set will be normalized (fig.40). The default value is the data set's arithmetic mean. Alternatively, one may invoke plotting of a contour map to obtain the integrated average, press the 'INTERUPT' special function key to return to the beginning of the program, and by re-using the same data, choose to normalize this data by the integrated average. It is therefore suggested that the same damping factor be used for the two runs. Furthermore, a normalizing factor of -1.0 will invert the data, giving the three dimensional view a different perspective (fig.47&48). Once this option is selected, the only way in which to plot the non-normalized data is to return to the begining of the data management routine (option '5- GO BACK TO START OF DATA ENTRY' in the DATA EDITING MENU or key 'k0- INPUT DATA' in the MAIN KEY MENU).

4- VERIFY/CHANGE NORMALIZED DATA

This option is similar to option #1 except that the data to be verified is a normalized data set. If the normalized data set shows an erroneous data point, it can edited just like the original (non-normalized) data set. The user after noting the discrepancy among the normalized data, may wish to re-edit the non-normalized data set by the further selection of item #1.

5- GO BACK TO START OF DATA ENTRY

This feature allows the user to abort continuation of the program run with the current data set and allows him to retrieve a data set from disc or to enter a new one at the keyboard. This option, under certain circumstances, may also be used in quicker data editing, as will be explained later (refer to ADDITIONAL NOTES FOR EDITING DATA).

6- LEAVE DATA EDITING MENU

This option allows the user to leave the data management routine, returning him to the MAIN KEY MENU.

3.3.2.2 DATA EDITING KEY MENU

This menu (fig.38) consists of the eight special function keys described below enabling the user to quickly edit the data. Also, use of the scrolling wheel (upper left of the keyboard) aids in this process.

As in manual data entry, a rectangular box indicates the pointer position within the data set, but in this case, the box position is controlled by the scrolling wheel. By rotating the scrolling wheel clockwise the box will travel down each of the columns (the way data is manually entered) or by giving the wheel a counter-clockwise rotation, the box moves in the reverse direction. Please note that once a SFK k0, k1 or k2 (described below) has been pressed, the scrolling wheel and the SFK's are disabled until the action specified by the particular SFK has been completed.

k0- CHANGE ITEM

To change the value of a data item, the viewing box must first be moved until it is positioned at the erroneous data point. Then 'k0' is pressed and the program will display the old value above the table and prompt the user to enter the correct value. Once the new value is entered, this value overwrites the old data item in the specified location, the wheel re-assumes control of the box, and the special function keys are re-enabled.

k1- ADD AN ITEM

To add a data value between two consecutive data items, the box is first positioned at the location where the new value should be stored. Then this SFK is pressed, and the computer prompts the user to enter the new data

value. Again, once the new value is entered, the wheel re-assumes control of the box, and one can select another editing key. Please note that the addition of a new data value only affects the items within the current data block, pushing each item further down the path and with the last item (lower right corner) being deleted from the data set. All data values in any other data block remain unaffected by this operation. Furthermore, the box position is reset to the upper right of the data block.

k2- DELETE AN ITEM

To delete a data value, the box is first positioned at the item to be deleted. This SFK is then pressed, and automatically the item is deleted and all items following its position are moved forward. Again, since this operation does not affect data in the other data blocks, the last data item (lower right) obtains a value of zero. Furthermore, the box position is reset to the upper right of the data block.

k3- QUIT EDITING

In order to exit from this data verifiction/edit routine, the user must press this SFK. The user is then returned to the DATA EDITING MENU wherein he may wish to return back to this routine.

VIEWING KEYS:
k5- VIEW UP k6- VIEW DOWN k7- VIEW LEFT k8- VIEW RIGHT

These four special function keys allow the user to scan the various data blocks. By selecting any of these keys, the old data block is erased and replaced by the one specified by the SFK that was pressed. The graphical representation of the data set and blocks is reset to show the position within the data set of the new data block. Also, one need not wait until the data block has been entirely printed on the screen in order to press another of these four scanning keys. In this manner, the user is able to quickly move from one section of the data set to another. Also, the box position is reset to the upper right of the newly viewed data block. If a data set is only one block wide in the transverse direction, the viewing keys k7 & k8 will be omitted. Likewise, if a data set is only one block high in the longitudinal direction keys k5 & k6 will be omitted. Furthermore, if a data set is entirely defined by a single data block, the graphical representation of the data set (upper right corner of figure 38) will also be omitted.

3.3.3 ADDITIONAL NOTES FOR EDITING DATA

As described, editing the data values is quick and simple, whereas, editing existing grid specifications has not been discussed and is a more complicated task and requires two steps. Firstly, one should access the keyboard data entry route, then select the appropriate grid to be corrected from the GRID MENU. Once this is accomplished, the program will prompt the user with the special function keys shown in figure 32, one of which allow him to return to the GRID MENU where he can then select item 'A - ENTER A NEW GRID' enabling the entry of a new grid. Since the computer retains the grid specifications obtained in the first step, and asks for all grid inputs for the entry of a new grid, one is able to change any of the grid specifiers (fig.29&30) one wishes and the previous values can be recalled by simply pressing 'CONTINUE'.

In a similar way, to edit the grid specifications of a disc file data set, the user selects the erroneous data file at the start of the data management routine (fig.27). This operation simply loads the data values into the program array. Then, by again returning back to the start of the data entry routine from the DATA EDITING MENU, the user opts for the keyboard data entry mode, selects an appropriate grid type and can quickly attain the DATA EDITING MENU by selecting key 'k4' shown in figure 32. The data items having been stored in the program's memory will be placed into the new data set, but will only occupy the appropriate data locations if and only if the old grid and the new one have the same number of transverse and longitudinal grid lines.

3.4 DRAWING A CONTOUR MAP

Once data has been entered, one may proceed to the drawing a contour map on the plotter or on the screen. To draw a contour map on the screen, key 'k2 - PLOT ON SCREEN' from the MAIN KEY MENU must be pressed. To draw a contour map on the plotter, key 'k3 - PLOT ON PAPER' should be selected. In either case, the user is offered the options within the PLOT MENU #1.

The contour maps on the screen will all be drawn to the scale specified by the grid co-ordinates. Cartesian maps are always drawn with its axes

along the 'x' and 'y' axes and the polar annular plots are drawn with an angle of zero along the positive 'x' axis and an angle of 90 degrees along the positive 'y' axis. Furthermore, to place the polar sector map on the screen, the sector is rotated so that it is symmetrical to the the 'y' axis and with the sector's outer radius nearer the top of the screen than its inner radius. On the plotter, however, the polar sector grids may be rotated as one wishes, obtaining any desired orientation.

3.4.1 PLOT MENU #1

This menu offers the user 17 different options which are described below along with their various default values. Items 1-7 & 9 are toggle switches which can be set to either a 'YES' or 'NO' response.

1&2 : 1- DATA VALUES PLOTTED default: NO
2- GRID POINTS PLOTTED default: YES

These two options are interactive since they cannot both have the value 'YES' at the same time. Option #1 prints the values of the plotted data set at their corresponding grid points (fig.14). Option #2 marks the grid point locations by printing a '+' at each of the grid points (fig.11).

3- DATA VALUES PRINTED default: NO

This option prints, on the hardcopy printer, a tabular list of the data set together with the specified grid co-ordinates, grid shape, file name and the current time and date. All the data sets presented in tables 1 to 6 have been printed using this option. The data set's computed volume, base area and mean height are also printed when this option is selected. These items will be printed below the data set list and after completion of the cross plot curve fitting.

4- CONTOUR POINTS PRINTED default: NO

This option should only be selected when it is critical to have a detailed list of all of the calculated contour points involved in creating the contour lines. If the contour map does not seem to be realistic, then the contour map may be drawn again; but, with this option selected, the user will be in a better position to decide what has gone wrong if indeed the contour map is in error. Table 7 shows the listing of the contour points obtained in the plotting of figure 18.

5- GRID FRAME DRAWN default: YES

This option draws a solid frame around the grid along the outermost grid lines and is available for all three grid shapes (fig.3).

6- VALUE OF AXES LABELLED default: NO

This option is only available for cartesian grids and denotes the grid lines by specifying them with their values (fig.12&13).

7- PLOT SPECIFIER default: YES

The plot specifier consists of the file name, time and date of the plot and, if selected, is printed at the bottom left corner of the grid frame (fig.11).

8- DAMPING FACTOR default: 2.0

All plots use rational spline (Spath 1974) curve-fitted cross plots and the damping factor required by the rational spline may be changed to suit the user's needs. The default damping factor is 2.0, whereas a value of zero generates a cubic spline, while a value of infinity degenerates the curve into a linearly interpolated one. The effect of the value of the damping factor upon a cross plot is shown in figure 15 while figures 16 to 22 show the factor's effect on the contour map of a typical data set. As shown by the specified figures, the cross plots and hence, contour maps and 3-D views will change much more rapidly under a small change of value at low damping factors than under the same change but at high values. Since extreme values of the damping factor may cause round-off errors within the program, the value of the damping factor is restricted to the range between 0.1 and 100. Furthermore, since the damping factor changes the shape of the fitted cross plots, the integrated volume and average height will also change.

9- POLAR PLOT TO SCALE default: YES

All plots on the screen are to the scale specified by the grid co-ordinates but polar plots on the hardcopy plotter may be selected to be scale or not. Since cartesian grids have their scale determined by the digitizing process, to simplify the digitizing process for polar plots, this parameter (plot to scale) must be known. In general the plots should be left to scale, but if the grid co-ordinates define a polar sector or annulus which is too narrow (radially or circumferentially) to give an adequate view

of the contour map, then the user may select to have the value of this option be 'NO'. If the polar plot is chosen not to scale, the digitizing process requires three operations rather than two. Figure 6 shows a typical polar annular contour plot drawn to scale while the same data is plotted with an exagerated decrease in its inner radius in figure 7. When a polar (annular or sector) grid is specified to be drawn not to scale, one may increase or decrease the inner radius with respect to the outer radius (fig.8). Furthermore, the average height determined by the integration of the fitted curves may be different from the result obtained with a scaled plot.

10-CONTOUR LEVELS

This option allows the selection of up to 40 contour levels having the same vertical interval between them. The default contour levels are calculated so that approximately 10 levels will cover the range of values within the data set. This is achieved by setting the vertical interval between the contour levels to have a mantissa of either 1.0, 2.0, 2.5 or 5.0 multiplied by a factor of ten raised to an exponent and each of the default contour levels will be multiples of this vertical interval. One may, however, specify any desired positive vertical interval and contour level range by specifying the lower and upper levels. If one had chosen this option, then decided to keep the default levels, this may be achieved by the entry of the number '0' for the vertical interval. To select a single contour level, the vertical interval must be any positive number and with the lower and upper levels having the value of the desired level to be plotted.

11-GRID & TITLE COLOUR default: pen #1

This option selects the desired colour for all the options that are available to enhance the grid before plotting is started. Thus the grid points ('+' or data values), frames, grid axes, plot identifier and the plot labels will all have this same colour. The default is the pen in slot #1 on the plotter. Any of the four plotter pens may be selected.

12-CONTOUR LINE COLOUR default: pen #2

This specifier determines the colour pen to draw the contour lines. Any of the four plotter pens may be selected.

13-LINE VALUE LABEL COLOUR default: pen #3

The only part of the contour map using this colour is the set of numeric contour level values which are printed within the particular contour line. Any of the four plotter pens may be selected.

14-CONTOUR LEVEL CHARACTER SIZE default: 1.6

This item determines the character size on the plotter for the labelling of the data values (item 1), of the grid axes values (item 6) and of the plot specifier (item 7).

15-TITLE CHARACTER SIZE default: 1.8

This item allows the user to select the appropriate character size for the title labels (item 4 of PLOT MENU #2).

16-GO BACK TO MAIN KEY MENU

This option allows the user to abort the process of obtaining a contour map so that he can select a different option within the MAIN KEY MENU.

17-LEAVE PLOT MENU

In order to obtain a contour map, this option must be selected (simply pressing 'CONTINUE' without entering an option number will also, by default, select this option). If the plot is on the screen, no more inputs are required by the user, but if the plot is sent to the plotter, the user has the additional options of PLOT MENU #2.

3.4.2 PLOT MENU #2

This menu (fig.42) is only offered to the user if a contour map has been chosen to be drawn on the plotter and only after exiting from PLOT MENU #1. The six options this menu offers are described below.

1- RECALL PREVIOUS FRAMES

This option relieves the user from re-digitizing the grid area on the platen every time he wishes to plot on the plotter, by setting up the grid area as was specified by the prior digitization. Unwanted results will occur if this item is selected using a grid shape other than the one used in the initial digitization of the grid area.

2- SETUP NEW GRID FRAME

The procedure to establish an area for the contour map to be drawn by the plotter is very specific for each of the three basic grid shapes (fig.3). In all three cases, however, the process requires that the plotter pen be moved by the appropriate keys on the plotter and the 'ENTER' key on the plotter pressed to digitize the location of the pen within the platen.

For a cartesian grid two digitizations must be performed : 1- lower left corner and 2- upper right corner (fig.43). These two digitizations are sufficient to define a rectangular area whose sides are parallel to the sides of the platen and which allows the user to choose any size, location and scale of the transverse and longitudinal axes he desires.

For the polar plots, two digitizations are required for a plot to scale and three are needed for those not to scale (fig.44). For a polar annulus, the first digitization locates the desired centre of the plot while the second determines the outer radius. It is suggested that the pen be moved only transversely or longitudinaly from the centre to the outer radius.

For a polar sector, since the centre of the sector to be drawn may fall outside the platen surface, a different digitizing scheme is required (fig.45). The first digitization locates the sector point whose radial value is the outer grid radius and whose angle is the largest specified by the circumferential grid co-ordinates (i.e. outer point which is the furthest counter-clockwise). The second digitization locates a point again on the outer grid radius but at the angle of least value specified by the grid co-ordinates. These two digitizations determine the size, location and orientation of the drawn sector (fig.5).

After the digitizations have been made, the computer waits until the user is ready to have the grid plotted. This allows one to place a guide sheet on the platen, which defines standard digitizing points in order to create standard grid areas, and once these locations have been digitized, the user has the opportunity to remove this guide sheet and to place a clean sheet on the plotter. Once the user is ready to plot on this clean sheet, he simply presses 'CONTINUE'.

3- DRAW ADDITIONAL RECTANGULAR FRAMES

As many as four additional rectangular frames may be drawn if the user wishes to define a boundary other than the outer grid frame for cartesian grids or to simply put a cosmetic frame around the contour map (fig.6&7 and 16-22).

4- PRINT LABELS

As many labels as desired may be printed by the plotter anywhere with respect to the grid area using this labelling process (fig.46). The program requests that the user specify the desired line origin index (LORG) whose value is an integer from 1 to 9. By specifying this index, it is possible to create labels which are centered or left, right, lower, or upper justified. Both the BASIC Language Reference and the BASIC Progamming Techniques manuals have detailed information concerning the LORG command. The default value for this input is LORG=5 (label will be centered about the pen location) and is obtained by simply pressing 'CONTINUE' rather than entering a numeric value. The next input is a character string of at most 80 characters to be labelled. The third input is to locate the desired label on the paper. This is accomplished by moving the pen to the desired line origin location with reference to the specified LORG value and then pressing 'CONTINUE' on the 9836's keyboard. These three steps may be repeated as often as one wishes and this process is terminated by entering the number '0' as the LORG input. Also, the experienced user/programmer may wish to press the 'PAUSE' key halting temporarily the program run and enabling him to perform any processing he desires. Thus, knowing some of the BASIC graphic commands, the user could have the labels printed at various angles, with different character sizes and with different pens. Once these commands have been performed, the 'CONTINUE' key must be pressed to resume the program run.

5- GO BACK TO PLOT MENU #1

This option allows the user the opportunity to return to the previous menu and from there he can select different plot selections, and then return again to the PLOT MENU #2 or he may wish to return to the MAIN KEY MENU to select an appropriate option. Upon his return to PLOT MENU #2, it will be necessary to select either item 1 or 2 to establish the grid area for the contour plot.

6- COMMENCE PLOTTING

Once this key is pressed, the program begins the determination and plotting of the selected contour levels. There may be a considerable amount of time spent before a line is plotted due to the time required to calculate the elaborate set fitted curves.

3.5 DRAWING A THREE DIMENSIONAL PLOT

Once data has been entered through the data management routine, all that is required to invoke the 3-D plotting of the data is to press key 'k4' in the MAIN KEY MENU. There are no user options for this type of plot, thus, after a brief delay in calculating the various cross plots, the 3-D plotting commences on the screen. As with the contour maps, the integrated average height of the data block is displayed on the screen (fig.47). The 3-D view is drawn representing a three dimensional block of which two plane sides are visible. The left one of these sides represents the bottom row of the data set, while the right side represents the rightmost column of the data set. The base of the 3-D block is drawn at a vertical height corresponding to the lowest data value along the two frontal sides of the block. Furthermore, only cartesian grids may be plotted as a three dimensional view.

Once the plot has been completed, the computer will prompt the user informing him that the CONTINUE key must be pressed to continue with the program. Once this key has been pressed, the screen is cleared of the above displays and the MAIN KEY MENU is again offered to the user. The 'DUMP GRAPHICS' and 'DUMP ALPHA' keys on the 9836 enable one to obtain copies of the screen displays on the printer. Prior to plotting the 3-D view, one may wish to normalize the data by a negative number, turning the 3-D plot upside down enabling the plotting of parts of the data block that may have been hidden (fig.48).

3.6 AUTOMATIC MODE MENU

In this menu (fig.49), one can enter the desired file names to be accessed, may edit them, and also select the type of plot to be drawn for all of these data files. Typical file name entry and editing is shown in figure 50. Once plotting is initiated, if an error occurs in attempting to retrieve data from any of the specified data files (e. g. file not present on disc, file format not compatible, etc.) this file will be disregarded and the computer will atttempt to plot the data from the next file. Once a plot has been completed on the screen, the image is then dumped onto the HP printer and the process is repeated until all file names have been accessed.

1- CONTOUR PLOT? YES
2- THREE DIMENSIONAL PLOT? NO

These two items act as toggle switches, with one having a value of 'YES' while the other must have a value of 'NO'. The default values are shown above.

3- ENTER / ADD THE NAMES OF THE FILES TO BE ACCESSED

As many as seventy five file names may be entered. There is no attempt by the program to verify that the entered file names are indeed valid names other than the restriction of their lengths to ten characters. The word 'QUIT' in either lower case or upper case characters must be entered in order to return to the AUTOMATIC MODE MENU. If this item is selected a second time, the process of entering file names is continued from the end of the current set of file names.

4- REVIEW / EDIT THE ENTERED FILE NAMES

The file names entered within item #3 may be edited and the list of file names are printed on the screen as they were entered. To change an erroneous entry, one enters the corresponding file name number, and then enters the correct file name. To delete an entry, the erroneous entry is again selected, but, rather than entering a new file name, the CONTINUE key is pressed without any other characters typed at the keyboard. This will delete the selected entry and all following names in the list will be renumbered to reflect the change. To stop the editing process, the number '0' is entered rather than a valid file name number.

5- COMMENCE PLOTTING

If a contour plot had been chosen to represent the data; once this item is selected, the user will be offered an abbreviated version of PLOT MENU #1 (fig.50). The items from that menu are selected as in the normal manner, and once the user leaves the PLOT MENU #1, the automatic plotting process begins. If the three dimensional plot is the chosen representation, upon the selection of this item, the automatic plotting begins immediately since ther are no options involved for the three dimensional plot.

6- RETURN TO THE MAIN KEY MENU

This option allows one to abort the automatic plotting process, returning him to the MAIN KEY MENU.

3.7 INTERUPT FACILITY

An interrupt facility exists (SFK k9) enabling the user to halt the process of drawing a contour map or 3-D view, and which returns the user back to the MAIN KEY MENU. Thus, if a plot is not satisfactory, the user need not wait until it is completed in order to attempt another trial.

4.0 FORMAT SPECIFICATIONS

4.1 FORMAT FOR KEYBOARD ENTRIES

To enter a response to a computer prompt, the user must only type the characters that are required for the necessary operation. Any additional characters will cause the program to misread your response. Numeric responses must not contain any alphabetical characters unless the user is using scientific notation to describe the numerical value. At various places within the program, the computer will prompt the user to respond by entering one of the following words: 'KEYBOARD', 'YES', 'NO' or 'QUIT'. It is paramount that the entire word be typed and not simply the first character. This stipulation exists to minimize the chance of erroneously pressing the wrong key.

4.2 OUTPUT FORMATS

Although the data specifying the grid locations and the grid point values may have up to twelve significant digits, there are specific output formats which will not display this number of digits. In the data editing sequence, data is printed with four digits before the decimal point and three digits after the point. Also, the set of plotted data values and the list of contour points also have this same format specification. Thus, if a number has more digits after the decimal point, only three will appear with rounding of the last digit. Thus, when editing data, although the data values may seem truncated, their full values remain in the computer's memory. Trailing zeroes are printed, whereas leading zeroes are replaced with blanks. However, if a number has more than four digits before the decimal point (including a negative sign '-') then this format cannot be used to represent the value but rather, the entire decimal expansion of the number will be printed. This may cause some overlap in the values shown on the screen and on the plotter.

When listing the data set on the printer, the output format is similar to the formats previously discussed except that four digits are printed after the decimal point. Furthermore, if a value has more than four digits before the point, a series of asterisks ('*********') will be printed to show that there has been a format overflow.

4.3 DATA DISC FORMATS

The data files created to store the data on disc have a specific format which must be followed. Thus, by using the specified format it is possible to create suitable data files externally from the context of 'CONTOURS'. To create a data file, the following lines of code are used in the program and hence, an externally created data file must have the same format in order for proper data retrieval by 'CONTOURS'.

```
1       DIM Form$[12],Labels$(3)[80],File$[10]
2       REAL Time,A(2100),Lx(50),Ly(50),F(2000)
3       INTEGER Imax,Jmax,K,Rec,Newmax,Total
                .
                .
11      Newmax=Imax+Jmax+Imax*Jmax
12      Total=280+Newmax*8
13      Rec=INT((Total+20)/256)+1
14      CREATE BDAT File$,Rec
15      REDIM A(Newmax)
                .
                .
21      FOR K=1 TO Imax
22        A(K)=Lx(K)
23      NEXT K
                .
                .
31      FOR K=1 TO Jmax
32        A(K+Imax)=Ly(K)
33      NEXT K
                .
                .
41      FOR K=1 TO Imax*Jmax
42        A(K+Imax+Jmax)=F(K)
43      NEXT K
                .
                .

51      ASSIGN @PATH TO File$
52      OUTPUT @PATH;Time;Form$;Labels$(*);Imax;Jmax;A(*)
53      ASSIGN @PATH TO *
```

Both the 'Time' and 'Labels$' variables are dummy variables since they do not have to be specified to any particular value for the program to use the data file effectively. In fact, although future updated versions of 'CONTOURS' may make use of these variables, the program does not utilize them at the present time. Although, these variables are not utilized, it is imperative that they be be present in the 'OUTPUT' statement of line 52.

The other four variables must be specified with the following guide lines. The variable 'Form$' must have one of the three values below:

```
1-  Form$="CARTESIAN"
2-  Form$="POLARSEC"
3-  Form$="POLARANN"
```

Undesired results may occur if the variable 'Form$' does not have a value equal to one shown above.

'Imax' has a value equal to the number of longitudinal grid lines for a cartesian grid (radial grid lines for a polar grids). 'Jmax' has a value equal to the number of transverse grid lines for a cartesian grid (circumferential grid lines for a polar grids). The data array 'A' contains the Imax and Jmax grid locations and the data values at the Imax*Jmax grid points. The manner in which these values must be ordered within the array 'A' is shown in lines 21 to 43 above. The array 'Lx' contains the Imax grid line positions which must be monotonically increasing in value. The array 'Ly' contains the Jmax grid line positions which must also be monotonically increasing in value. The array 'F' contains the Imax*Jmax data values which must be ordered in a manner shown in table 6. Thus the data items are ordered sequentially beginning at the grid point specified by the transverse grid line with the greatest value and by the longitudinal grid line with the lowest value (upper left corner of the data set) and moving down the longitudinal grid line filling it, and then moving again from top to bottom of each of the next longitudinal grid lines whose grid position is monotonically increasing. If 'I' and 'J' represent particular column and row of data respectively, the value at this position in the data matrix is mapped into position (I-1)*Jmax+J in the vector array 'F'.

The array 'A' is simply a storage array which contains the values of the working arrays 'Lx', 'Ly', and 'F'. Please note that the values within 'A' may be either positive or negative. However, although the program may accept very large or small data values, they will not be displayed properly due to format restrictions and more importantly may cause some round-off errors which may cause errors in the contour and three dimensional plots, or worse, may cause the computer to terminate all activity for that program run.

5.0 LIMITATIONS

When drawing figures on the plotter, the user should be aware of the plotter's inability of plotting beyond its 'P1' and 'P2' settings. If the grid area is digitized beyond the available plotter area, part of the figure will not be drawn. All items drawn on the plotter are subject to this limitation, including printed labels.

The rational splines have a first derivative boundary condition which is equal to the slope of a linear segment in the outer grid intervals. This causes extraneous inflection points in the fitted cross plot within the outer grid intervals (fig.15). A better boundary condition may be to set the outer first derivative to the slope at the boundary of a circular arc fitted to the three outer points (comprising of a boundary grid point and the two inner grid points along the same grid line).

Although the program has the capability of accepting a 2000 point data set, to plot its contour map would take quite a long time. Even with smaller data sets, the time taken may seem unusually large. This is due to the extensive curve fitting and the resulting complexity of the interpolation scheme to find the contour points and lines. Furthermore, the program's code is interpreted, rather than compiled into an object code. The computing time may be reduced by a factor of 5 to 10 if the code is rewritten into a language which may be compiled and run on a suitable computer. FORTRAN is a widely used language, and is suitable for this purpose since the structure of the transcribed code from BASIC to FORTRAN would be very similar. Also, the 9836 may be upgraded to include a FORTRAN compiler. While some machines offer a BASIC compiler, since BASIC does have a universal syntax, the code presented in this manual would nonetheless have to be translated.

It is also possible to make the searching for contour points more efficient but this would necessitate larger memory utilization and the resulting savings in computing time would only be a fraction of the savings offered by a compiled code.

To run the program as described in this manual, the computer requires enough memory for the following items:

		no. of bytes	
Loading of BASIC 2.0, AP2_1, GRAPH2_1	:	502 766	
Loading of program 'CONTOURS'	:	152 120	
Variable allocation	:	705 000	(approx.)
Total	:	1 360 000	(approx.)

Within the variable allocation, approximately 88% of the total memory requirement for variable allocation is taken by the following arrays:

Xcont	Ycont	A	F	
Crosshor	Ahor	Bhor	Chor	Dhor
Crossver	Aver	Bver	Cver	Dver
G	Gdum	(allocated in subroutine Data_input)		

The amount of required memory may be reduced by dimensioning the above arrays to a smaller size, thus decreasing the size of data set that may be accepted and plotted. Furthermore, approximately 50000 bytes may be saved when loading 'CONTOURS' into memory by using a version without comments. This will also decrease slightly the amount of computing time spent to produce the various plots.

6.0 REFERENCES

Dierckx, P., Algorithms for Smoothing Data with Periodic and Parametric Splines, Computer Graphics and Image Processing 20, pp 171-184, (1982)

Faber, D. H., Rutten-Keulemans, E. W. M., Altona, C., Computer Plotting of Contour Maps: An Improved Method, Comp. and Chem., 3, pp 51-55, (1979)

Mcallister, D. F., Roulier, J. A., An Algorithm for Computing a Shape-Preserving Osculatory Quadractic Spline, ACM Math. Softw. 7, 3, pp 383-386, (1981)

Reinsch, C. H., Smoothing by Spline Functions, Numerische Mathematik, Vol 10, pp 177-183, (1967)

Spath, H., Spline Algorithms for Curves and Surfaces, Utilitas Mathematica Publishing Inc., Winnipeg, (1974)

Table 1 Sample data set with cartesian grid

1 Jan 2000 13:43:52

>>>> HERE IS THE DATA FROM FILE: TEST <<<<

	0.0000	3.2500	6.5700	9.9200	13.3000	16.7500	20.1500	23.4200
29.2500	-.8900	-.8700	-.8600	-.8900	-.8900	-.8800	-.8500	-.9800
22.7500	-.9200	-.8700	-.8800	-.9200	-.9000	-.9000	-.9300	-.8600
16.2500	-.9300	-.8800	-.9000	-.9100	-.9100	-.9200	-.9000	-.8600
9.7500	-.9000	-.9300	-.9000	-.9100	-.9000	-.9000	-.9100	-.8400
3.2500	-1.0200	-1.0500	-1.0300	-.9800	-.9900	-.9500	-1.0400	-1.0100

Table 2 Sample data set with cartesian grid

1 Jan 2000 13:44:05

>>>> HERE IS THE DATA FROM FILE: TEST2 <<<<

	0.0000	3.2500	6.5700	9.9200	13.3000	16.7500	20.1500	23.4200
32.5000	39.6100	98.1900	131.2300	116.5800	120.7600	108.5800	116.8100	117.2200
26.0000	152.2300	178.4100	203.1700	201.4900	206.7800	203.5800	177.7800	180.8000
19.5000	176.4000	209.6400	202.1000	178.0000	170.2700	185.9800	205.3000	236.2900
13.0000	238.2100	196.0400	175.8100	167.6400	154.7000	183.1500	209.1200	229.2000
6.5000	188.5000	192.0100	197.9400	185.8000	188.5000	198.8100	163.0300	126.0900
0.0000	127.8400	109.3500	130.3400	129.4500	124.9100	125.1700	119.4200	57.3700

Table 3 Sample data set with a polar annular grid

1 Jan 2000 13:41:09

>>>> HERE IS THE DATA FROM FILE: TEST3 <<<<

	50.0000	130.0000	210.0000	330.0000
7.5770	.1360	.1440	.1970	.1910
6.9430	.0720	.1040	.1990	.1470
6.2030	.0790	.0820	.1740	.1620
5.2800	.1190	.1250	.2180	.2090
3.8750	.1560	.0840	.1680	.2600

Table 4 Sample data set with a polar sector grid

1 Jan 2000 13:41:48

>>>> HERE IS THE DATA FROM FILE: TEST5 <<<<

	0.0000	2.5700	5.1400	7.7100	10.2900	12.8600	15.4300	18.0000	20.5700
10.3710	.2304	.1322	.0508	.0232	.0146	.0271	.0680	.1705	.2259
10.2450	.1984	.1217	.0414	.0145	.0091	.0086	.0127	.0258	.0408
9.9930	.0953	.1304	.0596	.0257	.0139	.0081	.0065	.0072	.0133
9.7410	.1192	.1664	.1419	.0712	.0321	.0156	.0087	.0104	.0177
9.4890	.2646	.2624	.1470	.0684	.0351	.0178	.0119	.0092	.0228
9.2370	.2982	.3256	.1769	.0863	.0394	.0204	.0136	.0127	.0240
8.9850	.3316	.3716	.2151	.1099	.0475	.0226	.0163	.0170	.0165
8.7330	.3124	.4119	.2710	.1204	.0415	.0164	.0160	.0225	.0161
8.4810	.3063	.4162	.2749	.1008	.0252	.0043	.0090	.0307	.0247
8.2290	.3089	.3914	.1921	.0311	-.0105	-.0118	-.0029	.0237	.0204
7.9770	.4385	.3395	.1136	.0077	-.0220	-.0180	-.0140	-.0004	.0884
7.7900	.5871	.4683	.3316	.2574	.2058	.1629	.1385	.2217	.5984

	23.1400	25.7100	28.2900	30.8600	33.4300	36.0000	38.5700	41.1400
10.3710	.4174	.2979	.2304	.1322	.0508	.0232	.0146	.0271
10.2450	.0771	.1363	.1984	.1217	.0414	.0145	.0091	.0086
9.9930	.0163	.0255	.0953	.1304	.0596	.0257	.0139	.0081
9.7410	.0222	.0551	.1192	.1664	.1419	.0712	.0321	.0156
9.4890	.0317	.0937	.2646	.2624	.1470	.0684	.0351	.0178
9.2370	.0350	.1161	.2982	.3256	.1769	.0863	.0394	.0204
8.9850	.0266	.1209	.3316	.3716	.2151	.1099	.0475	.0226
8.7330	.0313	.1026	.3124	.4119	.2710	.1204	.0415	.0164
8.4810	.0369	.0887	.3063	.4162	.2749	.1008	.0252	.0043
8.2290	.0193	.0871	.3089	.3914	.1921	.0311	-.0105	-.0118
7.9770	.3125	.4699	.4385	.3395	.1136	.0077	-.0220	-.0180
7.7900	.6689	.6743	.5871	.4683	.3316	.2574	.2058	.1629

Table 5 Data set indicating the order in which data is entered manually at the keyboard within the various data blocks

1 Jan 2000 13:42:14

>>>> HERE IS THE DATA FROM FILE: TEST6 <<<<

	0.0000	2.5700	5.1400	7.7100	10.2900	12.8600	15.4300	18.0000	20.5700
10.3710	1.0000	7.0000	13.0000	19.0000	25.0000	31.0000	37.0000	43.0000	97.0000
10.2450	2.0000	8.0000	14.0000	20.0000	26.0000	32.0000	38.0000	44.0000	98.0000
9.9930	3.0000	9.0000	15.0000	21.0000	27.0000	33.0000	39.0000	45.0000	99.0000
9.7410	4.0000	10.0000	16.0000	22.0000	28.0000	34.0000	40.0000	46.0000	100.0000
9.4890	5.0000	11.0000	17.0000	23.0000	29.0000	35.0000	41.0000	47.0000	101.0000
9.2370	6.0000	12.0000	18.0000	24.0000	30.0000	36.0000	42.0000	48.0000	102.0000
8.9850	49.0000	55.0000	61.0000	67.0000	73.0000	79.0000	85.0000	91.0000	127.0000
8.7330	50.0000	56.0000	62.0000	68.0000	74.0000	80.0000	86.0000	92.0000	128.0000
8.4810	51.0000	57.0000	63.0000	69.0000	75.0000	81.0000	87.0000	93.0000	129.0000
8.2290	52.0000	58.0000	64.0000	70.0000	76.0000	82.0000	88.0000	94.0000	130.0000
7.9770	53.0000	59.0000	65.0000	71.0000	77.0000	83.0000	89.0000	95.0000	131.0000
7.7900	54.0000	60.0000	66.0000	72.0000	78.0000	84.0000	90.0000	96.0000	132.0000

	23.1400	25.7100	28.2900	30.8600	33.4300	36.0000	38.5700	41.1400
10.3710	103.0000	109.0000	115.0000	121.0000	157.0000	163.0000	169.0000	175.0000
10.2450	104.0000	110.0000	116.0000	122.0000	158.0000	164.0000	170.0000	176.0000
9.9930	105.0000	111.0000	117.0000	123.0000	159.0000	165.0000	171.0000	177.0000
9.7410	106.0000	112.0000	118.0000	124.0000	160.0000	166.0000	172.0000	178.0000
9.4890	107.0000	113.0000	119.0000	125.0000	161.0000	167.0000	173.0000	179.0000
9.2370	108.0000	114.0000	120.0000	126.0000	162.0000	168.0000	174.0000	180.0000
8.9850	133.0000	139.0000	145.0000	151.0000	181.0000	187.0000	193.0000	199.0000
8.7330	134.0000	140.0000	146.0000	152.0000	182.0000	188.0000	194.0000	200.0000
8.4810	135.0000	141.0000	147.0000	153.0000	183.0000	189.0000	195.0000	201.0000
8.2290	136.0000	142.0000	148.0000	154.0000	184.0000	190.0000	196.0000	202.0000
7.9770	137.0000	143.0000	149.0000	155.0000	185.0000	191.0000	197.0000	203.0000
7.7900	138.0000	144.0000	150.0000	156.0000	186.0000	192.0000	198.0000	204.0000

Table 6 Data set indicating the order in which data is stored in the program arrays F and A

1 Jan 2000 13:42:51

>>>> HERE IS THE DATA FROM FILE: TEST7 <<<<

	0.0000	2.5700	5.1400	7.7100	10.2900	12.8600	15.4300	18.0000	20.5700
10.3710	1.0000	13.0000	25.0000	37.0000	49.0000	61.0000	73.0000	85.0000	97.0000
10.2450	2.0000	14.0000	26.0000	38.0000	50.0000	62.0000	74.0000	86.0000	98.0000
9.9930	3.0000	15.0000	27.0000	39.0000	51.0000	63.0000	75.0000	87.0000	99.0000
9.7410	4.0000	16.0000	28.0000	40.0000	52.0000	64.0000	76.0000	88.0000	100.0000
9.4890	5.0000	17.0000	29.0000	41.0000	53.0000	65.0000	77.0000	89.0000	101.0000
9.2370	6.0000	18.0000	30.0000	42.0000	54.0000	66.0000	78.0000	90.0000	102.0000
8.9850	7.0000	19.0000	31.0000	43.0000	55.0000	67.0000	79.0000	91.0000	103.0000
8.7330	8.0000	20.0000	32.0000	44.0000	56.0000	68.0000	80.0000	92.0000	104.0000
8.4810	9.0000	21.0000	33.0000	45.0000	57.0000	69.0000	81.0000	93.0000	105.0000
8.2290	10.0000	22.0000	34.0000	46.0000	58.0000	70.0000	82.0000	94.0000	106.0000
7.9770	11.0000	23.0000	35.0000	47.0000	59.0000	71.0000	83.0000	95.0000	107.0000
7.7900	12.0000	24.0000	36.0000	48.0000	60.0000	72.0000	84.0000	96.0000	108.0000

	23.1400	25.7100	28.2900	30.8600	33.4300	36.0000	38.5700	41.1400
10.3710	109.0000	121.0000	133.0000	145.0000	157.0000	169.0000	181.0000	193.0000
10.2450	110.0000	122.0000	134.0000	146.0000	158.0000	170.0000	182.0000	194.0000
9.9930	111.0000	123.0000	135.0000	147.0000	159.0000	171.0000	183.0000	195.0000
9.7410	112.0000	124.0000	136.0000	148.0000	160.0000	172.0000	184.0000	196.0000
9.4890	113.0000	125.0000	137.0000	149.0000	161.0000	173.0000	185.0000	197.0000
9.2370	114.0000	126.0000	138.0000	150.0000	162.0000	174.0000	186.0000	198.0000
8.9850	115.0000	127.0000	139.0000	151.0000	163.0000	175.0000	187.0000	199.0000
8.7330	116.0000	128.0000	140.0000	152.0000	164.0000	176.0000	188.0000	200.0000
8.4810	117.0000	129.0000	141.0000	153.0000	165.0000	177.0000	189.0000	201.0000
8.2290	118.0000	130.0000	142.0000	154.0000	166.0000	178.0000	190.0000	202.0000
7.9770	119.0000	131.0000	143.0000	155.0000	167.0000	179.0000	191.0000	203.0000
7.7900	120.0000	132.0000	144.0000	156.0000	168.0000	180.0000	192.0000	204.0000

Table 7 List of contour points obtained by plotting the data from Table 1 normalized by its arithmetic mean and with a damping factor of 2.0. Actual contour plot defined by these contour points is shown in figure 18.

```
##########################################################
                L E V E L  =  .925

**************************************
LINE # 1  - # OF POINTS= 5        LEFT_JOIN: 0       RIGHT_JOIN: 0

   1.000      19.083      29.250
   2.000      19.197      29.167
   3.000      19.581      29.008
   4.000      19.965      29.075
   5.000      20.155      29.250

**************************************
LINE # 2  - # OF POINTS= 2        LEFT_JOIN: 3       RIGHT_JOIN: 0

   1.000      23.036       9.629
   2.000      23.420       9.105

**************************************
LINE # 3  - # OF POINTS= 5        LEFT_JOIN: 2       RIGHT_JOIN: 0

   1.000      22.972       9.750
   2.000      22.691      11.050
   3.000      22.808      12.350
   4.000      23.036      13.251
   5.000      23.420      14.249

##########################################################
                L E V E L  =  .95

**************************************
LINE # 1  - # OF POINTS= 16       LEFT_JOIN: 0       RIGHT_JOIN: 0

   1.000       2.718      29.250
   2.000       3.071      20.763
   3.000       3.455      18.868
   4.000       3.839      19.021
   5.000       4.223      19.816
   6.000       4.607      20.734
   7.000       4.991      21.580
   8.000       5.375      22.347
   9.000       5.759      23.104
  10.000       6.143      23.908
  11.000       6.527      24.813
  12.000       6.911      25.921
  13.000       7.295      27.068
  14.000       7.679      28.141
  15.000       8.063      29.140
  16.000       8.106      29.250

**************************************
LINE # 2  - # OF POINTS= 11       LEFT_JOIN: 0       RIGHT_JOIN: 0

   1.000      17.344      29.250
   2.000      17.661      28.564
   3.000      18.045      28.046
   4.000      18.429      27.727
   5.000      18.813      27.528
   6.000      19.197      27.415
   7.000      19.581      27.380
   8.000      19.965      27.446
   9.000      20.349      27.726
  10.000      20.732      28.441
  11.000      20.941      29.250
```

Table 7 cont'd

```
************************************
LINE # 3  - # OF POINTS= 5       LEFT_JOIN: 4      RIGHT_JOIN: 0

   1.000      21.884      10.045
   2.000      22.268       9.315
   3.000      22.652       8.822
   4.000      23.036       8.420
   5.000      23.420       8.087


************************************
LINE # 4  - # OF POINTS= 11      LEFT_JOIN: 3      RIGHT_JOIN: 0

   1.000      21.628      11.345
   2.000      21.633      12.645
   3.000      21.884      14.334
   4.000      22.268      15.787
   5.000      22.386      16.250
   6.000      22.554      17.550
   7.000      22.595      18.850
   8.000      22.616      20.150
   9.000      22.652      21.170
  10.000      23.036      23.234
  11.000      23.420      23.767


##################################################
            L E V E L  =  .975

************************************
LINE # 1  - # OF POINTS= 24      LEFT_JOIN: 0      RIGHT_JOIN: 2

   1.000       0.000      27.975
   2.000        .384      27.224
   3.000        .768      26.169
   4.000       1.152      24.569
   5.000       1.536      21.708
   6.000       1.622      20.750
   7.000       1.725      19.450
   8.000       1.821      18.150
   9.000       1.920      16.850
  10.000       2.304      14.571
  11.000       2.688      13.903
  12.000       3.071      13.591
  13.000       3.455      13.425
  14.000       3.839      13.320
  15.000       4.223      13.223
  16.000       4.607      13.101
  17.000       4.991      12.905
  18.000       5.375      12.493
  19.000       5.759      11.619
  20.000       6.143      10.754
  21.000       6.527      10.302
  22.000       6.911      10.212
  23.000       7.295      10.377
  24.000       7.679      10.992


************************************
LINE # 2  - # OF POINTS= 5       LEFT_JOIN: 3      RIGHT_JOIN: 1

   1.000       6.143      14.444
   2.000       6.527      13.786
   3.000       6.911      13.355
   4.000       7.295      12.833
   5.000       7.679      11.902


************************************
LINE # 3  - # OF POINTS= 43      LEFT_JOIN: 2      RIGHT_JOIN: 0

   1.000       6.019      16.250
   2.000       6.143      16.907
```

Table 7 cont'd

3.000	6.527	18.121
4.000	6.911	19.115
5.000	7.295	20.311
6.000	7.679	22.023
7.000	8.063	23.752
8.000	8.447	25.055
9.000	8.830	26.104
10.000	9.214	26.958
11.000	9.598	27.633
12.000	9.982	28.090
13.000	10.366	28.291
14.000	10.750	28.331
15.000	11.134	28.260
16.000	11.518	28.089
17.000	11.902	27.809
18.000	12.286	27.389
19.000	12.670	26.769
20.000	13.054	25.880
21.000	13.438	24.815
22.000	13.822	23.930
23.000	14.206	23.361
24.000	14.590	23.048
25.000	14.973	22.901
26.000	15.357	22.872
27.000	15.741	22.952
28.000	16.125	23.170
29.000	16.509	23.579
30.000	16.893	24.184
31.000	17.277	24.745
32.000	17.661	25.117
33.000	18.045	25.355
34.000	18.429	25.515
35.000	18.813	25.630
36.000	19.197	25.719
37.000	19.581	25.795
38.000	19.965	25.872
39.000	20.349	25.978
40.000	20.732	26.170
41.000	21.116	26.622
42.000	21.500	29.103
43.000	21.506	29.250

****
LINE # 4 - # OF POINTS= 9 LEFT_JOIN: 5 RIGHT_JOIN: 0

1.000	20.349	11.653
2.000	20.732	10.308
3.000	21.116	9.563
4.000	21.500	9.035
5.000	21.884	8.584
6.000	22.268	8.183
7.000	22.652	7.821
8.000	23.036	7.500
9.000	23.420	7.225

****
LINE # 5 - # OF POINTS= 11 LEFT_JOIN: 4 RIGHT_JOIN: 0

1.000	20.161	12.953
2.000	20.160	14.253
3.000	20.349	15.610
4.000	20.732	16.792
5.000	21.116	17.824
6.000	21.500	19.337
7.000	21.884	22.404
8.000	22.268	24.325
9.000	22.652	24.776
10.000	23.036	24.965
11.000	23.420	25.062

##########

Table 7 cont'd

LEVEL = 1

****
LINE # 1 - # OF POINTS= 62 LEFT_JOIN: 0 RIGHT_JOIN: 0

1.000	0.000	8.282
2.000	.384	8.547
3.000	.768	8.852
4.000	1.152	9.188
5.000	1.536	9.549
6.000	1.920	9.932
7.000	2.304	10.292
8.000	2.688	10.558
9.000	3.071	10.692
10.000	3.455	10.646
11.000	3.839	10.437
12.000	4.223	10.132
13.000	4.607	9.787
14.000	4.991	9.454
15.000	5.375	9.149
16.000	5.759	8.879
17.000	6.143	8.651
18.000	6.527	8.481
19.000	6.911	8.392
20.000	7.295	8.359
21.000	7.679	8.359
22.000	8.063	8.381
23.000	8.447	8.421
24.000	8.830	8.474
25.000	9.214	8.534
26.000	9.598	8.589
27.000	9.982	8.610
28.000	10.366	8.575
29.000	10.750	8.507
30.000	11.134	8.427
31.000	11.518	8.345
32.000	11.902	8.266
33.000	12.286	8.190
34.000	12.670	8.114
35.000	13.054	8.035
36.000	13.438	7.941
37.000	13.822	7.822
38.000	14.206	7.679
39.000	14.590	7.513
40.000	14.973	7.324
41.000	15.357	7.120
42.000	15.741	6.916
43.000	16.125	6.755
44.000	16.509	6.719
45.000	16.893	6.939
46.000	17.277	7.385
47.000	17.661	7.865
48.000	18.045	8.293
49.000	18.429	8.647
50.000	18.813	8.921
51.000	19.197	9.110
52.000	19.581	9.204
53.000	19.965	9.174
54.000	20.349	8.975
55.000	20.732	8.651
56.000	21.116	8.290
57.000	21.500	7.927
58.000	21.884	7.576
59.000	22.268	7.244
60.000	22.652	6.936
61.000	23.036	6.657
62.000	23.420	6.418

****
LINE # 2 - # OF POINTS= 5 LEFT_JOIN: 0 RIGHT_JOIN: 3

1.000	0.000	14.342

Table 7 cont'd

2.000	.384	15.021
3.000	.661	16.250
4.000	.678	17.550
5.000	.597	18.850

****
LINE # 3 - # OF POINTS= 2 LEFT_JOIN: 0 RIGHT_JOIN: 2

1.000	0.000	23.014
2.000	.384	20.874

****
LINE # 4 - # OF POINTS= 3 LEFT_JOIN: 5 RIGHT_JOIN: 5

1.000	9.982	20.518
2.000	10.366	20.634
3.000	10.698	21.934

****
LINE # 5 - # OF POINTS= 3 LEFT_JOIN: 4 RIGHT_JOIN: 4

1.000	9.658	21.818
2.000	9.982	23.263
3.000	10.366	23.201

****
LINE # 6 - # OF POINTS= 3 LEFT_JOIN: 7 RIGHT_JOIN: 7

1.000	16.125	15.903
2.000	16.509	15.636
3.000	16.893	15.747

****
LINE # 7 - # OF POINTS= 3 LEFT_JOIN: 6 RIGHT_JOIN: 6

1.000	16.125	16.907
2.000	16.509	17.275
3.000	16.893	17.352

****
LINE # 8 - # OF POINTS= 8 LEFT_JOIN: 9 RIGHT_JOIN: 9

1.000	18.429	19.177
2.000	18.813	18.910
3.000	19.197	18.888
4.000	19.581	18.987
5.000	19.965	19.237
6.000	20.349	19.802
7.000	20.732	20.971
8.000	20.906	22.271

****
LINE # 9 - # OF POINTS= 9 LEFT_JOIN: 8 RIGHT_JOIN: 8

1.000	18.182	20.477
2.000	18.244	21.777
3.000	18.429	22.682
4.000	18.813	23.408
5.000	19.197	23.799
6.000	19.581	24.032
7.000	19.965	24.131
8.000	20.349	24.028
9.000	20.732	23.533

****
LINE # 10 - # OF POINTS= 5 LEFT_JOIN: 0 RIGHT_JOIN: 0

Table 7 cont'd

1.000	22.027	29.250
2.000	22.268	27.666
3.000	22.652	26.793
4.000	23.036	26.421
5.000	23.420	26.217

##

L E V E L = 1.025

****

LINE # 1 - # OF POINTS= 62 LEFT_JOIN: 0 RIGHT_JOIN: 0

1.000	0.000	7.058
2.000	.384	7.280
3.000	.768	7.528
4.000	1.152	7.793
5.000	1.536	8.066
6.000	1.920	8.337
7.000	2.304	8.591
8.000	2.688	8.806
9.000	3.071	8.943
10.000	3.455	8.936
11.000	3.839	8.797
12.000	4.223	8.592
13.000	4.607	8.359
14.000	4.991	8.122
15.000	5.375	7.891
16.000	5.759	7.675
17.000	6.143	7.482
18.000	6.527	7.321
19.000	6.911	7.204
20.000	7.295	7.113
21.000	7.679	7.032
22.000	8.063	6.951
23.000	8.447	6.866
24.000	8.830	6.774
25.000	9.214	6.675
26.000	9.598	6.576
27.000	9.982	6.499
28.000	10.366	6.467
29.000	10.750	6.467
30.000	11.134	6.480
31.000	11.518	6.497
32.000	11.902	6.511
33.000	12.286	6.515
34.000	12.670	6.503
35.000	13.054	6.462
36.000	13.438	6.366
37.000	13.822	6.199
38.000	14.206	5.975
39.000	14.590	5.697
40.000	14.973	5.370
41.000	15.357	4.999
42.000	15.741	4.608
43.000	16.125	4.262
44.000	16.509	4.110
45.000	16.893	4.394
46.000	17.277	5.053
47.000	17.661	5.755
48.000	18.045	6.376
49.000	18.429	6.888
50.000	18.813	7.292
51.000	19.197	7.594
52.000	19.581	7.791
53.000	19.965	7.865
54.000	20.349	7.773
55.000	20.732	7.545
56.000	21.116	7.261
57.000	21.500	6.959
58.000	21.884	6.657
59.000	22.268	6.366

Table 7 cont'd

60.000	22.652	6.093
61.000	23.036	5.845
62.000	23.420	5.633

****
LINE # 2 - # OF POINTS= 4 LEFT_JOIN: 0 RIGHT_JOIN: 0

1.000	22.538	29.250
2.000	22.652	28.780
3.000	23.036	27.835
4.000	23.420	27.336

##########
L E V E L = 1.05

****
LINE # 1 - # OF POINTS= 41 LEFT_JOIN: 0 RIGHT_JOIN: 0

1.000	0.000	5.943
2.000	.384	6.145
3.000	.768	6.368
4.000	1.152	6.605
5.000	1.536	6.849
6.000	1.920	7.090
7.000	2.304	7.319
8.000	2.688	7.514
9.000	3.071	7.644
10.000	3.455	7.651
11.000	3.839	7.542
12.000	4.223	7.376
13.000	4.607	7.184
14.000	4.991	6.985
15.000	5.375	6.788
16.000	5.759	6.599
17.000	6.143	6.422
18.000	6.527	6.260
19.000	6.911	6.112
20.000	7.295	5.966
21.000	7.679	5.810
22.000	8.063	5.635
23.000	8.447	5.437
24.000	8.830	5.213
25.000	9.214	4.968
26.000	9.598	4.726
27.000	9.982	4.554
28.000	10.366	4.521
29.000	10.750	4.578
30.000	11.134	4.674
31.000	11.518	4.779
32.000	11.902	4.875
33.000	12.286	4.951
34.000	12.670	4.993
35.000	13.054	4.982
36.000	13.438	4.877
37.000	13.822	4.652
38.000	14.206	4.329
39.000	14.590	3.912
40.000	14.973	3.391
41.000	15.064	3.250

****
LINE # 2 - # OF POINTS= 17 LEFT_JOIN: 0 RIGHT_JOIN: 0

1.000	17.497	3.250
2.000	17.661	3.699
3.000	18.045	4.590
4.000	18.429	5.298
5.000	18.813	5.853
6.000	19.197	6.274
7.000	19.581	6.570

Table 7 cont'd

8.000	19.965	6.727
9.000	20.349	6.704
10.000	20.732	6.534
11.000	21.116	6.299
12.000	21.500	6.039
13.000	21.884	5.772
14.000	22.268	5.512
15.000	22.652	5.265
16.000	23.036	5.041
17.000	23.420	4.852

****
LINE # 3 - # OF POINTS= 2 LEFT_JOIN: 0 RIGHT_JOIN: 0

1.000	23.058	29.250
2.000	23.420	28.468

##########
L E V E L = 1.075

****
LINE # 1 - # OF POINTS= 25 LEFT_JOIN: 0 RIGHT_JOIN: 0

1.000	0.000	4.851
2.000	.384	5.042
3.000	.768	5.251
4.000	1.152	5.471
5.000	1.536	5.695
6.000	1.920	5.918
7.000	2.304	6.129
8.000	2.688	6.312
9.000	3.071	6.438
10.000	3.455	6.455
11.000	3.839	6.370
12.000	4.223	6.233
13.000	4.607	6.073
14.000	4.991	5.903
15.000	5.375	5.732
16.000	5.759	5.563
17.000	6.143	5.396
18.000	6.527	5.227
19.000	6.911	5.046
20.000	7.295	4.841
21.000	7.679	4.604
22.000	8.063	4.328
23.000	8.447	4.004
24.000	8.830	3.627
25.000	9.172	3.250

****
LINE # 2 - # OF POINTS= 6 LEFT_JOIN: 0 RIGHT_JOIN: 0

1.000	12.078	3.250
2.000	12.286	3.332
3.000	12.670	3.437
4.000	13.054	3.457
5.000	13.438	3.329
6.000	13.554	3.250

****
LINE # 3 - # OF POINTS= 15 LEFT_JOIN: 0 RIGHT_JOIN: 0

1.000	18.248	3.250
2.000	18.429	3.690
3.000	18.813	4.438
4.000	19.197	4.997
5.000	19.581	5.398
6.000	19.965	5.637
7.000	20.349	5.677

Table 7 cont'd

8.000	20.732	5.556
9.000	21.116	5.360
10.000	21.500	5.133
11.000	21.884	4.895
12.000	22.268	4.658
13.000	22.652	4.432
14.000	23.036	4.227
15.000	23.420	4.056

##
L E V E L = 1.1

****
LINE # 1 - # OF POINTS= 22 LEFT_JOIN: 0 RIGHT_JOIN: 0

1.000	0.000	3.726
2.000	.384	3.918
3.000	.768	4.124
4.000	1.152	4.336
5.000	1.536	4.550
6.000	1.920	4.761
7.000	2.304	4.961
8.000	2.688	5.134
9.000	3.071	5.258
10.000	3.455	5.286
11.000	3.839	5.222
12.000	4.223	5.112
13.000	4.607	4.979
14.000	4.991	4.834
15.000	5.375	4.685
16.000	5.759	4.530
17.000	6.143	4.367
18.000	6.527	4.186
19.000	6.911	3.962
20.000	7.295	3.685
21.000	7.679	3.347
22.000	7.776	3.250

****
LINE # 2 - # OF POINTS= 13 LEFT_JOIN: 0 RIGHT_JOIN: 0

1.000	18.955	3.250
2.000	19.197	3.695
3.000	19.581	4.223
4.000	19.965	4.552
5.000	20.349	4.656
6.000	20.732	4.579
7.000	21.116	4.419
8.000	21.500	4.219
9.000	21.884	4.002
10.000	22.268	3.782
11.000	22.652	3.570
12.000	23.036	3.376
13.000	23.327	3.250

##
L E V E L = 1.125

****
LINE # 1 - # OF POINTS= 15 LEFT_JOIN: 0 RIGHT_JOIN: 0

1.000	1.339	3.250
2.000	1.536	3.364
3.000	1.920	3.577
4.000	2.304	3.775
5.000	2.688	3.947
6.000	3.071	4.072
7.000	3.455	4.111
8.000	3.839	4.068

Table 7 cont'd

9.000	4.223	3.981
10.000	4.607	3.870
11.000	4.991	3.745
12.000	5.375	3.610
13.000	5.759	3.462
14.000	6.143	3.293
15.000	6.230	3.250

****

LINE # 2 - # OF POINTS= 7 LEFT_JOIN: 0 RIGHT_JOIN: 0

1.000	19.783	3.250
2.000	19.965	3.431
3.000	20.349	3.607
4.000	20.732	3.575
5.000	21.116	3.443
6.000	21.500	3.261
7.000	21.523	3.250

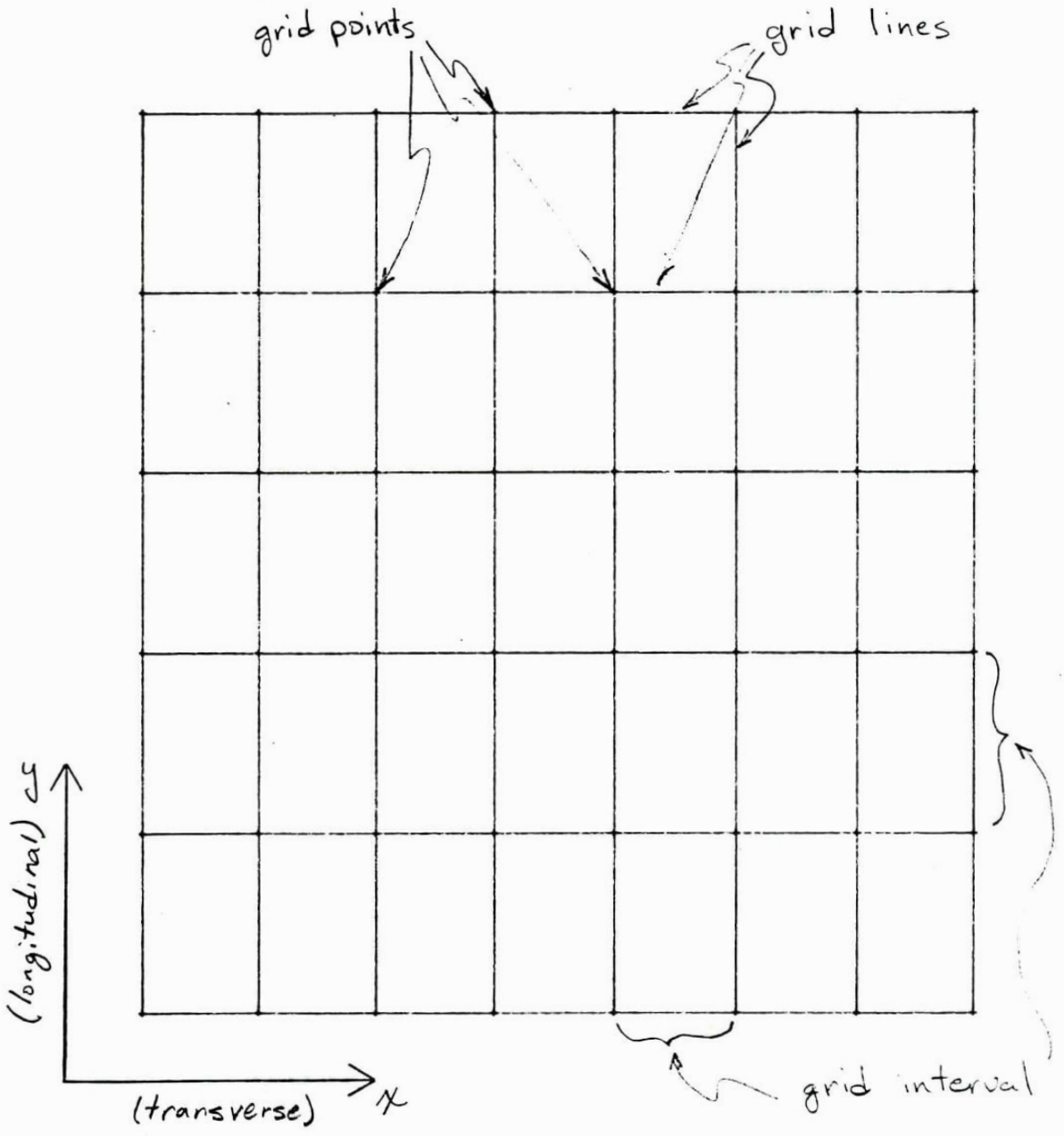


Figure 1 Typical grid.

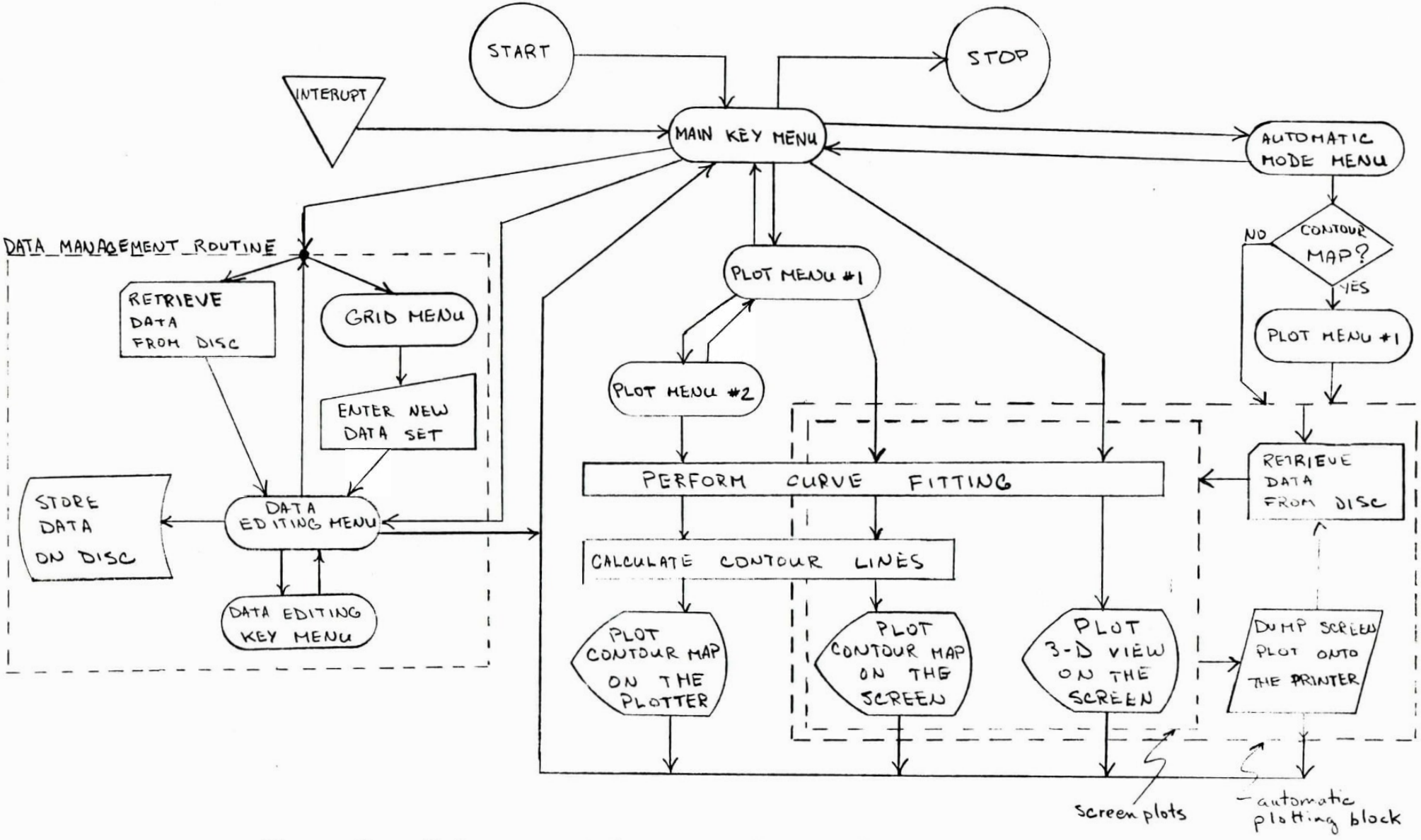
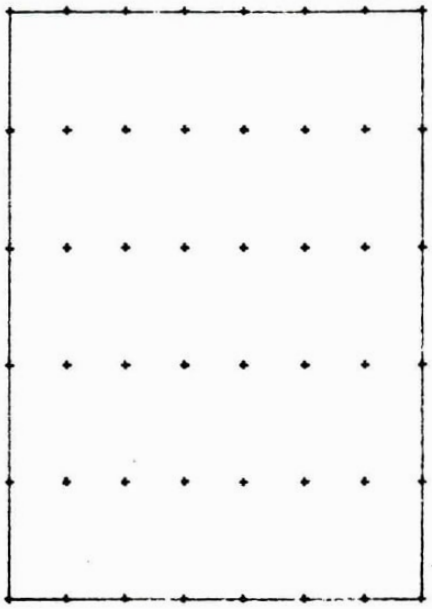
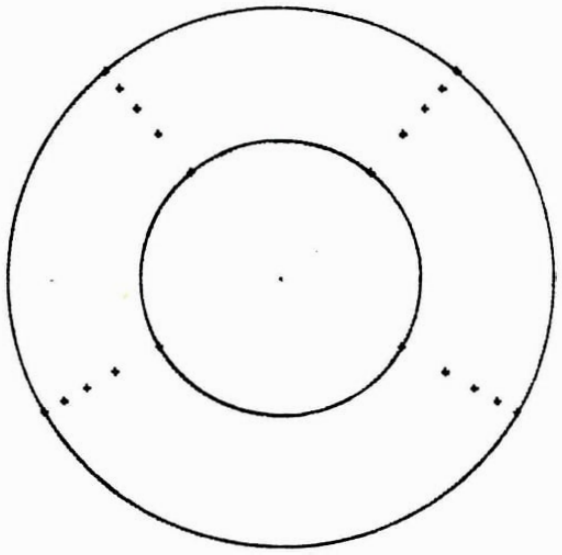


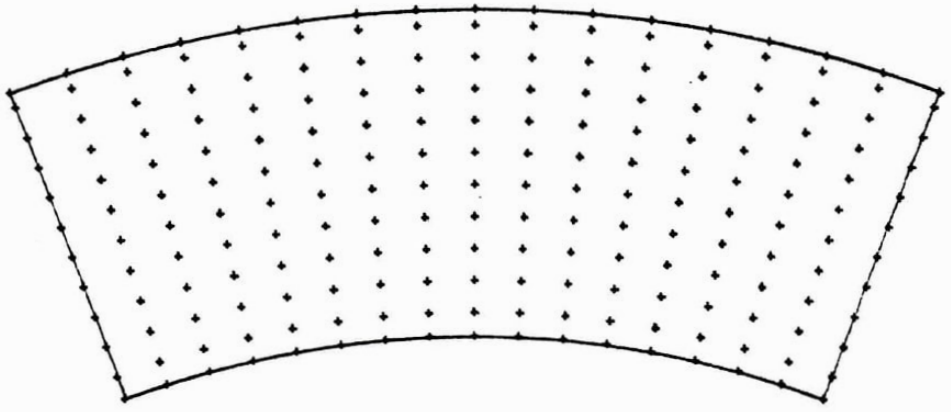
Figure 2 Major steps taken to produce various outputs.



A - Cartesian grid



B - polar annular grid



C - polar sector grid (less than 360°)

Figure 3 Basic grid shapes

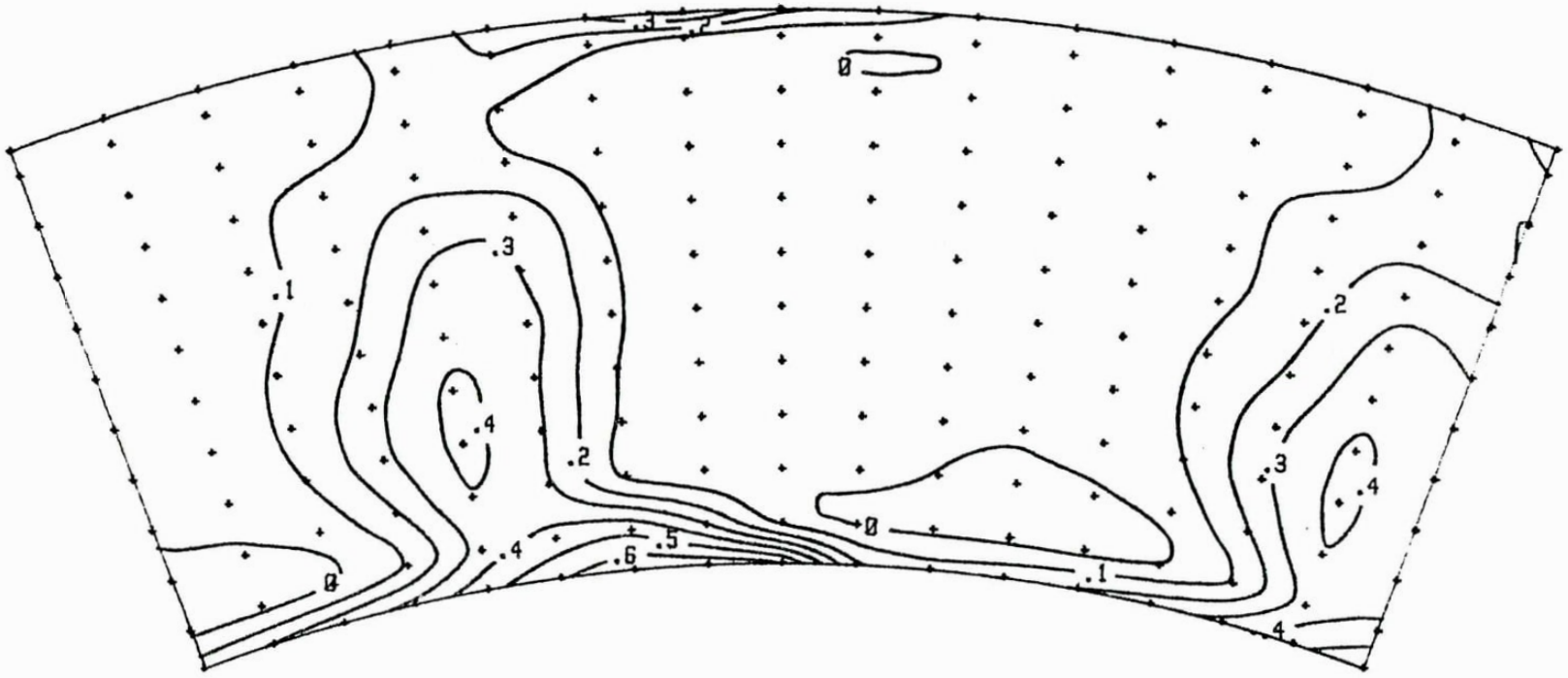
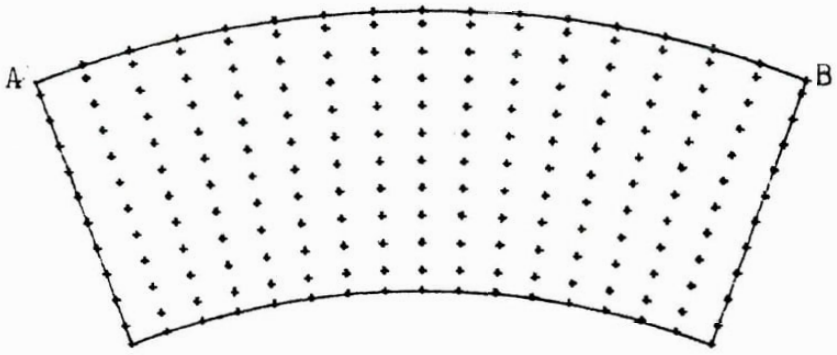
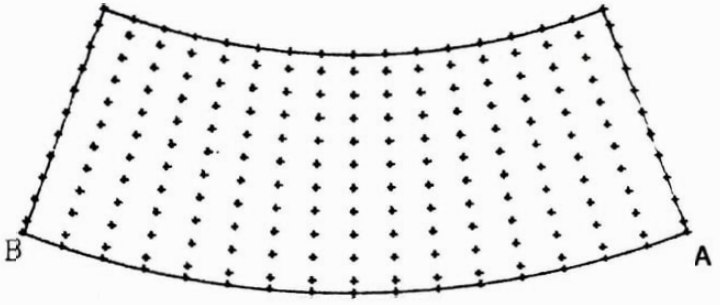


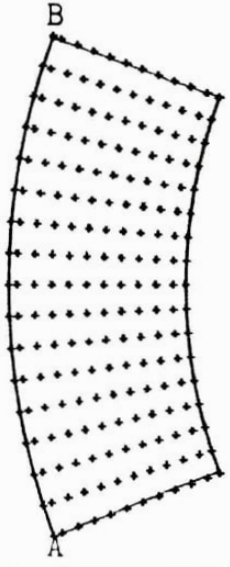
Figure 4 Contour plot displaying data from Table 4 plotted to scale.



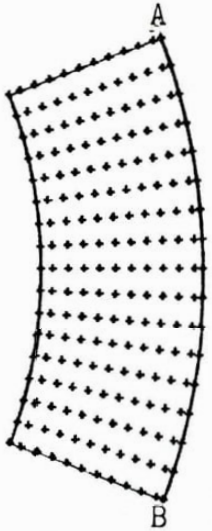
TEST5 1 Jan 2000 10:11:26
DAMPING = 2 NORMALIZING = 1



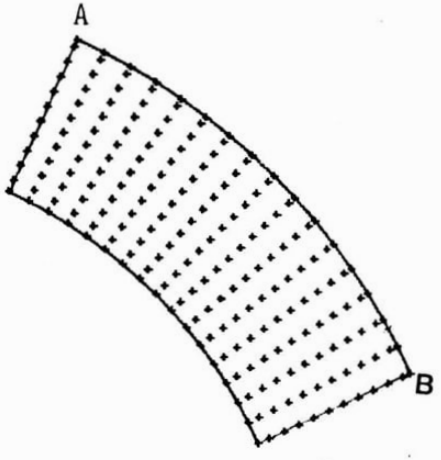
TEST5 1 Jan 2000 10:13:58
DAMPING = 2 NORMALIZING = 1



TEST5 1 Jan 2000 10:16:47
DAMPING = 2 NORMALIZING = 1



TEST5 1 Jan 2000 10:19:20
DAMPING = 2 NORMALIZING = 1



TEST5 1 Jan 2000 10:22:21
DAMPING = 2 NORMALIZING = 1

Figure 5 Polar sector grid defined in Table 4 plotted to scale but at various sizes and orientations obtained by digitizing the grid with firstly, point A, then point B.

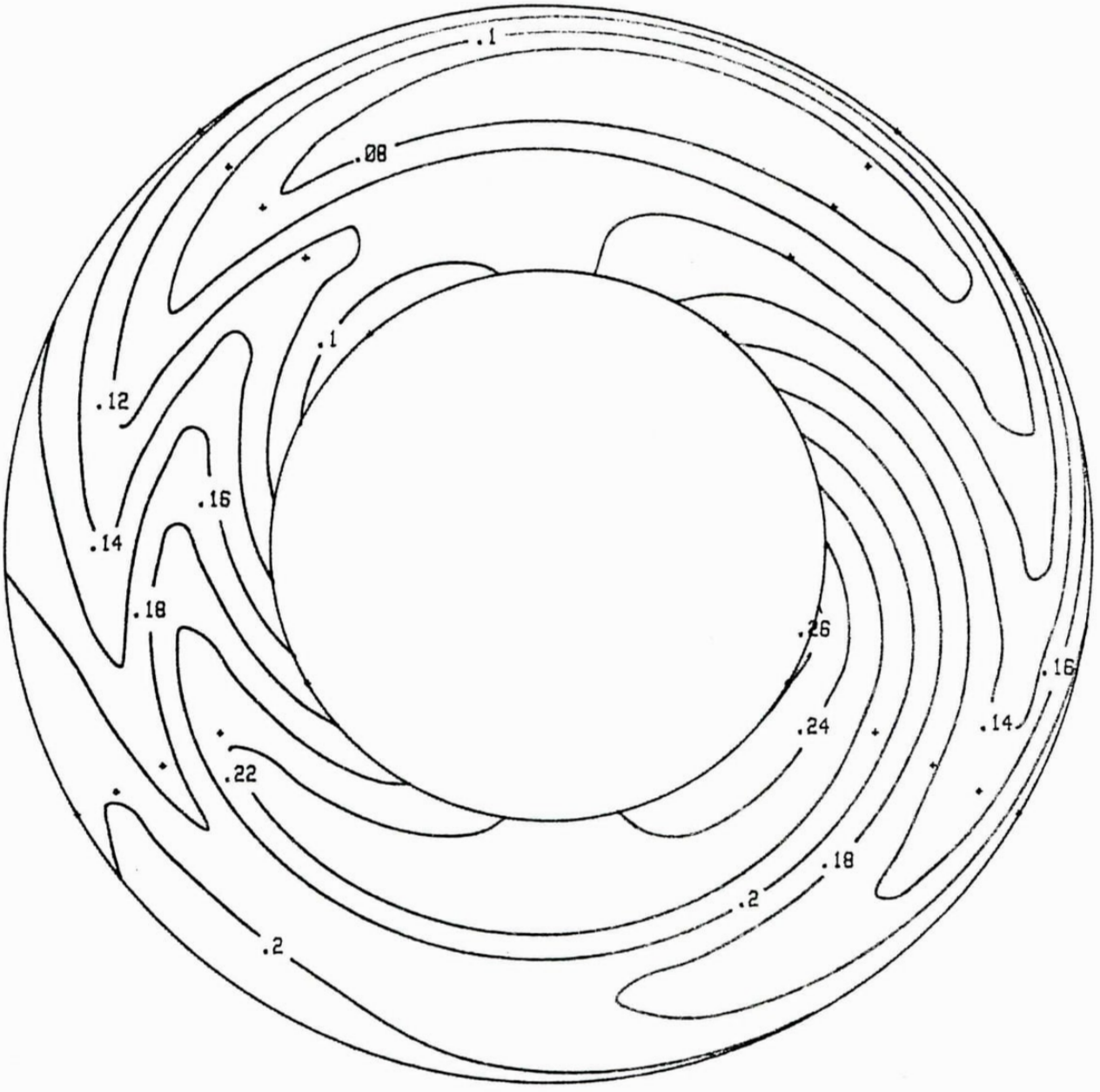


Figure 6 Contour map displaying the data from Table 3 plotted to scale.

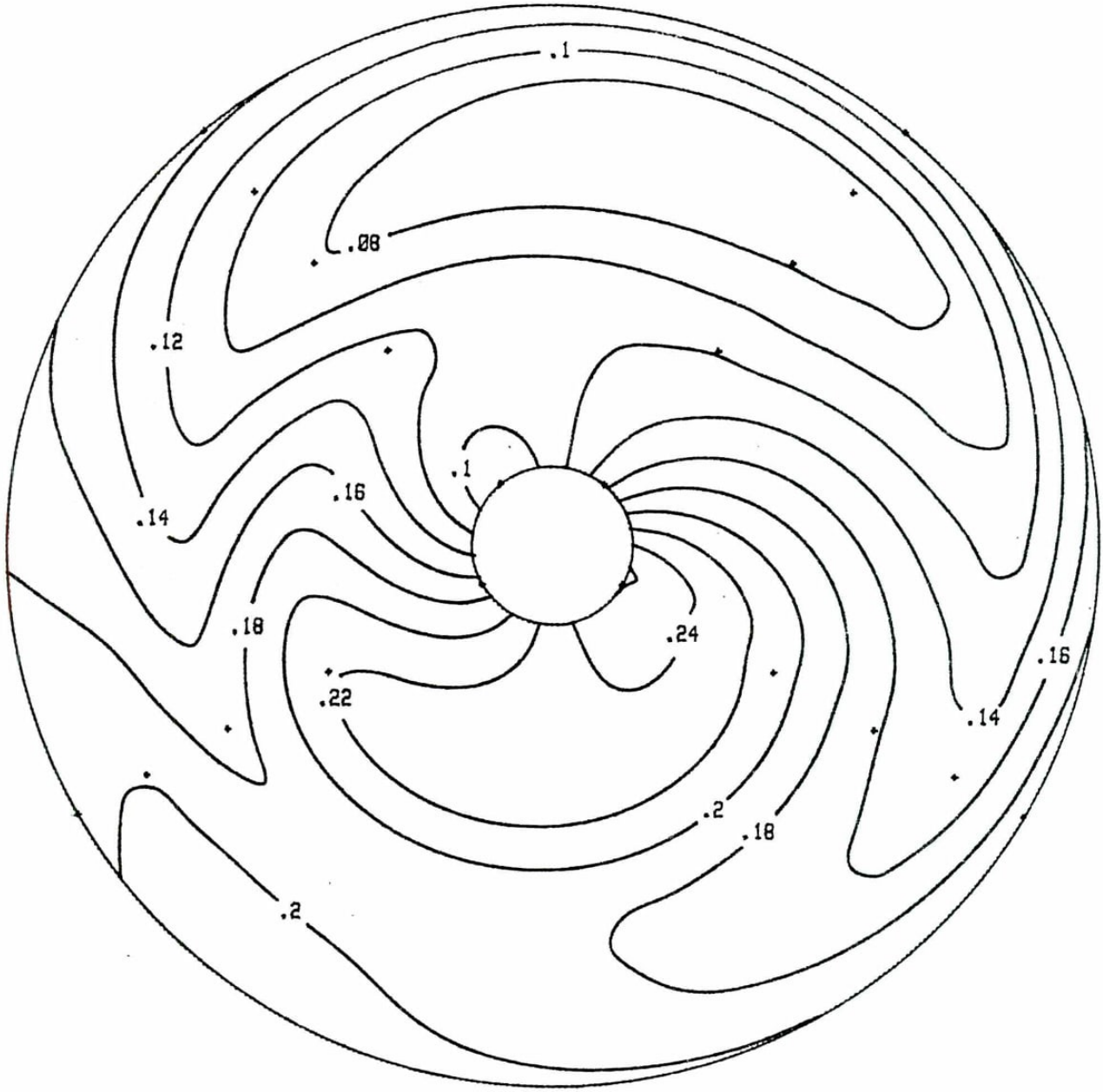


Figure 7 Contour map displaying the data from Table 3 not plotted to scale but with an exaggerated inner radius.

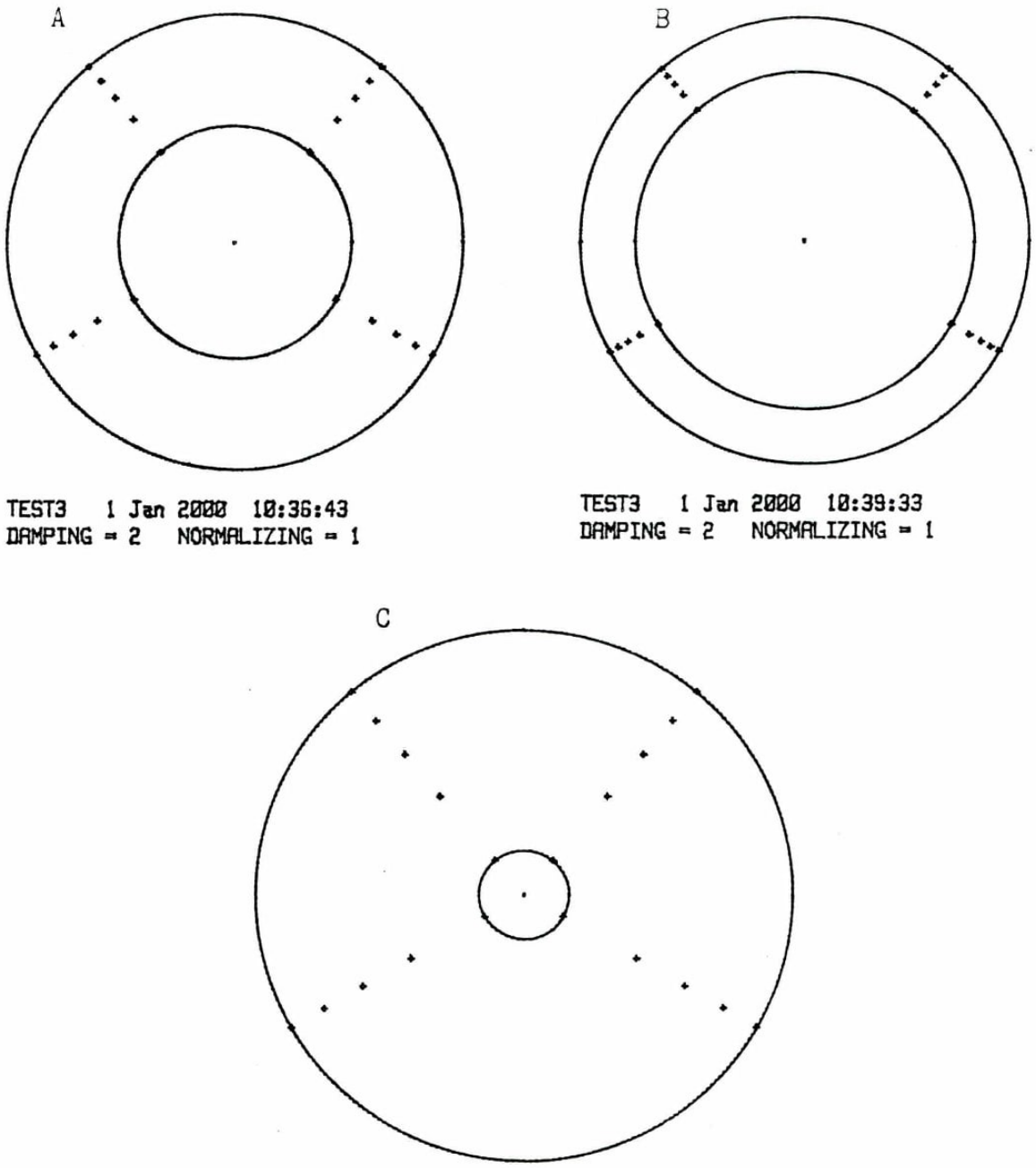


Figure 8 Polar annular grid defined in Table 3 drawn to scale in A, and not to scale in both B and C.

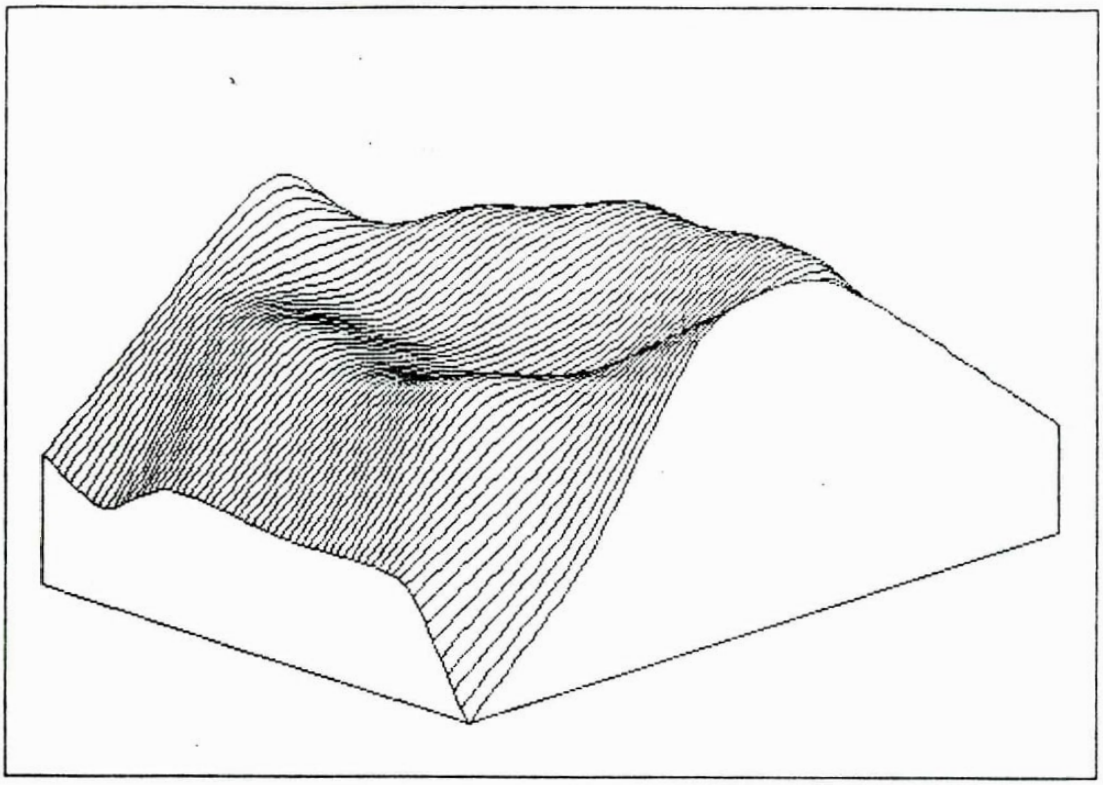


Figure 9 Three dimensional view displaying the data from Table 2 drawn on the 9836's video screen.

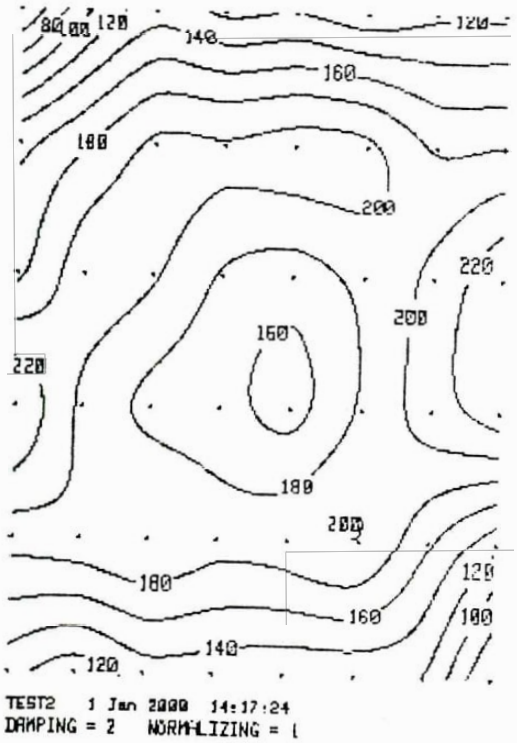


Figure 10

Contour map representing the data from Table 2 drawn on the 9836's video screen.

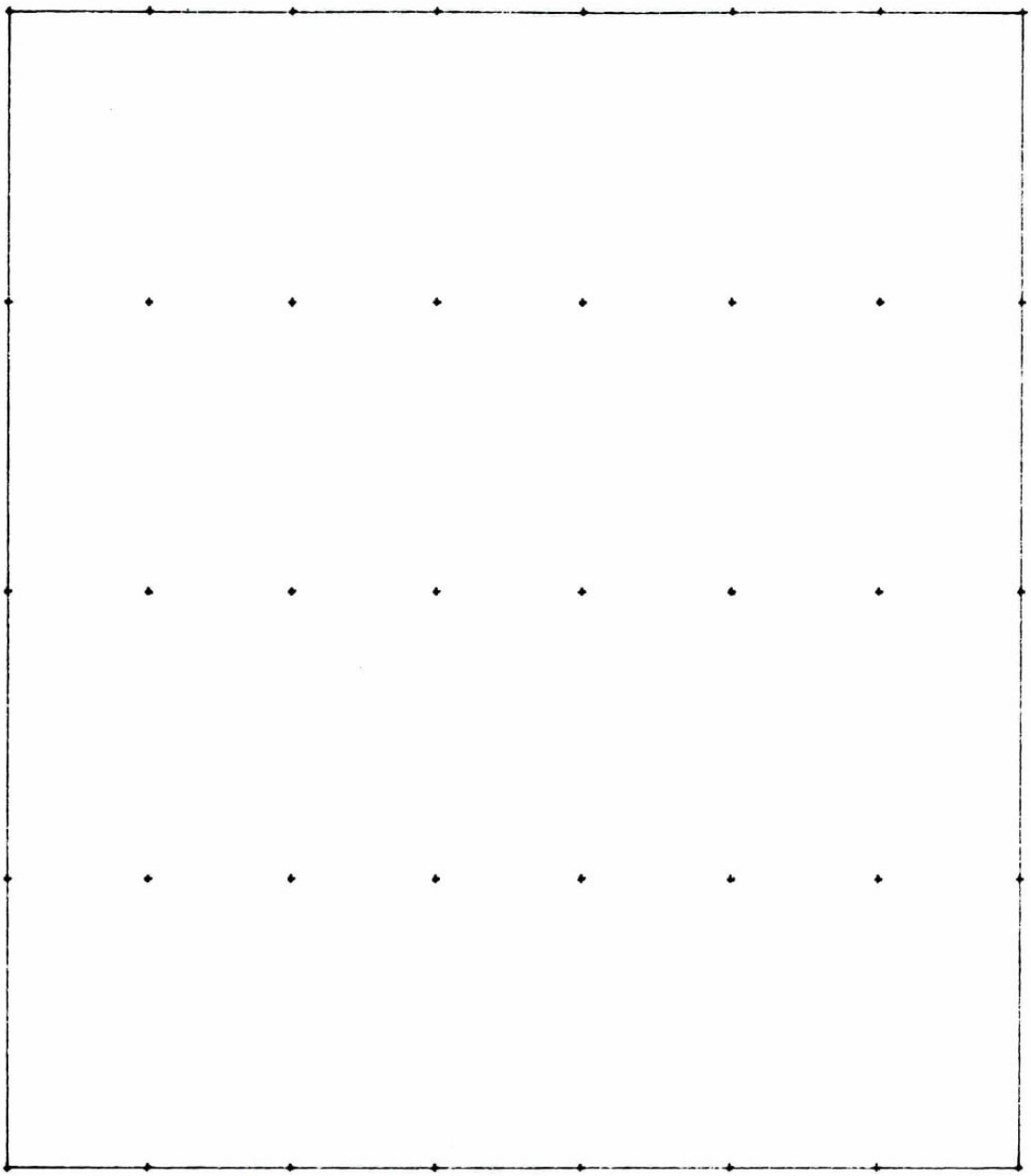


Figure 11 Grid defined in Table 1 shown by its grid points, with its boundary drawn and with its plot specifier (all default values of PLOT MENU # 1).

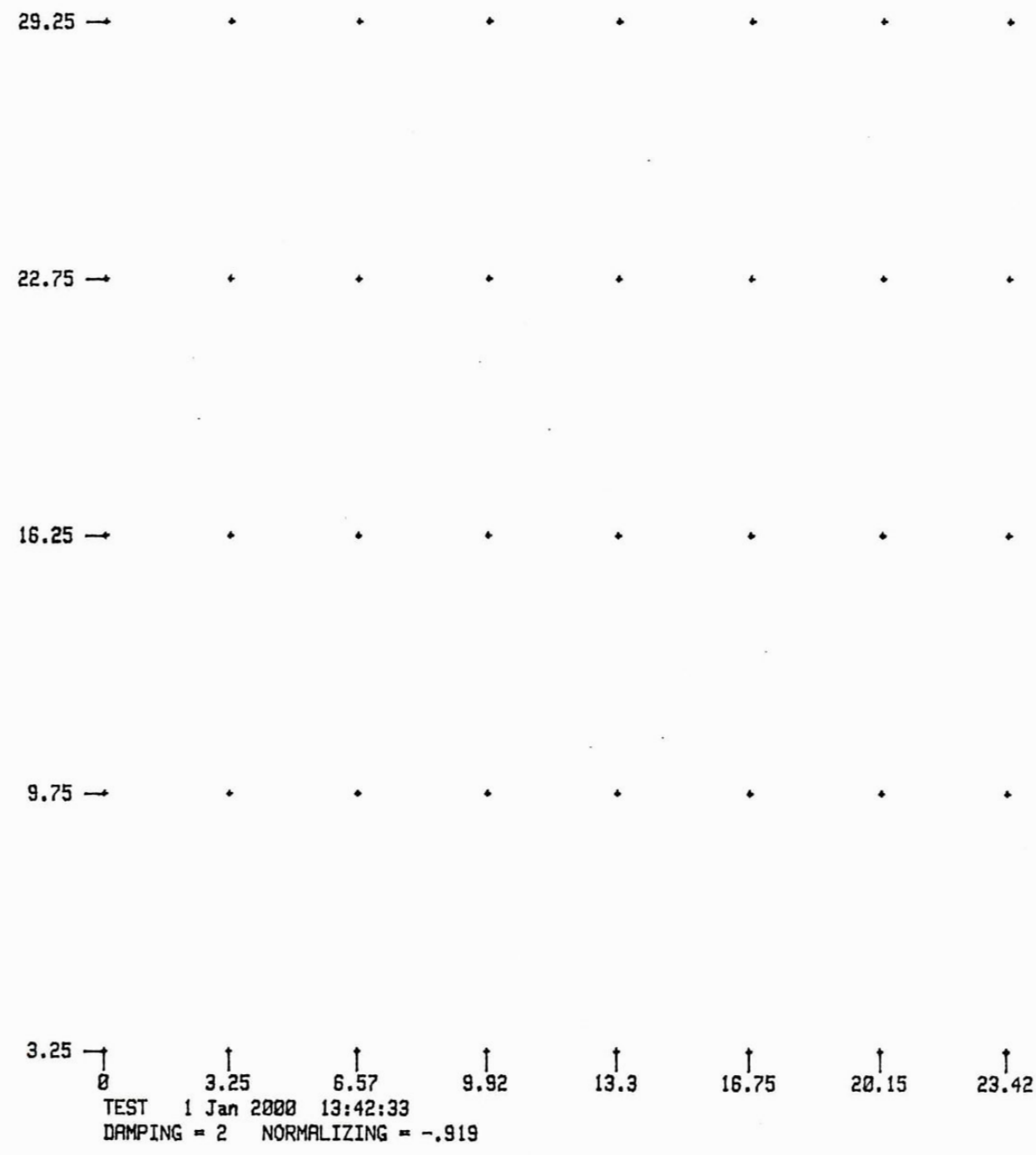


Figure 12 Grid defined in Table 1 shown by its grid points and labelled with its grid co-ordinates.

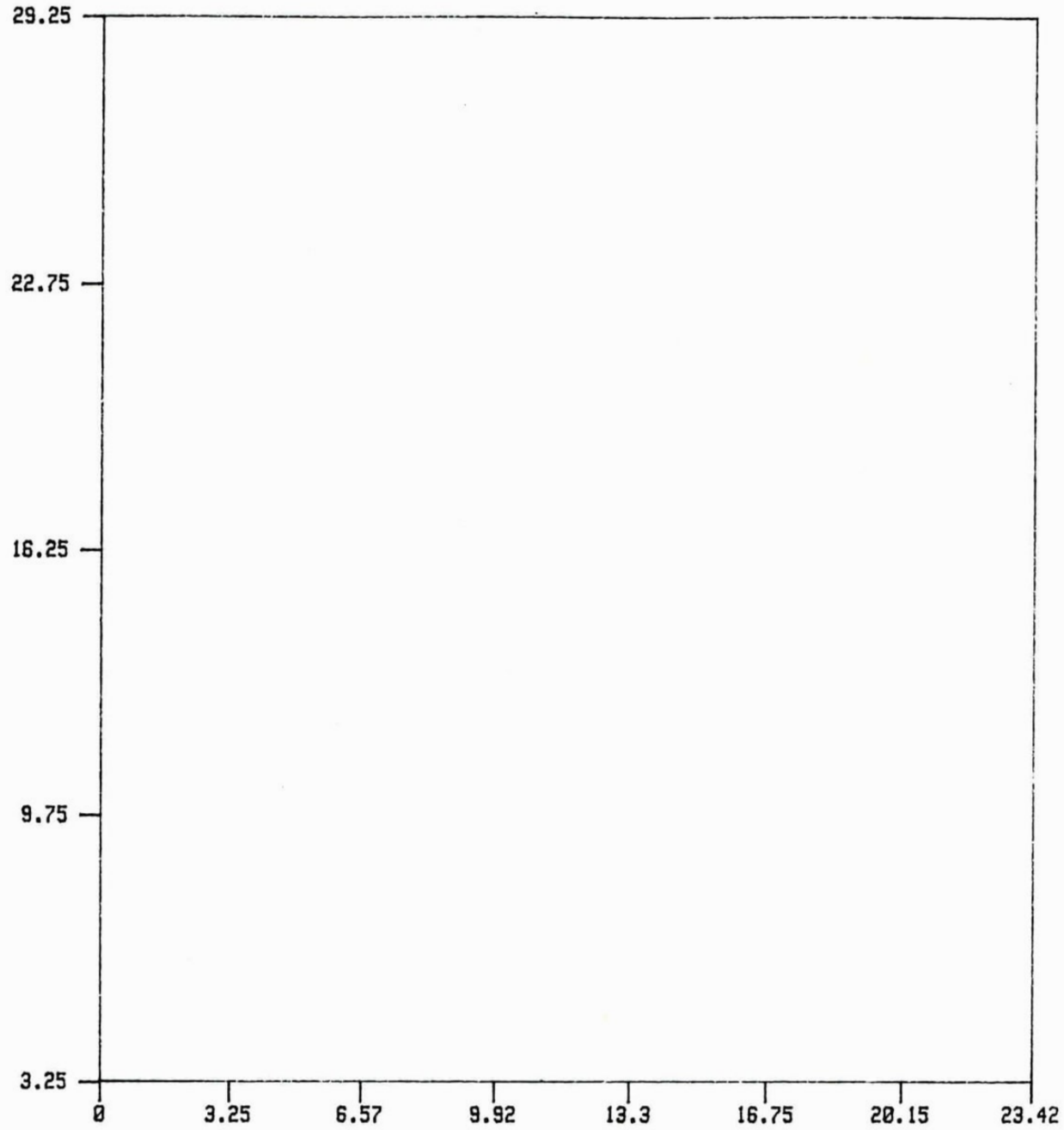


Figure 13 Grid defined in Table 1 displayed only by its frame and its labelled grid co-ordinates.

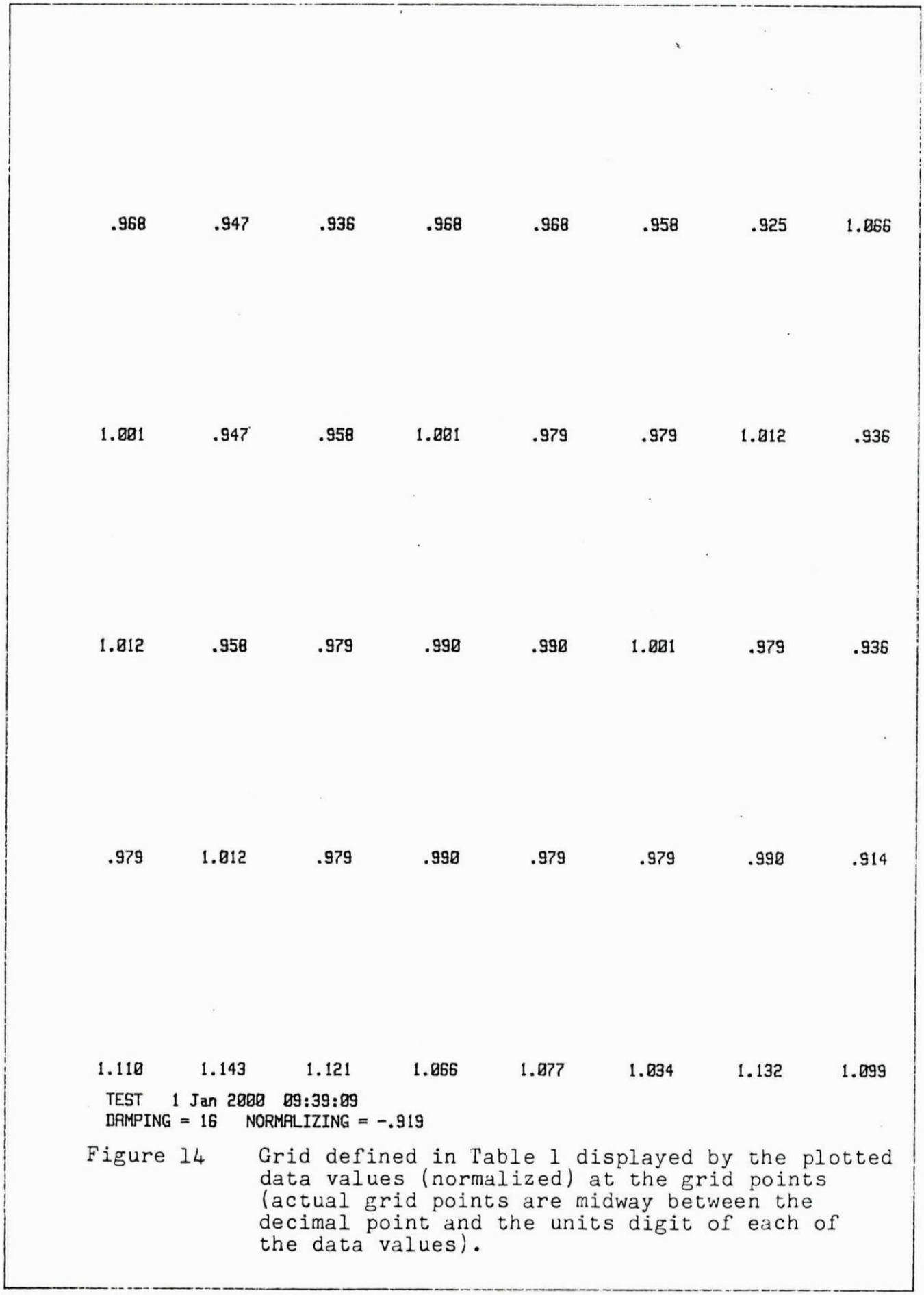
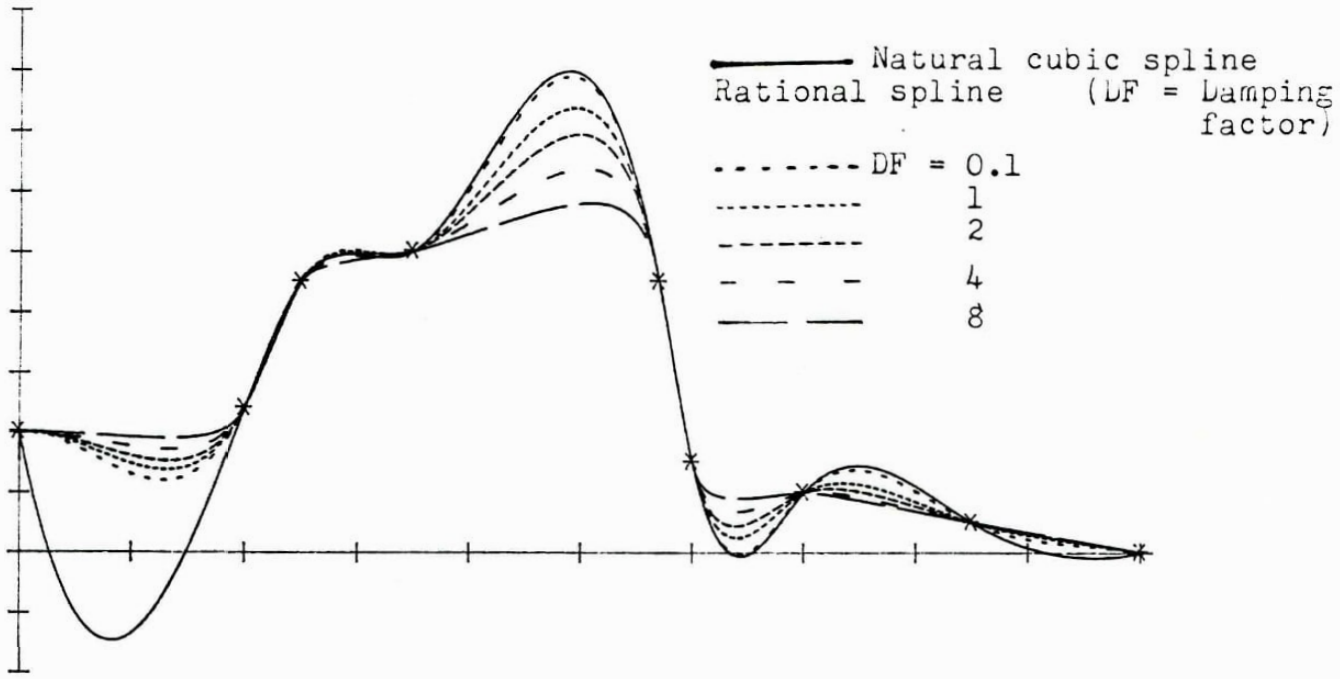


Figure 14 Grid defined in Table 1 displayed by the plotted data values (normalized) at the grid points (actual grid points are midway between the decimal point and the units digit of each of the data values).



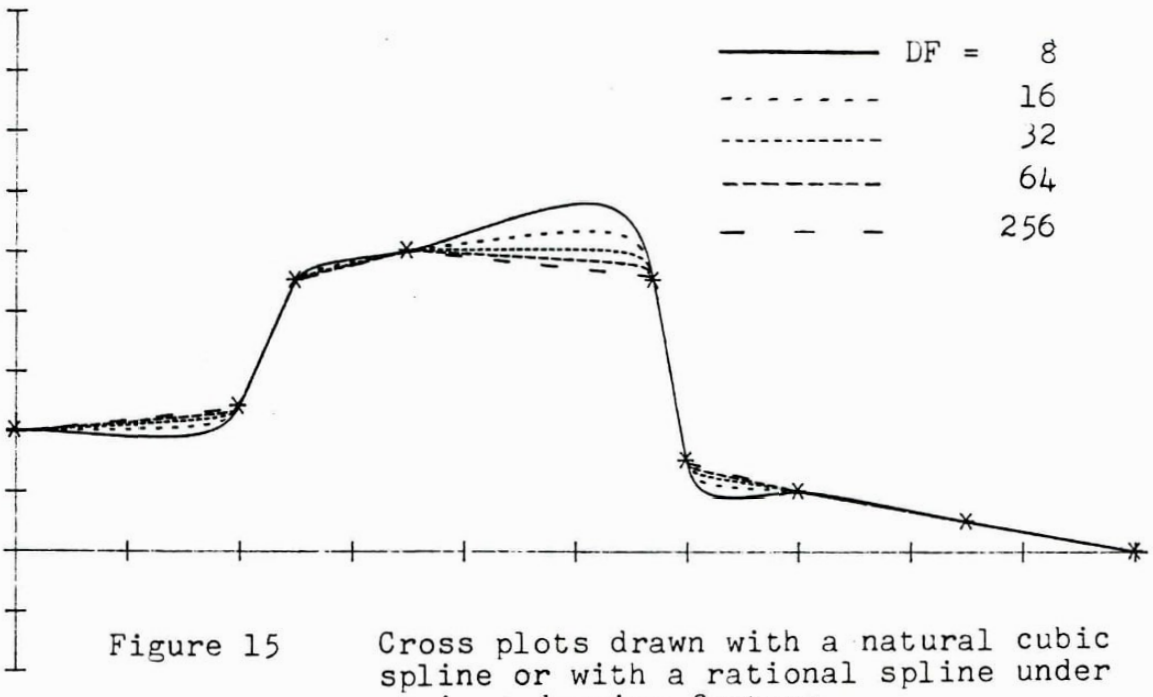


Figure 15 Cross plots drawn with a natural cubic spline or with a rational spline under various damping factors.

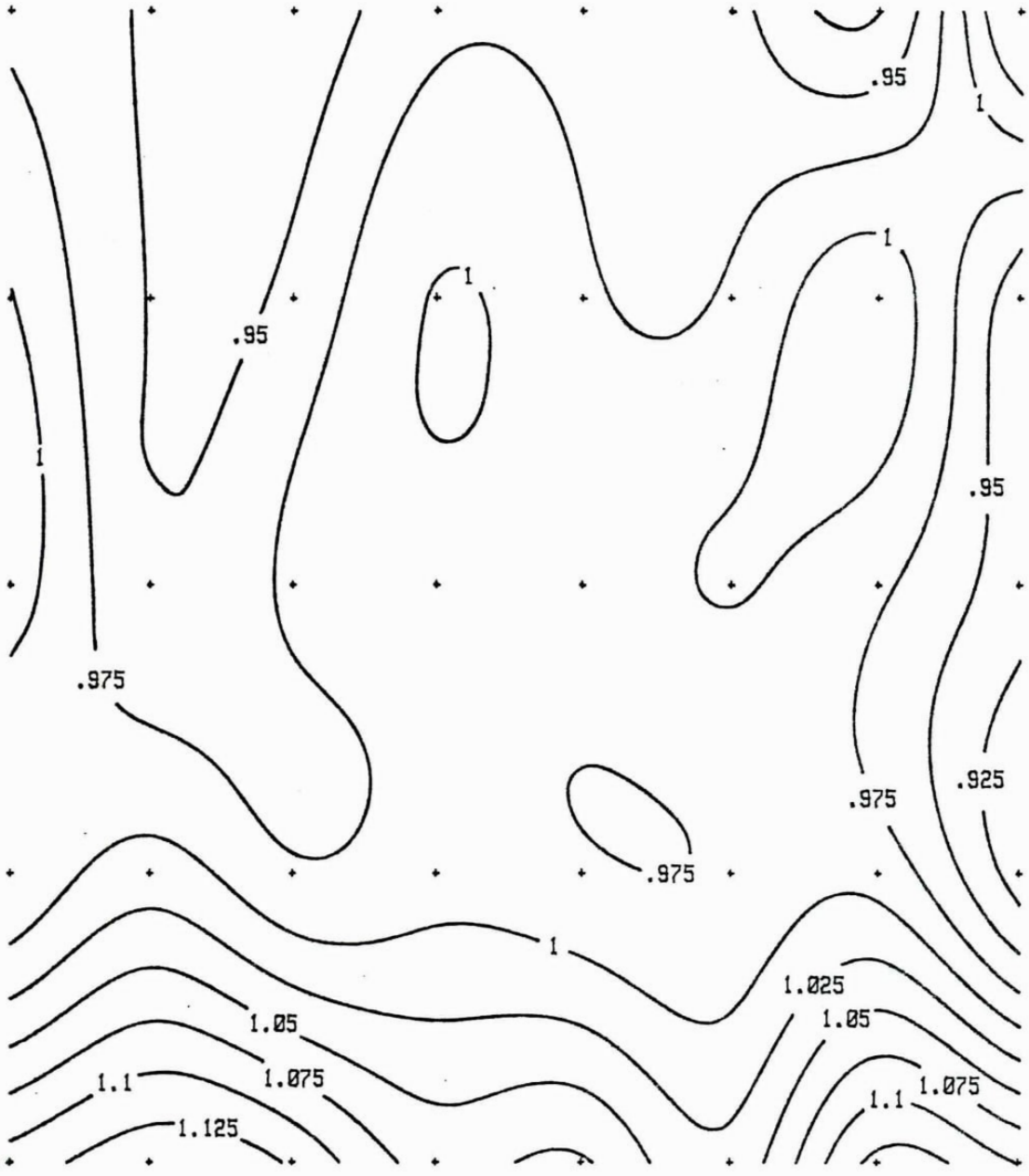


Figure 16 Contour plot of the data from Table 1 drawn with a damping factor of 0.1.

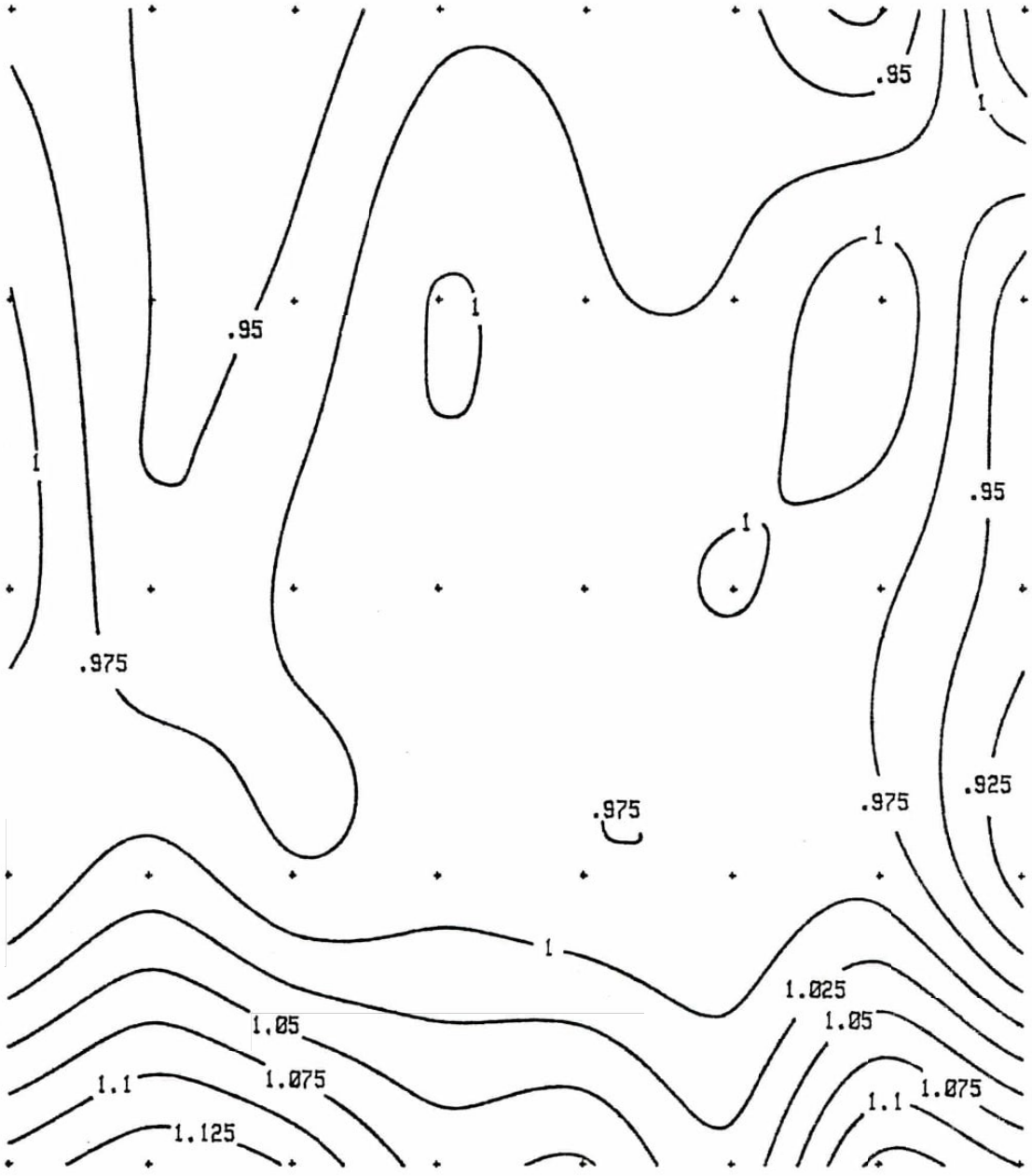


Figure 17 Contour plot of the data from Table 1 drawn with a damping factor of 1.0.

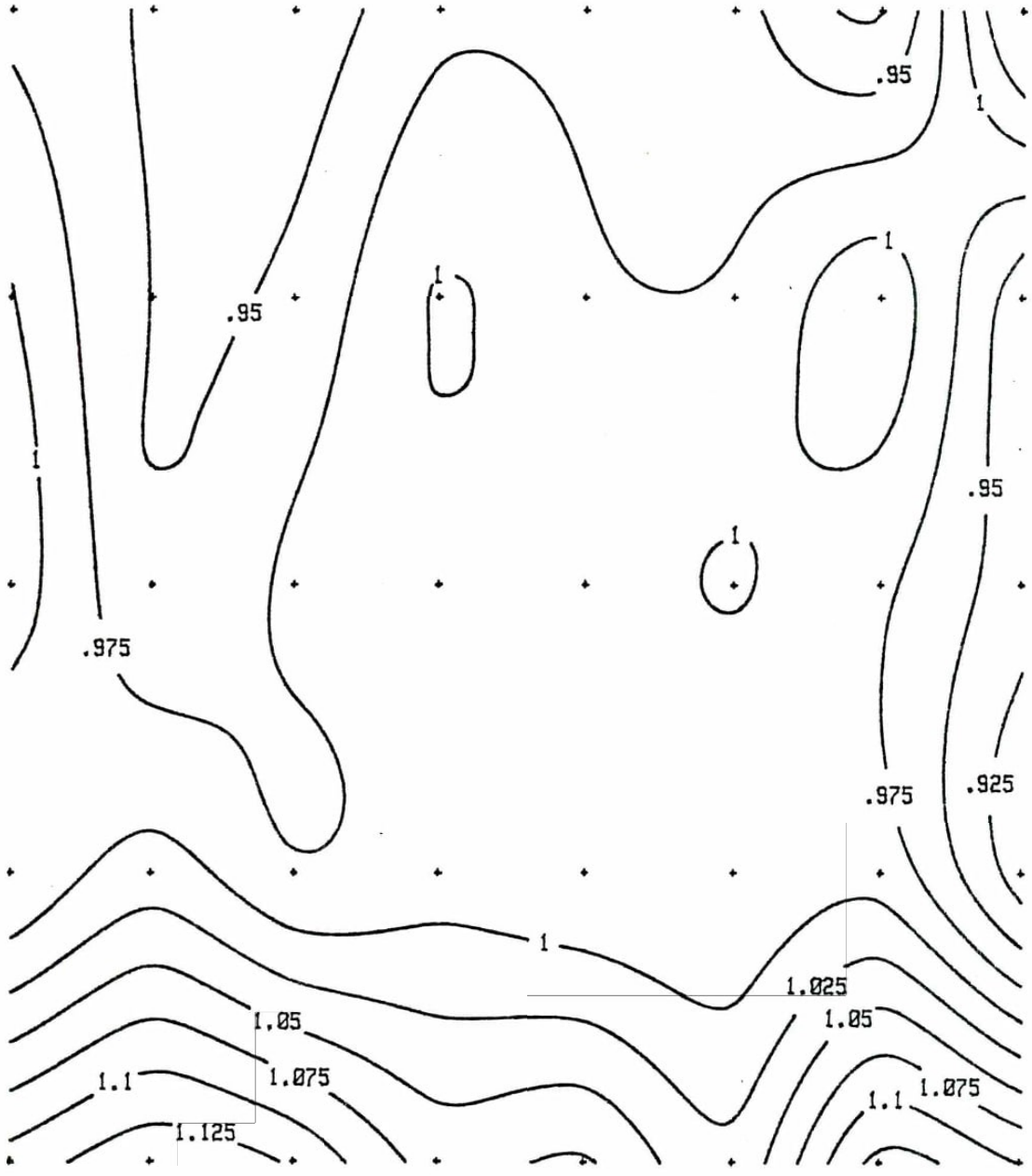


Figure 18 Contour plot of the data from Table 1 drawn with a damping factor of 2.0.

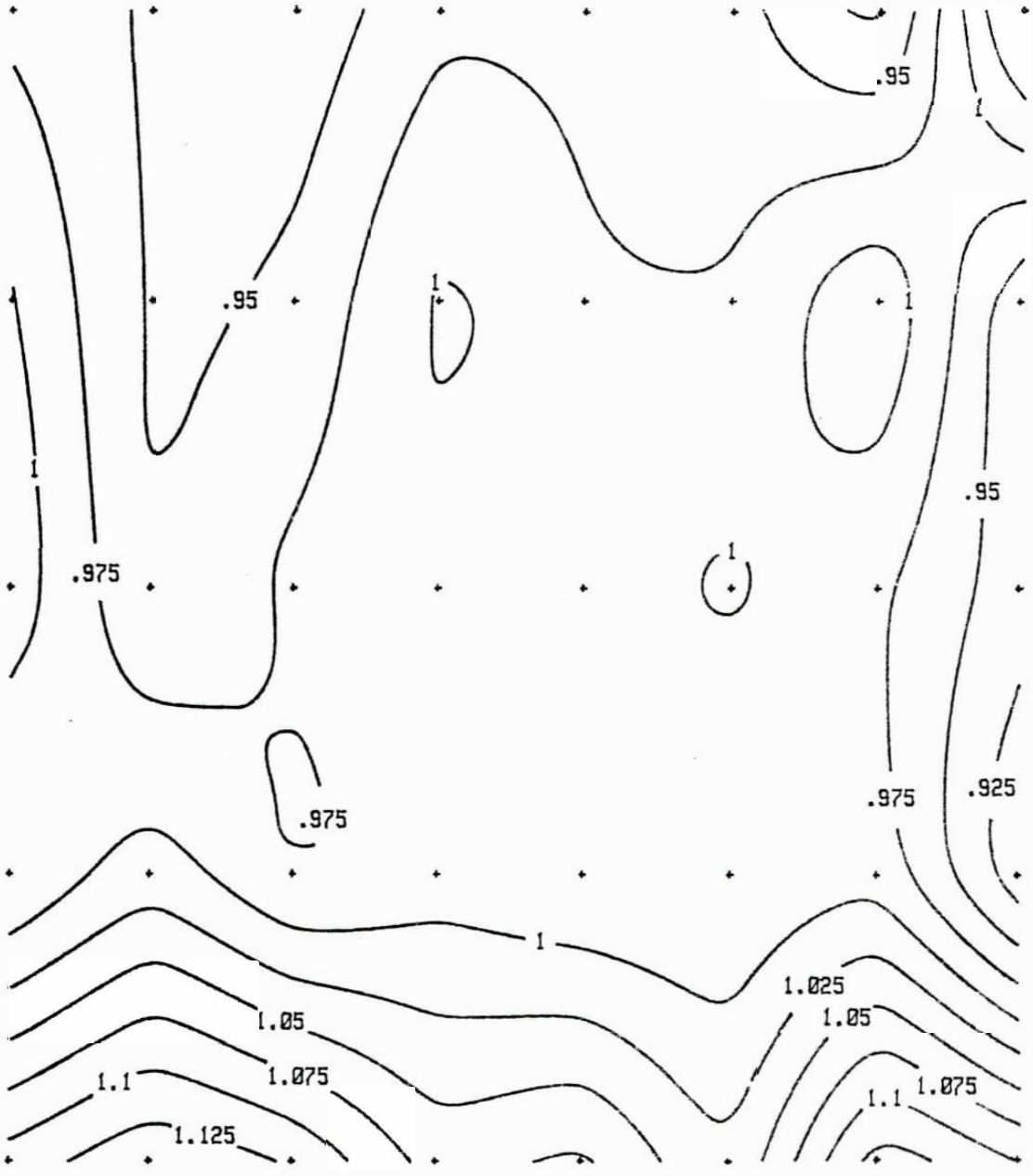


Figure 19 Contour plot of the data from Table 1 drawn with a damping factor of 4.0.

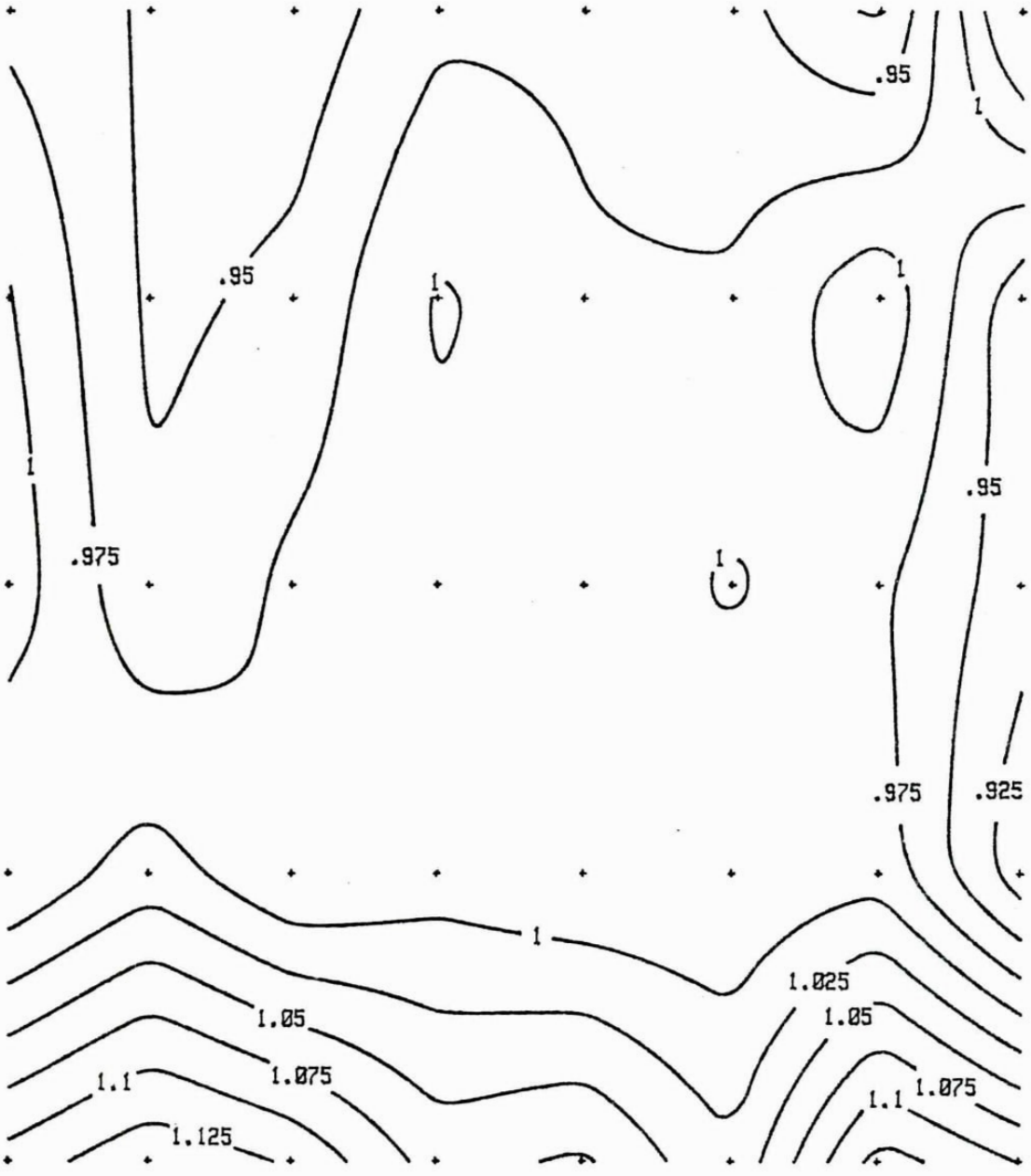


Figure 20 Contour plot of the data from Table 1 drawn with a damping factor of 8.0.

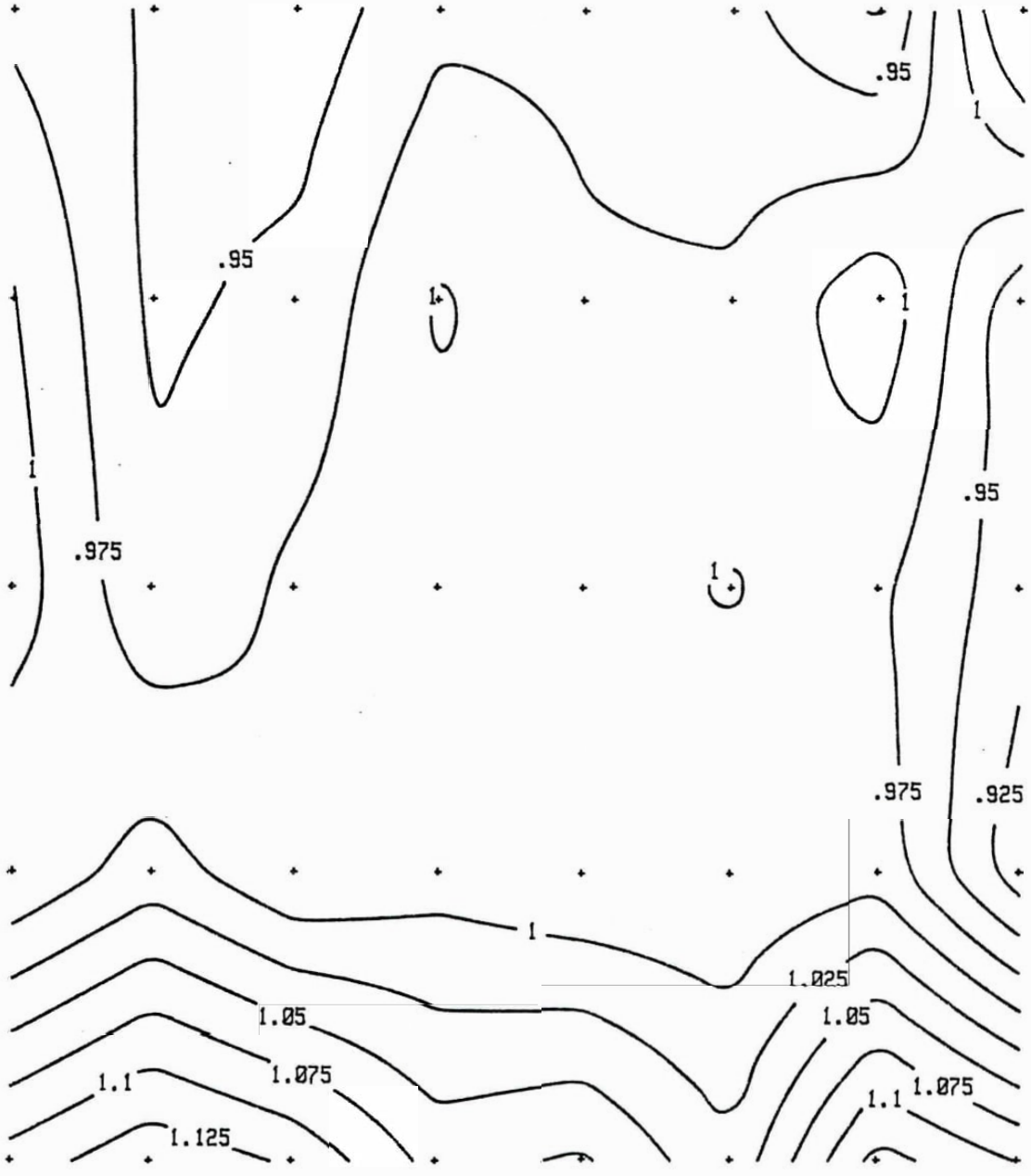


Figure 21 Contour plot of the data from Table 1 drawn with a damping factor of 16.0.

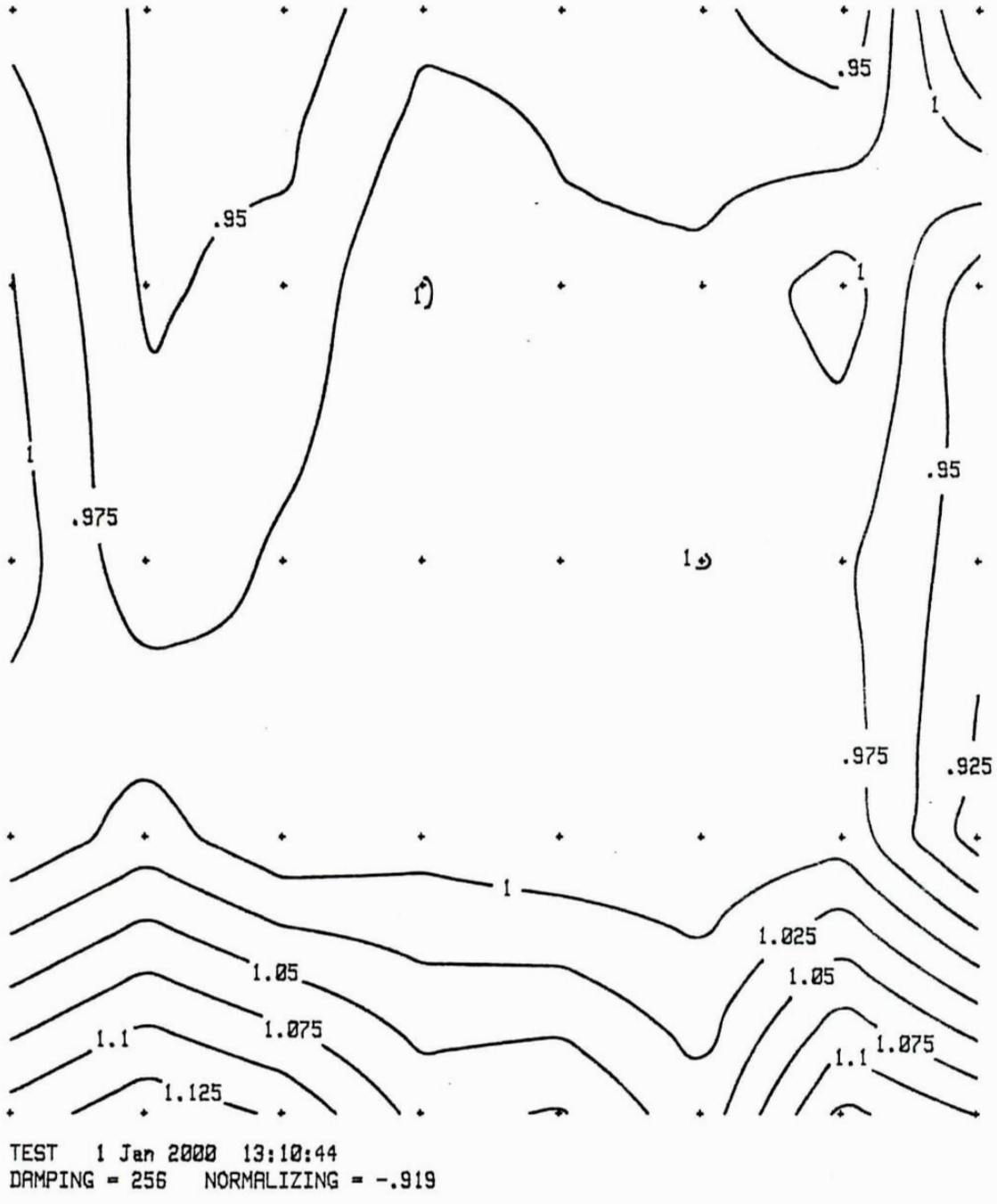


Figure 22 Contour plot of the data from Table 1 drawn with a damping factor of 256.

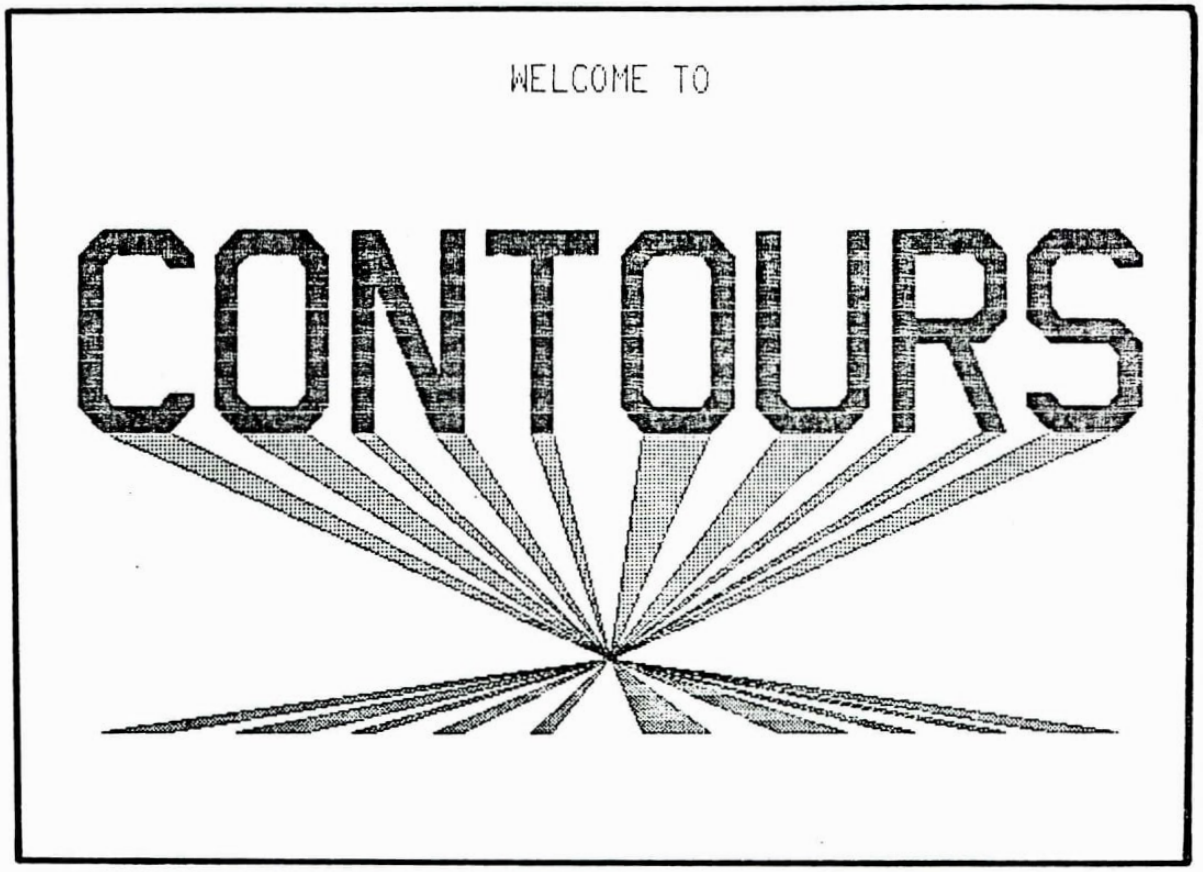


Figure 23 Initial screen displsy upon a program run.

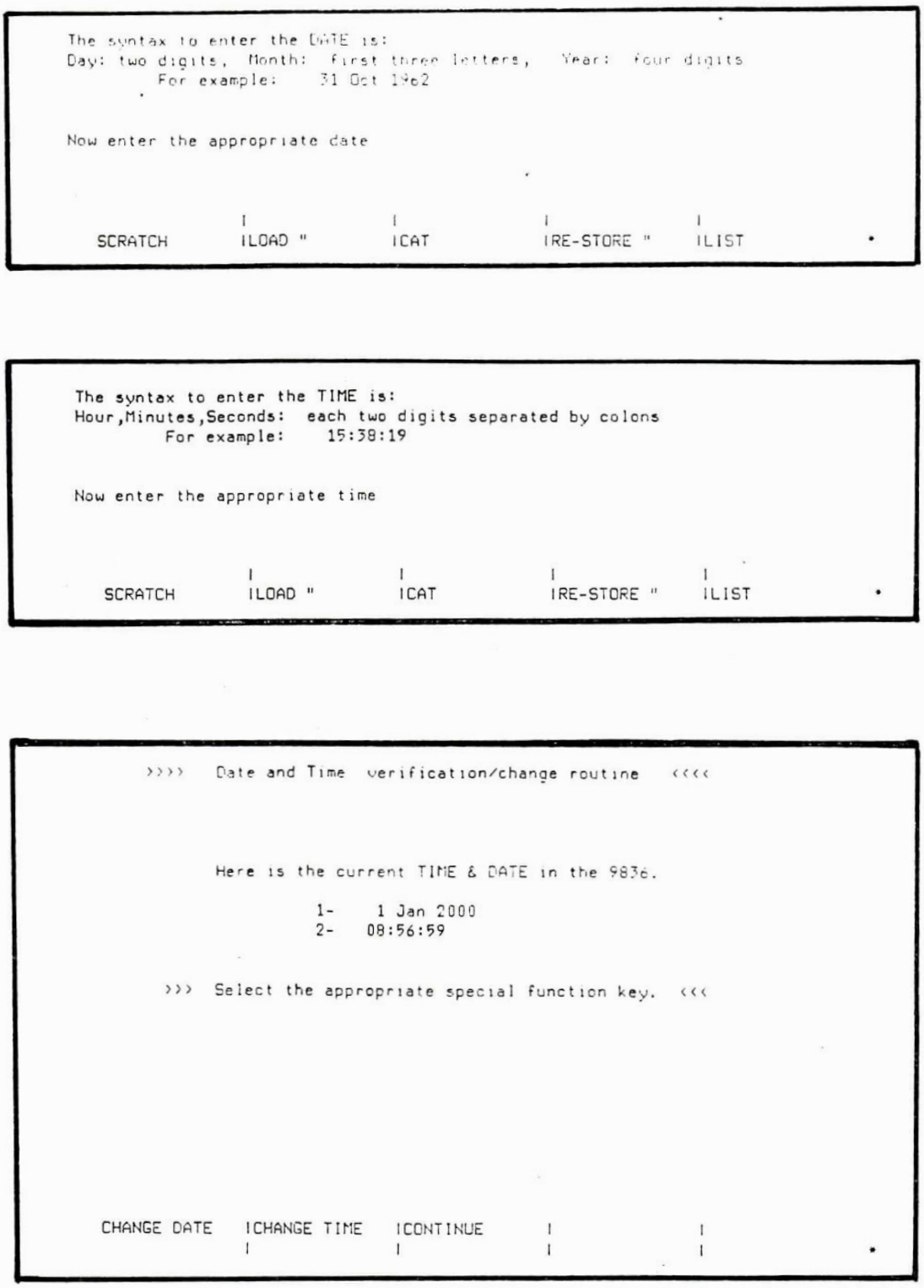


Figure 24 Video displays prompting the user to enter, verify and change the time and date within the computer's memory.

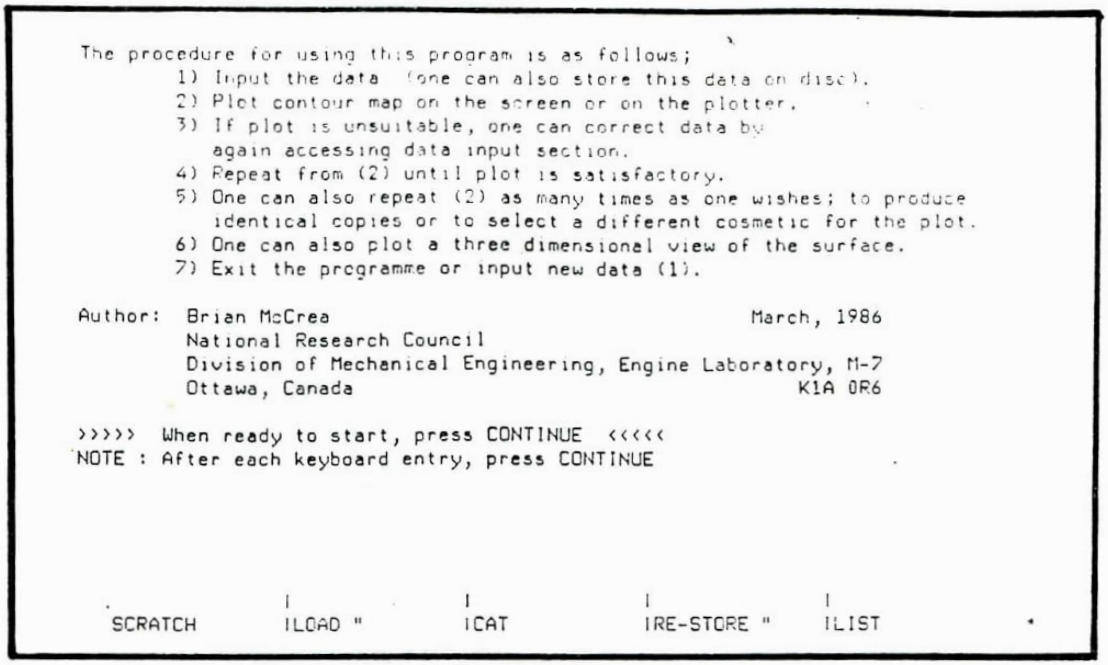


Figure 25 Informational display instructing the user of the capabilities of the program.

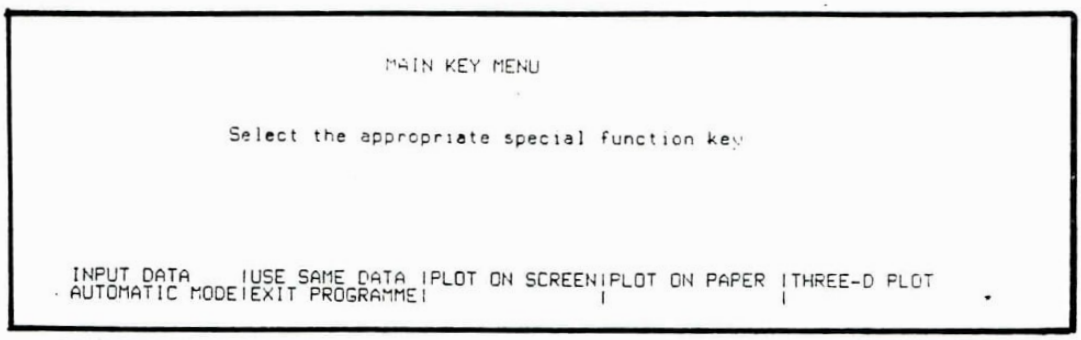


Figure 26 MAIN KEY MENU displaying its special function key selection of options. This is the major menu for the entire program.

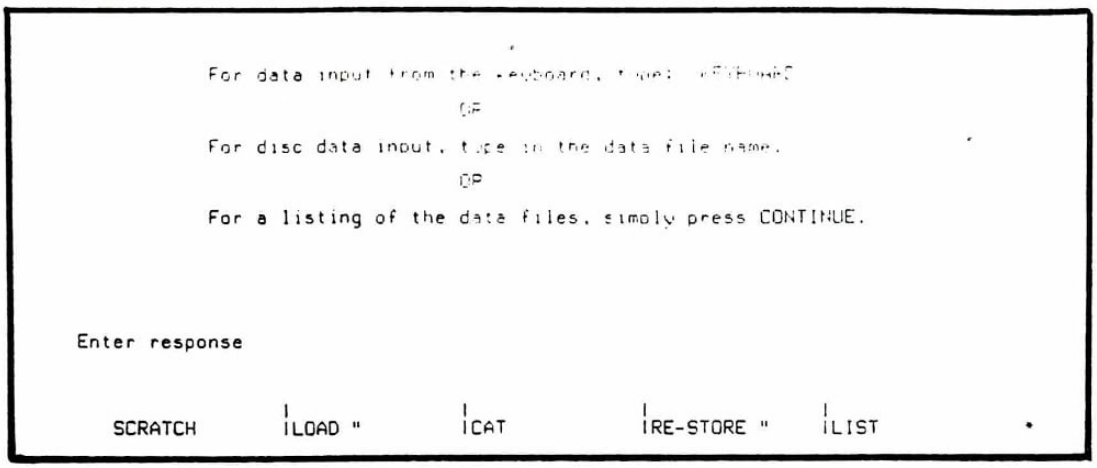


Figure 27 Start of Data Management routine prompting the user to enter new data or to retrieve data from a floppy disc.

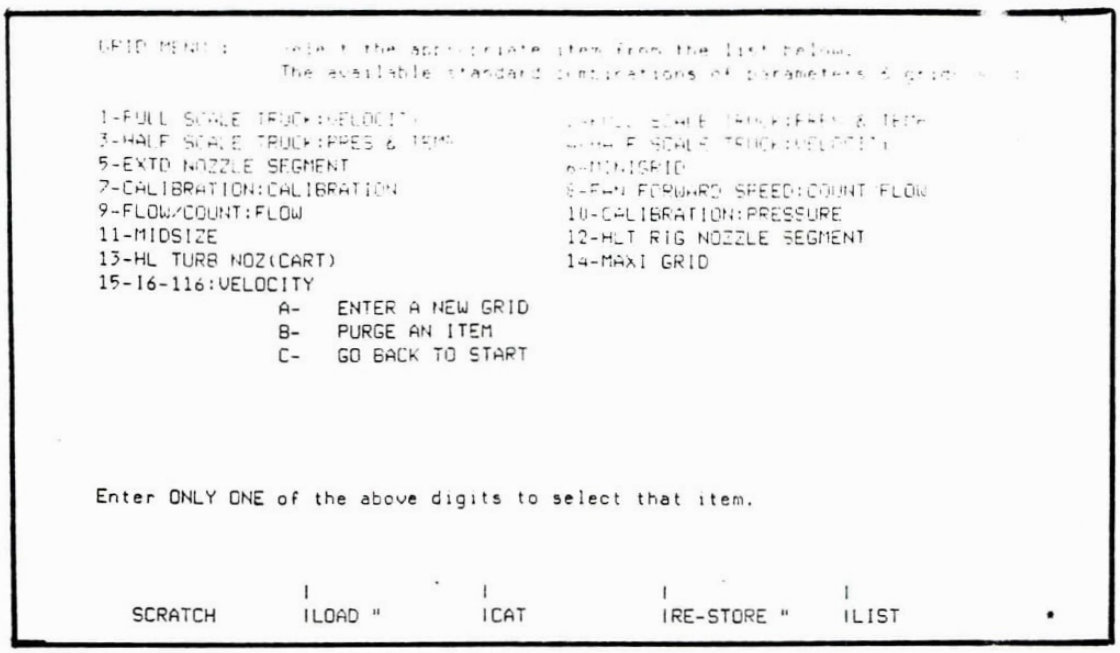


Figure 28 Typical GRID MENU listing existing grid layouts.

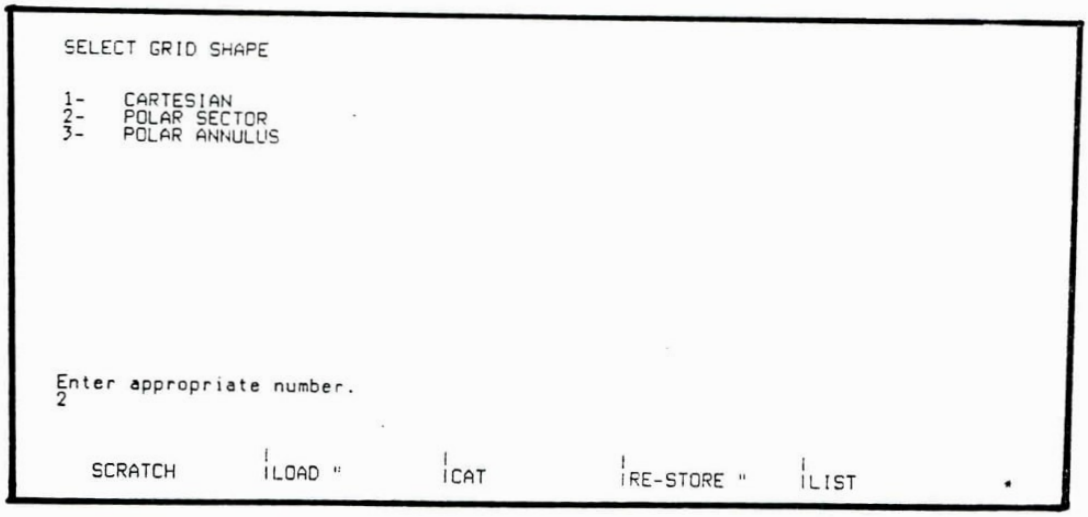


Figure 29 Start of entry of a new grid layout. Display prompting the selection of one of the basic grid shapes.

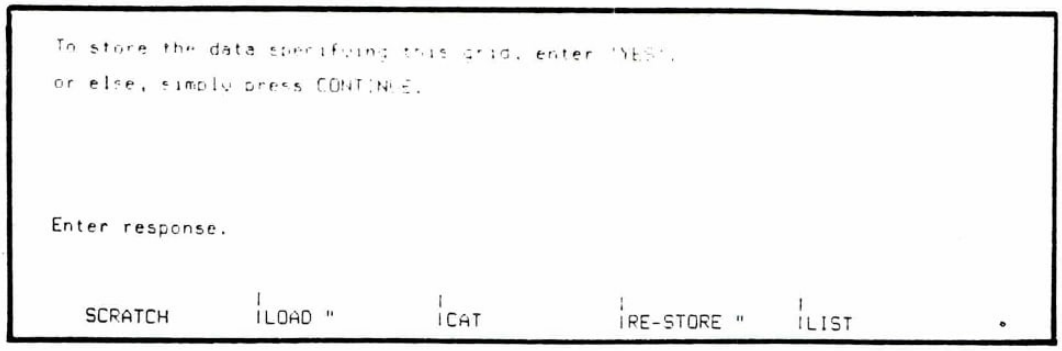
```
how many CIRCUMFERENTIAL - (ANGLE) coordinates are there?
?

   SCRATCH        |LOAD "        |CAT          |RE-STORE "    |LIST          .
```

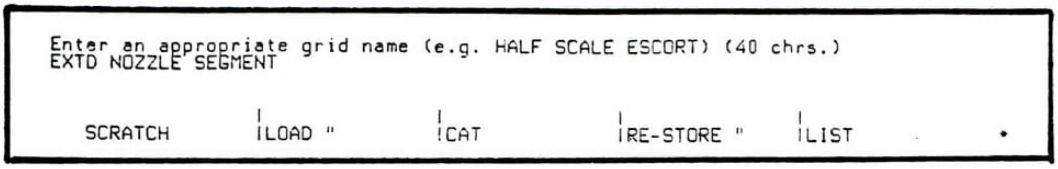
```
how many RADIAL - (R) coordinates are there?
?

   SCRATCH        |LOAD "        |CAT          |RE-STORE "    |LIST          .
```

```
1 -    X( 1 )= 0
2 -    X( 2 )= 2.57
3 -    X( 3 )= 5.14
4 -    X( 4 )= 7.71
5 -    X( 5 )= 10.29
6 -    X( 6 )= 12.86
7 -    X( 7 )= 15.43
8 -    X( 8 )= 18
9 -    X( 9 )= 20.57
10 -    X( 10 )= 23.14
11 -    X( 11 )= 25.71
12 -    X( 12 )= 28.29
13 -    X( 13 )= 30.86
14 -    X( 14 )= 33.43
15 -    X( 15 )= 36
16 -    X( 16 )= 38.57
17 -    X( 17 )= 41.14
If you wish to extend, or change the above, enter line number.

   SCRATCH        |LOAD "        |CAT          |RE-STORE "    |LIST          .
```

```
NOW ENTER THE  12  RADIAL - (R) GRID POSITIONS
 1 -      Y( 1 )= 7.79
 2 -      Y( 2 )= 7.927
 3 -      Y( 3 )= 8.229
 4 -      Y( 4 )= 8.481
 5 -      Y( 5 )= 8.733
 6 -      Y( 6 )= 8.985
 7 -      Y( 7 )= 9.237
 8 -      Y( 8 )= 9.489
 9 -      Y( 9 )= 9.741
 10 -      Y( 10 )= 9.993
 11 -      Y( 11 )= 10.245
 12 -      Y( 12 )= 10.371


If you wish to extend, or change the above, enter line number.

   SCRATCH        |LOAD "        |CAT          |RE-STORE "    |LIST          .
```

Figure 30 Displays prompting the entry of the number of grid lines and their co-ordinates, establishing the new grid layout. Shown display pertains to a polar plot. For a cartesian grid the prompts would include the terms 'transverse'and 'longitudinal' rather than 'circumferential' and 'radial'.





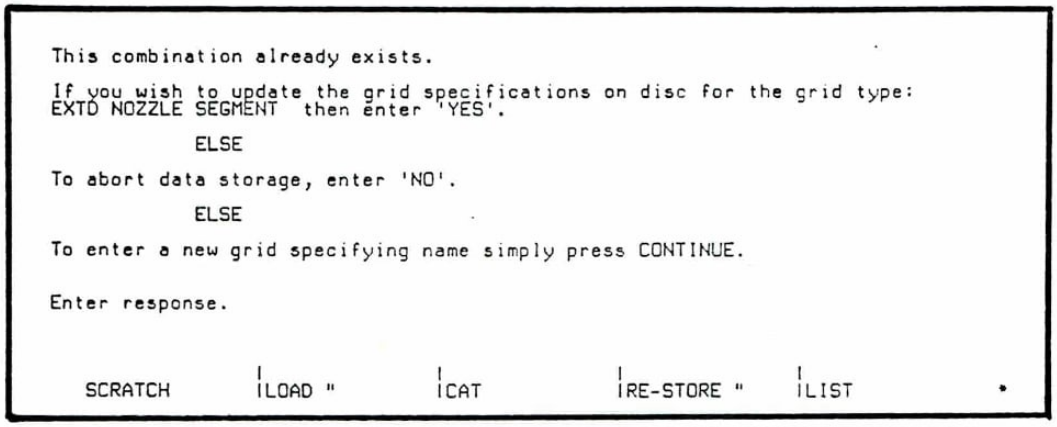


Figure 31 Displays pertaining to the storage of an entered grid. The bottom display informs the user that his chosen grid name already exists within the GRID MENU and allows the user to take an appropriate action.

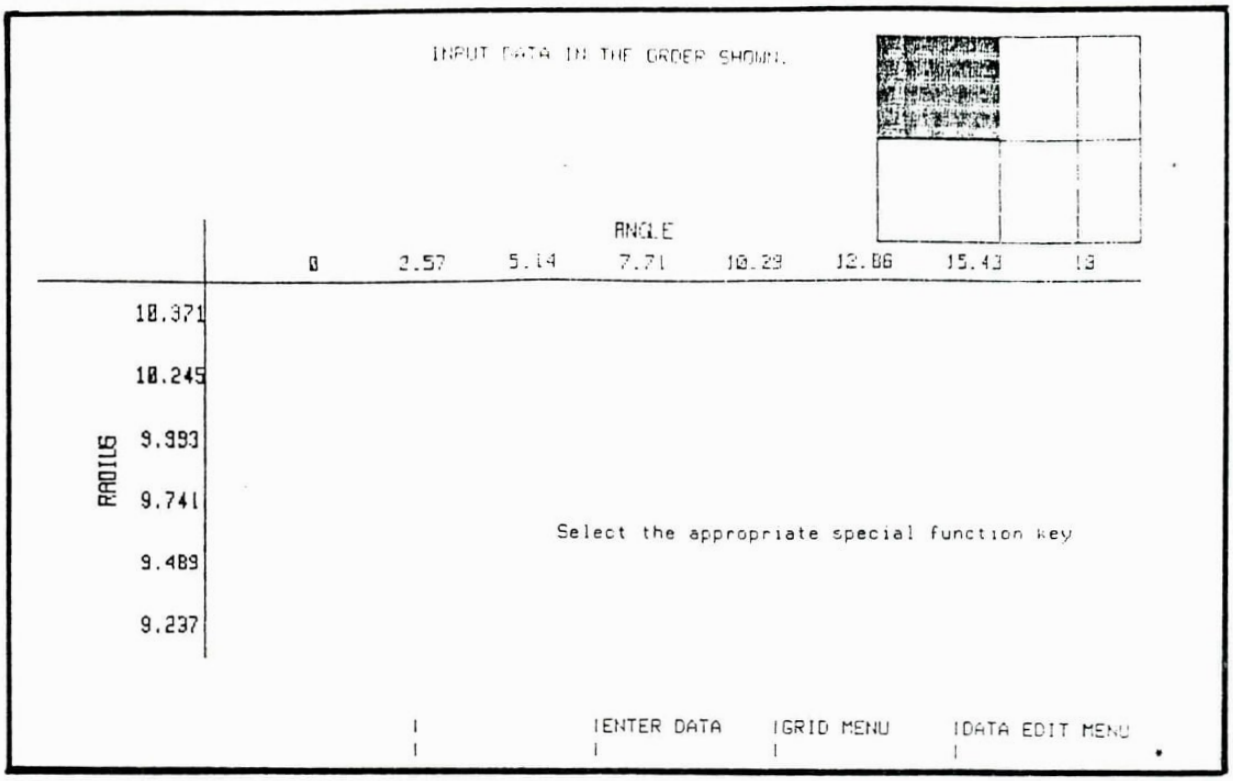


Figure 32 Special function key menu allowing the user to enter a data set, to return to the GRID MENU or to attain the DATA EDITING MENU. These last two options allow the user to verify existing grids from the GRID MENU without having to enter a whole data set.

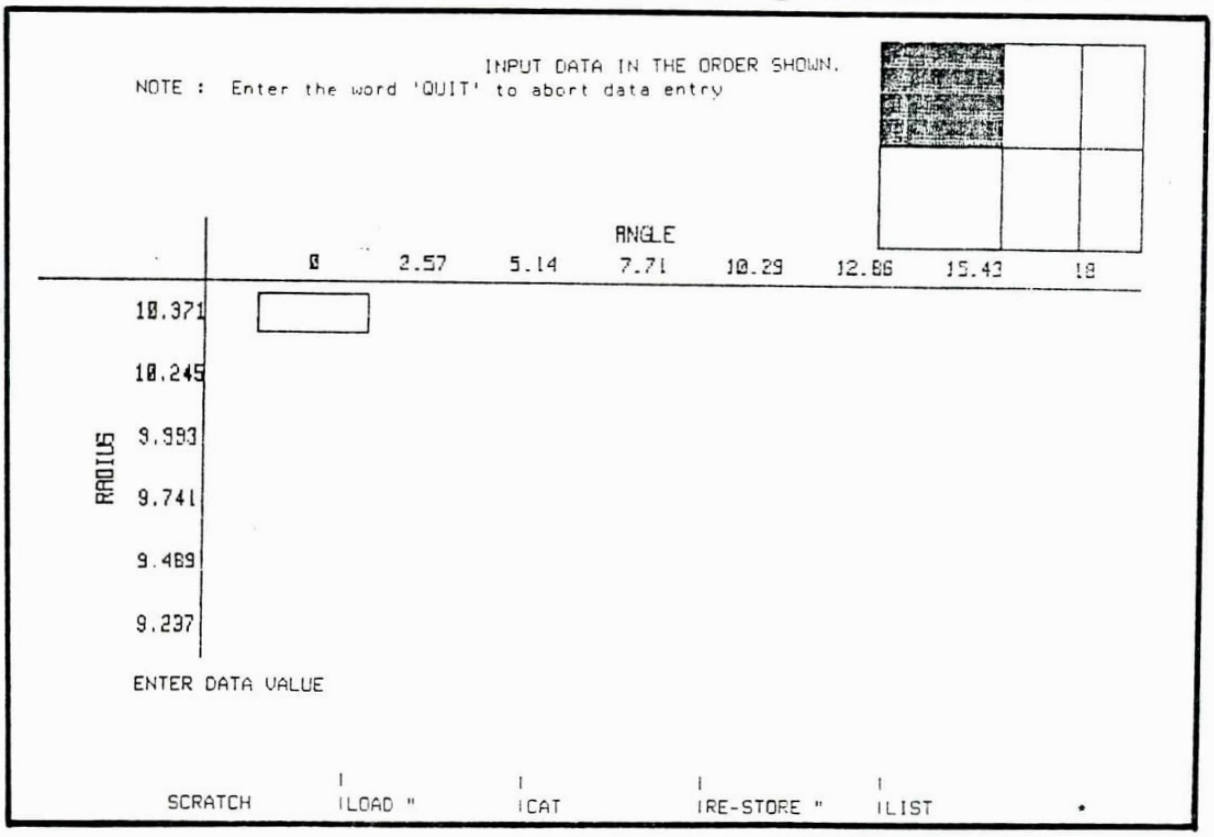


Figure 33 Start of the entry of a data set. The entered item will be placed at the spot denoted by the position of the box.

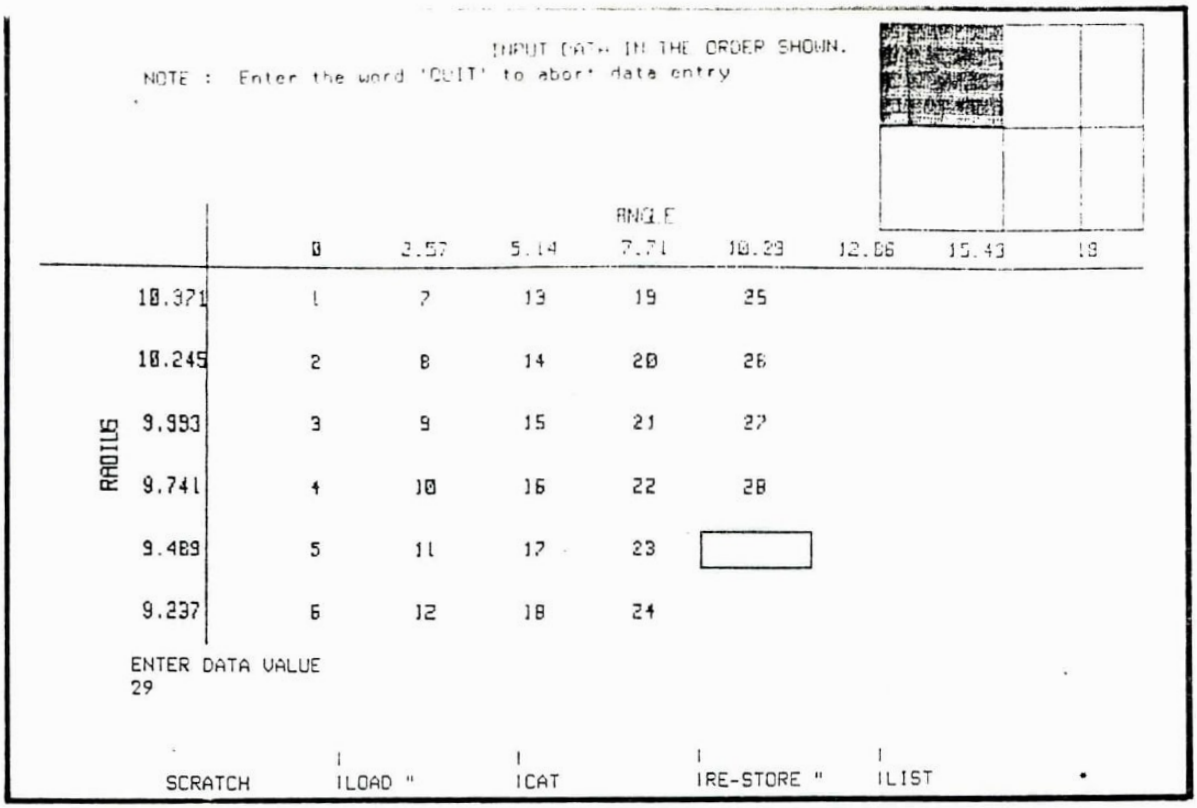


Figure 34 Continuation of data entry. The displayed items indicate the order in which data is to be entered.

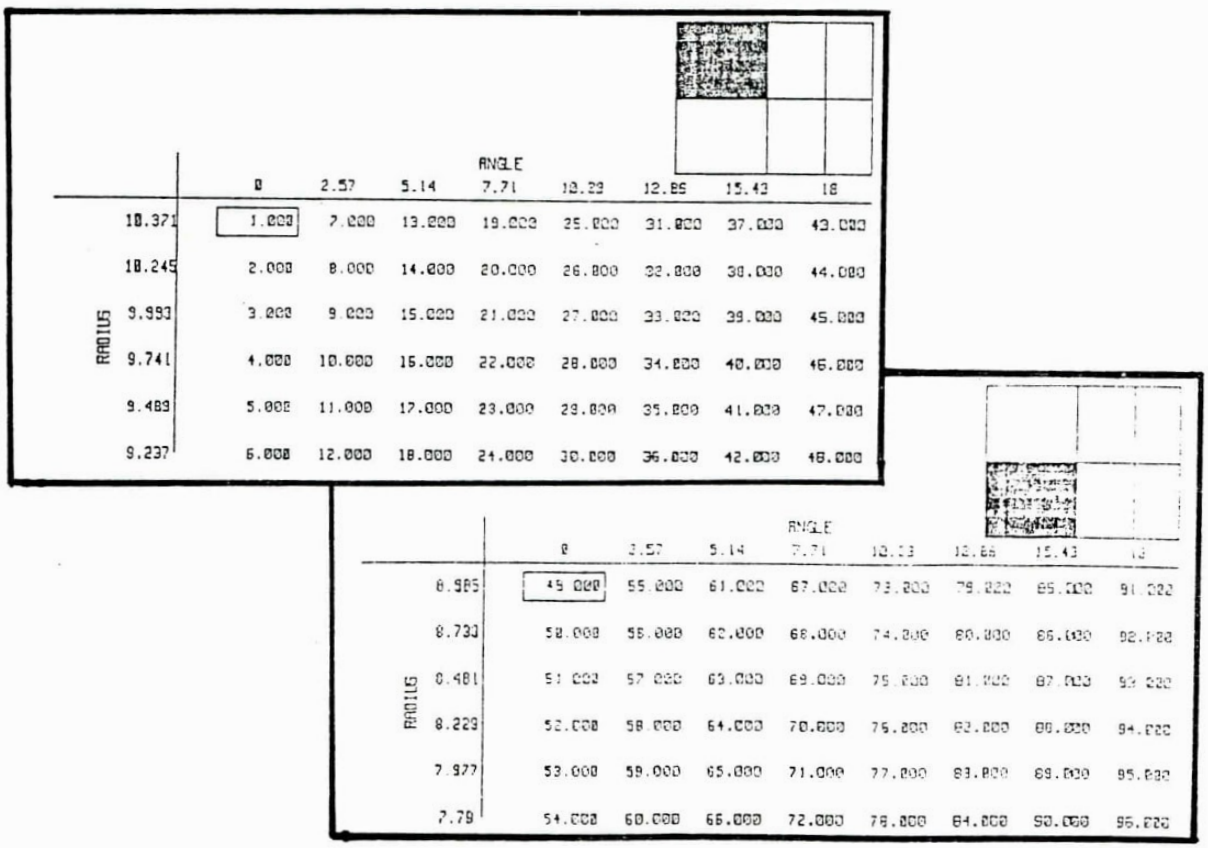


Figure 35 Display of the order in which data is entered and of the order in which the data blocks are selected to be filled with data.

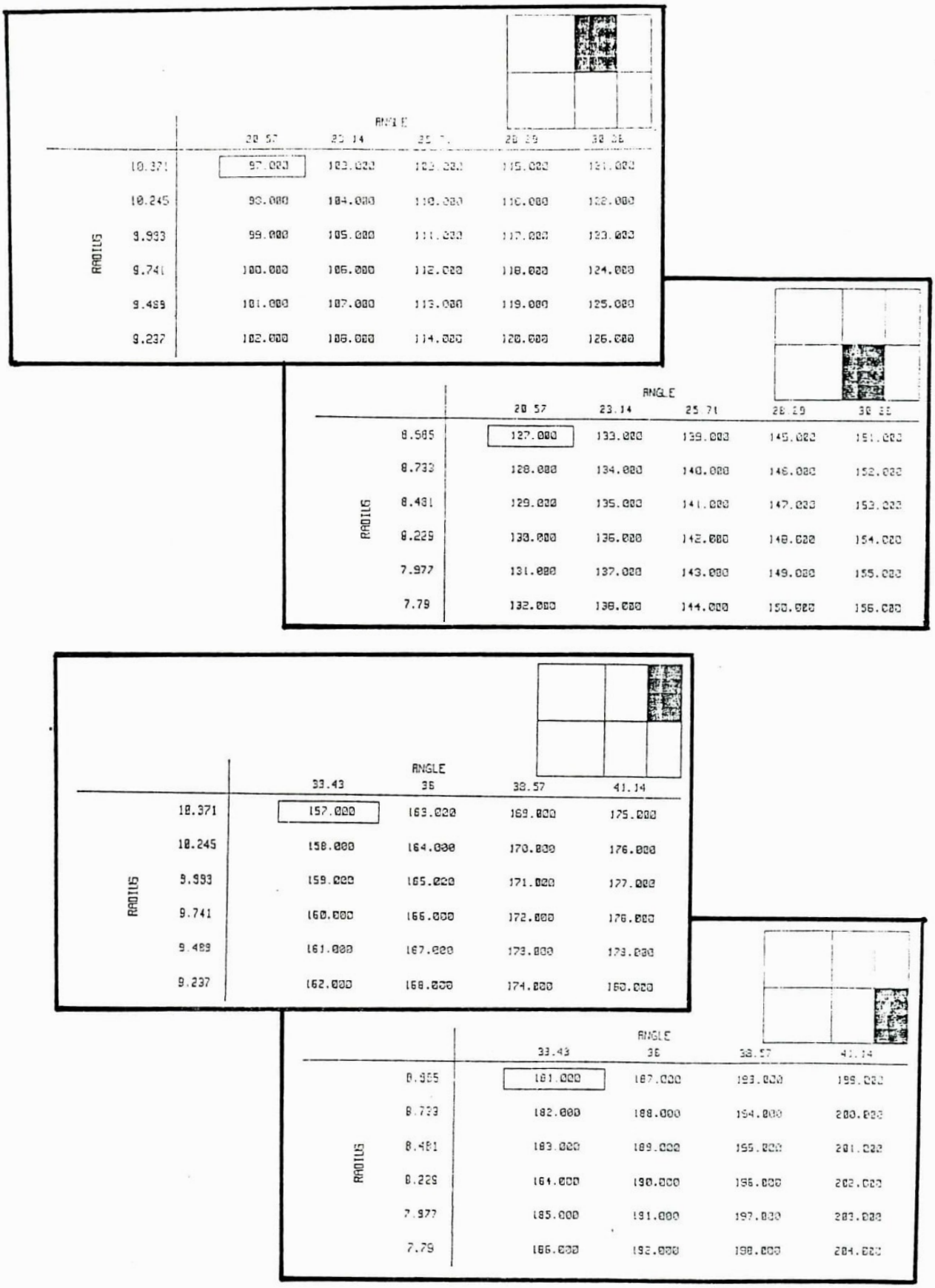


Figure 36 Display of the order in which data is entered and of the order in which the data blocks are selected to be filled with data. The data set shown above is also displayed in Table 5.

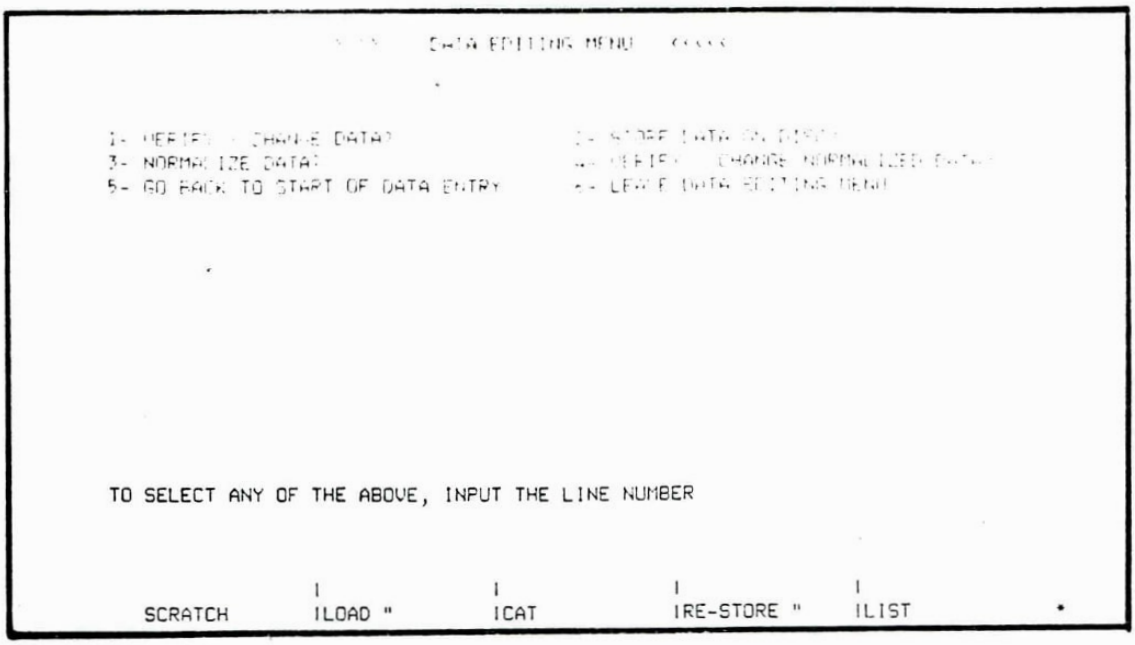


Figure 37 DATA EDITING MENU

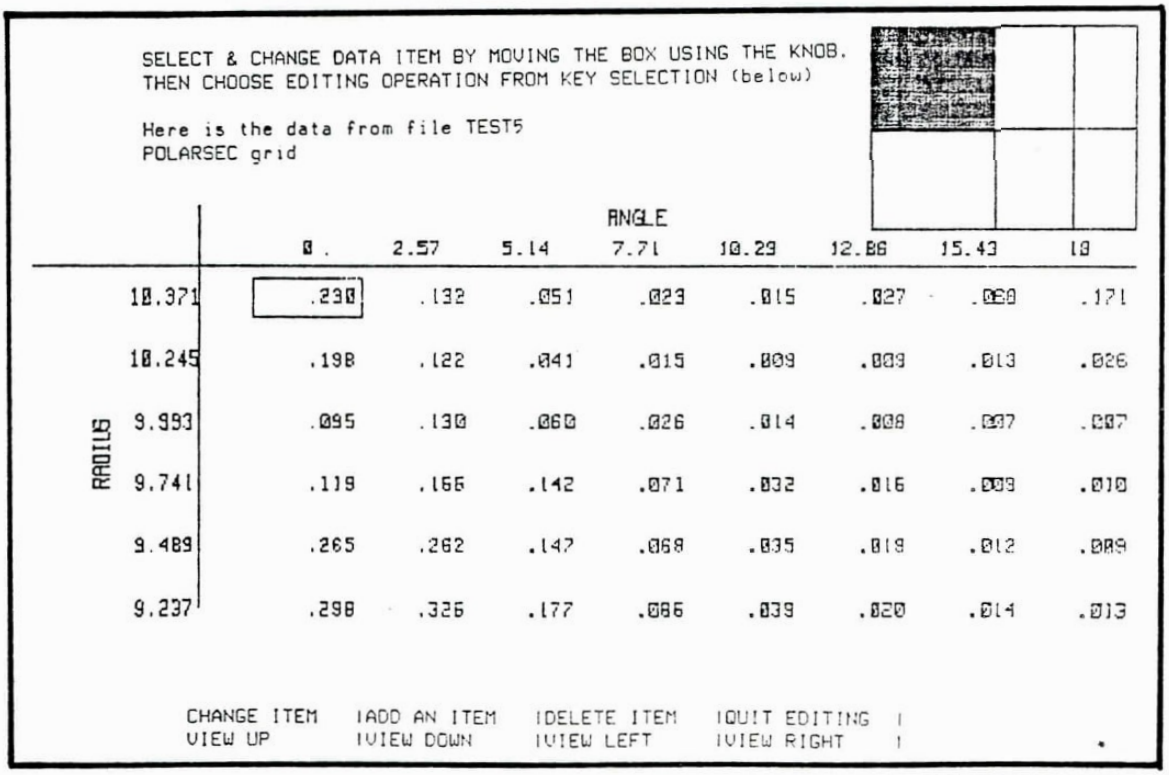


Figure 38 DATA EDITING MENU. For smaller data sets, not all of the viewing keys may be present. For a data set with only one data block, the graphic representation of the data set (upper right) will also be omitted.

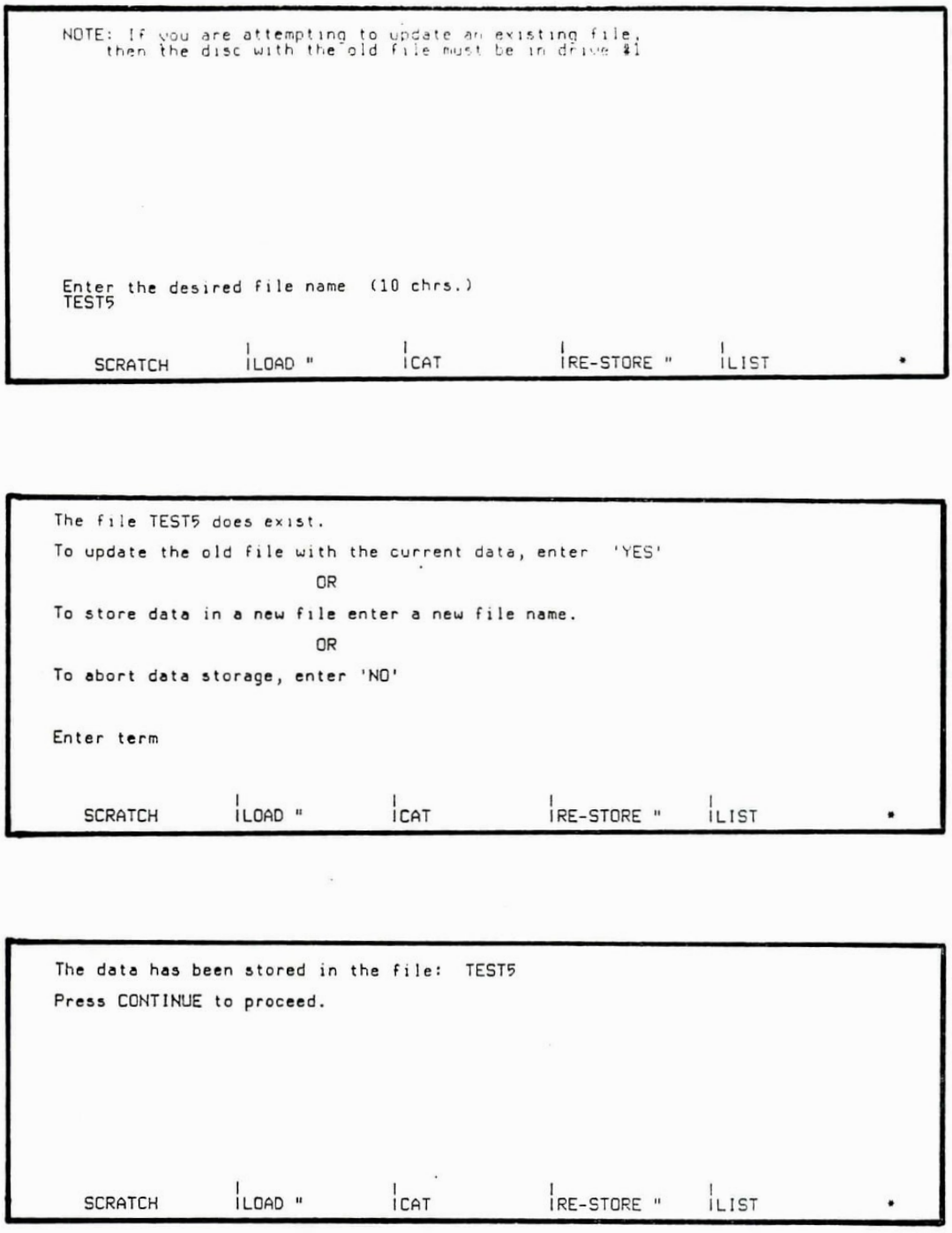


Figure 39 Displays pertaining to the storage of a data set. The middle display informs the user that the specified file name already exists on the floppy disc inserted in drive #1 and allows him to take an appropriate action. Once the data set has been stored the computer informs the user of this fact (bottom display).

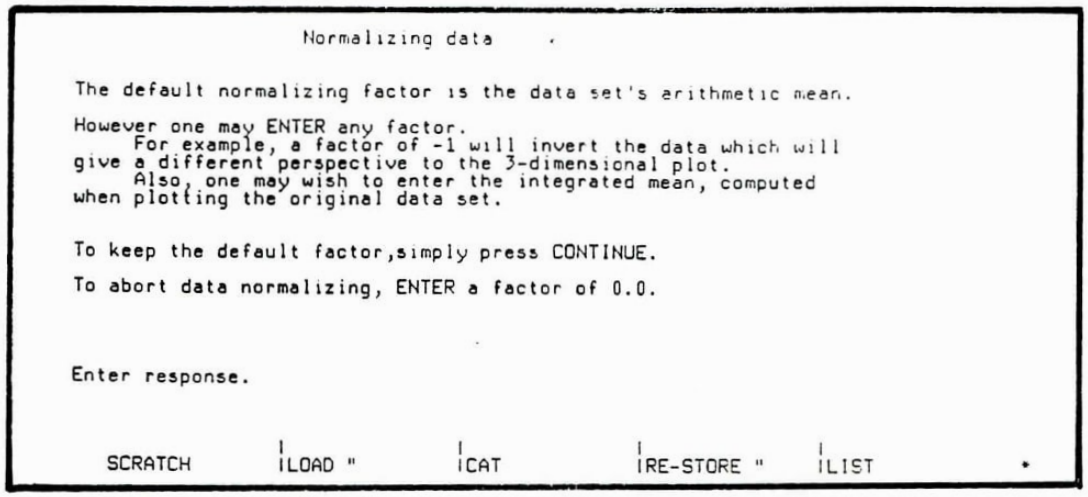


Figure 40 Option 3 from the DATA EDITING MENU allowing the data to be normalized (divided) by a given factor.

```
                >>>>>   PLOT MENU  #1   <<<<<

 1-DATA VALUES PLOTTED?          def:NO   2-GRID POINTS PLOTTED?       def:YES
 3-DATA VALUES PRINTED?          def:NO   4-CONTOUR POINTS PRINTED?    def:NO
 5-GRID FRAME DRAWN?             def:YES  6-VALUE OF AXES LABELED?     def:NO
 7-PLOT SPECIFIER?               def:YES  8-DAMPING FACTOR?            def: 2
 9-POLAR PLOT TO SCALE?            default: YES
10-CONTOUR LEVELS?                 default:multiples of .1
11-GRID & TITLE COLOUR?            default: PEN #  1
12-CONTOUR LINE COLOUR?            default: PEN #  2
13-LINE VALUE LABEL COLOUR?        default: PEN #  3
14-CONTOUR LEVEL CHARACTER SIZE?   default:  1.6
15-TITLE CHARACTER SIZE?           default:  1.8
16-GO BACK TO MAIN KEY MENU              17-LEAVE PLOT MENU



TO CHANGE ANY OF THE ABOVE, INPUT THE LINE NUMBER


              |              |              |              |
   SCRATCH    |LOAD "        |CAT           |RE-STORE "    |LIST
```

```
                >>>>>   PLOT MENU  #1   <<<<<

 1-DATA VALUES PLOTTED?          def:NO   2-GRID POINTS PLOTTED?       def:YES
 3-DATA VALUES PRINTED?          def:NO   4-CONTOUR POINTS PRINTED?    def:NO
 5-GRID FRAME DRAWN?             def:YES  6-VALUE OF AXES LABELED?     def:NO
 7-PLOT SPECIFIER?               def:YES  8-DAMPING FACTOR?            def: 2

10-CONTOUR LEVELS?                 default:multiples of .1
11-GRID & TITLE COLOUR?            default: PEN #  1
12-CONTOUR LINE COLOUR?            default: PEN #  2
13-LINE VALUE LABEL COLOUR?        default: PEN #  3
14-CONTOUR LEVEL CHARACTER SIZE?   default:  1.6
15-TITLE CHARACTER SIZE?           default:  1.8
16-GO BACK TO MAIN KEY MENU              17-LEAVE PLOT MENU



TO CHANGE ANY OF THE ABOVE, INPUT THE LINE NUMBER


              |              |              |              |
   SCRATCH    |LOAD "        |CAT           |RE-STORE "    |LIST
```

Figure 41 PLOT MENU #1 offering options available with a contour plot. The upper display occurs only for polar plots onto the plotter while the bottom display is given when using a cartesian grid or when plotting onto the screen.

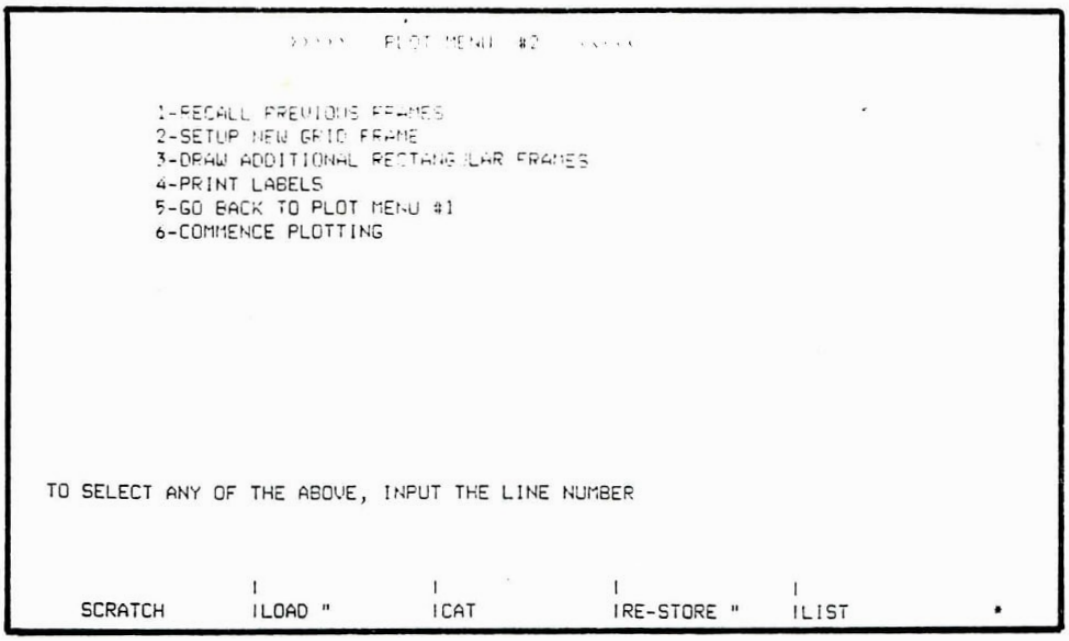
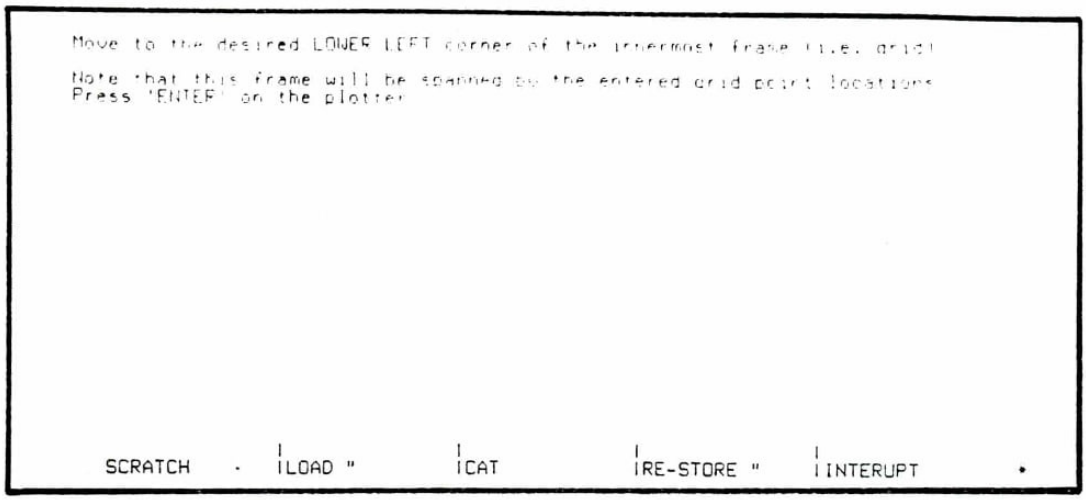


Figure 42 PLOT MENU #2. This menu is only attained if plotting onto the plotter which requires a digitizing process (item #2). Additional cosmetics for the contour plot are also available.



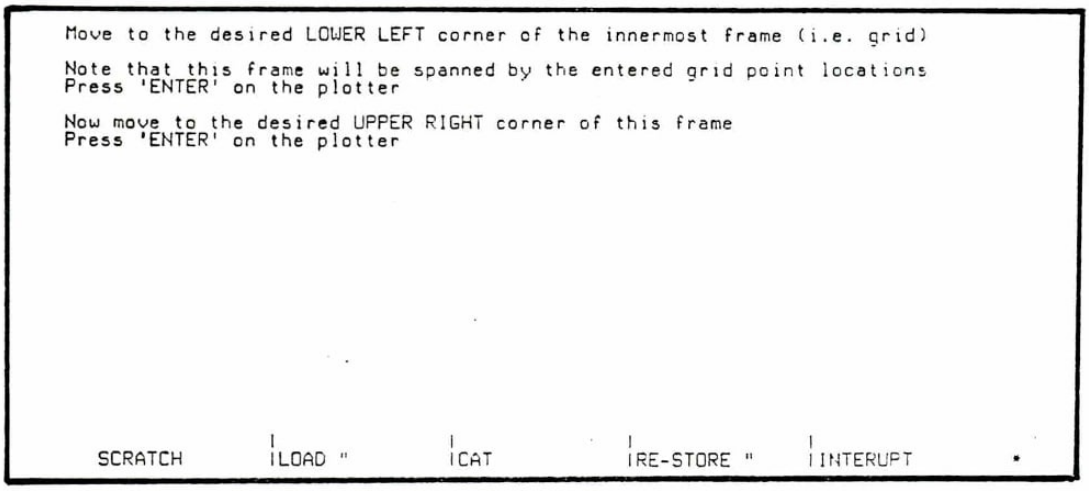
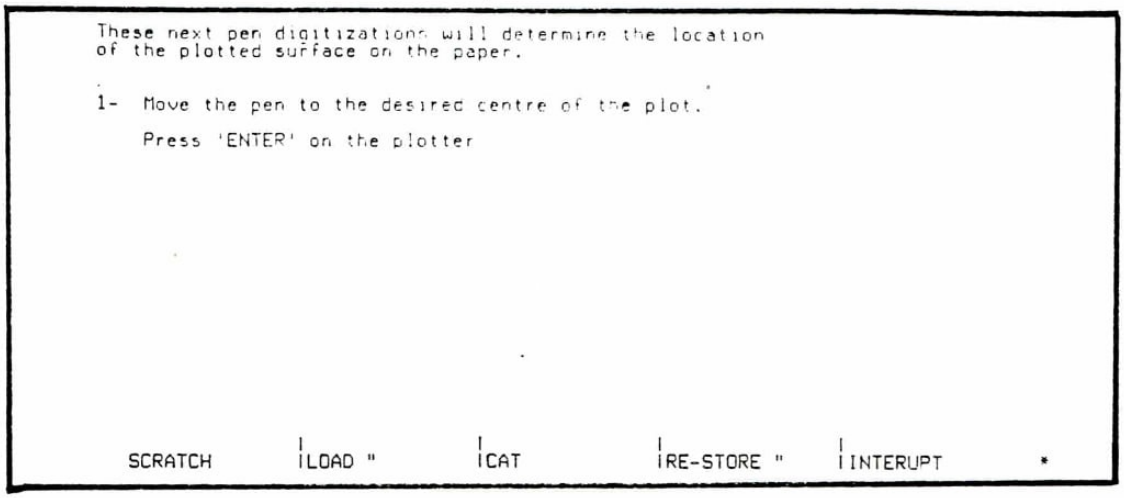
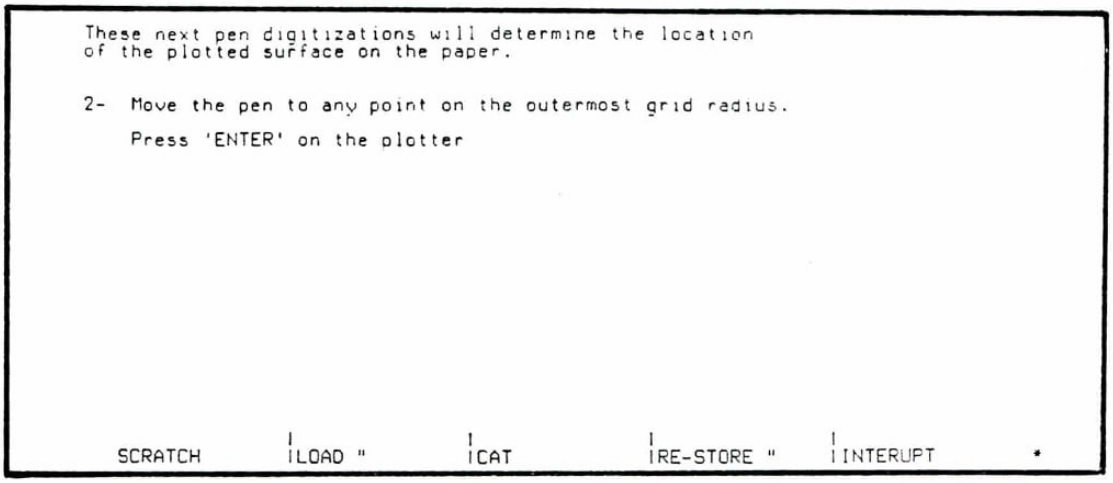


Figure 43 Displays prompting the user to digitize a cartesian grid by specifying its lower left and upper right corners. To draw additional rectangular frames (PLOT MENU #2, item #4) they must also be digitized in the same manner as above.





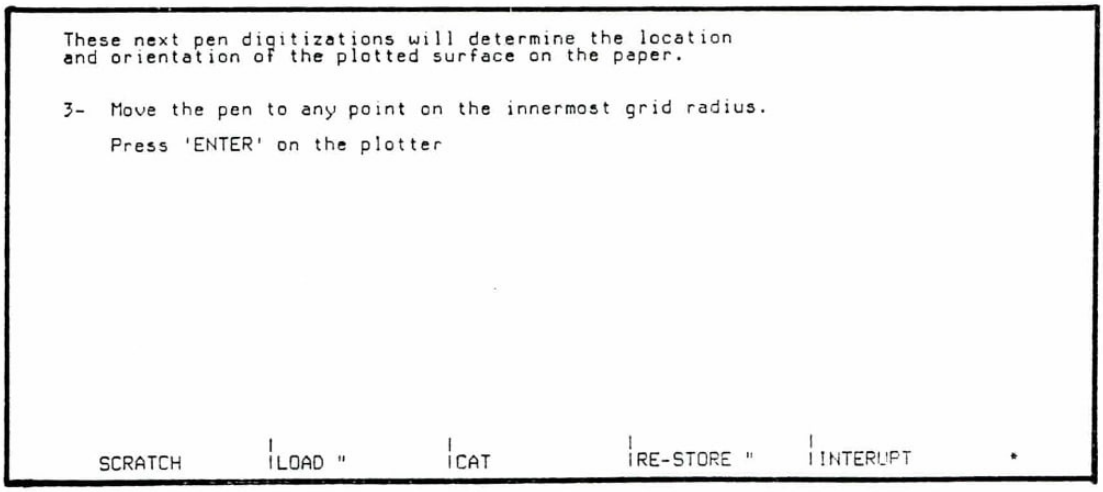
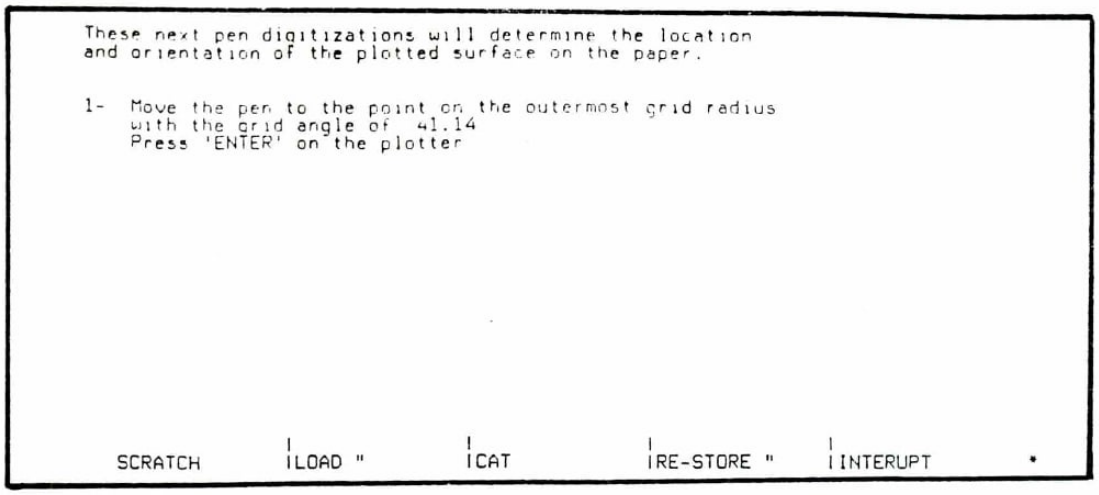
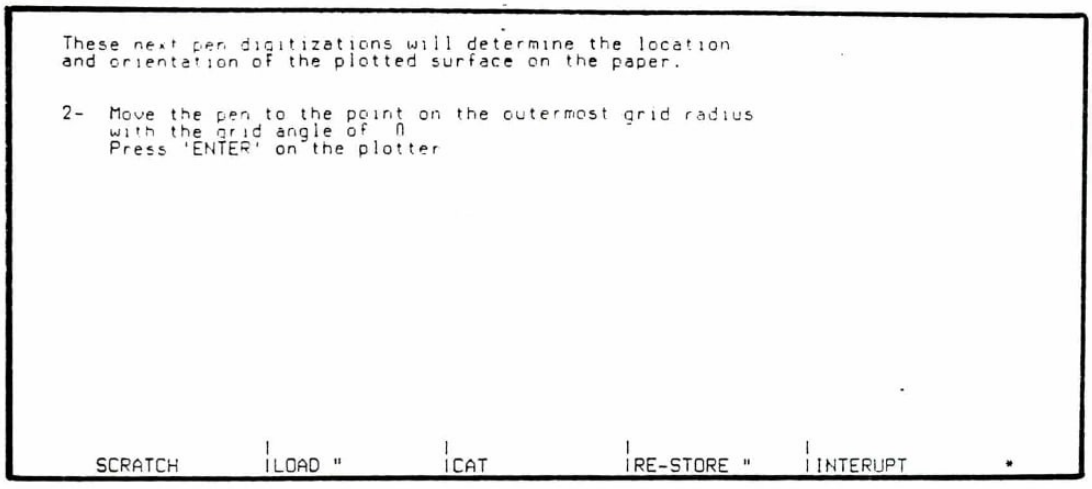


Figure 44 Displays prompting the user to digitize a polar annular grid by specifying its center and outer radius. If a polar plot is to be drawn not to scale (including polar section plots), the location of the inner radius must also be digitized.





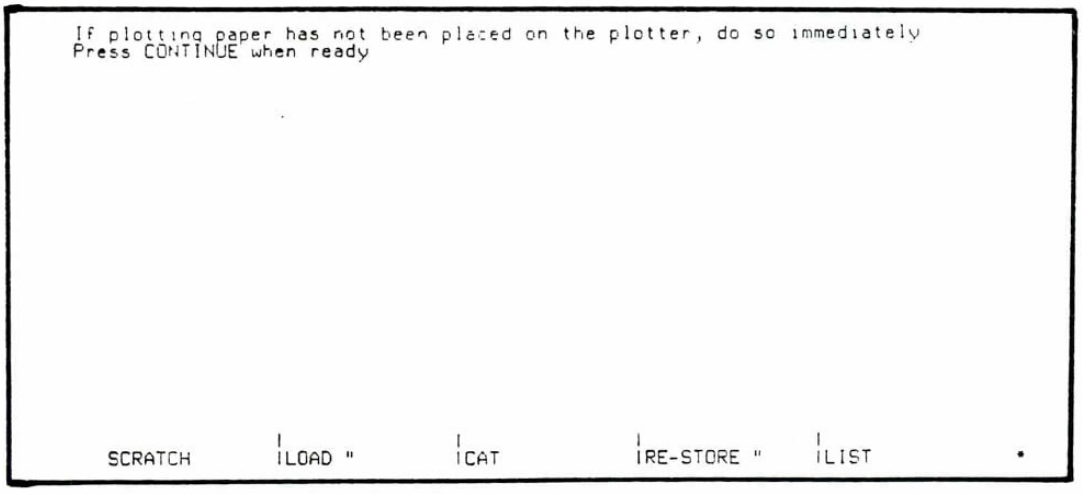


Figure 45 Displays prompting the digitization of a polar sector grid. The first digitization requires the placement of a point with the largest circumferential co-ordinate and then a point with the smallest circumferential co-ordinate. As shown, the grid is the one defined by Table 4 with bounding circumferential co-ordinates 41.14 and 0 degrees.

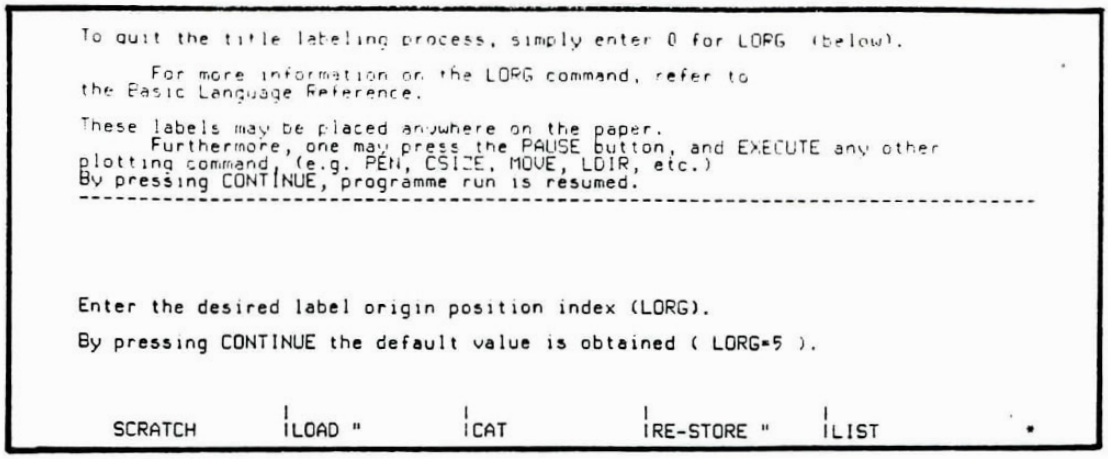


Figure 46 Title labelling process.

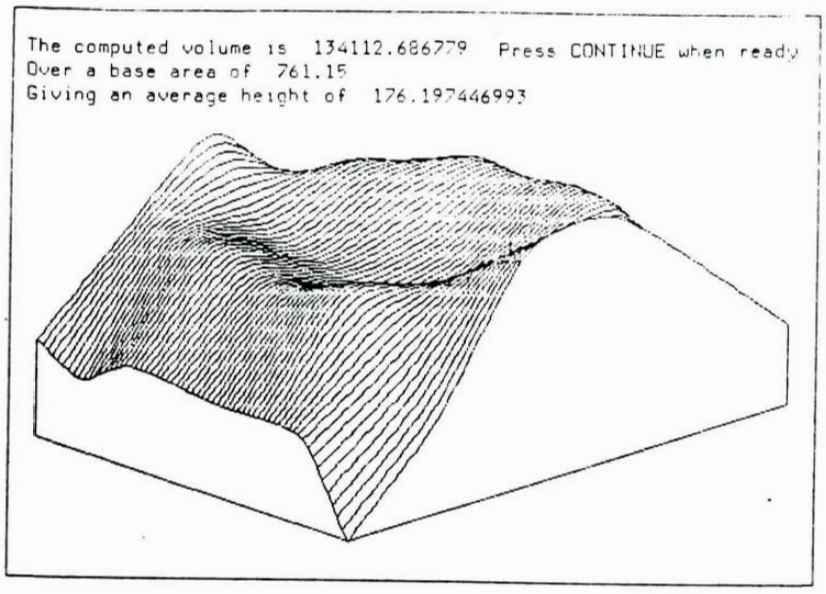


Figure 47 Screen display of the 3-D view of the data from Table 2 and its integrated volume and average height. This integrated volume and average height display is also printed for all contour plots.

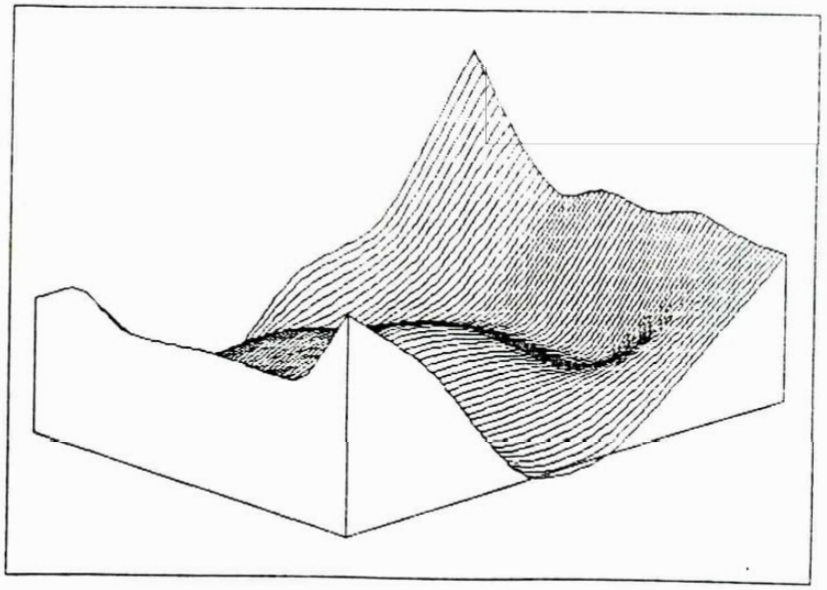


Figure 48 3-D view of the data from Table 2 but having been normalized by a factor of -1.0.

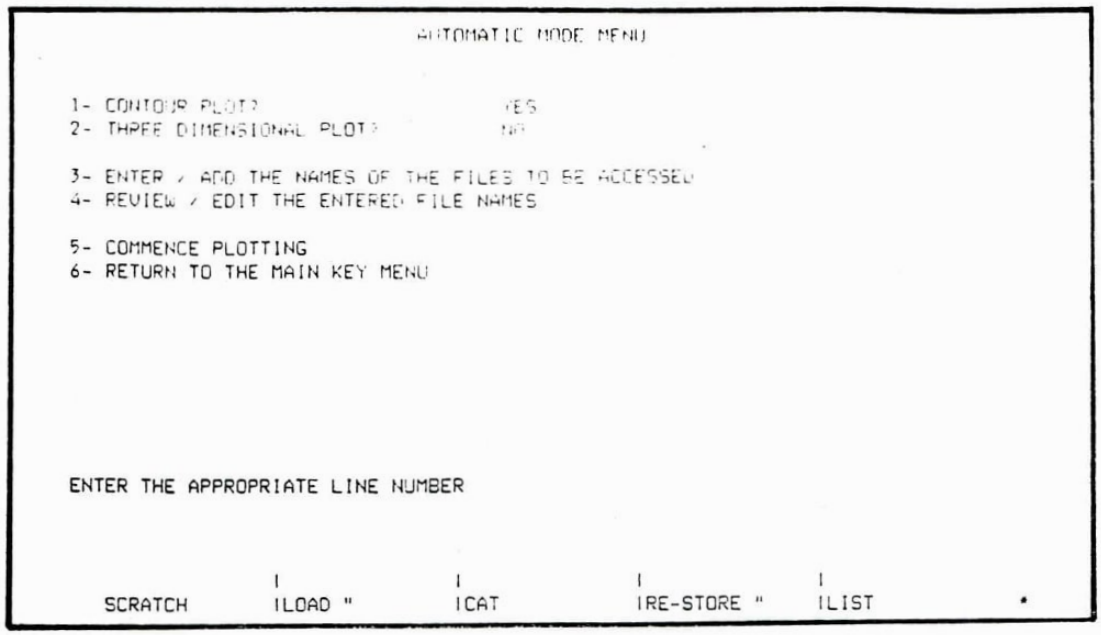


Figure 49 AUTOMATIC MODE MENU.

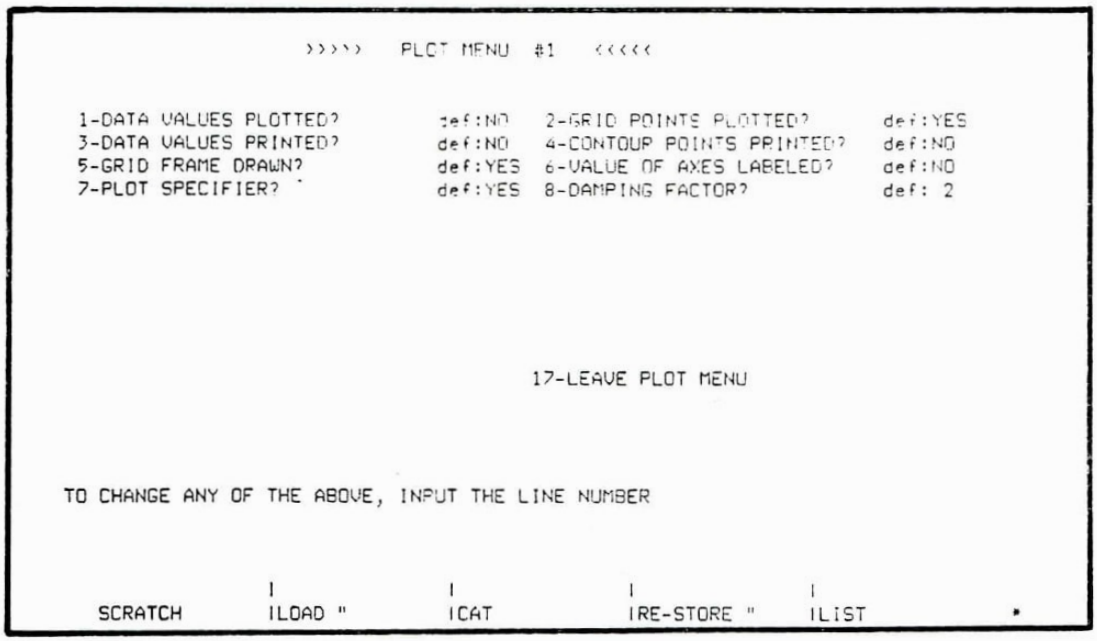
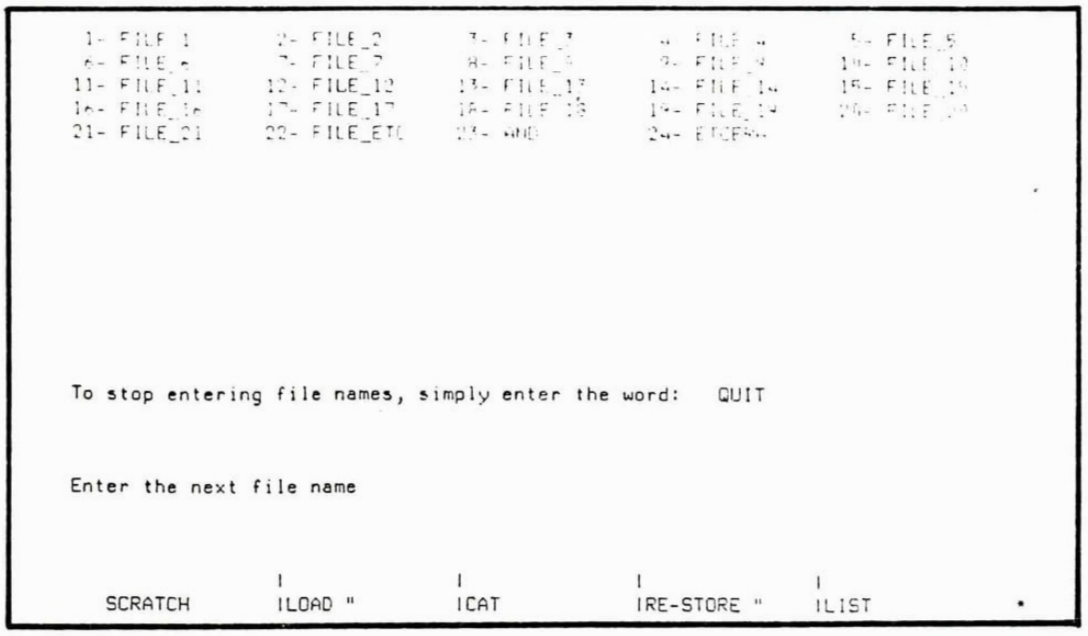


Figure 50 Reduced version of PLOT MENU #1 attained only after exiting from the AUTOMATIC MODE MENU and if contour maps have been chosen to be plotted.



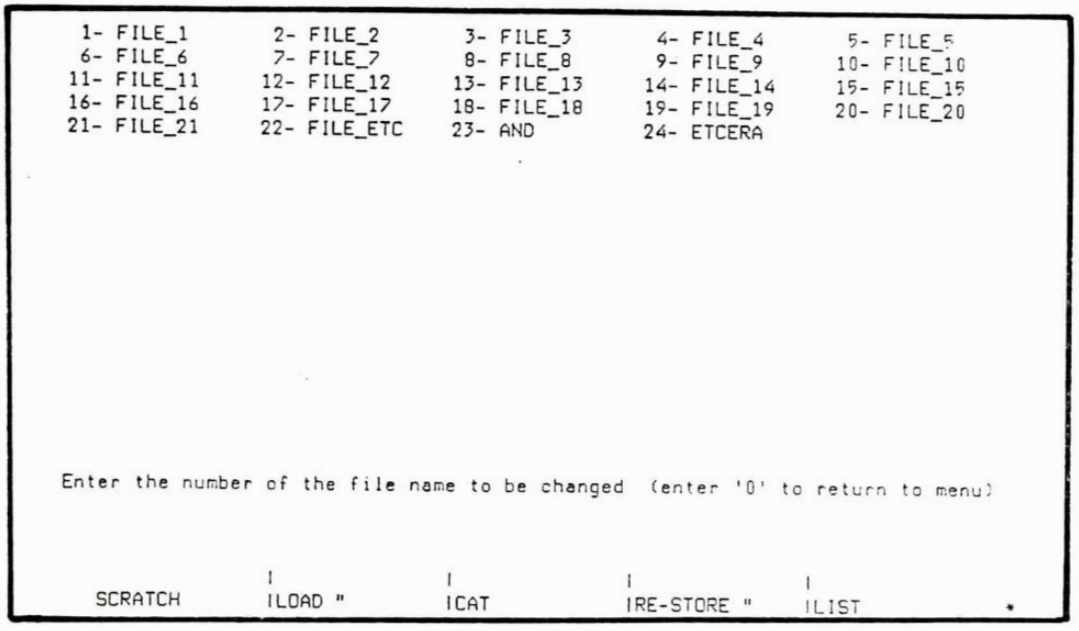


Figure 51 Screen displays of the entry and editing of file names within the AUTOMATIC MODE MENU.

LIST OF VARIABLES

TYPE CODE: A=ARRAY S=SIMPLE I=INTEGER R=REAL ST=STRING

VARIABLE	TYPE	DESCRIPTION
A	A,R	mass data storage array
Ahor,Bhor Chor,Dhor	A,R	rational spline coefficients for the set of transverse cross plots
Aver,Bver Cver,Dver	A,R	rational spline coefficients for the set of longitudinal cross plots
Ax,Bx, Cx,Dx	A,R	x-coefficients of the parametric cubic spline passed in the horizontal plane defining a contour line
Ay,By, Cy,Dy	A,R	y-coefficients of the parametric cubic spline passed in the horizontal plane defining a contour line
Bot	S,R	next bottom boundary point to be fitted
Bot_ct	S,I	number of bottom boundary points for a contour level
C	A,R	contour level values
Crosshor	A,R	vertical height of the transverse cross plots at the grid points
Crossver	A,R	vertical height of the longitudinal cross plots at each at each of the intersecting transverse grid lines
Distort	S,R	factor used on 'Ly' to distort the inner radii with respect to the outer radius in the polar contour plots
F	A,R	plotted data set
F_norm	A,R	normalized data set
F_org	A,R	original data set
Factor	S,R	scaling factor applied to the 'Ly' co-ordinates in relation to the 'Lx' values so that equal distances on the contour plot have the same numerical value in both the transverse and longitudinal directions
Form$	S,ST	grid shape specifier

Gamma	S,R	angular rotation given to the circumferential co-ordinates to properly place the polar sector plot within the specified viewing area
Imax	S,I	number of longitudinal grid lines
Istep	S,R	distance along a contour line between the interpolated values
Jmax	S,I	number of transverse grid lines
Left_join	A,I	flag selecting next line segment to join with the left end of the current line
Level	S,R	current contour level
Levmax	S,I	number of contour levels selected
Li_save	A,I	index specifying the line segment having as its latest point a particular contour point from the previous longitudinal cross plot
Limax	S,I	number of line segments on one contour level
Lower	S,R	lower interval bound for root search
Lx	A,R	longitudinal grid line co-ordinates
Ly	A,R	transverse grid line co-ordinates
New	S,I	number of contour points to be fitted from the current longitudinal cross plot
Old	S,I	number of open contour line segments from the previous longitudinal cross plot
Pcount	A,I	number of contour points on a line segment
Phor,Qhor Pver,Qver	A,R	damping factor for each spline interval along the cross plots
Right_join	A,I	flag selecting next line segment to join with the right end of the current line
Tin	S,R	value of the parametric variable
Top	S,R	next top boundary point to be fitted
Top_ct	S,I	number of top boundary points for a contour level
Tplot	A,R	value of the parametric variable at each of the contour points along the contour line

Upper	S,R	upper interval bound for root search
Values	S,I	number of contour lables per contour level
Vert	S,I	indice for current longitudinal cross plot
W	A,R	contour point locations found within a grid interval
X_bot	S,R	bottom boundary points
X_loc	A,R	x co-ordinate to place contour label
X_top	S,R	top boundary points
Xcont	A,R	x co-ordinate of line segment points
Xin	S,R	x co-ordinate of the current longitudinal cross plot
Xout	S,R	x interpolated value along the parametrically splined contour line
Xplot	A,R	x co-ordinate of contour line points
Xplot	A,R	x co-ordinate of the plotted point for the 3-D view
Xsave	S,R	x co-ordinate of thr current longitudinal cross plot
Xstep	S,R	increment between longitudinal cross plots
Y_loc	A,R	y co-ordinate to place contour label
Ycont	A,R	y co-ordinate of line segment points
Ynew	A,R	set of points from the current longitudinal cross plot within the 3-D view
Yold	A,R	set of uppermost points having been plotted to obtain the 3-D view
Yout	S,R	y interpolated value along the parametrically splined contour line
Yplot	A,R	y co-ordinate of contour line points
Yplot	A,R	y co-ordinate of the plotted point for the 3-D view
Ytemp	A,R	set of points defining the front transverse cross plot in the 3-D view

```
!*************************************************************************
!                                                                         *
!      C O N T O U R      P L O T T I N G      P R O G R A M M E         *
!                                                                         *
!*************************************************************************
!
!*************************************************************************
!                                                                         *
!                       Variable Description                              *
!                                                                         *
!    F_org(K)  = measured value at point K                                *
!    F_norm(*) = normalized F_org data set                                *
!    F(*)    => plotted data set                                          *
!    Lx(*)   => x-grid locations                                          *
!    Ly(*)   => y-grid locations                                          *
!    Xcont(line#,point#) => holds x-coordinates of each point on each     *
!                           line segment on a given contour level         *
!    Ycont(line#,point#) => holds y-coordinates of each point on each     *
!                           line segment on a given contour level         *
!          Pcount(Line#) => indicates the maximum number of points in     *
!                           the specified line segment                    *
!                                                                         *
! * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
!                                                                         *
!       This programme deals with one contour level at a time. It fits    *
!  a piecewise continuous rational spline along each of the transverse    *
!  grid lines to the given data.  Heights on all these cross plots are    *
!  read at a particuliar value of x.  By fitting a rational spline to     *
!  these heights, a longitudinal cross plot is obtained.  In this         *
!  manner, many longitudinal cross plots can be generated, by slowly      *
!  incrementing the value of the transverse co-ordinate.   Along these    *
!  longitudinal cross plots, the locations of the contour points for a    *
!  particular contour level are computed and these points are then        *
!  ordered into temporary line segments (all traversing from left to      *
!  right).  These segments are then selectively joined together into      *
!  full contour lines, which are then smoothed and plotted.  Finally,     *
!  the above search is repeated for the next contour level.               *
!                                                                         *
!*************************************************************************
!
OPTION BASE 1
DIM Xcont(50,200),Ycont(50,200),C(40),F_norm(2000),A(2100),F_org(2000)
COM /Gen/ Lx(50),Ly(50),Form$[12],INTEGER Imax,Jmax
COM /Gen2/ F(2000)
COM /Colours/ INTEGER Colour(3),Plot_flag
COM /Location/ X_loc(40,20),Y_loc(40,20),INTEGER Values(20)
COM /Bounds/ INTEGER Left(2),Right(2),Leftbound,Rightbound,Lineflag
COM /Global/ File$[160],Grid$[40],INTEGER Form
COM /Interupt/ INTEGER Interupt
COM /Change/ Factor,Distort,Gamma
COM /Damping/ Phor(150),Qhor(150),Pver(50),Qver(50)
INTEGER H,I,J,K,L,Limax,Levmax,Data_flag,Norm_flag,Cont_flag
INTEGER Same_flag,Lyflag,Lxflag,Kmax,Flag_3d,Total,Pm1,Auto_total,Auto_flag
INTEGER Slope_flag,Expon,Pcount(50),High_flag,Low_flag,Line,Next_int
DIM Flag$(17)[50],Def_flag$(17)[9],Device$[10],Ans$[18],Line$[18]
DIM Name$(75)[10],Labels$(3)[80]
REAL Sample_int(4),Digipt(2,3,4),Fdum(2000)
REAL Avg,Char_size,Char_title,Check,Damping,Hcont,Inter,Height
REAL Interval,Istep,Max,Min,Numlinnew,Numlinold,Slope,Test,Test_int
CALL Contours
CALL Datetime
Main:!
Same_flag=0
Cont_flag=0
GCLEAR
PRINTER IS 1
PRINT USING "(25/)"
PRINT TABXY(1,1);"The procedure for using this program is as follows;"
PRINT "        1) Input the data  (one can also store this data on disc)."
PRINT "        2) Plot contour map on the screen or on the plotter."
PRINT "        3) If plot is unsuitable, one can correct data by "
PRINT "           again accessing data input section."
PRINT "        4) Repeat from (2) until plot is satisfactory."
PRINT "        5) One can also repeat (2) as many times as one wishes; to p
roduce"
PRINT "           identical copies or to select a different cosmetic for th
e plot."
PRINT "        6) One can also plot a three dimensional view of the surface
."
PRINT "        7) Exit the programme or input new data (1)."
```

```
390   PRINT TABXY(1,12);"Author:  Brian McCrea
March, 1986"
395   PRINT TABXY(1,13);"         National Research Council"
400   PRINT TABXY(1,14);"         Division of Mechanical Engineering, Engine Labo
ratory, M-7"
405   PRINT TABXY(1,15);"         Ottawa, Canada
    K1A 0R6"
410   PRINT TABXY(1,17);">>>>>  When ready to start, press CONTINUE  <<<<<"
415   PRINT TABXY(1,18);"NOTE : After each keyboard entry, press CONTINUE "
420   PAUSE
425   FOR K=1 TO 8                        !\
430     Def_flag$(K)="def:"               ! \
435     Flag$(K)="NO"                     !  \
440   NEXT K                              !   \
445   FOR K=9 TO 15                       !    \
450     Def_flag$(K)="default: "          !     \
455   NEXT K                              !      \
460   Damping=2                           !       \INITIALIZE VARIABLES
465   Flag$(2)="YES"                      !       >
470   Flag$(5)="YES"                      !      /
475   Flag$(7)="YES"                      !     /
480   Flag$(11)="PEN # "                  !    /
485   Colour(1)=1                         !   /
490   Colour(2)=2                         !  /
495   Colour(3)=3                         ! /
500   Char_size=1.6                       !/
505   Char_title=1.8
510   Factor=1
515   Distort=1
520   Gamma=0
525   GOSUB Damping
530   GOTO Mainkeys
535 Damping: !                    !\
540   FOR I=1 TO 150              ! \
545     Phor(I)=Damping           !  \
550     Qhor(I)=Damping           !   \
555   NEXT I                      !    \INITIALIZE DAMPING COEFFICIENTS
560   FOR J=1 TO 50               !    /
565     Pver(J)=Damping           !   /
570     Qver(J)=Damping           !  /
575   NEXT J                      ! /
580   RETURN                      !/
585 Mainkeys: !
590   PRINT CHR$(12)
595   PRINTER IS 1
600   PRINT TABXY(17,16);"Select the appropriate special function key"
605   PRINT TABXY(30,12);"MAIN KEY MENU"
610   CONTROL 2,0;1 !   CAPS LOCK ON                !\
615   ON KEY 0 LABEL "INPUT DATA" GOSUB 920         ! \
620   ON KEY 1 LABEL "USE SAME DATA" GOSUB 915      !  \
625   ON KEY 2 LABEL "PLOT ON SCREEN" GOSUB 955     !   \
630   ON KEY 3 LABEL "PLOT ON PAPER" GOSUB 965      !    \
635   ON KEY 4 LABEL "THREE-D PLOT" GOSUB 895       !     \ SET UP KEY MENU
640   ON KEY 5 LABEL "AUTOMATIC MODE" GOSUB 670     !     /
645   ON KEY 6 LABEL "EXIT PROGRAMME" GOSUB 2420    !    /
650   ON KEY 7 GOTO 595                             !   /
655   ON KEY 8 GOTO 595                             !  /
660   ON KEY 9 GOTO 595                             ! /
665   GOTO 595                                      !/
670   OFF KEY
675   CALL Automatic(Name$(*),Auto_total,Line)
680   PRINT CHR$(12)
685   IF Line=6 THEN RETURN
690   IF Auto_total=0 THEN RETURN
695   Pm1=0
700   Avg=1
705   Auto_flag=1
710   IF Flag_3d<>1 THEN Showmenu
715   Plot_flag=0
720 Auto_loop:!
725   FOR T=1 TO Auto_total
730     ON ERROR GOTO Another
735     File$=Name$(T)
740     MASS STORAGE IS ":INTERNAL,4,1"
745     ASSIGN @Path TO File$
750     ENTER @Path;Time;Form$;Labels$(*);Imax;Jmax
755     REDIM A(Imax+Jmax*(Imax+1))
760     ENTER @Path;A(*)
765     ASSIGN @Path TO *
770     IF Form$<>"CARTESIAN" AND Flag_3d=1 THEN Another
```

```
775     FOR K=1 TO Imax
780       Lx(K)=A(K)
785     NEXT K
790     FOR K=1 TO Jmax
795       Ly(K)=A(K+Imax)
800     NEXT K
805     FOR K=1 TO Imax*Jmax
810       F(K)=A(K+Imax+Jmax)
815     NEXT K
820     GOTO 1030
825 Dump:!
830     ON ERROR GOTO Another
835     PRINTER IS 701
840     PRINT USING "@"
845     PRINTER IS 1
850     DUMP DEVICE IS 701
855     DUMP GRAPHICS
860 Another:!
865   NEXT T
870   GCLEAR
875   PRINTER IS 1
880   Auto_flag=0
885   Flag_3d=0
890   RETURN
895   PRINT CHR$(12)
900   OFF KEY
905   Flag_3d=1
910   GOTO 975
915   IF Same_flag=1 THEN Same_flag=2
920   OFF KEY
925   Avg=1
930   Auto_flag=0
935   Pm1=0
940   CALL Data_input(Avg,F_org(*),F_norm(*),A(*),Data_flag,Norm_flag,Same_flag,D
evice$)
945   GOSUB Damping
950   RETURN
955   Plot_flag=0       !!--------------PLOT CONTOURS ON SCREEN-----------------
960   GOTO 970
965   Plot_flag=1       !!--------------PLOT CONTOURS ON PLOTTER----------------
970   OFF KEY
975   IF Data_flag<>1 THEN
980     PRINT CHR$(12);TABXY(10,18);"No data has been entered, must first select
INPUT DATA key."
985     BEEP 400,.3
990     RETURN
995   END IF
1000  IF Flag_3d=1 AND Form$<>"CARTESIAN" THEN
1005    PRINT CHR$(12);TABXY(10,18);"Only cartesian grids may be plotted in the 3
-D view."
1010    BEEP 400,.3
1015    Flag_3d=0
1020    RETURN
1025  END IF
1030  Max=F(1)                                          !\
1035  Min=F(1)                                          ! \
1040  FOR K=2 TO Imax*Jmax                              !  \---DETERMINE THE MINIMUM---
1045    IF F(K)<Min THEN Min=F(K)                       !  /--AND MAXIMUM DATA VALUES--
1050    IF F(K)>Max THEN Max=F(K)                       ! /
1055  NEXT K                                            !/
1060  IF Min=Max THEN
1065    IF Auto_flag=1 THEN Another
1070    BEEP 555,.7
1075    PRINT CHR$(12);TABXY(1,8);"ALL DATA HAVE THE VALUE : ";Min
1080    PRINT TABXY(1,10);"Press CONTINUE to proceed"
1085    PAUSE
1090    Data_flag=0
1095    RETURN
1100  END IF
1105  IF Flag_3d=1 THEN 1955     !AFTER DATA HAS BEEN VERIFIED, CROSS PLOTS
1110                             !AND 3-D PLOT CAN BE IMMEDIATELY COMPUTED
1115  Inter=0
1120  Sample_int(1)=.1                                  !\
1125  Sample_int(2)=.2                                  ! \
1130  Sample_int(3)=.25                                 !  \
1135  Sample_int(4)=.5                                  !   \
1140  Numlinold=(Max-Min)/Sample_int(1)                 !    \
1145  Interval=Sample_int(1)                            !     >  INITIALIZE CONTOUR
1150  Check=ABS(Numlinold-10)                           !    /    LEVEL PARAMETERS
1155  Low_flag=0                                        !   /
```

```
1160  High_flag=0                                  !  /
1165  Next_int=2                                   ! /
1170  Expon=0                                      !/
1175  OFF ERROR
1180 Iterate:                                      |
1185  Test_int=Sample_int(Next_int)*10^Expon       |\
1190  Numlinnew=(Max-Min)/Test_int                 | \
1195  Test=ABS(Numlinnew-10)                       |  \
1200  IF Test<Check THEN                           |   \
1205    Numlinold=Numlinnew                        |    \
1210    Interval=Test_int                          |     \
1215    Check=Test                                 |      \
1220  ELSE                                         |       \
1225    IF Numlinnew<Numlinold THEN                |        \
1230      Low_flag=1                               |         >SELECT DEFAULT
1235    ELSE                                       |        / CONTOUR LEVELS:
1240      IF Numlinnew>Numlinold THEN              |       /
1245        High_flag=1                            |      /      THE VERTICAL
1250      END IF                                   |     <    INTERVAL HAVING A
1255    END IF                                     |      \   MANTISSA OF EITHER
1260  END IF                                       |       \  1.0,2.0,2.5 OR 5.0 AND
1265  IF Low_flag<>1 OR High_flag<>1 THEN          |        >MULTIPLIED BY A FACTOR
1270    IF Low_flag=1 THEN                         |       / OF TEN RAISED TO AN
1275      Next_int=Next_int-1                      |      /  EXPONENT IS COMPUTED
1280    ELSE                                       |     <   SO THAT AS CLOSE TO
1285      Next_int=Next_int+1                      |      \  TEN CONTOUR LEVELS AS
1290    END IF                                     |       \ POSSIBLE WILL BE
1295    IF Next_int=0 THEN                         |        \GENERATED.
1300      Next_int=4                               |         >
1305      Expon=Expon-1                            |        /
1310    ELSE                                       |       /
1315      IF Next_int=5 THEN                       |      /
1320        Next_int=1                             |     /
1325        Expon=Expon+1                          |    /
1330      END IF                                   |   /
1335    END IF                                     |  /
1340    GOTO Iterate                               | /
1345  END IF                                       |/
1350  IF Auto_flag=1 THEN Levels
1355 Plotmenu:  !\
1360            ! >----------------------SET UP PLOT MENU--------------------
1365            !/
1370  Def_flag$(10)="default:"
1375  Flag$(10)="multiples of "
1380  Flag$(10)[14]=VAL$(Interval)
1385  IF Form$[1,5]="POLAR" THEN Flag$(9)="YES"
1390 Showmenu:!
1395  PRINT CHR$(12)
1400  PRINT TABXY(1,1),"                              >>>>>   PLOT MENU  #1   <<<<<"
1405  PRINT TABXY(1,4);" 1-DATA VALUES PLOTTED?        ";Def_flag$(1);Flag$(1)
1410  PRINT TABXY(40,4);" 2-GRID POINTS PLOTTED?      ";Def_flag$(2);Flag$(2)
1415  PRINT TABXY(1,5);" 3-DATA VALUES PRINTED?        ";Def_flag$(3);Flag$(3)
1420  PRINT TABXY(40,5);" 4-CONTOUR POINTS PRINTED?   ";Def_flag$(4);Flag$(4)
1425  PRINT TABXY(1,6);" 5-GRID FRAME DRAWN?           ";Def_flag$(5);Flag$(5)
1430  PRINT TABXY(40,6);" 6-VALUE OF AXES LABELED?    ";Def_flag$(6);Flag$(6)
1435  PRINT TABXY(1,7);" 7-PLOT SPECIFIER?             ";Def_flag$(7);Flag$(7)
1440  PRINT TABXY(40,7);" 8-DAMPING FACTOR?           ";Def_flag$(8);Damping
1445  IF Auto_flag=1 THEN 1500
1450  IF Form$[1,5]="POLAR" AND Plot_flag=1 THEN
1455    PRINT TABXY(1,8);" 9-POLAR PLOT TO SCALE?          ";Def_flag$(9);Flag$(9
)
1460  END IF
1465  PRINT TABXY(1,9);"10-CONTOUR LEVELS?              ";Def_flag$(10);Flag$(10
)
1470  PRINT TABXY(1,10);"11-GRID & TITLE COLOUR?         ";Def_flag$(11);Flag$(1
1);Colour(1)
1475  PRINT TABXY(1,11);"12-CONTOUR LINE COLOUR?         ";Def_flag$(12);Flag$(1
1);Colour(2)
1480  PRINT TABXY(1,12);"13-LINE VALUE LABEL COLOUR?     ";Def_flag$(13);Flag$(1
1);Colour(3)
1485  PRINT TABXY(1,13);"14-CONTOUR LEVEL CHARACTER SIZE? ";Def_flag$(14);Char_si
ze
1490  PRINT TABXY(1,14);"15-TITLE CHARACTER SIZE?        ";Def_flag$(15);Char_ti
tle
1495  PRINT TABXY(1,15);"16-GO BACK TO MAIN KEY MENU"
1500  PRINT TABXY(40,15);"17-LEAVE PLOT MENU"
1505  ON ERROR GOTO Showmenu
1510  Line$=""
1515  BEEP
1520  INPUT "TO CHANGE ANY OF THE ABOVE, INPUT THE LINE NUMBER",Line$
```

```
1525  IF Auto_flag=1 AND Line$="" THEN Auto_loop
1530  IF Line$="" THEN Levels
1535  Line=VAL(Line$)
1540  IF Auto_flag=1 THEN
1545    IF Line=17 THEN Auto_loop
1550    IF Line<1 OR Line>8 THEN Showmenu
1555  END IF
1560  IF Line=17 THEN Levels
1565  IF Line=16 THEN
1570    PRINT CHR$(12)
1575    RETURN
1580  END IF
1585  IF Line>=1 AND Line<=15 THEN
1590    Def_flag$(Line)=""
1595            !!--------------DETERMINE NEW VALUE OF MENU ITEMS--------------
1600    IF Line=8 THEN
1605      INPUT "Enter new damping factor.",Damping     !\
1610      IF Damping<.1 THEN Damping=.1                 ! \ CHANGE THE
1615      IF Damping>100 THEN Damping=100               ! /DAMPING FACTOR
1620      GOSUB Damping                                 !/
1625    ELSE
1630      IF Line>0 AND Line<10 THEN
1635        IF Line=9 AND Form$[1,5]<>"POLAR" THEN Showmenu
1640        IF Line=9 AND Plot_flag=0 THEN Showmenu
1645        IF Flag$(Line)="YES" THEN                   !\
1650          Flag$(Line)="NO"                          ! \
1655        ELSE                                        !  >CHANGE ITEM ANSWER TO
1660          Flag$(Line)="YES"                         ! /    'YES'   OR   'NO'
1665        END IF                                      !/
1670        IF Line=1 OR Line=2 THEN        !\
1675          IF Flag$(Line)="YES" THEN     ! \
1680            Def_flag$(3-Line)=""        !  \SINCE ITEMS 2&3 ARE RELATED, BOTH
1685            Flag$(3-Line)="NO"          !  /  CANNOT HAVE 'YES' AS AN ANSWER
1690          END IF                        ! /
1695        END IF                          !/
1700      END IF
1705    END IF
1710    IF Line=10 THEN
1715      PRINT TABXY(1,20),"The minimum value is";Min     !\
1720      PRINT "The maximum value is";Max                 ! \
1725      INPUT "Enter the vertical interval.  (enter 0 to get back the defaults)
",Inter
1730      IF Inter<=0 THEN Plotmenu                        !  \
1735      INPUT "Lowest contour value?",C(1)               !   \
1740      INPUT "Highest contour value?",Hcont             !    \INPUT THE
1745      FOR I=2 TO 40                                    !    /CONTOUR LEVELS
1750        IF C(I-1)+Inter/1.000001>Hcont THEN 1765       !   /
1755        C(I)=C(I-1)+Inter                              !   \
1760      NEXT I                                           !    \
1765      Levmax=I-1                                       !    /
1770      Flag$(10)="from       to       stepping by "     !   /
1775      Flag$(10)[6;5]=VAL$(C(1))                        !  /
1780      Flag$(10)[15;5]=VAL$(C(Levmax))                  ! /
1785      Flag$(10)[33;5]=VAL$(Inter)                      !/
1790    END IF
1795    IF Line>10 AND Line<14 THEN                         !\
1800      Colour(Line-10)=Colour(Line-10)+1                 ! \CHANGE ITEM
1805      IF Colour(Line-10)=5 THEN Colour(Line-10)=1       ! / PEN COLOUR
1810    ELSE                                                !/
1815                              !INPUT NEW VALUES FOR ITEMS 14&15 OF MENU
1820      IF Line=14 THEN
1825        INPUT "Enter the line value label character size.",Char_size
1830      ELSE
1835        IF Line=15 THEN
1840          INPUT "Enter the character size for the title labels on the plotter
.",Char_title
1845        END IF
1850      END IF
1855    END IF
1860  END IF
1865  GOTO Showmenu               !GO BACK TO PLOT MENU WITH THE CORRECTIONS
1870 Levels:!
1875  IF Inter=0 THEN                                   !\
1880    Inter=Interval                                  ! \
1885    C(1)=INT((Min+Inter)/Inter)*Inter               !  \
1890    Hcont=INT(Max/Inter)*Inter                      !   \
1895    FOR I=2 TO 40                                   !    \---DETERMINE DEFAULT--
1900      IF C(I-1)+Inter/1.00001>Hcont THEN 1915       !    /----CONTOUR LEVELS----
1905      C(I)=C(I-1)+Inter                             !   /
1910    NEXT I                                          !  /
```

```
1915    Levmax=I-1                                          ! /
1920  END IF                                                !/
1925  OFF ERROR
1930        !!-------------CALL SUBROUTINES TO SET UP THE PLOT----------------
1935  IF Flag$(3)="YES" AND Pm1<>1 THEN CALL Data_set(F_org(*),File$)
1940  CALL Plot_grid(Istep,Digipt(*),Flag$(*),Char_size,Char_title,Damping,Avg,Co
nt_flag,Kmax,Pm1)
1945  IF Interupt<>0 THEN Re_initialize
1950  IF Pm1=1 THEN Showmenu
1955  Lxflag=0
1960  Lyflag=0
1965  IF Jmax=3 THEN
1970    Lyflag=1                                          !\
1975    Ly(4)=Ly(3)                                       ! \
1980    Ly(3)=Ly(2)+(Ly(4)-Ly(2))/1000                    !  \  EXPAND GRID (from 3 to 4) IN
1985    Ly(2)=Ly(2)-(Ly(2)-Ly(1))/1000                    !   \ THE LONGITUDINAL DIRECTION TO
1990    FOR K=1 TO Imax*Jmax                              !   / BE ABLE TO FIT THE SPLINES
1995      Fdum(K)=F(K)                                    !  /
2000    NEXT K                                            ! /
2005    Jmax=4                                            !/
2010    H=0                                         !\
2015    K=0                                         ! \
2020    FOR I=1 TO Imax                             !  \
2025      FOR J=1 TO Jmax                           !   \
2030        K=K+1                                   !    \
2035        IF K=(3+(I-1)*Jmax) THEN                !     \
2040          F(K)=F(K-1)                           !      >INITIALIZE EXPANDED DATA SET
2045          H=H+1                                 !     /
2050        ELSE                                    !    /
2055          F(K)=Fdum(K-H)                        !   /
2060        END IF                                  !  /
2065      NEXT J                                    ! /
2070    NEXT I                                      !/
2075  END IF
2080  IF Imax=3 THEN
2085    Lxflag=1                                          !\
2090    Lx(4)=Lx(3)                                       ! \  EXPAND GRID (from 3 to 4) IN
2095    Lx(3)=Lx(2)+(Lx(4)-Lx(2))/1000                    !  > THE TRANSVERSE DIRECTION TO
2100    Lx(2)=Lx(2)-(Lx(2)-Lx(1))/1000                    ! /  BE ABLE TO FIT THE SPLINES
2105    Imax=4                                            !/
2110    FOR K=Imax*Jmax TO 2*Jmax+1 STEP -1         !\
2115      F(K)=F(K-Jmax)                            ! >INITIALIZE EXPANDED DATA SET
2120    NEXT K                                      !/
2125  END IF
2130                        !COMPUTE TRANSVERSE AND LONGITUDINAL CROSS PLOTS
2135  PRINTER IS 1
2140  PRINT CHR$(12);TABXY(1,1);"PERFORMING CURVE FITTING"
2145  CALL Cross_setup(Height,Avg,Slope,Damping,Flag$(3),Norm_flag,Rightbound,Fla
g_3d)
2150  IF Interupt<>0 THEN Re_initialize
2155  IF Flag_3d=1 THEN
2160    Flag_3d=0
2165    CALL Three_d(Height,Damping,Avg,Max,Min,Rightbound,Auto_flag)
2170    IF Auto_flag=1 THEN Flag_3d=1
2175  ELSE
2180                          !INITIALIZE SEARCHING FOR CONTOUR POINTS TO START
2185    Leftbound=1           !ON THE LEFTMOST LONGITUDINAL CROSS PLOT
2190    Right(1)=0
2195    Left(1)=Rightbound
2200    Lineflag=0
2205          !\
2210          ! >-------------LOOP TO FIND AND PLOT CONTOUR LINES---------------
2215          !/
2220    FOR L=1 TO Levmax
2225      CALL Cross_plot(Xcont(*),Ycont(*),C(L),Pcount(*),L,Limax)
2230      IF Interupt<>0 THEN Re_initialize
2235      Total=0
2240      FOR K=1 TO Limax
2245        Total=Pcount(K)+Total
2250      NEXT K
2255      CALL Plot_cont(Xcont(*),Ycont(*),C(L),Istep,Char_size,L,Pcount(*),Limax
,Total,Flag$(4))
2260      IF Interupt<>0 THEN Re_initialize
2265      IF Lineflag=2 THEN 2275
2270    NEXT L
2275    CALL Plot_values(C(*),Char_size,Levmax,Auto_flag)
2280  END IF
2285  IF Auto_flag=1 THEN Dump
2290 Re_initialize:!
2295  PENUP
```

```
2300  GCLEAR
2305  PRINTER IS 1
2310  PRINT CHR$(12)
2315  IF Form$="POLARANN" AND Interupt<>2 THEN Imax=Imax-1
2320  Interupt=0  !RESET INTERUPT FLAG
2325  IF Lxflag=1 THEN                !\
2330    Imax=3                        ! \
2335    Lx(2)=(Lx(3)+Lx(2))/2         !  \
2340    Lx(3)=Lx(4)                   !   \
2345  END IF                          !    \RETURN TO ORIGINAL GRID FROM THE
2350  IF Lyflag=1 THEN                !    /  EXPANDED GRID REQUIRED FOR
2355    Jmax=3                        !   /   PASSING THE RATIONAL SPLINES
2360    Ly(2)=(Ly(3)+Ly(2))/2         !  /
2365    Ly(3)=Ly(4)                   ! /
2370  END IF                          !/
2375  IF Lxflag+Lyflag>0 THEN   !\
2380    FOR K=1 TO Imax*Jmax    ! \
2385      F(K)=Fdum(K)          !  \
2390    NEXT K                  !   >RETURN TO ORIGINAL DATA SET FROM THE
2395    Lxflag=0                !  /  EXPANDED ONE REQUIRED FOR PLOTTING
2400    Lyflag=0                ! /
2405  END IF                    !/
2410  CALL Initialize(Flag$(*))
2415  RETURN          !GO BACK TO KEY MENU
2420  PRINT CHR$(12)
2425  MASS STORAGE IS ":INTERNAL,4,0"
2430  PRINT TABXY(12,10);"*************************************************"
2435  PRINT TABXY(12,12);"         E N D    O F    P R O G R A M M E"
2440  PRINT TABXY(12,14);"*************************************************"
2445  END
2450  !
2455  !
2460  !!*************************************************************************
2465  !!                                                                        *
2470  !!           Data_input :   1st level DATA ENTRY SUBROUTINE               *
2475  !!                                                                        *
2480  !!*************************************************************************
2485  SUB Data_input(Avg,F_org(*),F_norm(*),A(*),INTEGER Data_flag,Norm_flag,Same
_flag,Device$)
2490    OPTION BASE 1
2495    COM /Gen/ Lx(*),Ly(*),Form$,INTEGER Imax,Jmax
2500    COM /Gen2/ F(*)
2505    COM /Global/ File$,Grid$,INTEGER Form
2510    DIM Term$(2,3)[35],Heading$(2,3)[6],Newtype$[40],Labels$(3)[80],Line$[160
]
2515    REAL Box(4,3),Box2(4,3),Gdum(50,100),Temp(64)
2520    REAL Dummy,Time,Total
2525    DIM Gridname$(50)[40],Gridform$(50)[12],G(50,100)
2530    INTEGER Numgrids,Numgrels
2535    INTEGER I,J,K,L,N,Line,Dis(9),Abort_flag,Man_flag,Dum,Rec,Table(150,4)
2540    INTEGER Sect,Width,Height,Row,Col,Check,Start,Type
2545    INTEGER Newmax,Len,Old,Flag_num
2550    PRINT CHR$(12)
2555    READ Box(*),Box2(*)
2560    DATA 0,.6,-1,     1,0,-1,     0,-.6,-1,     -1,0,-1
2565    DATA 0, 0,-2,     0,0,-1,     0,  0,-1,      0,0,-1
2570    Term$(1,1)="TRANSVERSE - (X / ROW)"       !\
2575    Term$(2,1)="LONGITUDINAL - (Y / COLUMN)"  ! \
2580    Term$(1,2)="CIRCUMFERENTIAL - (ANGLE / ROW)"
2585    Term$(2,2)="RADIAL - (R / COLUMN)"        !   \INITIALIZE STINGS TO BE
2590    Heading$(1,1)="X AXIS"                    !   /USED LATER AS QUALIFIERS
2595    Heading$(2,1)="Y AXIS"                    !  /IN PRINTOUTS ON THE SCREEN
2600    Heading$(1,2)="ANGLE"                     ! /
2605    Heading$(2,2)="RADIUS"                    !/
2610    Term$(1,3)=Term$(1,2)
2615    Term$(2,3)=Term$(2,2)
2620    Heading$(1,3)=Heading$(1,2)
2625    Heading$(2,3)=Heading$(2,2)
2630  Start:!
2635    Data_flag=1
2640    Norm_flag=0
2645    Old=1                       !\
2650    FOR I=1 TO 9                ! \
2655      Dis(I)=128                ! /  INITIALIZE ITEMS FOR MENU
2660    NEXT I                      !/
2665    IF Same_flag=2 THEN           !RE-USE PREVIOUSLY ENTERED DATA
2670      GOSUB Initialize
2675      GOTO Menu
2680    END IF
2685    Device$="DISC"
```

```
2690     PRINT CHR$(12)
2695     PRINT TABXY(12,5);"For data input from the keyboard, type:  KEYBOARD"
2700     PRINT TABXY(12,7);"                                    OR"
2705     PRINT TABXY(12,9);"For disc data input, type in the data file name."
2710     PRINT TABXY(12,11);"                                   OR"
2715     PRINT TABXY(12,13);"For a listing of the data files, simply press CONTINU
E."
2720     File$=""
2725     INPUT "Enter response",File$
2730     IF File$="" THEN
2735       ON ERROR GOSUB Error
2740       CAT ":INTERNAL,4,1"
2745       PRINT
2750       PRINT "It may be necessary to use the scrolling wheel to views all the
items."
2755       PRINT "Press CONTINUE, when ready."
2760       PAUSE
2765       GOTO 2690
2770     END IF
2775     PRINT CHR$(12)
2780     IF UPC$(File$)="KEYBOARD" THEN Device$="KEYBOARD"
2785     IF Device$="DISC" THEN
2790       ON ERROR GOTO Error      !!----CHECK FOR NON-EXISTENT DATA FILES----
2795       MASS STORAGE IS ":INTERNAL,4,1"
2800       ASSIGN @Path TO File$
2805       ENTER @Path;Time;Form$;Labels$(*);Imax;Jmax
2810       REDIM A(Imax+Jmax*(1+Imax))
2815       ENTER @Path;A(*)
2820       ASSIGN @Path TO *
2825       OFF ERROR
2830       Form=1
2835       IF Form$[1,5]="POLAR" THEN Form=2
2840       FOR K=1 TO Imax                    !\
2845         Lx(K)=A(K)                       ! \
2850       NEXT K                             !  \
2855       FOR K=1 TO Jmax                    !   \
2860         Ly(K)=A(K+Imax)                  !    > TRANSFER DATA FROM DISC MASS
2865       NEXT K                             !   / STORAGE ARRAY TO PROGRAM ARRAY
2870       FOR K=1 TO Imax*Jmax               !  /
2875         F_org(K)=A(K+Imax+Jmax)          ! /
2880       NEXT K                             !/
2885       GOSUB Initialize
2890       OFF ERROR
2895       GOTO Menu
2900     END IF    !\
2905               ! >---SELECT ITEMS PERTAINING TO KEYBOARD DATA ENTRY
2910 Readgrids:   !/
2915     ON ERROR GOSUB Error1        !!----LOAD ALL EXISTING GRIDS FROM DISC
2920     MASS STORAGE IS ":INTERNAL"
2925     ASSIGN @Path TO "GRIDS"
2930     ENTER @Path;Numgrids,Numgrels
2935     IF Numgrids>0 THEN
2940       REDIM Gridname$(Numgrids),Gridform$(Numgrids),G(Numgrids,Numgrels)
2945       ENTER @Path;Gridname$(*),Gridform$(*),G(*)
2950     END IF
2955     ASSIGN @Path TO *
2960     OFF ERROR
2965     GOTO Over
2970 Error1:!
2975     PRINT CHR$(12);TABXY(1,4);"You must have a satisfactory 'GRIDS' file on t
he disc in drive #0."
2980     PRINT CHR$(132);TABXY(1,6);"Press CONTINUE when ready."
2985     PRINT CHR$(128)
2990     PRINT TABXY(1,10);"NOTE :  To create a new 'GRIDS' file, type 'GRIDS' and
 press CONTINUE."
2995     PRINT TABXY(1,12);"Make sure an un-full initialized disc is presently ins
erted in drive #0."
3000     PRINT TABXY(1,14);"Otherwise, creation of 'GRIDS' will be aborted."
3005     Line$=""
3010     INPUT "Enter response",Line$
3015     IF UPC$(Line$)="GRIDS" THEN
3020       ON ERROR GOTO Error2
3025       ASSIGN @Path TO *
3030       CREATE BDAT "GRIDS",170
3035       ASSIGN @Path TO "GRIDS"
3040       Numgrids=0
3045       Numgrels=0
3050       OUTPUT @Path;Numgrids,Numgrels
3055       ASSIGN @Path TO *
3060       ASSIGN @Path TO "GRIDS"
```

```
3065      RETURN
3070 Error2:PRINT CHR$(132);TABXY(1,15);"Creation of 'GRIDS' aborted due to user
error."
3075      PRINT CHR$(128)
3080      BEEP
3085      GOTO 3005
3090    END IF
3095    RETURN
3100 Over:!
3105    OFF KEY
3110    GCLEAR
3115    PRINT CHR$(12);"GRID MENU :    Select the appropriate item from the list
below."
3120    PRINT "                The available standard combinations of parameters &
 grids are:"
3125    PRINT ""
3130    IF Numgrids=0 THEN
3135      PRINT "This is a blank 'GRIDS' file, thus there are no existing grids t
o choose from."
3140      PRINT "Therefore, the user may enter new grid specifications. (item A)"
3145      PRINT "Or, if he wishes to use the grid specifications from a different
 'GRIDS' file,"
3150      PRINT "he should select item C and as before, enter KEYBOARD to select
manual data"
3155      PRINT "entry, but this should be done after the present disc has been r
emoved and"
3160      PRINT "replaced by a disc with a non-empty 'GRIDS' file on it."
3165      PRINT ""
3170      PRINT ""
3175    END IF
3180    J=0
3185    FOR K=1 TO Numgrids
3190      Line$=VAL$(K)
3195      Len=LEN(Line$)+1
3200      Line$[Len]="-"
3205      Line$[Len+1]=Gridname$(K)
3210      Len=LEN(Line$)+1
3215      Line$[Len]="                                                          "
3220      IF Len<40 THEN
3225        IF K<>Numgrids THEN
3230          Line$[40]=VAL$(K+1)
3235          Len=LEN(Line$)+1
3240          Line$[Len]="-"
3245          Line$[Len+1]=Gridname$(K+1)
3250        END IF
3255        K=K+1
3260      END IF
3265      PRINT Line$
3270      J=J+1
3275    NEXT K
3280    PRINT "                A-   ENTER A NEW GRID"
3285    PRINT "                B-   PURGE AN ITEM"
3290    PRINT "                C-   GO BACK TO START"
3295    IF Numgrids=50 THEN
3300      PRINT "N.B.  You have reached the maximum number of grid files allowed"
3305      PRINT "No new grid shall be stored unless you PURGE an existing grid."
3310    END IF
3315    IF J>13 THEN
3320      PRINT "Use scrolling wheel to view all the menu items if necessary."
3325    END IF
3330    Line$=""
3335    INPUT "Enter ONLY ONE of the above digits to select that item.",Line$
3340    IF UPC$(Line$)="A" THEN Newgrid
3345    IF UPC$(Line$)="B" THEN Purge
3350    IF UPC$(Line$)="C" THEN Start
3355    ON ERROR GOTO 3375
3360    Type=VAL(Line$)
3365    OFF ERROR
3370    IF Type<0 OR Type>Numgrids THEN
3375      BEEP
3380      PRINT CHR$(12);TABXY(20,10);"Illegal response, try again."
3385      PRINT TABXY(20,12);"Press CONTINUE to proceed."
3390      PAUSE
3395      GOTO Over
3400    END IF
3405    Grid$=Gridname$(Type)
3410    Form$=Gridform$(Type)
3415    Imax=INT(G(Type,1))
3420    Jmax=INT(G(Type,2))
3425    FOR I=1 TO Imax
```

```
3430      Lx(I)=G(Type,I+2)                          !      \ ITEMS FROM SELECTED
3435    NEXT I                                       !      / GRID STORED ON DISC
3440    FOR J=1 TO Jmax                              !     /
3445      Ly(J)=G(Type,J+Imax+2)                     !    /
3450    NEXT J                                       !   /
3455    Form=1                                       !  /
3460    IF Form$[1,5]="POLAR" THEN Form=2            !/
3465    GOTO Manualdata
3470 Purge:!
3475    INPUT "Enter the item number to be purged  (enter 0 to abort purging)",Ty
pe
3480    IF Type<=0 OR Type>Numgrids THEN Over
3485    IF Type<Numgrids THEN
3490      Gridname$(Type)=Gridname$(Numgrids)
3495      Gridform$(Type)=Gridform$(Numgrids)
3500      FOR K=1 TO Numgrels
3505        G(Type,K)=G(Numgrids,K)
3510      NEXT K
3515    END IF
3520    Numgrels=0
3525    Numgrids=Numgrids-1
3530    IF Numgrids=0 THEN
3535      Numgrels=0
3540      PURGE "GRIDS"
3545      CREATE BDAT "GRIDS",170
3550      ASSIGN @Path TO "GRIDS"
3555      OUTPUT @Path;Numgrids,Numgrels
3560      ASSIGN @Path TO *
3565    ELSE
3570      GOSUB Checksize
3575      GOSUB Storedummy2
3580      GOSUB Record
3585    END IF
3590    GOTO Readgrids
3595                        !\
3600 Newgrid:!              ! >--SET UP NEW GRID SPECIFICATIONS---
3605                        !/
3610    PRINT CHR$(12);TABXY(1,1);"SELECT GRID SHAPE"
3615    PRINT TABXY(1,4);"1-   CARTESIAN"                !\
3620    PRINT TABXY(1,5);"2-   POLAR SECTOR"             !  \
3625    PRINT TABXY(1,6);"3-   POLAR ANNULUS"            !    \
3630    ON ERROR GOTO 3610                               !      \
3635    Line$=""                                         !       \
3640    INPUT "Enter appropriate number.",Line$          !        \ SPECIFY GRID SHAPE
3645    IF UPC$(Line$)="QUIT" THEN Start                 !        \FOR NEW GRID SPECS
3650    Form=VAL(Line$)                                  !         > TO BE ENTERED
3655    IF Form<1 OR Form>3 THEN 3610                    !        / MANUALLY ON THE
3660    IF Form=2 THEN Form$="POLARSEC"                  !       /     KEYBOARD
3665    IF Form=3 THEN                                   !      /
3670      Form$="POLARANN"                               !     /
3675      Form=2                                         !    /
3680    END IF                                           !   /
3685    IF Form=1 THEN Form$="CARTESIAN"                 !  /
3690    OFF ERROR                                        !/
3695    PRINT CHR$(12);TABXY(1,18);"how many ";Term$(1,Form);" co-ordinates are t
here?"
3700    INPUT Imax
3705    IF Imax<3 THEN
3710      PRINT "You must have at least 3 ";Term$(1,Form);" co-ordinates"
3715      BEEP
3720      GOTO 3700
3725    END IF
3730    IF Imax>50 THEN
3735      PRINT "You cannot have more than 50 ";Term$(1,Form);" co-ordinates"
3740      BEEP
3745      GOTO 3700
3750    END IF
3755    PRINT CHR$(12);TABXY(1,18);"how many ";Term$(2,Form);" co-ordinates are t
here?"
3760    INPUT Jmax
3765    IF Jmax<3 THEN
3770      PRINT "You must have at least 3 ";Term$(2,Form);" co-ordinates"
3775      BEEP
3780      GOTO 3760
3785    END IF
3790    IF Jmax>50 THEN
3795      PRINT "You cannot have more than 50 ";Term$(2,Form);" co-ordinates"
3800      BEEP
3805      GOTO 3760
3810    END IF
```

```
3815     IF Imax*Jmax>2000 THEN
3820       PRINT CHR$(12);TABXY(1,10);"Specified grid is too large. Max number of
data points is 2000."
3825       PRINT TABXY(1,12);"Entry of new grid has been aborted."
3830       PRINT TABXY(1,14);"Press CONTINUE to return to the GRID MENU."
3835       BEEP
3840       PAUSE
3845       GOTO Over
3850     END IF
3855     PRINT CHR$(12);"NOW ENTER THE ";Imax;" ";Term$(1,Form);" GRID POSITIONS"
3860     PRINT "The co-ordinates must be the running dimension from a common datum
."
3865     PRINT ""
3870     FOR I=1 TO Imax
3875       IF I=1 THEN
3880         INPUT "Enter first co-ordinate",Lx(1)
3885       ELSE
3890         INPUT "Enter next co-ordinate",Lx(I)
3895       END IF
3900       PRINT I;"-   ";"X(";I;")=";Lx(I);"           "
3905     NEXT I
3910     ON ERROR GOTO 3960
3915     Line$=""
3920     INPUT "If you wish to extend, or change the above, enter line number.",Li
ne$
3925     Line=VAL(Line$)
3930     OFF ERROR                              !!--CORRECT HORIZONTAL GRID LOCATIONS
3935     IF Line<1 OR Line>Imax+1 THEN 3960
3940     IF Line=Imax+1 THEN Imax=Imax+1
3945     INPUT "ENTER CORRECT VALUE",Lx(Line)
3950     PRINT Line;"-   ";"X(";Line;")=";Lx(Line);"          "
3955     GOTO 3910
3960     FOR I=1 TO Imax-1
3965       IF Lx(I+1)<=Lx(I) THEN
3970         BEEP
3975         PRINT CHR$(12);TABXY(1,1);Term$(1,Form);" positions conflict"
3980         PRINT TABXY(10,2);"-  must re-enter ";Term$(1,Form);" positions."
3985         PRINT TABXY(1,4);"NOTE:   The ";I+1;" grid position is not greater th
an the ";I;" grid position."
3990         PRINT TABXY(1,5);"        The grid positions must increase in value."
3995         PRINT TABXY(9,11);"Press CONTINUE to re-enter the grid positions."
4000         PRINT TABXY(1,13);"                                   OR"
4005         PRINT TABXY(9,15);"To return to the GRID MENU, type 'QUIT', then pres
s CONTINUE."
4010         Ans$=""
4015         INPUT "Enter response",Ans$
4020         IF Ans$="" THEN 3855
4025         GOTO Over
4030       END IF
4035     NEXT I
4040     PRINT CHR$(12);"NOW ENTER THE ";Jmax;" ";Term$(2,Form);" GRID POSITIONS"
4045     PRINT "The co-ordinates must be the running dimension from a common datum
."
4050     PRINT ""
4055     FOR J=1 TO Jmax
4060       IF J=1 THEN
4065         INPUT "Enter first co-ordinate",Ly(1)
4070       ELSE
4075         INPUT "Enter next co-ordinate",Ly(J)
4080       END IF
4085       PRINT J;"-     ";"Y(";J;")=";Ly(J);"           "
4090     NEXT J
4095     ON ERROR GOTO 4145
4100     Line$=""
4105     INPUT "If you wish to extend, or change the above, enter line number.",Li
ne$
4110     Line=VAL(Line$)
4115     OFF ERROR                              !!----CORRECT VERTICAL GRID LOCATIONS
4120     IF Line<1 OR Line>Jmax+1 THEN 4145
4125     IF Line=Jmax+1 THEN Jmax=Jmax+1
4130     INPUT "ENTER Y",Ly(Line)
4135     PRINT Line;"-     ";"Y(";Line;")=";Ly(Line);"       "
4140     GOTO 4095
4145     FOR J=1 TO Jmax-1
4150       IF Ly(J+1)<=Ly(J) THEN
4155         BEEP
4160         PRINT CHR$(12);TABXY(1,1);Term$(2,Form);" positions conflict"
4165         PRINT TABXY(10,2);"-  must re-enter ";Term$(2,Form);" positions."
4170         PRINT TABXY(1,4);"NOTE:   The ";J+1;" grid position is not greater th
an the ";J;" grid position."
```

```
4175        PRINT TABXY(1,5);"        The grid positions must increase in value."
4180        PRINT TABXY(9,11);"Press CONTINUE to re-enter the grid positions."
4185        PRINT TABXY(1,13);"                                    OR"
4190        PRINT TABXY(9,15);"To return to the GRID MENU, type 'QUIT', then pres
s CONTINUE."
4195        Ans$=""
4200        INPUT "Enter response",Ans$
4205        IF Ans$="" THEN 4040
4210        GOTO Over
4215      END IF
4220    NEXT J
4225    OFF ERROR
4230    PRINT CHR$(12);TABXY(1,10);"To store the data specifying this grid, enter
 'YES'."
4235    PRINT TABXY(1,12);"or else, simply press CONTINUE."
4240    Line$=""
4245    INPUT "Enter response.",Line$
4250    IF UPC$(Line$)="YES" THEN
4255      IF Numgrids=50 THEN
4260        PRINT CHR$(12);TABXY(1,10);"Space exceeded for the 'GRIDS' file."
4265        PRINT TABXY(1,12);"Storage of this new grid is aborted."
4270        PRINT TABXY(1,14);"Press CONTINUE to proceed."
4275        PAUSE
4280        GOTO Manualdata
4285      END IF
4290      PRINT CHR$(12)
4295      Grid$=""
4300      INPUT "Enter an appropriate grid name (e.g. HALF SCALE ESCORT) (40 chrs
. max.)",Grid$
4305      IF Grid$="" THEN
4310        BEEP
4315        PRINT TABXY(20,10);"You are required to enter an apprppriate name."
4320        GOTO 4300
4325      END IF
4330      IF LEN(Grid$)>40 THEN
4335        BEEP
4340        PRINT CHR$(12);TABXY(1,10);"Total length of the grid name exceeds max
imum of 40 characters."
4345        PRINT TABXY(20,13);"Try again."
4350        PRINT TABXY(10,16);"Press CONTINUE to proceed."
4355        PAUSE
4360        GOTO 4300
4365      END IF
4370      PRINT CHR$(12)
4375      Newtype$=Grid$
4380      FOR K=1 TO Numgrids
4385        IF Newtype$=Gridname$(K) THEN
4390          PRINT CHR$(12);TABXY(1,6);"This combination already exists."
4395          PRINT TABXY(1,8);"If you wish to update the grid specifications on
disc for the grid type:"
4400          PRINT TABXY(1,9);Newtype$;"  then enter 'YES'."
4405          PRINT TABXY(1,11),"              ELSE        "
4410          PRINT TABXY(1,13);"To abort data storage, enter 'NO'."
4415          PRINT TABXY(1,15),"              ELSE        "
4420          PRINT TABXY(1,17);"To enter a new grid specifying name simply press
 CONTINUE."
4425          Line$=""
4430          INPUT "Enter response.",Line$
4435          IF Line$="" THEN 4290
4440          IF UPC$(Line$)="NO" THEN Manualdata
4445          IF UPC$(Line$)="YES" THEN
4450            Gridname$(K)=Gridname$(Numgrids)
4455            Gridform$(K)=Gridform$(Numgrids)
4460            FOR N=1 TO Numgrels
4465              G(K,N)=G(Numgrids,N)
4470            NEXT N
4475            Numgrids=Numgrids-1
4480            GOTO Direct
4485          END IF
4490          BEEP
4495          PRINT CHR$(12);TABXY(10,10);"Try again."
4500          PRINT TABXY(10,12);"Press CONTINUE to proceed."
4505          GOTO 4390
4510        END IF
4515      NEXT K
4520    ELSE
4525      IF Line$="" THEN
4530        GOTO Manualdata
4535      ELSE
4540        PRINT TABXY(10,15);"Try again."
```

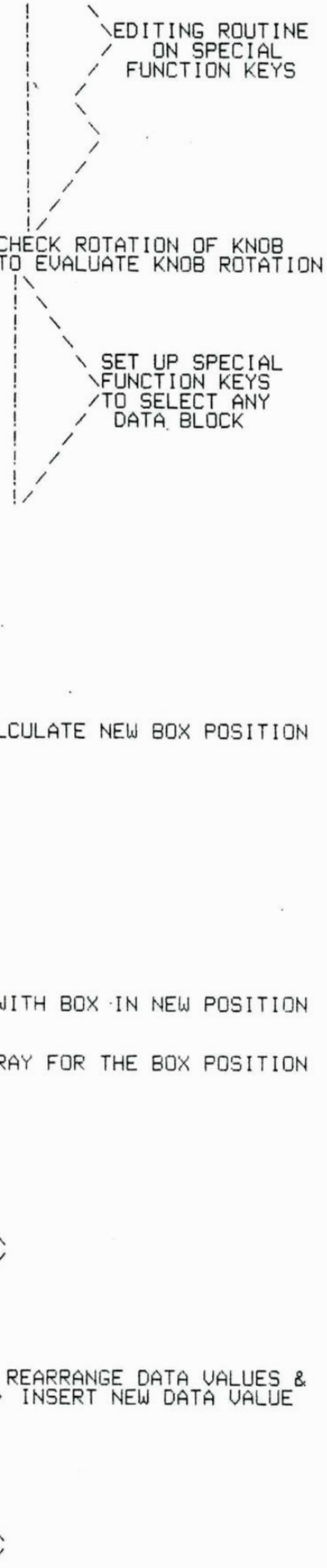
```
4545        GOTO 4240
4550      END IF
4555    END IF
4560 Direct:!
4565    Numgrels=Imax+Jmax+2
4570    GOSUB Checksize
4575    GOSUB Storedummy1
4580    GOSUB Add_on
4585    GOTO Manualdata
4590 Checksize:!
4595    FOR J=1 TO Numgrids
4600      Check=G(J,1)+G(J,2)+2
4605      IF Check>Numgrels THEN Numgrels=Check
4610    NEXT J
4615    RETURN
4620 Storedummy1:!
4625    Numgrids=Numgrids+1
4630    Flag_num=1
4635 Storedummy2:!
4640    FOR I=1 TO Numgrids-Flag_num
4645      FOR K=1 TO G(I,1)+G(I,2)+2
4650        Gdum(I,K)=G(I,K)
4655      NEXT K
4660      FOR J=K TO Numgrels
4665        Gdum(I,K)=0
4670      NEXT J
4675    NEXT I
4680    REDIM Gridname$(Numgrids),Gridform$(Numgrids),G(Numgrids,Numgrels)
4685    FOR I=1 TO Numgrids-Flag_num
4690      FOR K=1 TO Numgrels
4695        G(I,K)=Gdum(I,K)
4700      NEXT K
4705    NEXT I
4710    Flag_num=0
4715    RETURN
4720 Add_on:!
4725    Dum=Numgrids
4730    Gridname$(Dum)=Newtype$          !\
4735    Gridform$(Dum)=Form$             ! \
4740    G(Dum,1)=Imax                    !  \
4745    G(Dum,2)=Jmax                    !   \
4750    FOR I=1 TO Imax                  !    \STORE DATA ITEMS IN
4755      G(Dum,I+2)=Lx(I)               !    /MASS STORAGE ARRAYS
4760    NEXT I                           !   /
4765    FOR J=1 TO Jmax                  !  /
4770      G(Dum,J+2+Imax)=Ly(J)          ! /
4775    NEXT J                           !/
4780    FOR K=Jmax+2+Imax TO Numgrels
4785      G(Dum,K)=0
4790    NEXT K
4795 Record:!
4800    MASS STORAGE IS ":INTERNAL"
4805    Rec=170          !Rec=INT((Numgrids*(60+Numgrels*8)+20)/256)+1
4810    PURGE "GRIDS"
4815    CREATE BDAT "GRIDS",Rec
4820    ASSIGN @Path TO "GRIDS"
4825    OUTPUT @Path;Numgrids,Numgrels,Gridname$(*),Gridform$(*),G(*)
4830    ASSIGN @Path TO *
4835    RETURN
4840 Manualdata:     !\
4845                 ! >----------KEYBOARD DATA ENTRY-----------------
4850                 !/
4855    GOSUB Initialize
4860    FOR Sect=1 TO Width*Height
4865      PRINT CHR$(12);CHR$(133);TABXY(30,1);"INPUT DATA IN THE ORDER SHOWN."
4870      PRINT CHR$(128)
4875      Man_flag=1                  !\
4880      GOSUB Viewtable             ! \
4885      MOVE -5,-5                  !  \INITIALIZE FOR MANUAL DATA ENTRY
4890      K=0                         !  /
4895      PEN -1                      ! /
4900      MOVE Table(Sect,2),1        !/
4905      IF Sect=1 THEN
4910        PRINT TABXY(30,16);"Select the appropriate special function key"
4915        ON KEY 0 GOTO Loop1
4920        ON KEY 1 GOTO Loop1
4925        ON KEY 5 GOTO Loop1
4930        ON KEY 6 GOTO Loop1
4935        ON KEY 7 GOTO Loop1
4940        ON KEY 8 GOTO Loop1
```

```
4945        ON KEY 9 GOTO Loop1
4950        ON KEY 2 LABEL "ENTER DATA",15 GOTO Enter_data
4955        ON KEY 3 LABEL "GRID MENU",15 GOTO Over
4960        ON KEY 4 LABEL "DATA EDIT MENU",15 GOTO Menu
4965 Loop1: GOTO Loop1
4970 Enter_data:!
4975        PRINT TABXY(30,16);"                                                "
4980        OFF KEY
4985      END IF
4990      PRINT TABXY(1,2);"NOTE :  Enter the word 'QUIT' to abort data entry"
4995      FOR I=Table(Sect,1) TO Table(Sect,2)          !\
5000        FOR J=Table(Sect,3) TO Table(Sect,4)        ! \
5005          K=(I-1)*Jmax+J                            !  \
5010          IPLOT Box(*)                              !   \
5015          PEN 1                                     !    \
5020          MOVE I-.5,J-.3                            !    /
5025          IPLOT Box(*)                              !   /
5030          Line$=VAL$(F_org(K))                      !   \
5035          INPUT "ENTER DATA VALUE",Line$            !    \
5040          IF UPC$(Line$)="QUIT" THEN Menu           !     >MANUAL DATA ENTRY
5045          ON ERROR GOTO Err9                        !    /   AND DISPLAY
5050          F_org(K)=VAL(Line$)                       !   /
5055          PRINT TABXY(20,4);RPT$(" ",20)            !   \
5060          MOVE I,J                                  !    \
5065          LABEL F_org(K)                            !    /
5070          MOVE I-.5,J-.3                            !   /
5075          PEN -1                                    !  /
5080        NEXT J                                      ! /
5085        BEEP                                        !/
5090      NEXT I
5095      IPLOT Box(*)          !\
5100      WAIT 1.0              ! \END OF MANUAL DATA ENTRY DISPLAY
5105      GCLEAR                ! /
5110      GRAPHICS OFF          !/
5115    NEXT Sect
5120    OFF ERROR
5125    GOTO Menu
5130 Err9:!
5135    PRINT CHR$(133);TABXY(20,4);"TRY AGAIN"
5140    PRINT CHR$(128)
5145    BEEP
5150    GOTO 5030
5155 Initialize:    !\BREAK A LARGE DATA SET INTO BLOCKS SUITABLE FOR VIEWING
5160                !/
5165    PRINT CHR$(12)
5170    Width=INT(Imax/8)                           !\
5175    IF Imax MOD 8>0 THEN Width=Width+1          ! \ COMPUTE THE NUMBER OF X
5180    Height=INT(Jmax/8)                          !  > AND Y BLOCKS NEEDED TO
5185    IF Jmax MOD 8>0 THEN Height=Height+1        ! /     DISPLAY THE DATA
5190    Sect=0                                      !/
5195    FOR I=1 TO Width                                    !\
5200      FOR J=1 TO Height                                 ! \
5205        Sect=Sect+1                                     !  \
5210        Table(Sect,1)=1+(I-1)*8                         !   \
5215        Table(Sect,2)=I*8                               !    \
5220        IF I=Width THEN                                 !    /
5225          Table(Sect,2)=Imax                            !   /
5230          IF Width>1 THEN                               !  /
5235            Table(Sect-Height,2)=INT((Table(Sect-Height,1)+Imax)/2)
5240            Table(Sect,1)=Table(Sect-Height,2)+1        !  \
5245          END IF                                        !   \COMPUTE THE X AND Y
5250        END IF                                          !    \POSITIONS FOR THE
5255        Table(Sect,3)=1+(J-1)*8                         !    /START AND END OF
5260        Table(Sect,4)=J*8                               !   /EACH DISPLAY BLOCK
5265        IF J=Height THEN                                !  /
5270          Table(Sect,4)=Jmax                            ! /
5275          IF Height>1 THEN                              !  \
5280            Table(Sect-1,4)=INT((Table(Sect-1,3)+Jmax)/2)
5285            Table(Sect,3)=Table(Sect-1,4)+1             !   \
5290          END IF                                        !   /
5295        END IF                                          !  /
5300      NEXT J                                            ! /
5305    NEXT I                                              !/
5310    Sect=1
5315    RETURN
5320 Menu:      !\
5325            ! >---------------------SET UP MENU-------------------
5330            !/
5335    OFF KEY
5340    GCLEAR
```

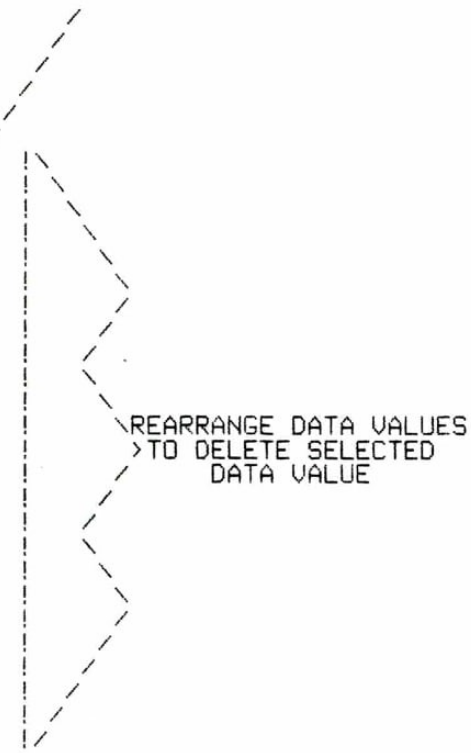
```
5345     Man_flag=0
5350     Same_flag=1
5355     Row=Table(Sect,3)
5360     Col=Table(Sect,1)
5365     PRINT CHR$(12);TABXY(20,1);">>>>>   DATA EDITING MENU   <<<<<"
5370     PRINT CHR$(Dis(1));TABXY(1,5);"1- VERIFY / CHANGE DATA?"
5375     PRINT CHR$(Dis(2));TABXY(40,5);"2- STORE DATA ON DISC?"
5380     PRINT CHR$(Dis(3));TABXY(1,6);"3- NORMALIZE DATA?"
5385     PRINT CHR$(Dis(4));TABXY(40,6);"4- VERIFY / CHANGE NORMALIZED DATA?"
5390     PRINT CHR$(Dis(8));TABXY(1,7);"5- GO BACK TO START OF DATA ENTRY"
5395     PRINT CHR$(Dis(9));TABXY(40,7);"6- LEAVE DATA EDITING MENU"
5400     PRINT CHR$(128)
5405     ON ERROR GOTO 5410
5410     Line$=""
5415     INPUT "TO SELECT ANY OF THE ABOVE, INPUT THE LINE NUMBER",Line$
5420     IF Line$="" THEN End
5425     Line=VAL(Line$)
5430     OFF ERROR
5435                       !!---------DETERMINE NEW VALUE OF MENU ITEMS----------
5440     IF Line=1 THEN
5445       GOTO Original         !!GOTO ROUTINE TO VERIFY & CORRECT ORIGINAL DATA
5450     ELSE
5455       IF Line=2 THEN
5460         GOTO Store                  !!--GOTO ROUTINE TO STORE KEYBOARD DATA
5465       ELSE
5470         IF Line=3 THEN
5475           GOTO Normalize            !!------SETUP DATA AS RATIOS
5480         ELSE
5485           IF Line=4 THEN
5490             GOTO Derived    !GOTO ROUTINE TO VERIFY/CORRECT NORMALIZED DATA
5495           ELSE
5500             IF Line=5 THEN
5505               GOTO Start            !GO BACK TO START OF DATA INPUT
5510             ELSE
5515               IF Line=6 THEN
5520                 GOTO End            !!----------LEAVE SUBPROGRAM----------
5525               ELSE
5530                 GOTO 5405              !GO BACK & SELECT VALID MENU ITEM
5535               END IF
5540             END IF
5545           END IF
5550         END IF
5555       END IF
5560     END IF
5565 B:  Dis(Old)=128          !\
5570     Dis(Line)=129         ! >DETERMINE SHADING OF MENU ITEM
5575     Old=Line              !/
5580     GOTO Menu                    !!-------------RETURN TO MENU-------------
5585 Original:            !
5590     FOR K=1 TO Imax*Jmax         !\
5595       F(K)=F_org(K)              ! \
5600     NEXT K                       !  \
5605     GOSUB Verify                 !   >SELECT ORIGINAL DATA TO BE CHECKED
5610     FOR K=1 TO Imax*Jmax         !  /
5615       F_org(K)=F(K)              ! /
5620     NEXT K                       !/
5625     GOTO B
5630 Derived:             !
5635     FOR K=1 TO Imax*Jmax         !\
5640       F(K)=F_norm(K)             ! \
5645     NEXT K                       !  \
5650     GOSUB Verify                 !   >SELECT NORMALIZED DATA TO BE CHECKED
5655     FOR K=1 TO Imax*Jmax         !  /
5660       F_norm(K)=F(K)             ! /
5665     NEXT K                       !/
5670     GOTO B
5675 Verify:                         !!---------ROUTINE TO CORRECT DATA---------
5680     IF Device$="DISC" THEN
5685       PRINT CHR$(12);TABXY(1,4);"Here is the data from file ";File$
5690     ELSE
5695       PRINT CHR$(12);TABXY(1,4);"Here is the data entered at the keyboard"
5700     END IF
5705     PRINT TABXY(1,5);Form$;" grid"
5710     Sect=1
5715 Viewtable:  !\
5720             ! >   OUTPUT DATA ARRAY ON SCREEN
5725             !/
5730     GRAPHICS ON
5735     GINIT
5740     IF Width*Height>1 THEN
```

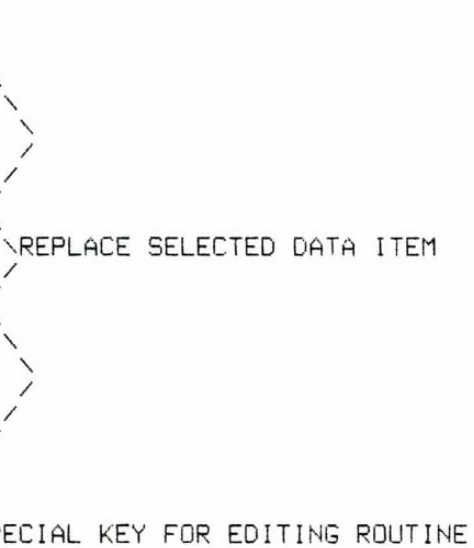
```
5745      VIEWPORT 100,140,75,100              !!----SETUP VIEWPORT & WINDOW----
5750      WINDOW .5,Imax+.5,Jmax+.5,.5
5755      FOR I=1 TO Width-1                          !\
5760        K=I*Height+1                              ! \
5765        MOVE Table(K,1)-.5,.5                     !  \
5770        DRAW Table(K,1)-.5,Jmax+.5                !   \
5775      NEXT I                                      !    \
5780      FOR J=2 TO Height                           !     \
5785        MOVE .5,Table(J,3)-.5                     !      \
5790        DRAW Imax+.5,Table(J,3)-.5                !       \PLOT DATA BLOCKS IN
5795      NEXT J                                      !       /A REFERENCE RECTANGLE
5800      FRAME                                       !      /
5805      FOR K=1 TO 4                                !     /
5810        Box2(K,1)=Table(Sect,2.5-ABS(K-2.5))+1-ABS(K-2.5)
5815        Box2(K,2)=Table(Sect,3+INT((K-1)/2))-.5+INT((K-1)/2)
5820      NEXT K                                      !   /
5825      AREA INTENSITY 1,1,1                        !  /
5830      PLOT Box2(*),FILL                           !/
5835    END IF
5840    VIEWPORT 0,140,25,78                  !!----SETUP VIEWPORT & WINDOW----
5845    WINDOW Table(Sect,1)-2.5,Table(Sect,2)+.5,Table(Sect,4)+.5,Table(Sect,3)-
1.5
5850    AXES 0,0,Table(Sect,1)-1,Table(Sect,3)-.5     !!-------DRAW AXES-------
5855    CSIZE 3.2,.45
5860    MOVE (Table(Sect,2)+Table(Sect,1))/2-.5,Table(Sect,3)-1.5
5865    LORG 6                                      !\
5870    LABEL Heading$(1,Form)                      ! \
5875    FOR I=Table(Sect,1) TO Table(Sect,2)        !  \---SETUP COLUMN HEADING---
5880      MOVE I,Table(Sect,3)-1                    !  /
5885      LABEL Lx(I)                               ! /
5890    NEXT I                                      !/
5895    MOVE Table(Sect,1)-2.4,(Table(Sect,3)+Table(Sect,4))/2
5900    DEG                                         !\
5905    LDIR 90                                     ! \
5910    LABEL Heading$(2,Form)                      !  \
5915    LDIR 0                                      !   \
5920    LORG 5                                      !    > SETUP ROW HEADING
5925    FOR J=Table(Sect,3) TO Table(Sect,4)        !   /
5930      MOVE Table(Sect,1)-1.7,J                  !  /
5935      LABEL Ly(Jmax-J+1)                        ! /
5940    NEXT J                                      !/
5945    CSIZE 2.9,.51
5950    IF Man_flag=1 THEN RETURN  !!RETURN BACK TO KEYBOARD DATA ENTRY SECTION
5955    ON ERROR GOTO 5990
5960    FOR I=Table(Sect,1) TO Table(Sect,2)          !\
5965      FOR J=Table(Sect,3) TO Table(Sect,4)        ! \
5970        MOVE I,J                                  !  \
5975        K=(I-1)*Jmax+J                            !   \
5980        LABEL USING "DDDD.DDD";F(K)               !    >OUTPUT DATA TABLE
5985        GOTO 5995                                 !   /
5990        LABEL F(K)                                !  /
5995      NEXT J                                      ! /
6000    NEXT I                                        !/
6005    OFF ERROR
6010    PEN -1
6015    MOVE 0,Table(Sect,4)
6020    IDRAW 0,.5
6025    PEN 1
6030    PRINT CHR$(132)
6035    PRINT TABXY(1,1);"SELECT & CHANGE DATA ITEM BY MOVING THE BOX USING THE K
NOB."
6040    PRINT TABXY(1,2);"THEN CHOOSE EDITING OPERATION FROM KEY SELECTION (below
)"
6045    PRINT CHR$(128)
6050    BEEP
6055    Row=Table(Sect,3)       !\
6060    Col=Table(Sect,1)       ! \
6065    K=(Col-1)*Jmax+Row      !  \RE-INITIALIZE BOX INDICES
6070    Total=0                 !  /
6075    MOVE Col-.5,Row-.3      ! /
6080    IPLOT Box(*)            !/
6085 Keys:!
6090    ON KEY 0 LABEL "CHANGE ITEM",3 GOTO Change        !\
6095    ON KEY 1 LABEL "ADD AN ITEM",3 GOTO Insert        ! \
6100    ON KEY 2 LABEL "DELETE ITEM",3 GOTO Delete        !  \
6105    ON KEY 3 LABEL "QUIT EDITING",3 GOTO Ret          !   \
6110    ON KEY 4 GOTO Spin                                !    \
6115    IF Height>1 THEN                                  !    /
6120      ON KEY 5 LABEL "VIEW UP",3 GOTO Viewup          !   /
6125      ON KEY 6 LABEL "VIEW DOWN",3 GOTO Viewdown      !   \
```

```
    ELSE
      ON KEY 5 GOTO Spin
      ON KEY 6 GOTO Spin
    END IF
    IF Width>1 THEN
      ON KEY 7 LABEL "VIEW LEFT",3 GOTO Viewleft
      ON KEY 8 LABEL "VIEW RIGHT",3 GOTO Viewright
    ELSE
      ON KEY 7 GOTO Spin
      ON KEY 8 GOTO Spin
    END IF
    ON KEY 9 GOTO Spin
    ON KNOB .01,2 GOSUB Move_box   !EVERY SO OFTEN CHECK ROTATION OF KNOB
Spin:GOTO Spin  !LOOP WHILE TIMER DETERMINES WHEN TO EVALUATE KNOB ROTATION
Viewup:!
    IF Sect MOD Height<>1 THEN Sect=Sect-1
    GOTO Viewtable
Viewdown:!
    IF Sect MOD Height<>0 THEN Sect=Sect+1
    GOTO Viewtable
Viewleft:!
    IF Sect>Height THEN Sect=Sect-Height
    GOTO Viewtable
Viewright:!
    IF Sect<=(Width-1)*Height THEN Sect=Sect+Height
    GOTO Viewtable
Move_box:!
    Total=Total+KNOBX
    IF ABS(Total)>20 THEN
      Row=Row+ABS(Total)/Total
      Total=0
      IF Row>Table(Sect,4) THEN
        Row=Table(Sect,3)
        Col=Col+1
      ELSE
        IF Row<Table(Sect,3) THEN
          Row=Table(Sect,4)
          Col=Col-1
        END IF
      END IF
      IF Col<Table(Sect,1) THEN
        Row=Table(Sect,3)
        Col=Table(Sect,1)
      ELSE
        IF Col>Table(Sect,2) THEN
          Col=Table(Sect,2)
          Row=Table(Sect,4)
        END IF
      END IF
      PEN -1
      IPLOT Box(*)
      MOVE Col-.5,Row-.3
      PEN 1
      IPLOT Box(*)
      K=(Col-1)*Jmax+Row    !COMPUTE LOCATION IN ARRAY FOR THE BOX POSITION
    END IF
    RETURN
Insert:
    OFF KNOB
    OFF KEY
    L=0
    FOR I=Col TO Table(Sect,2)
      Start=Table(Sect,3)
      IF I=Col THEN Start=Row
      FOR J=Start TO Table(Sect,4)
        L=L+1
        N=(I-1)*Jmax+J
        Temp(L)=F(N)
      NEXT J
    NEXT I
    L=0
    FOR I=Col TO Table(Sect,2)
      Start=Table(Sect,3)
      IF I=Col THEN Start=Row+1
      FOR J=Start TO Table(Sect,4)
        L=L+1
        N=(I-1)*Jmax+J
        F(N)=Temp(L)
      NEXT J
    NEXT I
```



```
6530     ON ERROR GOTO 6535
6535     Line$=""
6540     INPUT "Enter new data value.",Line$
6545     F(K)=VAL(Line$)
6550     OFF ERROR
6555     GOTO Verify
6560 Delete:
6565     OFF KNOB
6570     OFF KEY
6575     L=0
6580     FOR I=Col TO Table(Sect,2)
6585       Start=Table(Sect,3)
6590       IF I=Col THEN Start=Row+1
6595       FOR J=Start TO Table(Sect,4)
6600         L=L+1
6605         N=(I-1)*Jmax+J
6610         Temp(L)=F(N)
6615       NEXT J
6620     NEXT I
6625     L=0
6630     FOR I=Col TO Table(Sect,2)
6635       Start=Table(Sect,3)
6640       IF I=Col THEN Start=Row
6645       FOR J=Start TO Table(Sect,4)
6650         L=L+1
6655         N=(I-1)*Jmax+J
6660         F(N)=Temp(L)
6665       NEXT J
6670     NEXT I
6675     F((Table(Sect,2)-1)*Jmax+Table(Sect,4))=0
6680     GOTO Verify
6685 Change:!
6690     OFF KNOB
6695     OFF KEY
6700     PRINT TABXY(1,4);"The old value at this location is: ";F(K);"         "
6705     Dummy=F(K)
6710     ON ERROR GOTO 6720
6715     Line$=""
6720     INPUT "Enter new data value.",Line$
6725     IF Line$="" THEN Ret
6730     F(K)=VAL(Line$)
6735     OFF ERROR
6740     LORG 5
6745     PEN -1
6750     MOVE Col,Row
6755     ON ERROR GOTO 6770
6760     LABEL USING "DDDD.DDD";Dummy
6765     GOTO 6775
6770     LABEL Dummy
6775     PEN 1
6780     MOVE Col,Row
6785     ON ERROR GOTO 6800
6790     LABEL USING "DDDD.DDD";F(K)
6795     GOTO 6805
6800     LABEL Dummy
6805     OFF ERROR
6810     MOVE Col-.5,Row-.3
6815     GOTO Keys              !GO BACK TO SELECT SPECIAL KEY FOR EDITING ROUTINE
6820 Ret:OFF ERROR
6825     OFF KNOB
6830     OFF KEY
6835     RETURN      !!----END OF DATA CORRECTION ROUTINE----
6840 Store:                              !!ROUTINE TO STORE KEYBOARD ENTERED DATA
6845     Abort_flag=0
6850     PRINT CHR$(12);TABXY(1,4);"NOTE: If you are attempting to update an exist
ing file,"
6855     PRINT TABXY(1,5);"    then the disc with the old file must be in drive #1
"
6860     IF Device$<>"KEYBOARD" THEN
6865       PRINT TABXY(1,9);"The current file name is ";File$
6870       PRINT TABXY(1,12);"You have the option of specifying a new file name"
6875       PRINT TABXY(1,13);"                                    OR"
6880       PRINT TABXY(1,14);"by simply pressing CONTINUE, you can keep the curren
t name."
6885       Line$=""
6890       INPUT "Enter response.",Line$
6895       IF Line$<>"" THEN File$=Line$
6900     ELSE
6905       INPUT "Enter the desired file name  (10 chrs. max.)",File$
6910       IF File$="" OR LEN(File$)>10 THEN
```





```
6915          PRINT CHR$(12);TABXY(10,10);"Illegal file name, try again."
6920          BEEP
6925          GOTO 6905
6930        END IF
6935      END IF
6940      ON ERROR GOSUB Error
6945      MASS STORAGE IS ":INTERNAL,4,1"
6950      Newmax=Imax+Jmax*(1+Imax)
6955      Total=280+Newmax*8
6960      Rec=INT((Total+20)/256)+1
6965      IF Abort_flag=0 THEN CREATE BDAT File$,Rec
6970      OFF ERROR
6975      IF Abort_flag=1 THEN
6980        PRINT CHR$(12);TABXY(1,10);"Data storage has been aborted."
6985        BEEP 200,1.0
6990      ELSE
6995        REDIM A(Newmax)                        !\
7000        FOR K=1 TO Imax                        ! \
7005          A(K)=Lx(K)                           !  \
7010        NEXT K                                 !   \
7015        FOR K=1 TO Jmax                        !    \STORE NEW DATA ON DISC
7020          A(K+Imax)=Ly(K)                      !    /  MASS STORAGE ARRAYS
7025        NEXT K                                 !   /
7030        FOR K=1 TO Imax*Jmax                   !  /
7035          A(K+Imax+Jmax)=F_org(K)              ! /
7040        NEXT K                                 !/
7045        ASSIGN @Path TO File$
7050        OUTPUT @Path;Time;Form$;Labels$(*);Imax;Jmax;A(*)
7055        ASSIGN @Path TO *
7060        PRINT CHR$(12);TABXY(1,10);"The data has been stored in the file:  ";Fi
le$
7065      END IF
7070      PRINT TABXY(1,12);"Press CONTINUE to proceed."
7075      PAUSE
7080      GOTO B          !
7085          !\
7090          ! >MESSAGES AND ROUTINES FOR POSSIBLE SYSTEM ERRORS
7095          !/
7100 Error:!
7105      IF ERRN=54 THEN         !--DUPLICATE FILE NAME
7110                              !--SET UP MENU
7115        PRINT CHR$(12);TABXY(1,4);"The file ";File$;" does exist."
7120        PRINT TABXY(1,6);"To update the old file with the current data, enter
'YES'"
7125        PRINT TABXY(1,8);"                                OR"
7130        PRINT TABXY(1,10);"To store data in a new file enter a new file name."
7135        PRINT TABXY(1,12);"                                OR"
7140        PRINT TABXY(1,14);"To abort data storage, enter 'NO'"
7145                              !--SELECT OPTION FROM MENU
7150        Line$=""
7155        INPUT "Enter term",Line$
7160        IF Line$="YES" THEN
7165          PURGE File$
7170        ELSE
7175          IF Line$="NO" THEN
7180            Abort_flag=1          !SET FLAG TO ABORT DATA STORAGE ROUTINE
7185          ELSE
7190            File$=Line$
7195            IF File$="" OR LEN(File$)>10 THEN
7200              PRINT TABXY(1,16);"Illegal response, try again  (10 chrs max. in
a file name)."
7205              BEEP
7210              GOTO 7150
7215            END IF
7220          END IF
7225        END IF
7230      ELSE
7235        IF ERRN=64 OR ERRN=55 THEN
7240          PRINT CHR$(12);TABXY(1,10);"No more space on the present disc to crea
te a new file"
7245          PRINT TABXY(1,11);"Remove present disc from drive #1 and insert a new
 one."
7250          PRINT TABXY(1,12);"Press CONTINUE once the disc has been inserted."
7255        ELSE
7260          IF ERRN=84 THEN
7265            PRINT CHR$(12);TABXY(1,10);"This disc has yet to be initialized."
7270            PRINT TABXY(1,12);"Please be patient since this process will take a
 few minutes."
7275            BEEP
7280            WAIT 2.0
```

```
7285          ON ERROR GOTO Error
7290          INITIALIZE ":INTERNAL,4,1"
7295          BEEP
7300          PRINT CHR$(12);TABXY(1,8);"The disc has now been initialized."
7305          PRINT TABXY(1,10);"To continue, press CONTINUE."
7310        ELSE
7315          IF ERRN=66 THEN
7320            PRINT CHR$(12);TABXY(1,8);"This new disc has too many bad tracks
and cannot be initialized."
7325            PRINT TABXY(1,10);"Remove present disc from drive #1 and insert a
 new one."
7330            PRINT TABXY(1,12);"Press CONTINUE once the disc has been inserted
."
7335          ELSE
7340            IF ERRN=80 THEN
7345              PRINT CHR$(12);TABXY(1,8);"Disc door is open, please close it."
7350              PRINT TABXY(20,10);"OR"
7355              PRINT TABXY(1,12);"There is no disc present in drive #1, please
 insert one."
7360              PRINT TABXY(1,14);"Press CONTINUE once the error has been corre
cted."
7365            ELSE
7370              IF ERRN=81 THEN
7375                PRINT CHR$(12);TABXY(1,8);"Mass storage hardware failure."
7380                PRINT TABXY(1,10);"Try removing present disc from drive #1 an
d then re-inserting it."
7385                PRINT TABXY(1,12);"Press CONTINUE once the disc has been inse
rted."
7390              ELSE
7395                IF ERRN=56 OR ERRN=53 THEN
7400                  IF ERRL(6965)=1 THEN
7405                    PRINT CHR$(12);TABXY(1,10);"Illegal file name  (max. of 1
0 chars.)"
7410                  ELSE
7415                    PRINT CHR$(12),TABXY(1,3);"The file ";File$;" does not ex
ist on this disc."
7420                    IF ERRL(2800)=1 THEN 2695
7425                    PRINT TABXY(1,5);"Enter a new file name and/or insert a n
ew disc in drive #1"
7430                  END IF
7435                  BEEP
7440                  File$=""
7445                  INPUT "Re-enter data file name.",File$
7450                  GOTO 7515
7455                ELSE
7460                  PRINT CHR$(12),TABXY(1,10);"Error # ";ERRN;" has occurred."
7465                  BEEP
7470                  PAUSE
7475                END IF
7480              END IF
7485            END IF
7490          END IF
7495        END IF
7500      END IF
7505      BEEP
7510      PAUSE
7515    END IF
7520    IF ERRL(2800)=1 THEN 2800
7525    RETURN      !!RETURN AFTER APPROPRIATE ERROR MESSAGE HAS BEEN DISPLAYED
7530 Normalize:                 !!----RE-EVALUATE DATA AS RATIOS FOR GREATER---
7535                            !!       EASE OF COMPARISION BETWEEN PLOTS
7540    PRINT CHR$(12);TABXY(20,1);"Normalizing data"
7545    PRINT TABXY(1,4);"The default normalizing factor is the data set's arithm
etic mean."
7550    PRINT TABXY(1,6);"However one may ENTER any factor."
7555    PRINT TABXY(1,7);"     For example, a factor of -1 will invert the data w
hich will "
7560    PRINT TABXY(1,8);"give a different perspective to the 3-dimensional plot.
"
7565    PRINT TABXY(1,9);"     Also, one may wish to enter the integrated mean, c
omputed"
7570    PRINT TABXY(1,10);"when plotting the original data set."
7575    PRINT TABXY(1,13);"To keep the default factor,simply press CONTINUE."
7580    PRINT TABXY(1,15);"To abort data normalizing, ENTER a factor of 0.0."
7585    Ans$=""
7590    INPUT "Enter response.",Ans$
7595    IF Ans$="" THEN
7600      Total=0                       !\
7605      FOR K=1 TO Imax*Jmax          ! \
7610        Total=Total+F_org(K)        !  >  COMPUTE ARRAY AVERAGE
```

```
7615      NEXT K                                  ! /
7620      Avg=Total/(Imax*Jmax)                   !/
7625    ELSE
7630      ON ERROR GOTO 7685
7635      Avg=VAL(Ans$)
7640      OFF ERROR
7645      IF Avg=0 THEN Menu
7650    END IF
7655    Norm_flag=1           !--SET NORMALIZING FLAG
7660    FOR K=1 TO Imax*Jmax                    !\
7665      F_norm(K)=F_org(K)/Avg                ! >NORMALIZE DATA POINTS
7670    NEXT K                                  !/
7675    OFF ERROR
7680    GOTO B          !----RETURN TO MENU
7685    PRINT TABXY(1,18);"Illegal response, try again."
7690    GOTO 7585
7695 End:                                       !!------------LEAVE SUBPROGRAM-------------
7700    IF Norm_flag=1 THEN                     !\
7705      FOR K=1 TO Imax*Jmax                  ! \
7710        F(K)=F_norm(K)                      !  \
7715      NEXT K                                !   \PUT NORMALIZED DATA INTO PLOTTING
7720    ELSE                                    !    >ARRAY IF ORIGINAL DATA HAD BEEN
7725      FOR K=1 TO Imax*Jmax                  !   /  NORMALIZED EARLIER, ELSE USE
7730        F(K)=F_org(K)                       !  /       THE ORIGINAL DATA
7735      NEXT K                                ! /
7740    END IF                                  !/
7745    PRINT CHR$(12)        !\CLEAR THE SCREEN
7750    GRAPHICS OFF          !/
7755  SUBEND
7760  !
7765  !
7770  !!*************************************************************************
7775  !!                                                                        *
7780  !!          Plot_grid :   1st level SUBROUTINE TO PLOT BASIC GRID         *
7785  !!                                                                        *
7790  !!*************************************************************************
7795  SUB Plot_grid(Istep,Digipt(*),Flag$(*),Char_size,Char_title,Damp,Avg,INTEGE
R Cont_flag,Kmax,Pm1)
7800    OPTION BASE 1
7805    COM /Gen/ Lx(*),Ly(*),Form$,INTEGER Imax,Jmax
7810    COM /Gen2/ F(*)
7815    COM /Colours/ INTEGER Colour(*),Plot_flag
7820    COM /Interupt/ INTEGER Interupt
7825    COM /Global/ File$,Grid$,INTEGER Form
7830    COM /Change/ Factor,Distort,Gamma
7835    INTEGER I,J,K,Flag,Lorg,Line,Grid_flag
7840    DIM R(2),U(4),W(4),String$[100]
7845    REAL Alpha,Beta,Check,Cx,Cy,D,Dif12x,Dif12y,Dif13x,Dif13y,Dum
7850    REAL Ratio,St,Step,Tic,X,Xm,Y,Ym
7855    ON KEY 9 LABEL "INTERUPT",15 GOTO Interupt
7860 Startgrid: !
7865    DEG
7870    Pm1=0
7875    PRINT CHR$(12)
7880    Gamma=0
7885    Grid_flag=0
7890    GRAPHICS ON
7895    GINIT
7900    PEN Colour(1)
7905    IF Plot_flag=0 THEN     !!-----------SELECT PLOTTING DEVICE-------------
7910      IF Form$="CARTESIAN" THEN
7915        Ratio=(Lx(Imax)-Lx(1))/(Ly(Jmax)-Ly(1))   !\
7920        IF Ratio>1.5 THEN                         ! \
7925          VIEWPORT 10,130,55-60/Ratio,55+60/Ratio!  \
7930          D=120*.09875                            !   \ SET UP SCREEN
7935        ELSE                                      !    >   VIEWPORT
7940          VIEWPORT 10,10+Ratio*80,15,90           !   /
7945          D=Ratio*80*.09875                       !  /
7950        END IF                                    ! /
7955        Istep=.02*(Lx(Imax)-Lx(1))/D              !/
7960        WINDOW Lx(1),Lx(Imax),Ly(1),Ly(Jmax)
7965        IF Flag$(5)="YES" THEN FRAME
7970        GOSUB Cosmetics
7975      ELSE
7980        IF Form$="POLARANN" THEN
7985          VIEWPORT 0,131,0,100    !\
7990          WINDOW 0,131,0,100      ! \
7995          Digipt(1,1,1)=60        !  \
8000          Digipt(2,1,1)=55        !   \SET UP POLAR ANNULAR
8005          Digipt(1,2,1)=60        !   /   VIEWPORT
```

```
8010          Digipt(2,2,1)=95          !  /
8015          Flag$(9)="YES"            ! /
8020          GOSUB Gridplot            !/
8025        ELSE
8030          Gamma=90-(Lx(Imax)+Lx(1))/2
8035          FOR I=1 TO Imax
8040            Lx(I)=Lx(I)+Gamma
8045          NEXT I
8050          IF Lx(1)<0 THEN
8055            Ratio=2/(1-SIN(Lx(1)))
8060          ELSE
8065            Ratio=2*COS(Lx(1))/((1-Ly(1)/Ly(Jmax))*SIN(Lx(1)))
8070          END IF
8075          Factor=360/((Ly(Jmax)+Ly(1))*PI)
8080          GOSUB Factor
8085          IF Ratio>1.5 THEN
8090            VIEWPORT 10,130,55-60/Ratio,55+60/Ratio
8095          ELSE
8100            VIEWPORT 10,10+Ratio*80,15,95
8105          END IF
8110          IF Lx(1)<0 THEN                  !\
8115            W(1)=-Ly(Jmax)                 ! \
8120            W(3)=Ly(Jmax)*SIN(Lx(1))       !  \
8125          ELSE                             !   \
8130            W(1)=-Ly(Jmax)*COS(Lx(1))      !    \SET UP POLAR
8135            W(3)=Ly(1)*SIN(Lx(1))          !    /SECTOR VIEWPORT
8140          END IF                           !   /
8145          W(2)=-W(1)                       !  /
8150          W(4)=Ly(Jmax)                    ! /
8155          WINDOW W(1),W(2),W(3),W(4)       !/
8160          CLIP OFF
8165          Istep=.2025
8170          GOSUB Cosmetics
8175        END IF
8180      END IF
8185      SUBEXIT
8190    ELSE
8195      GOTO 8225
8200 Timeout:!
8205      PRINT CHR$(12),TABXY(10,10);"Plotter has not been powered up."
8210      PRINT TABXY(10,12);"Press CONTINUE when the plotter is ready."
8215      BEEP
8220      PAUSE
8225      ON TIMEOUT 7,.5 GOTO Timeout
8230      PLOTTER IS 705,"HPGL"            !\
8235      OFF TIMEOUT                      ! \
8240      GRAPHICS INPUT IS 705,"HPGL"     !  \SET UP ENTIRE PLOTTER PLATEN
8245      VIEWPORT 0,152,0,100             !  / AS A PLOTTING SURFACE TO
8250      WINDOW 0,152,0,100               ! /  ACCEPT ANY DIGITIZATIONS
8255      K=0                              !/
8260            !\
8265            ! >----------------------SET UP PLOT MENU--------------------
8270 Menu:     !/
8275      PRINT CHR$(12)
8280 Menu1:!
8285      PRINT TABXY(1,1),"                          >>>>>   PLOT MENU  #2   <<<<<"
8290      PRINT TABXY(10,4);"1-RECALL PREVIOUS FRAMES"
8295      PRINT TABXY(10,5);"2-SETUP NEW GRID FRAME"
8300      PRINT TABXY(10,6);"3-DRAW ADDITIONAL RECTANGULAR FRAMES"
8305      PRINT TABXY(10,7);"4-PRINT LABELS"
8310      PRINT TABXY(10,8);"5-GO BACK TO PLOT MENU #1"
8315      PRINT TABXY(10,9);"6-COMMENCE PLOTTING"
8320      ON ERROR GOTO Menu
8325      Line$=""
8330      BEEP
8335      INPUT "TO SELECT ANY OF THE ABOVE, INPUT THE LINE NUMBER",Line$
8340      IF Line$="" THEN Line$="6"
8345      Line=VAL(Line$)
8350      IF Line=1 OR Line=2 THEN
8355        IF Line=1 THEN
8360          IF Cont_flag=1 THEN
8365            Flag=1
8370            GOSUB Gridplot
8375            GOSUB Framing
8380          ELSE
8385            PRINT TABXY(10,14);"A previous frame does not exist.       "
8390            GOTO Menu1
8395          END IF
8400        ELSE
8405          Flag=0
```

```
8410          GOSUB Gridplot
8415          Cont_flag=1
8420        END IF
8425      ELSE
8430        IF Grid_flag=1 THEN
8435          IF Line=3 THEN
8440            GOSUB Digi
8445            GOSUB Framing
8450          END IF
8455          IF Line=4 THEN GOSUB Title
8460          IF Line=5 THEN
8465            CALL Initialize(Flag$(*))
8470            PRINT CHR$(12)
8475            Pm1=1
8480            SUBEXIT
8485          END IF
8490          IF Line=6 THEN
8495            VIEWPORT V(1),V(2),V(3),V(4)
8500            WINDOW W(1),W(2),W(3),W(4)
8505            SUBEXIT
8510          END IF
8515        ELSE
8520          IF Line=5 THEN 8465
8525          PRINT TABXY(10,14);"You must select item 1 or 2 to begin.   "
8530          GOTO Menu1
8535        END IF
8540      END IF
8545      GOTO Menu
8550    END IF
8555 Gridplot:   !
8560    CALL Initialize(Flag$(*))
8565    Grid_flag=1
8570    IF Form$[1,5]="POLAR" THEN
8575      DEG
8580      IF Form$="POLARANN" THEN
8585        IF Flag=0 AND Plot_flag=1 THEN GOSUB Annulus
8590        Dif12x=Digipt(1,2,1)-Digipt(1,1,1)
8595        Dif12y=Digipt(2,2,1)-Digipt(2,1,1)
8600        R(2)=SQR(Dif12x^2+Dif12y^2)
8605        DISABLE
8610        IF Flag$(9)="NO" THEN
8615          Dif13x=Digipt(1,3,1)-Digipt(1,1,1)
8620          Dif13y=Digipt(2,3,1)-Digipt(2,1,1)
8625          GOSUB Distort
8630        ELSE
8635          R(1)=R(2)*Ly(1)/Ly(Jmax)
8640        END IF
8645        Factor=360*(R(2)-R(1))/((R(1)+R(2))*PI*(Ly(Jmax)-Ly(1)))
8650        GOSUB Factor
8655        ENABLE
8660        V(1)=Digipt(1,1,1)-R(2)
8665        V(2)=Digipt(1,1,1)+R(2)
8670        V(3)=Digipt(2,1,1)-R(2)
8675        V(4)=Digipt(2,1,1)+R(2)
8680        W(1)=-Ly(Jmax)
8685        W(2)=Ly(Jmax)
8690        W(3)=-Ly(Jmax)
8695        W(4)=Ly(Jmax)
8700      ELSE
8705        IF Flag=0 THEN GOSUB Polar
8710        Dif12x=Digipt(1,2,1)-Digipt(1,1,1)
8715        Dif12y=Digipt(2,2,1)-Digipt(2,1,1)
8720        Xm=SQR(Dif12x^2+Dif12y^2)/2
8725        IF Dif12x<>0 THEN
8730          Alpha=ATN(Dif12y/Dif12x)
8735          IF Dif12x<0 THEN Alpha=Alpha+180
8740        ELSE
8745          Alpha=90
8750          IF Dif12y<0 THEN Alpha=Alpha+180
8755        END IF
8760        Beta=90+Alpha+(Lx(Imax)-Lx(1))/2
8765        Ym=Xm*TAN(90-(Lx(Imax)-Lx(1))/2)
8770        R(2)=SQR(Xm^2+Ym^2)
8775        Cx=-R(2)*COS(Beta)+Digipt(1,1,1)
8780        Cy=-R(2)*SIN(Beta)+Digipt(2,1,1)
8785        DISABLE
8790        IF Flag$(9)="NO" THEN
8795          Dif13x=Digipt(1,3,1)-Cx
8800          Dif13y=Digipt(2,3,1)-Cy
8805          GOSUB Distort
```

CALCULATE SIZE AND LOCATION OF THE POLAR ANNULUS IN PLOTTER UNITS

INITIALIZE LIMITS ON THE POLAR ANNULAR VIEWPORT

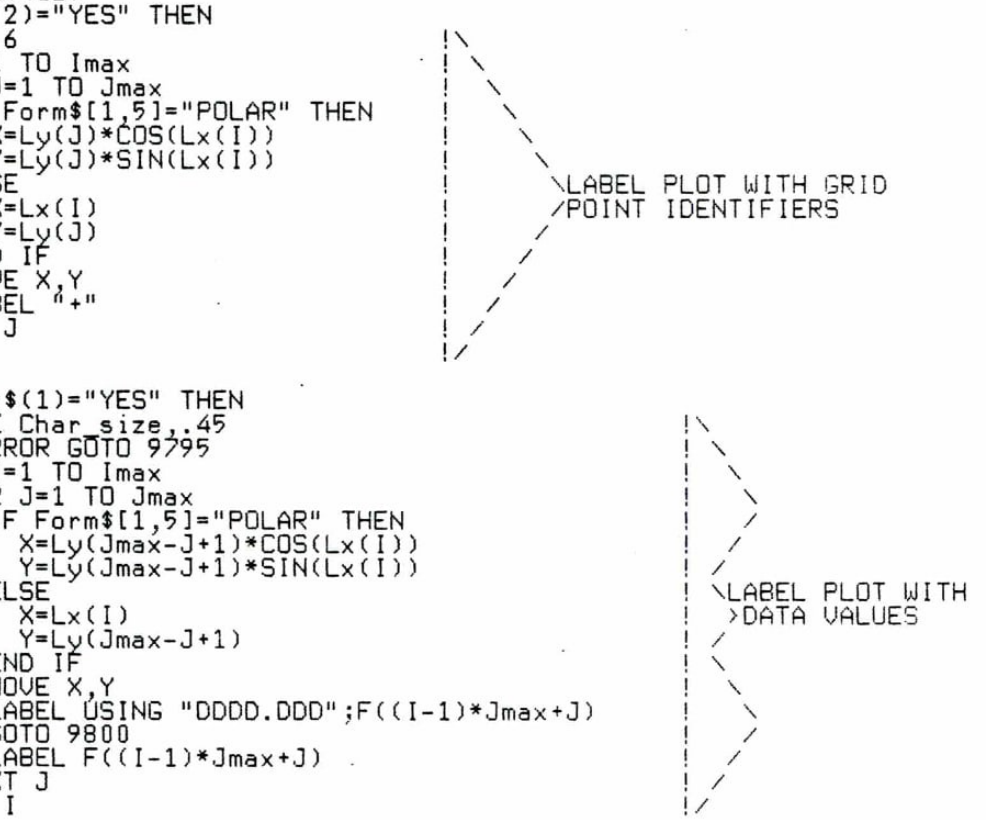
CALCULATE ORIENTATION, SIZE AND LOCATION OF POLAR SECTOR PLOTS

COMPUTE INNER RADIUS (to scale or distorted)

```
8810        ELSE                                    !  /
8815          R(1)=R(2)*Ly(1)/Ly(Jmax)              ! /
8820        END IF                                  !/
8825        Factor=360*(R(2)-R(1))/((R(1)+R(2))*PI*(Ly(Jmax)-Ly(1)))
8830        GOSUB Factor
8835        Gamma=Beta-Lx(Imax)                     !\
8840        FOR I=1 TO Imax                         ! \ROTATE GRID LOCATIONS FOR
8845          Lx(I)=Lx(I)+Gamma                     ! /APPROPRIATE ORIENTATION
8850        NEXT I                                  !/
8855        ENABLE
8860        Dum=Ly(1)*COS(Lx(1))                    !\
8865        W(1)=Dum                                ! \
8870        V(1)=Dum*R(1)/Ly(1)+Cx                  !  \
8875        W(2)=Dum                                !   \
8880        V(2)=V(1)                               !    \INITIALIZE LIMITS ON
8885        Dum=Ly(1)*SIN(Lx(1))                    !    /POLAR SECTOR VIEWPORT
8890        W(3)=Dum                                !   /
8895        V(3)=Dum*R(1)/Ly(1)+Cy                  !  /
8900        W(4)=Dum                                ! /
8905        V(4)=V(3)                               !/
8910        Step=(Lx(Imax)-Lx(1))/50                                    !\
8915        FOR X=Lx(1) TO Lx(Imax)+Step/2 STEP Step                    ! \
8920          FOR J=1 TO Jmax STEP Jmax-1                               !  \
8925            Check=Ly(J)*COS(X)                                      !   \
8930            IF Check<W(1) THEN                                      !    \
8935              W(1)=Check                                            !     \
8940              V(1)=R(1+(J-1)/(Jmax-1))*COS(X)+Cx                    !      /
8945            ELSE                                                    !     /
8950              IF Check>W(2) THEN                                    !    /
8955                W(2)=Check                                          !   /
8960                V(2)=R(1+(J-1)/(Jmax-1))*COS(X)+Cx                  !  /
8965              END IF                                                ! \ COMPUTE LIMITS
8970            END IF                                                  !  \ (rectangular
8975            Check=Ly(J)*SIN(X)                                      !   >co-ordinates on
8980            IF Check<W(3) THEN                                      !  / the plotter) TO
8985              W(3)=Check                                            ! /OBTAIN THE VIEWPORT
8990              V(3)=R(1+(J-1)/(Jmax-1))*SIN(X)+Cy                    ! \   AND WINDOW
8995            ELSE                                                    !  \
9000              IF Check>W(4) THEN                                    !   \
9005                W(4)=Check                                          !    \
9010                V(4)=R(1+(J-1)/(Jmax-1))*SIN(X)+Cy                  !     /
9015              END IF                                                !    /
9020            END IF                                                  !   /
9025          NEXT J                                                    !  /
9030        NEXT X                                                      !/
9035      END IF
9040              !\COMPUTE STEPSIZE BETWEEN INTERPOALTED POINTS ON PARAMETRIC
9045              ! >SPLINE (in the horizontal plane) NEEDED TO GIVE THE DRAWN
9050              !/ CONTOUR LINES A SMOOTH APPEARANCE
9055      Istep=.02*(Ly(1)+Ly(Jmax))/((R(2)+R(1))*.09875)
9060      K=1
9065    ELSE
9070      IF Flag=0 THEN
9075        K=0
9080        GOSUB Digi
9085      END IF
9090      DISABLE
9095      Factor=(Lx(Imax)-Lx(1))*(Digipt(2,2,1)-Digipt(2,1,1))/((Ly(Jmax)-Ly(1))
*(Digipt(1,2,1)-Digipt(1,1,1)))
9100      GOSUB Factor
9105      ENABLE
9110      D=(Digipt(1,2,1)-Digipt(1,1,1))*.09875
9115      Istep=.02*(Lx(Imax)-Lx(1))/D
9120      V(1)=Digipt(1,1,1)                    !\
9125      V(2)=Digipt(1,2,1)                    ! \
9130      V(3)=Digipt(2,1,1)                    !  \
9135      V(4)=Digipt(2,2,1)                    !   \
9140      W(1)=Lx(1)                            !    \
9145      W(2)=Lx(Imax)                         !     >SET UP CARTESIAN VIEWPORT
9150      W(3)=Ly(1)                            !    /     ON THE PLOTTER
9155      W(4)=Ly(Jmax)                         !   /
9160    END IF                                  !  /
9165    VIEWPORT V(1),V(2),V(3),V(4)            ! /
9170    WINDOW W(1),W(2),W(3),W(4)              !/
9175    GOSUB Cosmetics
9180    RETURN
9185    !
9190 Factor:                                    !\
9195    FOR J=1 TO Jmax                         ! \RE-SCALE LONGITUDINAL GRID DIMENSION SO
9200      Ly(J)=Ly(J)*Factor                    !  >THAT THE PLOT IS NOW TO SCALE
```

```
9205     NEXT J                          ! /
9210     RETURN                          !/
9215 Distort:                                                          !\DISTORT RADIAL
9220     R(1)=SQR(Dif13x^2+Dif13y^2)                                   ! \DIMENSION FOR
9225     Distort=(Ly(Jmax)-Ly(1))*R(2)/(Ly(Jmax)*(R(2)-R(1)))          !  \BETTER PLOT
9230     FOR J=Jmax-1 TO 1 STEP -1                                     !   >VISIBILITY
9235       Ly(J)=Ly(Jmax)-(Ly(Jmax)-Ly(J))/Distort                     !  /
9240     NEXT J                                                        ! /
9245     RETURN                                                        !/
9250 Annulus: !
9255     PRINT CHR$(12);TABXY(1,1);"These next pen digitizations will determine th
e location"
9260     PRINT TABXY(1,2);"of the plotted surface on the paper."
9265     PRINT TABXY(1,5);"1-  Move the pen to the desired centre of the plot."
9270     GOTO 9300
9275 Polar: !
9280     PRINT CHR$(12);TABXY(1,1);"These next pen digitizations will determine th
e location"
9285     PRINT TABXY(1,2);"and orientation of the plotted surface on the paper."
9290     PRINT TABXY(1,5);"1-  Move the pen to the point on the outermost grid rad
ius      "
9295     PRINT TABXY(1,6);"    with the grid angle of ";Lx(Imax)
9300     PRINT TABXY(1,7);"    Press 'ENTER' on the plotter"
9305     BEEP
9310     DIGITIZE Digipt(1,1,1),Digipt(2,1,1)
9315     IF Form$="POLARSEC" THEN
9320       PRINT TABXY(1,5);"2-  Move the pen to the point on the outermost grid r
adius
9325       PRINT TABXY(1,6);"    with the grid angle of ";Lx(1);"
     "
9330     ELSE
9335       PRINT TABXY(1,5);"2-  Move the pen to any point on the outermost grid r
adius.
9340     END IF
9345     BEEP
9350     DIGITIZE Digipt(1,2,1),Digipt(2,2,1)
9355     Factor=1
9360     IF Flag$(9)="NO" THEN
9365       PRINT TABXY(1,5);"3-  Move the pen to any point on the innermost grid r
adius.
9370       PRINT TABXY(1,6);"                                                    "
9375       BEEP
9380       DIGITIZE Digipt(1,3,1),Digipt(2,3,1)
9385     END IF
9390     RETURN
9395             !\
9400 Digi:       ! > DIGITIZING ROUTINE TO TRANSFER FRAME SIZE FROM THE PLOTTER
9405             !/
9410     VIEWPORT 0,152,0,100
9415     WINDOW 0,152,0,100
9420     K=K+1
9425     IF K=1 THEN
9430       PRINT CHR$(12);"Move to the desired LOWER LEFT corner of the innermost
frame (i.e. grid)"
9435       PRINT TABXY(1,3);"Note that this frame will be spanned by the entered g
rid point locations"
9440     ELSE
9445       PRINT CHR$(12);"Move to the desired LOWER LEFT corner of this new frame
"
9450     END IF
9455     PRINT "Press 'ENTER' on the plotter"
9460     BEEP
9465     DIGITIZE Digipt(1,1,K),Digipt(2,1,K)
9470     PRINT TABXY(1,6);"Now move to the desired UPPER RIGHT corner of this fram
e"
9475     PRINT "Press 'ENTER' on the plotter"
9480     BEEP
9485     DIGITIZE Digipt(1,2,K),Digipt(2,2,K)
9490     Kmax=K
9495     RETURN
9500 Framing:!
9505     IF Flag=1 THEN
9510       FOR K=2 TO Kmax
9515         GOSUB Frame
9520       NEXT K
9525     ELSE
9530       GOSUB Frame
9535     END IF
9540     VIEWPORT 0,152,0,100
9545     WINDOW 0,152,0,100
```

```
9550     RETURN
9555 Frame:!
9560     VIEWPORT 0,152,0,100
9565     WINDOW 0,152,0,100
9570     VIEWPORT Digipt(1,1,K),Digipt(1,2,K),Digipt(2,1,K),Digipt(2,2,K)
9575     FRAME
9580     RETURN
9585 Cosmetics:!
9590     IF Plot_flag=1 THEN
9595       PRINT CHR$(12);"If plotting paper has not been placed on the plotter, d
o so immediately"
9600       PRINT "Press CONTINUE when ready"
9605       BEEP
9610       PAUSE
9615     END IF
9620     LORG 5
9625     CLIP OFF
9630     DEG
9635     PRINT CHR$(12)
9640     IF Flag$(2)="YES" THEN
9645       CSIZE .6
9650       FOR I=1 TO Imax
9655         FOR J=1 TO Jmax
9660           IF Form$[1,5]="POLAR" THEN
9665             X=Ly(J)*COS(Lx(I))
9670             Y=Ly(J)*SIN(Lx(I))
9675           ELSE
9680             X=Lx(I)
9685             Y=Ly(J)
9690           END IF
9695           MOVE X,Y
9700           LABEL "+"
9705         NEXT J
9710       NEXT I
9715     ELSE
9720       IF Flag$(1)="YES" THEN
9725         CSIZE Char_size,.45
9730         ON ERROR GOTO 9795
9735         FOR I=1 TO Imax
9740           FOR J=1 TO Jmax
9745             IF Form$[1,5]="POLAR" THEN
9750               X=Ly(Jmax-J+1)*COS(Lx(I))
9755               Y=Ly(Jmax-J+1)*SIN(Lx(I))
9760             ELSE
9765               X=Lx(I)
9770               Y=Ly(Jmax-J+1)
9775             END IF
9780             MOVE X,Y
9785             LABEL USING "DDDD.DDD";F((I-1)*Jmax+J)
9790             GOTO 9800
9795             LABEL F((I-1)*Jmax+J)
9800           NEXT J
9805         NEXT I
9810       END IF
9815     END IF
9820     PENUP
9825     IF Flag$(5)="YES" THEN
9830       IF Form$="POLARSEC" THEN
9835         St=Istep*Ly(Jmax)/Ly(1)
9840         CALL Arcplot(Ly(1),Lx(1),Lx(Imax),St)
9845         X=Ly(Jmax)*COS(Lx(Imax))
9850         Y=Ly(Jmax)*SIN(Lx(Imax))
9855         DRAW X,Y
9860         St=-Istep
9865         CALL Arcplot(Ly(Jmax),Lx(Imax),Lx(1),St)
9870         X=Ly(1)*COS(Lx(1))
9875         Y=Ly(1)*SIN(Lx(1))
9880         DRAW X,Y
9885         PENUP
9890       ELSE
9895         IF Form$="POLARANN" THEN
9900           St=Istep*Ly(Jmax)/Ly(1)
9905           CALL Arcplot(Ly(1),0,360,St)
9910           PENUP
9915           CALL Arcplot(Ly(Jmax),0,360,Istep)
9920           PENUP
9925         ELSE
9930           IF Form$="CARTESIAN" AND Plot_flag=1 THEN
9935             VIEWPORT 0,152,0,100
9940             WINDOW 0,152,0,100
```



```
9945            VIEWPORT V(1),V(2),V(3),V(4)    !  \DRAW CARTESIAN FRAME
9950            FRAME                           ! /
9955            WINDOW W(1),W(2),W(3),W(4)      !/
9960            CLIP OFF
9965          END IF
9970        END IF
9975      END IF
9980    END IF
9985    Tic=(Lx(Imax)+Ly(Jmax)-Lx(1)-Ly(1))/100
9990    IF Flag$(6)="YES" AND Form$="CARTESIAN" THEN
9995      CSIZE Char_size,.45
10000     LORG 6                                  !\
10005     FOR I=1 TO Imax                         ! \
10010       MOVE Lx(I),Ly(1)                      !  \
10015       DRAW Lx(I),Ly(1)-Tic                  !  /
10020       LABEL Lx(I)                           !  \LABEL PLOT AXES WITH
10025     NEXT I                                  !   >GRID POINT LOCATIONS
10030     LORG 8                                  !  /ON THE CARTESIAN GRID
10035     FOR J=1 TO Jmax                         !  \
10040       MOVE Lx(1),Ly(J)                      !   \
10045       DRAW Lx(1)-Tic,Ly(J)                  !   /
10050       LABEL Ly(J)/Factor                    !  /
10055     NEXT J                                  ! /
10060   END IF                                    !/
10065   CSIZE Char_size,.45
10070   IF Plot_flag=0 THEN
10075     CSIZE 2.7,.45
10080   END IF
10085   IF Flag$(7)="YES" THEN
10090     LORG 3
10095     MOVE W(1),W(3)
10100     LABEL " "
10105     IF Flag$(6)="YES" AND Form$="CARTESIAN" THEN LABEL " "
10110     LABEL File$&"  "&DATE$(TIMEDATE)&"  "&TIME$(TIMEDATE)
10115     LABEL "DAMPING = "&VAL$(Damp)&"   NORMALIZING = "&VAL$(Avg)
10120   END IF
10125   PENUP
10130   RETURN
10135 Title:!
10140   Lorg=5
10145   CSIZE Char_title,.45
10150   DEG
10155   PRINT TABXY(20,1);">>>>   T I T L E    L A B E L I N G   <<<<"
10160   PRINT CHR$(12);CHR$(132);TABXY(1,3);"To quit the title labeling process,
simply enter 0 for LORG  (below)."
10165   PRINT CHR$(128)
10170   PRINT TABXY(1,5);"      For more information on the LORG command, refer t
o "
10175   PRINT TABXY(1,6);"the Basic Language Reference."
10180   PRINT TABXY(1,8);"These labels may be placed anywhere on the paper."
10185   PRINT TABXY(1,9);"      Furthermore, one may press the PAUSE button, and
EXECUTE any other"
10190   PRINT TABXY(1,10);"plotting command, (e.g. PEN, CSIZE, MOVE, LDIR, etc.)"
10195   PRINT TABXY(1,11);"By pressing CONTINUE, programme run is resumed."
10200   PRINT TABXY(1,12);RPT$("-",80)
10205   BEEP
10210   PRINT TABXY(1,18);"Enter the desired label origin position index (LORG)."
10215   INPUT "By pressing CONTINUE the default value is obtained ( LORG=5 ).",Lo
rg
10220   IF Lorg=0 THEN 10325
10225   IF Lorg<0 OR Lorg>9 THEN
10230     PRINT "Try again."
10235     BEEP
10240     GOTO 10215
10245   END IF
10250   LORG Lorg
10255   BEEP
10260   INPUT "Enter the selected character string to be plotted.",String$
10265   PRINT CHR$(132);TABXY(1,14);"MOVE the pen to the desired origin location
on the paper."
10270   PRINT CHR$(128)
10275   PRINT TABXY(1,15);"When ready, press CONTINUE."
10280   BEEP
10285   PAUSE
10290   READ LOCATOR X,Y
10295   MOVE X,Y
10300   LABEL String$
10305   PENUP
10310   PRINT TABXY(1,14);RPT$(" ",80)
10315   PRINT TABXY(1,15);RPT$(" ",80)
```

```
10320   GOTO 10215
10325   OFF ERROR
10330   LDIR 0
10335   RETURN
10340 Interupt:Interupt=2
10345 SUBEND
10350 !
10355 !
10360 !!************************************************************************
10365 !!                                                                        *
10370 !!      Cross_setup :   1st level SUBROUTINE TO DETERMINE SPLINE          *
10375 !!                      COEFFICIENTS FOR BOTH THE TRANSVERSE AND          *
10380 !!                      THE LONGITUDINAL CROSS PLOTS                      *
10385 !!                                                                        *
10390 !!************************************************************************
10395 SUB Cross_setup(Avgheight,Avg,Slope,Damping,Flag$,INTEGER Norm_flag,Rightbo
und,Flag_3d)
10400   OPTION BASE 1
10405   COM /Gen/ Lx(*),Ly(*),Form$,INTEGER Imax,Jmax
10410   COM /Gen2/ F(*)
10415   COM /Horiz/ Crosshor(50,51),Ahor(50,50),Bhor(50,50),Chor(50,50),Dhor(50,5
0),Xstep
10420   COM /Vertical/ Crossver(126,50),Aver(126,50),Bver(126,50),Cver(126,50),Dv
er(126,50),Ystep
10425   COM /Interupt/ INTEGER Interupt
10430   COM /Damping/ Phor(*),Qhor(*),Pver(*),Qver(*)
10435   COM /Change/ Factor,Distort,Gamma
10440   DIM Aa(150),Bb(150),Cc(150),Dd(150),Hor_cross(150),Ver_cross(50),Yderiv1(
150)
10445   REAL Area,Base_area,Upper,Volume,X_diff,Xin,Lz(150)
10450   INTEGER I,J,K,M,Vert,Num,Num_plots,Flag
10455   ON KEY 9 LABEL "INTERUPT",15 GOTO Interupt1
10460   Flag=0
10465   FOR I=1 TO Imax
10470     Lz(I)=Lx(I)
10475   NEXT I
10480   IF Form$="POLARANN" THEN
10485     Lx(Imax+1)=360+Lx(1)
10490     Lz(Imax+1)=Lx(Imax+1)                              !\
10495     FOR K=2 TO Imax*2                                  ! \CREATE OVERLAP IN ANGULAR
10500       Lz(K+Imax)=Lz(K+Imax-1)+Lz(K)-Lz(K-1)            !  > CO-ORDINATE FOR POLAR
10505     NEXT K                                             ! /  ANNULAR PLOTS TO GIVE
10510     Num=Imax*3 !\                                      !/THREE REPEATED DATA CYCLES
10515   ELSE         ! >STORE NUMBER OF ELEMENTS
10520     Num=Imax   !/TO BE FITTED BY THE SPLINE
10525   END IF
10530   FOR J=1 TO Jmax
10535     FOR I=1 TO Imax                                    !\
10540       Hor_cross(I)=F(I*Jmax-J+1)*1.000001              ! \SET UP DATA AS A SERIES OF
10545       Crosshor(J,I)=Hor_cross(I)                       ! /  TRANSVERSE DATA SETS
10550     NEXT I                                             !/
10555     IF Form$="POLARANN" THEN                 !\
10560       FOR K=1 TO Imax*2                      ! \OVERLAP THE MEASURED PARAMETER IN
10565         Hor_cross(K+Imax)=Hor_cross(K)       !  >THE CIRCUMFERENTIAL DIRECTION
10570       NEXT K                                 ! /FOR THE ANNULAR PLOTS
10575     END IF                                   !/
10580             !COMPUTE THE BOUNDARY SLOPES REQUIRED BY THE RATIONAL SPLINE
10585     Yderiv1(1)=(Hor_cross(2)-Hor_cross(1))/(Lz(2)-Lz(1))
10590     Yderiv1(Num)=(Hor_cross(Num)-Hor_cross(Num-1))/(Lz(Num)-Lz(Num-1))
10595             !\COMPUTE THE TRANSVERSE CROSS PLOT
10600             !/FOR EACH TRANSVERSE SET OF DATA
10605     CALL Raspl1(Lz(*),Hor_cross(*),Phor(*),Qhor(*),Yderiv1(*),Aa(*),Bb(*),C
c(*),Dd(*),Num)
10610     IF Interupt=1 THEN Interupt1
10615     IF Form$="POLARANN" THEN
10620       Crosshor(J,Imax+1)=Crosshor(J,1)   !\
10625       FOR K=1 TO Imax                    ! \
10630         Ahor(J,K)=Aa(Imax+K)             !  \    USE INNER CYCLE OF
10635         Bhor(J,K)=Bb(Imax+K)             !   >  OVERLAPPED SPLINE FOR
10640         Chor(J,K)=Cc(Imax+K)             !  / THE POLAR CIRCUMFERENTIAL
10645         Dhor(J,K)=Dd(Imax+K)             ! /         CROSS PLOTS
10650       NEXT K                             !/
10655     ELSE
10660       FOR K=1 TO Num-1             !\
10665         Ahor(J,K)=Aa(K)            ! \
10670         Bhor(J,K)=Bb(K)            !  \STORE RATIONAL SPLINE COEFFICIENTS
10675         Chor(J,K)=Cc(K)            !  /FOR THE TRANSVERSE CROSS PLOTS
10680         Dhor(J,K)=Dd(K)            ! /
10685       NEXT K                       !/
10690     END IF
```

```
10695   NEXT J
10700   OFF KEY
10705   IF Form$="POLARANN" THEN Imax=Imax+1      !
10710   ON KEY 9 LABEL "INTERUPT",15 GOTO Interupt2
10715   Num_plots=INT(Imax*(1.8+57.5/Imax-81.5/(Imax*Imax)))
10720   IF Imax>25 THEN Num_plots=75+Imax    !\COMPUTE X-INCREMENT BETWEEN THE
10725   Xstep=(Lx(Imax)-Lx(1))/Num_plots     ! > LONGITUDINAL CROSS PLOTS AND
10730   Rightbound=Num_plots+1               !/CALCULATE THE TOTAL NUMBER OF THEM
10735   Ystep=(Ly(Jmax)-Ly(1))/20
10740   Vert=0
10745   M=0
10750           !\
10755           ! \    LOOP TRAVERSING (at a fine increment) FROM RIGHT
10760           !  \TO LEFT, INTERPOLATING ALONG EACH OF THE TRANSVERSE
10765           !  /CROSS PLOTS, TO OBTAIN LONGITUDINAL SETS OF HEIGHTS
10770           ! / WHICH THEN GENERATE LONGITUDINAL CROSS PLOTS.
10775           !/
10780   FOR Xin=Lx(1) TO Lx(Imax)+Xstep/100 STEP Xstep
10785     FOR I=M+1 TO Imax-1          !\
10790       IF Lx(I)>Xin THEN 10800    ! \SELECT APPROPRIATE TRANSVERSE
10795     NEXT I                       ! /GRID INTERVAL
10800     M=I-1                        !/
10805     Vert=Vert+1
10810     X_diff=Xin-Lx(M)
10815     Upper=Lx(M+1)-Lx(M)
10820             !!----DETERMINE HEIGHTS ALONG THE TRANSVERSE CROSS PLOTS----
10825     FOR J=1 TO Jmax
10830       Ver_cross(J)=FNRasplint(Ahor(J,M),Bhor(J,M),Chor(J,M),Dhor(J,M),Phor(
M),Qhor(M),X_diff,Upper)
10835       Crossver(Vert,J)=Ver_cross(J)
10840     NEXT J
10845             !COMPUTE THE BOUNDARY SLOPES REQUIRED BY THE RATIONAL SPLINE
10850     Yderiv1(1)=(Ver_cross(2)-Ver_cross(1))/(Ly(2)-Ly(1))
10855     Yderiv1(Jmax)=(Ver_cross(Jmax)-Ver_cross(Jmax-1))/(Ly(Jmax)-Ly(Jmax-1))
10860             !\COMPUTE THE LONGITUDINAL CROSS PLOT RESULTING FROM THE
10865             !/INTERPOLATED HEIGHTS ALONG THE TRANSVERSE CROSS PLOTS
10870     CALL Raspl1(Ly(*),Ver_cross(*),Pver(*),Qver(*),Yderiv1(*),Aa(*),Bb(*),C
c(*),Dd(*),Jmax)
10875     IF Interupt=1 THEN Interupt2
10880     Area=0
10885     FOR K=1 TO Jmax-1
10890       Aver(Vert,K)=Aa(K)         !\
10895       Bver(Vert,K)=Bb(K)         ! \STORE RATIONAL SPLINE COEFFICIENTS
10900       Cver(Vert,K)=Cc(K)         ! /   FOR LONGITUDINAL CROSS PLOTS
10905       Dver(Vert,K)=Dd(K)         !/
10910       Upper=Ly(K+1)-Ly(K)
10915       New=FNArea(Aa(K),Bb(K),Cc(K),Dd(K),Pver(K),Qver(K),Upper)
10920       IF Form$[1,5]="POLAR" THEN New=New*(Ly(K+1)+Ly(K))
10925       Area=Area+New
10930     NEXT K
10935     IF Vert=1 OR Vert=Rightbound THEN Area=Area/2
10940     Volume=Volume+Area
10945   NEXT Xin
10950   Volume=Volume*Xstep/Factor                                  !\
10955   Base_area=(Lx(Imax)-Lx(1))*(Ly(Jmax)-Ly(1))/Factor          ! \
10960   IF Form$[1,5]="POLAR" THEN                                  !  \ COMPUTE
10965     Volume=Volume*PI/360/Factor                               !   >AVERAGE
10970     Base_area=Base_area*(Ly(Jmax)+Ly(1))*PI/360/Factor        !  / HEIGHT
10975   END IF                                                      ! /
10980   Avgheight=Volume/Base_area                                  !/
10985   PRINT CHR$(12)
10990   PRINT "The computed volume is ";Volume
10995   PRINT "Over a base area of ";Base_area
11000   PRINT "Giving an average height of ";Avgheight
11005   IF Norm_flag=1 THEN
11010     PRINT "Data has been normalized by a factor of ";Avg
11015   END IF
11020   IF Flag$="YES" AND Flag=0 THEN
11025     ON TIMEOUT 7,.5 GOSUB Timeout
11030     PRINTER IS 701
11035     PRINT " "
11040     PRINT "The plotting of the above data set with a damping factor of ";Da
mping
11045     PRINT "gives the following results:"
11050     PRINT " "
11055     Flag=1
11060     GOTO 10990
11065   END IF
11070   OFF TIMEOUT
11075   PRINTER IS 1
```

```
11080   SUBEXIT
11085 Interupt1:Interupt=2
11090   SUBEXIT
11095 Interupt2:Interupt=1
11100   SUBEXIT
11105 Timeout:!
11110   CALL Printerset
11115   ON TIMEOUT 7,.5 GOSUB Timeout
11120   RETURN
11125 SUBEND
11130 !
11135 !
11145 !!*************************************************************************
11150 !!                                                                        *
11155 !!      Cross_plot :   1st level SUBROUTINE TO FIND CONTOUR POINTS        *
11160 !!                                                                        *
11165 !!*************************************************************************
11170 SUB Cross_plot(Xcont(*),Ycont(*),Level,INTEGER Pcount(*),L,Limax)
11175   COM /Gen/ Lx(*),Ly(*),Form$,INTEGER Imax,Jmax
11180   COM /Horiz/ Crosshor(*),Ahor(*),Bhor(*),Chor(*),Dhor(*),Xstep
11185   COM /Vertical/ Crossver(*),Aver(*),Bver(*),Cver(*),Dver(*),Ystep
11190   COM /Form/ INTEGER Left_join(50),Right_join(50)
11195   COM /Bounds/ INTEGER Left(*),Right(*),Leftbound,Rightbound,Lineflag
11200   COM /Interupt/ INTEGER Interupt
11205   COM /Damping/ Phor(*),Qhor(*),Pver(*),Qver(*)
11210   REAL Y(20),X_bot(50),X_top(50),W(4)
11215   REAL Bot,Li,Top,Upper,Xin
11220   INTEGER G,H,I,J,K,M,N,Count,Li_save(50),New,Old,Bot_ct,Top_ct,Good,Vert
11225   INTEGER Leftflag
11230   ON KEY 9 LABEL "INTERUPT",15 GOTO Interupt
11235   FOR I=1 TO 50                  !\
11240     Left_join(I)=0               ! \
11245     Right_join(I)=0              !  \INITIALIZE FLAGS AND COUNTERS
11250   NEXT I                         !  /
11255   Bot_ct=0                       ! /
11260   Top_ct=0                       !/
11265   PRINTER IS 1
11270   BEEP
11275   PRINT TABXY(42,1),"CALCULATING CONTOUR LEVEL ";Level;"       "
11280         !\
11285         ! >LOOP TO OBTAIN CONTOUR POINTS ALONG THE BOTTOM BOUNDARY
11290         !/
11295   FOR I=1 TO Imax-1
11300     Upper=Lx(I+1)-Lx(I)
11305     CALL Interpnew(W(*),Lx(I),Ahor(1,I),Bhor(1,I),Chor(1,I),Dhor(1,I),Phor(
I),Qhor(I),Level,0,Upper,Good)
11310     IF Interupt=1 THEN Interupt
11315     IF Bot_ct<>0 AND Good<>0 THEN                !\
11320       IF PROUND(X_bot(Bot_ct)-W(1),-6)=0. THEN   ! \
11325         Good=Good-1                              !  \ CHECK FOR ROUND-OFF
11330         FOR K=1 TO Good                          !   \ ERROR AND DISMISS
11335           W(K)=W(K+1)                            !   /POSSIBLE EXTRANEOUS
11340         NEXT K                                   !  /BOTTOM BOUNDARY POINT
11345       END IF                                     ! /
11350     END IF                                       !/
11355     FOR K=1 TO Good            !\
11360       X_bot(Bot_ct+K)=W(K)     ! >STORE BOTTOM BOUNDARY POINTS
11365     NEXT K                     !/
11370     Bot_ct=Bot_ct+Good    !TOTAL NUMBER OF BOTTOM BOUNDARY POINTS
11375   NEXT I
11380   IF Bot_ct>0 THEN !\
11385     Bot=X_bot(1)   ! \
11390     G=1            !  \INITIALIZE BOTTOM BOUNDARY VARIABLES
11395   ELSE             !  /FOR USAGE IN THE SUBROUTINE 'Order'
11400     Bot=Lx(1)-1    ! /
11405     G=0            !/
11410   END IF
11415         !\
11420         ! >LOOP TO OBTAIN CONTOUR POINTS ALONG THE TOP BOUNDARY
11425         !/
11430   FOR I=1 TO Imax-1
11435     Upper=Lx(I+1)-Lx(I)
11440     CALL Interpnew(W(*),Lx(I),Ahor(Jmax,I),Bhor(Jmax,I),Chor(Jmax,I),Dhor(J
max,I),Phor(I),Qhor(I),Level,0,Upper,Good)
11445     IF Interupt=1 THEN Interupt
11450     IF Top_ct<>0 AND Good<>0 THEN                !\
11455       IF PROUND(X_top(Top_ct)-W(1),-6)=0. THEN   ! \
11460         Good=Good-1                              !  \ CHECK FOR ROUND-OFF
11465         FOR K=1 TO Good                          !   \ ERROR AND DISMISS
11470           W(K)=W(K+1)                            !   /POSSIBLE EXTRANEOUS
```

```
11475         NEXT K                                          !  / TOP BOUNDARY POINT
11480       END IF                                            ! /
11485     END IF                                              !/
11490     FOR K=1 TO Good           !\
11495       X_top(Top_ct+K)=W(K)    ! >STORE TOP BOUNDARY POINTS
11500     NEXT K                    !/
11505     Top_ct=Top_ct+Good    !TOTAL NUMBER OF TOP BOUNDARY POINTS
11510   NEXT I
11515   IF Top_ct>0 THEN !\
11520     Top=X_top(1)   ! \
11525     H=1            !  \INITIALIZE TOP BOUNDARY VARIABLES
11530   ELSE             !  /FOR USAGE IN THE SUBROUTINE 'Order'
11535     Top=Lx(1)-1    ! /
11540     H=0            !/
11545   END IF
11550   IF L>2 AND Lineflag<>0 THEN                           ! CHECK AREA
11555     IF Left(1)>=Left(2) THEN Leftbound=Left(1)          ! AVAILABLE FOR
11560     IF Right(1)<Right(2) THEN Rightbound=Right(1)+1     ! FURTHER CONTOUR
11565   END IF                                                ! LINES BASED ON THE
11570   Left(2)=Left(1)                                       ! ON THE LOCATION
11575   Right(2)=Right(1)                                     ! OF THE PREVIOUS
11580   Leftflag=0                                            ! CONTOUR LINES
11585   Xin=(Leftbound-1)*Xstep+Lx(1)
11590           !\
11595           ! \MAIN LOOP OBTAINING CONTOUR POINTS FROM
11600           ! /ONE LONGITUDINAL CROSS PLOT AT A TIME
11605           !/
11610   FOR Vert=Leftbound TO Rightbound
11615     Count=0
11620     FOR K=1 TO Jmax-1          !\FIND CONTOUR POINTS FROM ONE GRID INTERVAL
11625       Upper=Ly(K+1)-Ly(K)      !/AT A TIME ON THE CURRENT CROSS PLOT
11630       CALL Interpnew(W(*),Ly(K),Aver(Vert,K),Bver(Vert,K),Cver(Vert,K),Dver
(Vert,K),Pver(K),Qver(K),Level,0,Upper,Good)
11635       IF Interupt=1 THEN Interupt
11640       IF Count<>0 AND Good<>0 THEN            !\
11645         IF PROUND(Y(Count)-W(1),-6)=0. THEN    ! \
11650           Good=Good-1                          !  \ CHECK FOR ROUND-OFF
11655           FOR N=1 TO Good                      !   \ ERROR AND DISMISS
11660             W(N)=W(N+1)                        !   /POSSIBLE EXTRANEOUS
11665           NEXT N                               !  / CONTOUR POINT AT THE
11670         END IF                                 ! /   INTERVAL BOUNDARIES
11675       END IF                                   !/
11680       FOR N=1 TO Good      !\
11685         Y(Count+N)=W(N)    ! >STORE CONTOUR POINTS FROM THIS
11690       NEXT N               !/ CURRENT LONGITUDINAL CROSS PLOT
11695       Count=Count+Good  !TOTAL NUMBER OF CONTOUR POINTS ON THE CROSS PLOT
11700     NEXT K
11705     IF Count>0 AND Y(1)=Ly(1) THEN
11710       FOR N=2 TO Count
11715         Y(N-1)=Y(N)
11720       NEXT N
11725       Count=Count-1
11730     END IF
11735     IF Count>0 AND Y(Count)=Ly(Jmax) THEN Count=Count-1
11740     IF Leftflag=0 AND Count<>0 THEN   !\
11745       Leftflag=1                      ! \KEEP TRACK OF FLAGS AND INDICES
11750       Left(1)=Vert                    !  >TO LOCATE AREA AVAILABLE FOR
11755     END IF                            ! /    FURTHER CONTOUR LEVELS
11760     IF Count<>0 THEN Right(1)=Vert    !/
11765     IF Bot_ct>G THEN
11770       IF X_bot(G+1)>=Xin-Xstep AND X_bot(G+1)<Xin THEN
11775         G=G+2                       !\
11780         IF G>Bot_ct THEN            ! \
11785           Bot=Lx(1)-1               !  \
11790           G=Bot_ct                  !   \
11795         ELSE                        !    \
11800           Bot=X_bot(G)              !     \
11805           GOTO 11765                !      \ CHECK IF THERE ARE PAIRS OF
11810         END IF                      !       \BOUNDARY POINTS BETWEEN
11815       END IF                        !       /CONSECUTIVE LONGITUDINAL
11820     END IF                          !      / CROSS PLOTS  ...
11825     IF Top_ct>H THEN                !     /
11830       IF X_top(H+1)>=Xin-Xstep AND X_top(H+1)<Xin THEN
11835         H=H+2                       !      \
11840         IF H>Top_ct THEN            !       \  ...IF THIS IS THE CASE, THEN
11845           Top=Lx(1)-1               !        > THEY FORM A CONTOUR LINE
11850           H=Top_ct                  !       /  BETWEEN THEM
11855         ELSE                        !      /
11860           Top=X_top(H)              !     /
11865           GOTO 11825                !    /
```

```
11870         END IF                          !  /
11875       END IF                            ! /
11880     END IF                              !/
11885     IF Xin=Lx(1) THEN             !\
11890       FOR Li=1 TO Count           ! \
11895         Xcont(Li,1)=Xin           !  \
11900         Ycont(Li,1)=Y(Li)         !   \   EACH POINT ON THE LEFT BOUNDARY
11905         Pcount(Li)=1              !    >  MUST START ITS OWN LINE AND HENCE
11910         Li_save(Li)=Li            !   /   THE SUBROUTINE 'Order' DOES NOT NEED
11915       NEXT Li                     !  /    TO BE ACCESSED
11920       Limax=Count                 ! /
11925       Old=Count                   !/
11930     ELSE                                  !ORDER NEW POINTS WITH EXISTING LINES
11935                 !\KEEP TRACK OF THE NUMBER OF CONTOUR
11940       New=Count ! >POINTS TO BE ORDERED ON THIS
11945                 !/CURRENT LONGITUDINAL CROSS PLOT
11950       CALL Order(Level,Xcont(*),Ycont(*),Xin,Y(*),Xstep,Bot,Top,New,Old,Lim
ax,Pcount(*),Li_save(*),Vert)
11955       IF Interupt=1 THEN Interupt     !CHECK INTERUPT FLAG
11960                 !\SAVE NUMBER OF CONTOUR POINTS (HAVE ALL BEEN ORDERED) ON
11965       Old=New   ! >THE CURRENT LONGITUDINAL CROSS PLOT TO AID IN ORDERING
11970                 !/ THE CONTOUR POINTS ON THE NEXT LONGITUDINAL CROSS PLOT
11975       IF Bot<Lx(1) AND G<Bot_ct THEN      !\
11980         G=G+1                             ! \
11985         Bot=X_bot(G)                      !  \
11990       END IF                              !   \RECHECK TOP AND BOTTOM
11995       IF Top<Lx(1) AND H<Top_ct THEN      !   /BOUNDARY CONTOUR POINTS
12000         H=H+1                             !  /
12005         Top=X_top(H)                      ! /
12010       END IF                              !/
12015     END IF
12020     Xin=Xin+Xstep
12025   NEXT Vert !!---OBTAIN NEXT CONTOUR POINTS FROM NEXT VERTICAL CROSS PLOT
12030   IF Limax=0 AND Lineflag=1 THEN Lineflag=2
12035   IF Limax<>0 AND Lineflag=0 THEN Lineflag=1
12040   SUBEXIT
12045 Interupt:Interupt=1
12050 SUBEND
12055 !
12060 !
12065 !!*************************************************************************
12070 !!                                                                         *
12075 !!      Order :   2nd level SUBROUTINE TO ORDER INTERPOLATED POINTS        *
12080 !!                                                                         *
12085 !!*************************************************************************
12090 SUB Order(Level,Xcont(*),Ycont(*),Xsave,Y(*),Xstep,Bot,Top,INTEGER New,Old,
Li,Pcount(*),Li_save(*),Vert)
12095   OPTION BASE 1
12100   COM /Gen/ Lx(*),Ly(*),Form$,INTEGER Imax,Jmax
12105   COM /Form/ INTEGER Left_join(*),Right_join(*)
12110   COM /Interupt/ INTEGER Interupt
12115   REAL V1(2),V2(2),Keepx(50),Keepy(50)
12120   REAL Angle,Check,Dist,Dist1,Dist2,Py,Y1,Y2,Ytest
12125   INTEGER H,J,K,N,Templisave(50),Number,Next,Start,Stop
12130   ON KEY 9 LABEL "INTERUPT",15 GOTO Interupt
12135     !\
12140     ! > CHECK IF NEXT BOTTOM BOUNDARY POINT LIES WITHIN MINOR X-INTERVAL
12145     !/
12150   IF Old=0 AND New=0 THEN
12155     IF Bot>=Xsave-Xstep AND Bot<=Xsave THEN
12160       IF Top>=Xsave-Xstep AND Top<=Xsave THEN
12165         Li=Li+1
12170         Y1=(Ly(1)+Ly(2))/2
12175         Y2=(Ly(Jmax)+Ly(Jmax-1))/2
12180         Xcont(Li,1)=Bot
12185         Ycont(Li,1)=Ly(1)
12190         CALL Newpoints(Y1,Y2,Xsave,Keepx(*),Keepy(*),Level,Number,Vert)
12195         FOR N=1 TO Number
12200           Xcont(Li,N+1)=Keepx(N)
12205           Ycont(Li,N+1)=Keepy(N)
12210         NEXT N
12215         Pcount(Li)=N+1
12220         Xcont(Li,N+1)=Top
12225         Ycont(Li,N+1)=Ly(Jmax)
12230       END IF
12235     END IF
12240     SUBEXIT
12245   END IF
12250   Start=1
12255   Next=1
```

```
12260   Stop=New
12265   IF Bot>=Xsave-Xstep AND Bot<Xsave THEN
12270     IF Old=0 THEN Bot_start
12275     IF New=0 THEN Bot_end
12280     IF Y(1)<Ycont(Li_save(1),Pcount(Li_save(1))) THEN
12285 Bot_start:                  !\
12290                             !/ START NEW LINE WITH THIS BOUNDARY POINT
12295       Li=Li+1
12300       Pcount(Li)=2
12305       Xcont(Li,1)=Bot
12310       Ycont(Li,1)=Ly(1)
12315       Xcont(Li,2)=Xsave
12320       Ycont(Li,2)=Y(1)
12325       Next=2
12330       Templisave(1)=Li
12335     ELSE   !\
12340 Bot_end:   ! >END EXISTING LINE WITH THIS BOUNDARY POINT
12345            !/
12350       Pcount(Li_save(1))=Pcount(Li_save(1))+1
12355       Xcont(Li_save(1),Pcount(Li_save(1)))=Bot
12360       Ycont(Li_save(1),Pcount(Li_save(1)))=Ly(1)
12365       Start=2
12370     END IF
12375     Bot=Lx(1)-1
12380   END IF
12385       !\
12390       ! > CHECK IF NEXT TOP BOUNDARY POINT LIES WITHIN MINOR X-INTERVAL--
12395       !/
12400   IF Top>=Xsave-Xstep AND Top<Xsave THEN
12405     IF Old=0 THEN Top_start
12410     IF New=0 THEN Top_end
12415     IF Y(New)>Ycont(Li_save(Old),Pcount(Li_save(Old))) THEN
12420 Top_start:                  !\
12425                             !/ START NEW LINE WITH THIS BOUNDARY POINT
12430       Li=Li+1
12435       Pcount(Li)=2
12440       Xcont(Li,1)=Top
12445       Ycont(Li,1)=Ly(Jmax)
12450       Xcont(Li,2)=Xsave
12455       Ycont(Li,2)=Y(New)
12460       Stop=New-1
12465       Templisave(New)=Li
12470     ELSE     !\
12475 Top_end:     ! >END EXISTING LINE WITH THIS BOUNDARY POINT
12480              !/
12485       Pcount(Li_save(Old))=Pcount(Li_save(Old))+1
12490       Xcont(Li_save(Old),Pcount(Li_save(Old)))=Top
12495       Ycont(Li_save(Old),Pcount(Li_save(Old)))=Ly(Jmax)
12500       Old=Old-1
12505     END IF
12510     Top=Lx(1)-1
12515   END IF
12520             !\
12525             ! \ LOOP WHICH ACCESSES EACH OPEN CONTOUR LINE
12530             !  \(one at a time) FITTED WITH A CONTOUR POINT
12535             !  /FROM THE LAST LONGITUDINAL CROSS PLOT TO ORDER
12540             ! / THE NEW SET OF LONGITUDINAL CONTOUR POINTS.
12545             !/
12550   Flag=0
12555   FOR J=Start TO Old
12560 Start:!
12565     IF Next>Stop AND J<Old THEN Rightjoin
12570     Py=Ycont(Li_save(J),Pcount(Li_save(J)))
12575     Check=(Py-Y(Next))^2
12580     IF Py<Y(Next) THEN
12585       IF J=Old THEN Continue
12590       Dist=(Py-Ycont(Li_save(J+1),Pcount(Li_save(J+1))))^2
12595     ELSE
12600       IF Next=Stop THEN Continue
12605       Dist=(Y(Next)-Y(Next+1))^2
12610     END IF
12615     IF Dist<Check THEN
12620       IF Py<Y(Next) THEN
12625                   !\ THE TWO CONTOUR POINTS FROM THE CURRENT LONGITUDINAL
12630 Rightjoin:      ! \CROSS PLOT ESTABLISH A ')' SHAPED CURVE, HENCE, A
12635                 ! /'RIGHT_JOIN' AND THE INSERTION OF ADDITIONAL CONTOUR
12640                 !/ POINTS BETWEEN THEM, ALONG THE TRANSVERSE GRID LINES
12645         Right_join(Li_save(J))=Li_save(J+1)
12650         Right_join(Li_save(J+1))=Li_save(J)
12655         Y1=Ycont(Li_save(J),Pcount(Li_save(J)))
```

```
12660         Y2=Ycont(Li_save(J+1),Pcount(Li_save(J+1)))
12665         CALL Newpoints(Y1,Y2,Xsave,Keepx(*),Keepy(*),Level,Number,Vert)
12670         IF Interupt=1 THEN Interupt
12675         FOR N=1 TO Number
12680           Ycont(Li_save(J),Pcount(Li_save(J))+N)=Keepy(N)
12685           Xcont(Li_save(J),Pcount(Li_save(J))+N)=Keepx(N)
12690         NEXT N
12695         Pcount(Li_save(J))=Pcount(Li_save(J))+Number
12700         J=J+1
12705       ELSE
12710         K=Next                    !\
12715         GOSUB Leftjoin            ! \
12720         Next=Next+2               !  \THE TWO CONTOUR POINTS FROM THE
12725         Templisave(K-1)=Li-1      !  /PREVIOUS CROSS PLOT ESTABLISH A '('
12730         Templisave(K)=Li          ! / SHAPED CURVE, HENCE, A 'LEFT_JOIN'
12735         GOTO Start                !/
12740       END IF
12745     ELSE
12750 Continue:       !\ CONTINUE AN EXISTING LINE SEGMENT WITH A POINT FROM THE
12755                 !/ SET OF NEW POINTS ON THE CURRENT LONGITUDINAL CROSS PLOT
12760       IF ABS(Py-Y(Next))>Ly(Jmax)/Jmax/1.5 THEN
12765         IF Py<Y(Next) THEN        !\
12770           Y1=Py                   ! \PUT THE LAST TWO POINTS ON THIS CONTINUING
12775           Y2=Y(Next)              !  \LINE SEGMENT IN INCREASING ORDER,
12780         ELSE                      !   >SO THAT, IF THEY ARE FAR APART,
12785           Y1=Y(Next)              !  /ADDITIONAL CONTOUR POINTS FROM THE
12790           Y2=Py                   ! /TRANSVERSE CROSS PLOTS MAY BE INSERTED
12795         END IF                    !/
12800         CALL Newpoints(Y1,Y2,Xsave,Keepx(*),Keepy(*),Level,Number,Vert)
12805         IF Number>0 THEN
12810           IF Py<Y(Next) THEN
12815             FOR N=1 TO Number
12820               Xcont(Li_save(J),Pcount(Li_save(J))+N)=Keepx(N)
12825               Ycont(Li_save(J),Pcount(Li_save(J))+N)=Keepy(N)
12830             NEXT N
12835           ELSE
12840             FOR N=1 TO Number
12845               Xcont(Li_save(J),Pcount(Li_save(J))+N)=Keepx(Number-N+1)
12850               Ycont(Li_save(J),Pcount(Li_save(J))+N)=Keepy(Number-N+1)
12855             NEXT N
12860           END IF
12865           Pcount(Li_save(J))=Pcount(Li_save(J))+Number
12870         END IF
12875       END IF
12880       Pcount(Li_save(J))=Pcount(Li_save(J))+1
12885       Xcont(Li_save(J),Pcount(Li_save(J)))=Xsave
12890       Ycont(Li_save(J),Pcount(Li_save(J)))=Y(Next)
12895       Templisave(Next)=Li_save(J)
12900       Next=Next+1
12905     END IF
12910   NEXT J
12915   FOR K=1 TO Next-1               !\SAVE THE LINE SEGMENTS ON WHICH
12920     Li_save(K)=Templisave(K)      ! >   EACH OF THE POINTS FROM
12925   NEXT K                          !/  THE CURRENT CROSS PLOT LIES
12930   Flag=1
12935   FOR K=Next TO Stop
12940 Leftjoin:!
12945     Li=Li+1                       !\
12950     Pcount(Li)=1                  ! \
12955     Xcont(Li,1)=Xsave             !  \CREATE NEW LINE SEGMENTS WHICH
12960     Ycont(Li,1)=Y(K)              !  /FORM A '(' SHAPED CURVE
12965     Left_join(Li)=Li+1            ! /
12970     Left_join(Li+1)=Li            !/
12975     Y1=Y(K)
12980     Y2=Y(K+1)
12985     CALL Newpoints(Y1,Y2,Xsave,Keepx(*),Keepy(*),Level,Number,Vert)
12990     IF Interupt=1 THEN Interupt
12995     FOR N=1 TO Number               !\
13000       Ycont(Li+1,N)=Keepy(N)        ! \
13005       Xcont(Li+1,N)=Keepx(N)        !  \
13010     NEXT N                          !   \STORE THE ADDITIONAL POINTS
13015     Li=Li+1                         !    >FOUND ALONG THE TRANSVERSE
13020     K=K+1                           !   /CROSS PLOTS
13025     Pcount(Li)=N                    !  /
13030     Xcont(Li,N)=Xsave               ! /
13035     Ycont(Li,N)=Y(K)                !/
13040     IF Flag=0 THEN RETURN
13045     Li_save(K-1)=Li-1
13050     Li_save(K)=Li
13055   NEXT K
```

```
13060    IF Stop<New THEN Li_save(New)=Templisave(New)
13065    IF Form$="POLARANN" THEN     !\SET 'JOIN' FLAGS TO CONTINUE LINES
13070                                 ! >ACROSS THE FIRST GRID ANGLE
13075                                 !/IN THE POLAR ANNULAR PLOTS
13080      IF PROUND((Xsave-Lx(Imax))/Xsave,-8)=0 THEN
13085        FOR K=1 TO New
13090          FOR J=1 TO Li
13095            IF Xcont(J,1)=Lx(1) AND PROUND((Ycont(Li_save(K),Pcount(Li_save(K
))-Ycont(J,1))/Ycont(J,1),-4)=0 THEN
13100              Left_join(J)=-Li_save(K)
13105              Right_join(Li_save(K))=-J
13110              GOTO 13125
13115            END IF
13120          NEXT J
13125        NEXT K
13130      END IF
13135    END IF
13140    SUBEXIT
13145  Interupt:Interupt=1
13150  SUBEND
13155  !
13160  !
13165  !!*************************************************************************
13170  !!                                                                        *
13175  !!       Plot_cont :   1st level SUBROUTINE TO PLOT CONTOUR LINES         *
13180  !!                                                                        *
13185  !!*************************************************************************
13190  SUB Plot_cont(Xcont(*),Ycont(*),Level,Istep,Char_size,INTEGER L,Pcount(*),L
imax,Total,Flag$)
13195    OPTION BASE 1
13200    COM /Form/ INTEGER Left_join(*),Right_join(*)
13205    COM /Gen/ Lx(*),Ly(*),Form$,INTEGER Imax,Jmax
13210    COM /Location/ X_loc(*),Y_loc(*),INTEGER Values(*)
13215    COM /Colours/ INTEGER Colour(3),Plot_flag
13220    COM /Interupt/ INTEGER Interupt
13225    COM /Change/ Factor,Distort,Gamma
13230    REAL Dist1,Dist2,Dummy_x,Dummy_y,Space,Spot,T_diff,Tin,X,Xout,Xx,Yout,Yy,
Z
13235    INTEGER H,I,K,M,N,Q,R,S,Check,Li,Next_li,First,Li_flag(40),Counter,Flag
13240    INTEGER Err_flag
13245    ON KEY 9 LABEL "INTERUPT",15 GOTO Interupt
13250    ALLOCATE Xplot(Total+6),Yplot(Total+6)
13255    Level$=VAL$(Level)
13260    Space=(1.1*LEN(Level$)+2.5)*Char_size*Istep*40/27
13265    IF Plot_flag=0 THEN Space=(1.1*LEN(Level$)+2.5)*Istep*4.0
13270    Counter=0
13275    FOR Li=1 TO Limax            !\
13280      Li_flag(Li)=0              ! >----INITIALIZE VECTOR
13285    NEXT Li                      !/
13290    IF Flag$="YES" THEN                  !!-------PRINT CONTOUR POINTS--------
13295      ON TIMEOUT 7,.5 GOSUB Timeout
13300      PRINTER IS 701
13305      GOTO 13330
13310  Timeout:!
13315      CALL Printerset
13320      ON TIMEOUT 7,.5 GOSUB Timeout
13325      RETURN
13330      PRINT ""
13335      PRINT ""
13340      PRINT "#####################################################"
13345      PRINT "                 L E V E L  = ";Level
13350      FOR Li=1 TO Limax
13355        PRINT ""
13360        PRINT ""
13365        PRINT "************************************"
13370        PRINT "LINE #";Li;" - # OF POINTS=";Pcount(Li);"      LEFT_JOIN:";Lef
t_join(Li);"     RIGHT_JOIN:";Right_join(Li)
13375        PRINT ""
13380        ON ERROR GOTO 13425
13385        FOR R=1 TO Pcount(Li)
13390                     !\ RESET CONTOUR POINTS TO THE SCALE SET BY THE ORIGINAL
13395                     ! >GRID POINT SPECIFICATIONS BY INVERTING THE USAGE OF
13400                     !/ THE VARIABLES:    Gamma, Distort & Factor
13405          Xx=Xcont(Li,R)-Gamma
13410          Yy=(Ly(Jmax)*(1-Distort)+Ycont(Li,R)*Distort)/Factor
13415          PRINT USING "DDDD.DDD,XXX";R,Xx,Yy
13420          GOTO 13430
13425          PRINT R;Xx;Yy
13430        NEXT R
13435        OFF ERROR
```

```
13440     NEXT Li
13445     OFF TIMEOUT
13450     PRINTER IS 1
13455   END IF
13460   IF Form$="POLARANN" THEN                    !\
13465     FOR Li=1 TO Limax                         ! \
13470       IF Left_join(Li)<0 THEN                 !  \
13475         IF Left_join(Li)<>-Li THEN            !   \
13480           Q=-Left_join(Li)                    !    \
13485           N=Pcount(Q)                         !     \
13490           FOR M=1 TO Pcount(Li)-1             !     /
13495             Xcont(Q,N+M)=Xcont(Li,M+1)        !    /
13500             Ycont(Q,N+M)=Ycont(Li,M+1)        !   /
13505           NEXT M                              !  /  FOR ANNULAR PLOTS, IF TWO
13510           Pcount(Q)=N+M-1                     !  \  SEGMENTS MEET AT THE FIRST
13515           Right_join(Q)=Right_join(Li)        !   > GRID ANGLE ('LX(1)'), THEN
13520           Li_flag(Li)=1                       !  /  THEY MUST BE JOINED TOGETHER
13525           FOR I=Li TO Limax                   !  \  TO FORM ONE CONTINUOUS
13530             IF Right_join(I)=Li THEN          !   \ SEGMENT CROSSING OVER THE
13535               Right_join(I)=Q                 !    \FIRST GRID ANGLE
13540               GOTO 13555                      !     \
13545             END IF                            !     /
13550           NEXT I                              !    /
13555         END IF                                !   /
13560       END IF                                  !  /
13565     NEXT Li                                   ! /
13570   END IF                                      !/
13575                       !\
13580   FOR Li=1 TO Limax   ! \    ----SELECT LINE SEGMENTS TO FORM CONTOUR LINE
13585                       ! /
13590                       !/            !\IF A SEGMENT HAS ALREADY BEEN USED IN
13595     IF Li_flag(Li)=1 THEN Continue    ! >THE FORMATION OF A FULL CONTOUR LINE
13600     Flag=0                            !/ THEN PROCEED TO THE NEXT SEGMENT
13605     FOR M=1 TO Pcount(Li)         !\
13610       Xplot(M)=Xcont(Li,M)        ! \
13615       Yplot(M)=Ycont(Li,M)        !  >SETUP CONTOUR LINE WITH FIRST SEGMENT
13620     NEXT M                        ! /
13625     N=Pcount(Li)                  !/
13630     IF Right_join(Li)=-Li THEN Next
13635     Next_li=Li                !!-----INITIALIZE DUMMY VARIABLE------
13640 Again:!
13645     IF Right_join(Next_li)<>0 THEN
13650       Li_flag(Right_join(Next_li))=1                !\
13655       FOR M=1 TO Pcount(Right_join(Next_li))        ! \   CONTINUE CONTOUR
13660         S=Pcount(Right_join(Next_li))-M+1           !  \  LINE WITH NEXT
13665         Xplot(M+N)=Xcont(Right_join(Next_li),S)     !   > SEGMENT ATTACHED
13670         Yplot(M+N)=Ycont(Right_join(Next_li),S)     !  /  ON THE RIGHT SIDE
13675       NEXT M                                        ! /
13680       N=N+Pcount(Right_join(Next_li))               !/
13685       IF Left_join(Right_join(Next_li))=Li THEN
13690         IF N<4 THEN Continue
13695         N=N+1                         !\
13700         Xplot(N)=Xplot(1)             ! \----IF CONTOUR LINE IS A CLOSED LOOP
13705         Yplot(N)=Yplot(1)             ! /----THEN CLOSE LOOP WITH FIRST POINT
13710         GOTO Next                     !/
13715       ELSE
13720         IF Left_join(Right_join(Next_li))<>0 THEN
13725           Li_flag(Left_join(Right_join(Next_li)))=1           ! CONTINUE
13730           FOR M=1 TO Pcount(Left_join(Right_join(Next_li)))   ! CONTOUR
13735             Q=M+N                                             !LINE WITH
13740             Xplot(Q)=Xcont(Left_join(Right_join(Next_li)),M)  !   NEXT
13745             Yplot(Q)=Ycont(Left_join(Right_join(Next_li)),M)  ! SEGMENT
13750           NEXT M                                            ! ATTACHED
13755           N=Q                                               !  ON THE
13760           Next_li=Left_join(Right_join(Next_li))            !LEFT SIDE
13765           GOTO Again                                        !
13770         ELSE
13775           GOTO Check
13780         END IF
13785       END IF
13790     ELSE
13795                           !IF SEGMENT HAS BEEN JOINED THEN CONTOUR LINE IS
13800 Check:                    !FINISHED AND IS READY TO BE PLOTTED
13805       IF Flag=1 THEN Next
13810       IF Left_join(Li)=Li+1 THEN
13815         FOR I=1 TO N/2                !\
13820           Dummy_x=Xplot(I)            ! \
13825           Dummy_y=Yplot(I)            !  \
13830           Xplot(I)=Xplot(N-I+1)       !   \  REVERSE ORDER OF POINTS TO
13835           Yplot(I)=Yplot(N-I+1)       !   /ACCOMODATE OTHER LINE SEGMENTS
```

```
13840           Xplot(N-I+1)=Dummy_x
13845           Yplot(N-I+1)=Dummy_y
13850         NEXT I
13855         Li_flag(Li+1)=1
13860         Flag=1
13865         FOR M=1 TO Pcount(Li+1)
13870           Q=M+N
13875           Xplot(Q)=Xcont(Li+1,M)
13880           Yplot(Q)=Ycont(Li+1,M)
13885         NEXT M
13890         N=Q
13895         Next_li=Li+1
13900         GOTO Again
13905       END IF
13910     END IF
13915 Next:!
13920     IF N<4 THEN                    !!----------TEST FOR SHORT CONTOUR LINES
13925       IF N=3 THEN
13930         Z=3
13935         Xplot(4)=Xplot(3)
13940         Yplot(4)=Yplot(3)
13945         Xplot(3)=Xplot(2)+(Xplot(4)-Xplot(2))/Z
13950         Yplot(3)=Yplot(2)+(Yplot(4)-Yplot(2))/Z
13955         Xplot(2)=Xplot(2)-(Xplot(2)-Xplot(1))/Z
13960         Yplot(2)=Yplot(2)-(Yplot(2)-Yplot(1))/Z
13965         N=4
13970       ELSE
13975         GOTO Continue        !!THROW OUT LINES WITH LESS THAN THREE POINTS
13980       END IF
13985     END IF
13990     IF Form$[1,5]="POLAR" THEN
13995       DEG
14000       FOR I=1 TO N
14005         X=Xplot(I)
14010         Xplot(I)=Yplot(I)*COS(X)
14015         Yplot(I)=Yplot(I)*SIN(X)
14020       NEXT I
14025     END IF
14030     IF Xplot(1)=Xplot(N) AND Yplot(1)=Yplot(N) THEN
14035       FOR M=1 TO 4
14040         Xplot(N+M)=Xplot(M+1)
14045         Yplot(N+M)=Yplot(M+1)
14050       NEXT M
14055       N=N+4
14060       First=3
14065     ELSE
14070       FOR M=N TO 1 STEP -1
14075         Yplot(M+1)=Yplot(M)
14080         Xplot(M+1)=Xplot(M)
14085       NEXT M
14090       Yplot(2)=.9*Yplot(1)+.1*Yplot(3)
14095       Xplot(2)=.9*Xplot(1)+.1*Xplot(3)
14100       N=N+1
14105       Yplot(N+1)=Yplot(N)
14110       Xplot(N+1)=Xplot(N)
14115       Yplot(N)=.9*Yplot(N+1)+.1*Yplot(N-1)
14120       Xplot(N)=.9*Xplot(N+1)+.1*Xplot(N-1)
14125       N=N+1
14130       First=1
14135     END IF
14140     ALLOCATE Tplot(N),Bx(N),Cx(N),Dx(N),By(N),Cy(N),Dy(N)
14145     Tplot(1)=0
14150               ! >SETUP INDEPENDENT PARAMETRIC VARIABLE
14155               !/
14160     FOR I=2 TO N
14165       Tplot(I)=Tplot(I-1)+SQR((Xplot(I)-Xplot(I-1))^2+(Yplot(I)-Yplot(I-1))
^2)
14170     NEXT I
14175         !\
14180         ! >CALCULATE PARAMETRIC CURVES DETERMINED BY THE CONTOUR POINTS
14185         !/
14190     CALL Spline(Tplot(*),Xplot(*),Bx(*),Cx(*),Dx(*),N,Err_flag)
14195     IF Interupt=1 THEN Interupt
14200     IF Err_flag=1 THEN Cont
14205     CALL Spline(Tplot(*),Yplot(*),By(*),Cy(*),Dy(*),N,Err_flag)
14210     IF Interupt=1 THEN Interupt
14215     IF Err_flag=1 THEN Cont
14220     Spot=(Tplot(N-First+1)-Tplot(First)-Space)*((INT(2*L-1) MOD 5)+1)/6+Spa
ce/2
14225     H=0
```

```
14230     Check=0
14235     PEN Colour(2)
14240     MOVE Xplot(First),Yplot(First)
14245                   !\
14250                   ! \PLOT THE CONTOUR LINE USING THE
14255                   ! /PARAMETRICALLY CUBIC SPLINES
14260                   !/
14265     FOR Tin=Tplot(First) TO Tplot(N-First+1)+Istep/2 STEP Istep
14270                   !\
14275                   ! >PLOT VALUE OF LEVEL & BREAK LINE AT THIS SPOT
14280                   !/
14285       IF Tin>Spot-Space/2 AND Check=0 THEN
14290         Check=1
14295         IF Tplot(N)>2*Space THEN    !\ONLY CREATE A SPACE IF THE LINE
14300                                     ! >IS LONG ENOUGH TO ACCEPT ONE
14305                                     !/
14310           Tin=Tin+Space/2
14315           Xout=FNSplint(Tplot(*),Xplot(*),Bx(*),Cx(*),Dx(*),Tin,N)
14320           Yout=FNSplint(Tplot(*),Yplot(*),By(*),Cy(*),Dy(*),Tin,N)
14325           Counter=Counter+1
14330           X_loc(L,Counter)=Xout
14335           Y_loc(L,Counter)=Yout
14340           Tin=Tin+Space/2
14345           Xout=FNSplint(Tplot(*),Xplot(*),Bx(*),Cx(*),Dx(*),Tin,N)
14350           Yout=FNSplint(Tplot(*),Yplot(*),By(*),Cy(*),Dy(*),Tin,N)
14355           MOVE Xout,Yout
14360         END IF
14365       END IF
14370       FOR K=H+1 TO N-First          !\  KEEP TRACK OF CURRENT CONTOUR
14375         IF Tplot(K)>Tin THEN 14385  ! \ POINT DURING PLOTTING TO SELECT
14380       NEXT K                        !  >THE APPROPRIATE SPLINE INTERVAL
14385       H=K-1                         ! / WITHIN THE PARAMETRIC SPLINE
14390                                     !/  FITTED ON THE HORIZONTAL PLANE.
14395       T_diff=Tin-Tplot(H)
14400       Xout=FNSplintnew(Xplot(H),Bx(H),Cx(H),Dx(H),T_diff)
14405       Yout=FNSplintnew(Yplot(H),By(H),Cy(H),Dy(H),T_diff)
14410       DRAW Xout,Yout
14415     NEXT Tin
14420 Cont:!
14425     PENUP
14430     DEALLOCATE Tplot(*),Bx(*),Cx(*),Dx(*),By(*),Cy(*),Dy(*)
14435 Continue:!
14440   NEXT Li
14445   Values(L)=Counter     !SAVE NUMBER OF DISTINCT CONTOUR LINES
14450                         !TO BE LABELED FOR THIS CONTOUR LEVEL
14455   FOR Li=1 TO Limax
14460     Pcount(Li)=0
14465   NEXT Li
14470   Limax=0
14475   SUBEXIT
14480 Interupt:Interupt=1
14485   GOTO 14445
14490 SUBEND
14495 !
14500 !
14505 !!*************************************************************************
14510 !!                                                                        *
14515 !!      Plot_values :   1st level SUBROUTINE TO PLOT LINE VALUES          *
14520 !!                                                                        *
14525 !!*************************************************************************
14530 SUB Plot_values(C(*),Char_size,INTEGER Levmax,Auto_flag)
14535   COM /Gen/ Lx(*),Ly(*),Form$,INTEGER Imax,Jmax
14540   COM /Colours/ INTEGER Colour(*),Plot_flag
14545   COM /Location/ X_loc(*),Y_loc(*),INTEGER Values(*)
14550   COM /Interupt/ INTEGER Interupt
14555   INTEGER K,L
14560   ON KEY 9 LABEL "INTERUPT",15 GOTO Interupt
14565   CLIP OFF
14570   PEN Colour(3)
14575   CSIZE Char_size,.45
14580   IF Plot_flag=0 THEN CSIZE 3.0,.44
14585   LORG 5
14590   FOR L=1 TO Levmax                 !\
14595     FOR K=1 TO Values(L)            ! \
14600       MOVE X_loc(L,K),Y_loc(L,K)    !  \LABEL PLOT WITH CONTOUR VALUES
14605       LABEL C(L)                    !  /
14610     NEXT K                          ! /
14615   NEXT L                            !/
14620   PEN 0
14625   IF Auto_flag=1 THEN SUBEXIT
```

```
14630   PRINTER IS 1
14635   PRINT CHR$(129)
14640   PRINT TABXY(42,1);"Contour plot is completed.            "
14645   PRINT TABXY(42,1);"Press CONTINUE when ready.            "
14650   PRINT CHR$(128)
14655   BEEP
14660   PAUSE
14665   PRINT CHR$(12)
14670   GINIT
14675   GCLEAR
14680   SUBEXIT
14685 Interupt:Interupt=1
14690 SUBEND
14695 !
14700 !
14705 !!*************************************************************************
14710 !!                                                                        *
14715 !!                S P L I N E   M O D U L E                               *
14720 !!                                                                        *
14725 !!                AUTHOR:  J E NEILSON                                    *
14730 !!                DATE: SEPT 1978.                                        *
14735 !!                                                                        *
14740 !!*************************************************************************
14745 SUB Spline(X(*),Y(*),B(*),C(*),D(*),INTEGER N,Err_flag)
14750   COM /Interupt/ INTEGER Interupt
14755   INTEGER I,Istop,J
14760   DIM Delta(500)
14765   ON KEY 9 LABEL "INTERUPT",15 GOTO Interupt
14770   ON ERROR GOTO Error
14775   Err_flag=0
14780   Istop=N-1
14785   FOR I=1 TO Istop
14790     D(I)=X(I+1)-X(I)
14795     Delta(I)=(Y(I+1)-Y(I))/D(I)
14800   NEXT I
14805   B(1)=-D(1)
14810   FOR I=2 TO Istop
14815     B(I)=2*(D(I-1)+D(I))-D(I-1)^2/B(I-1)
14820   NEXT I
14825   B(N)=-D(N-1)-D(N-1)^2/B(N-1)
14830   C(1)=D(1)^2*((Delta(3)-Delta(2))/(X(4)-X(2))-(Delta(2)-Delta(1))/(X(3)-X(
1)))/(X(4)-X(1))
14835   FOR I=2 TO Istop
14840     C(I)=(Delta(I)-Delta(I-1))-D(I-1)*C(I-1)/B(I-1)
14845   NEXT I
14850   C(N)=-D(N-1)^2*((Delta(N-1)-Delta(N-2))/(X(N)-X(N-2))-(Delta(N-2)-Delta(N
-3))/(X(N-1)-X(N-3)))/(X(N)-X(N-3))-D(N-1)*C(N-1)/B(N-1)
14855   C(N)=C(N)/B(N)
14860   FOR J=1 TO Istop
14865     I=N-J
14870     C(I)=(C(I)-D(I)*C(I+1))/B(I)
14875   NEXT J
14880   FOR I=1 TO Istop
14885     B(I)=Delta(I)-D(I)*(C(I+1)+2*C(I))
14890     D(I)=(C(I+1)-C(I))/D(I)
14895     C(I)=3*C(I)
14900   NEXT I
14905   SUBEXIT
14910 Interupt:Interupt=1
14915   SUBEXIT
14920 Error:Err_flag=1
14925 SUBEND
14930 !
14935 !
14940 !!*************************************************************************
14945 !!                                                                        *
14950 !!        CUBIC SPLINE FUNCTION EVALUATION WITH BINARY SEARCH             *
14955 !!                                                                        *
14960 !!*************************************************************************
14965 DEF FNSplint(X(*),Y(*),B(*),C(*),D(*),Xin,INTEGER N)
14970   REAL Dx,Splint
14975   INTEGER Left,Right,Mid
14980   Left=1
14985   Right=N
14990   Mid=(Left+Right)/2
14995   IF Mid=Right THEN 15025
15000   IF Xin<X(Mid) THEN 15015
15005   Left=Mid
15010   GOTO 14990
15015   Right=Mid
```

```
15020   GOTO 14990
15025   Dx=Xin-X(Left)
15030   Splint=Y(Left)+Dx*(B(Left)+Dx*(C(Left)+Dx*D(Left)))
15035   RETURN Splint
15040 FNEND
15045 !!*************************************************************************
15050 !!                                                                        *
15055 !!      CUBIC SPLINE INTERPOLATING FUNCTION                               *
15060 !!                                                                        *
15065 !!*************************************************************************
15070 DEF FNSplintnew(Depterm,Bterm,Cterm,Dterm,T)
15075   REAL Splint
15080   Splint=Depterm+T*(Bterm+T*(Cterm+T*Dterm))
15085   RETURN Splint
15090 FNEND
15095 !
15100 !
15105 !!*************************************************************************
15110 !!                                                                        *
15115 !!                          TITLE DRAWING ROUTINE                         *
15120 !!                                                                        *
15125 !!*************************************************************************
15130 SUB Contours
15135   OPTION BASE 1
15140   REAL Array(30,3),Sh_top(3,3),Sh_bot(3,3)
15145   INTEGER Shades,Flag,Points,I,J
15150   PRINTER IS 1
15155   PRINT CHR$(12)
15160   GINIT
15165   GRAPHICS ON
15170   VIEWPORT 0,140,9,100
15175   WINDOW 5,54,-10,20
15180   READ Sh_top(*),Sh_bot(*)
15185   DATA 29.5,-5,-1,  0,4,-1,   0,4,-1
15190   DATA 29.5,-5,-1,  0,-8,-1,   0,-8,-1
15195   !!----------------------------------------
15200   CSIZE 6,.4
15205   MOVE 29.5,18
15210   LORG 5
15215   LABEL "WELCOME TO"
15220   FOR J=1 TO 8
15225     READ Shades,Flag,Points
15230     REDIM Array(Points,3)
15235     READ Array(*)
15240     IF Flag=1 THEN
15245       AREA INTENSITY 0,0,0
15250       J=J-1
15255     ELSE
15260       AREA INTENSITY 1,1,1
15265     END IF
15270     MOVE J*6,4
15275     IPLOT Array(*),FILL,EDGE
15280     FOR I=1 TO Shades
15285       AREA INTENSITY .25,.25,.25
15290       READ Sh_top(2,1),Sh_top(3,1)
15295       MOVE Sh_top(1,1),Sh_top(1,2)
15300       PLOT Sh_top(*),FILL,EDGE
15305       Sh_bot(2,1)=Sh_top(2,1)
15310       Sh_bot(3,1)=Sh_top(3,1)
15315       AREA INTENSITY .5,.5,.5
15320       MOVE Sh_bot(1,1),Sh_bot(1,2)
15325       PLOT Sh_bot(*),FILL,EDGE
15330     NEXT I
15335   NEXT J
15340   WAIT 3
15345   GCLEAR
15350   SUBEXIT
15355 C:DATA 1,0,19,0,1,-2,     0,6,-1,      1,1,-1,      3,0,-1,      1,-1,-1
15360   DATA       0,-.5,-1,   -1,0,-1,     -.5,.5,-1,   -2,0,-1,    -.5,-.5,-1
15365   DATA       0,-5,-1,    .5,-.5,-1,   2,0,-1,      .5,.5,-1,    1,0,-1
15370   DATA       0,-.5,-1,   -1,-1,-1,    -3,0,-1,     -1,1,-1,     7,10
15375 O:DATA 1,0,9,0,1,-2,     0,6,-1,      1,1,-1,      3,0,-1,      1,-1,-1
15380   DATA       0,-6,-1,    -1,-1,-1,    -3,0,-1,     -1,1,-1,     13,16
15385   DATA 0,1,9,1,1.5,-2,   0,5,-1,    .5,.5,-1,    2,0,-1,      .5,-.5,-1
15390   DATA       0,-5,-1,    -.5,-.5,-1, -2,0,-1,     -.5,.5,-1
15395 N:DATA 2,0,10,0,8,-1,    1.3,0,-1,    2.7,-6,-1,   0,6,-1,      1,0,-1
15400   DATA       0,-8,-1,    -1.3,0,-1,   -2.7,6,-1,   0,-6,-1,     -1,0,-1
15405   DATA       18,19,      21.7,23
15410 T:DATA 1,0,9,0,7,-2,     0,1,-1,      5,0,-1,      0,-1,-1,     -2,0,-1
15415   DATA       0,-7,-1,    -1,0,-1,     0,7,-1,      -2,0,-1,     26,27
```

```
15420    DATA 1,0,9,0,1,-2,      0,6,-1,       1,1,-1,       3,0,-1,       1,-1,-1
15425    DATA       0,-6,-1,     -1,-1,-1,     -3,0,-1,      -1,1,-1,     31,34
15430    DATA 0,1,9,1,1.5,-2,    0,5,-1,     .5,.5,-1,     2,0,-1,       .5,-.5,-1
15435    DATA       0,-5,-1,     -.5,-.5,-1, -2,0,-1,       -.5,.5,-1
15440 U:DATA 1,0,13,0,1,-2,      0,7,-1,       1,0,-1,       0,-6.5,-1,    .5,-.5,-1
15445    DATA       2,0,-1,      .5,.5,-1,     0,6.5,-1,     1,0,-1,       0,-7,-1
15450    DATA       -1,-1,-1,    -3,0,-1,      -1,1,-1,      37,40
15455 R:DATA 2,0,12,0,8,-1,      4,0,-1,       1,-1,-1,      0,-2.5,-1,    -1,-1,-1
15460    DATA       -.6,0,-1,    1.6,-3.5,-1,  -1.1,0,-1,    -1.6,3.5,-1,  -1.3,0,-1
15465    DATA       0,-3.5,-1,   -1,0,-1,       42,43,       46,47
15470    DATA 0,1,7,  1,4.5,-2,    0,2.5,-1,     2.5,0,-1
15475    DATA       .5,-.5,-1,    0,-1.5,-1,    -.5,-.5,-1,   -2.5,0,-1
15480 S:DATA 1,0,27,0,1,-2,      0,.5,-1,      1,0,-1,       .5,-.5,-1,    2,0,-1
15490    DATA       .5,.5,-1,    0,1.5,-1,     -.5,.5,-1,    -2.5,0,-1,    -1,1,-1
15495    DATA       0,2.5,-1,    1,1,-1,       3,0,-1,       1,-1,-1,      0,-.5,-1
15500    DATA       -1,0,-1,     -.5,.5,-1,    -2,0,-1,      -.5,-.5,-1,   0,-1.5,-1
15505    DATA       .5,-.5,-1,   2.5,0,-1,     1,-1,-1,      0,-2.5,-1,    -1,-1,-1
15510    DATA       -3,0,-1,     -1,1,-1,      49,52
15515 SUBEND
15520 !
15525 !
15530 !!*************************************************************************
15535 !!                                                                         *
15540 !!            3-D  PLOTTING ROUTINE                                        *
15545 !!                                                                         *
15550 !!*************************************************************************
15555 SUB Three_d(Height,Damping,Avg,Max,Min,INTEGER Rightbound,Auto_flag)
15560    COM /Vertical/ Crossver(*),Aver(*),Bver(*),Cver(*),Dver(*),Ystep
15565    COM /Gen/ Lx(*),Ly(*),Form$,INTEGER Imax,Jmax
15570    COM /Horiz/ Crosshor(*),Ahor(*),Bhor(*),Chor(*),Dhor(*),Xstep
15575    COM /Interupt/ INTEGER Interupt
15580    COM /Damping/ Phor(*),Qhor(*),Pver(*),Qver(*)
15585    COM /Global/ File$,Grid$,INTEGER Form
15590    REAL Aspect,Bottom,Cophi,Cothe,Extra,Factor,Left,M1,M2,Phi,Right,Siphi
15595    REAL Sithe,Start,Stop,Theta,Top,Upper,X1,X2,Xin,Xrange,Xrel,Xx,Xx1,Xx2
15600    REAL Y1,Y2,Y3,Yin,Yout,Yrange,Yrel,Ystep,Yy,Yy1,Yy2
15605    INTEGER H,I,J,K,M,N,N1,N2,N3,N4,Vert,Total,Flag,Dif,Next,Backflag,Rflag
15610    ON KEY 9 LABEL "INTERUPT",15 GOTO Interupt
15615    Mini=Crosshor(1,1)
15620    FOR K=2 TO Imax
15625       IF Crosshor(1,K)<Mini THEN Mini=Crosshor(1,K)
15630    NEXT K
15635    FOR K=2 TO Jmax
15640       IF Crossver(Rightbound,K)<Mini THEN Mini=Crossver(Rightbound,K)
15645    NEXT K
15650    GRAPHICS ON
15655    GINIT
15660    DEG
15665    Xrange=Lx(Imax)-Lx(1)          !\
15670    Yrange=Ly(Jmax)-Ly(1)          ! \
15675    Extra=(Xrange+Yrange)/30       !  \
15680    Theta=25                       !   \
15685    Phi=25                         !    >INITIALIZE VARIABLES
15690    Cothe=COS(Theta)               !   /
15695    Cophi=COS(Phi)                 !  /
15700    Sithe=SIN(Theta)               ! /
15705    Siphi=SIN(Phi)                 !/
15710    Dif=INT(Xstep*60/Yrange*Cophi)+1
15715    Ystep=Xstep/Dif
15720    Aspect=1
15725    Factor=(Sithe*Xrange+Siphi*Yrange)/(Max-Mini)*Aspect
15730    VIEWPORT 0,130,7,100
15735    Left=Cothe*Lx(1)+Cophi*Ly(1)-Extra
15740    Right=Cothe*Lx(Imax)+Cophi*Ly(Jmax)+Extra
15745    Bottom=Sithe*Lx(1)+Siphi*Ly(1)+Mini*Factor-2*Extra
15750    Top=Sithe*Lx(Imax)+Siphi*Ly(Jmax)+Max*Factor+2*Extra
15755    WINDOW Left,Right,Bottom,Top
15760    N1=INT(Xrange/Ystep)+4
15765    N2=INT(Yrange/Ystep)+4
15770    N3=N2+Dif
15775    N4=MAX(N1,N2)
15780    ALLOCATE Xplot(N4),Yplot(N4),Ynew(N2),Yold(N3),Ydummy(N2),Ytemp(N1),Ykeep
(Dif+1)
15785    OFF ERROR
15790    FRAME
15795    CSIZE 2.7,.45
15800    LORG 1
15805    MOVE Left,Bottom
15810    LABEL File$&"  "&DATE$(TIMEDATE)&"  "&TIME$(TIMEDATE)&"   DAMPING = "&VAL
$(Damping)&"   NORMALIZING = "&VAL$(Avg)&"   AVERAGE HEIGHT = "&VAL$(Height)
```

```
15815   H=0          !\
15820   K=0          ! >INITIALIZE COUNTERS
15825   M=1          !/
15830                                                  !\
15835   FOR Xin=Lx(1) TO Lx(Imax)+Yystep/2 STEP Yystep  ! >PLOT FRONT
15840     IF Xin>Lx(Imax) THEN Xin=Lx(Imax)            !/ CROSS PLOT
15845     FOR I=M+1 TO Imax-1          !\
15850       IF Lx(I)>Xin THEN 15860    ! \SELECT APPROPRIATE TRANSVERSE
15855     NEXT I                       ! /GRID INTERVAL
15860     M=I-1                        !/
15865     Xrel=Xin-Lx(M)
15870     Upper=Lx(M+1)-Lx(M)
15875     Yout=FNRasplint(Ahor(1,M),Bhor(1,M),Chor(1,M),Dhor(1,M),Phor(M),Qhor(M)
,Xrel,Upper)
15880     K=K+1
15885     Ytemp(K)=Yout*Factor+Sithe*(Lx(Imax)-Xin)+Siphi*Ly(1)
15890     Xplot(K)=Cothe*Xin+Cophi*Ly(1)
15895   NEXT Xin
15900   MOVE Xplot(1),Ytemp(1)       !\
15905   FOR N=1 TO K                 ! \PLOT FRONT TRANSVERSE CROSS PLOT
15910     DRAW Xplot(N),Ytemp(N)     ! / AND KEEP THE HEIGHTS IN ARRAY
15915   NEXT N                       !/   'Ytemp' FOR FUTURE REFERENCE
15920   PENUP
15925   Y3=Ytemp(1)  !\SAVE CO-ORDINATE OF THE LEFTMOST POINT ON THE FRONT
15930                !/TRANSVERSE CROSS PLOT TO BE LEFT CORNER OF THE 3-D BLOCK
15935   Next=K        !\NUMBER OF POINTS DRAWN TO OBTAIN THEFRONT CROSS PLOT
15940   Backflag=0    !/
15945                                        !\
15950   FOR Vert=Rightbound TO 1 STEP -1     ! >PLOT EACH OF THE LONGITUDINAL CROSS
15955                                        !/ PLOTS STARTING FROM THE RIGHT
15960     X1=(Vert-1)*Xstep+Lx(1)    !\KEEP TRACK OF X CO-ORDINATE
15965     X2=Lx(Imax)-X1             !/
15970     M=1          !\
15975     K=0          ! \INITIALIZE COUNTERS AND FLAGS
15980     I=0          ! /
15985     Flag=0       !/
15990     FOR Yin=Ly(1) TO Ly(Jmax)+Yystep/2 STEP Yystep
15995       IF Yin>Ly(Jmax) THEN Yin=Ly(Jmax)
16000       FOR J=M+1 TO Jmax-1          !\
16005         IF Ly(J)>Yin THEN 16015    ! \SELECT APPROPRIATE LONGITUDINAL
16010       NEXT J                       ! /GRID INTERVAL
16015       M=J-1                        !/
16020       I=I+1       !COUNT THE POINTS ALONG A LONGITUDINAL CROSS PLOT
16025       Yrel=Yin-Ly(M)
16030       Upper=Ly(M+1)-Ly(M)
16035       Yout=FNRasplint(Aver(Vert,M),Bver(Vert,M),Cver(Vert,M),Dver(Vert,M),P
ver(M),Qver(M),Yrel,Upper)
16040       Ynew(I)=Siphi*Yin+Sithe*X2+Yout*Factor
16045       IF Vert=Rightbound THEN      !\
16050         K=K+1                      ! \STORE ALL POINTS FROM THE RIGHTMOST
16055         Yplot(I)=Ynew(I)           !  \CROSS-PLOT IN THE PLOTTING ARRAYS
16060         Xplot(I)=Cophi*Yin+Cothe*X1 !  /AND KEEP THESE NEW HEIGHTS AS THE
16065         Ydummy(I)=Ynew(I)          ! /  UPPERMOST HEIGHTS ON THE SCREEN
16070       ELSE                         !/
16075         IF Flag=0 THEN
16080           IF Ynew(I)<(Yold(I)-1.E-10) THEN
16085             Flag=1
16090             IF K=I-1 THEN          !\NO INTERPOLATION REQUIRED SINCE
16095               MOVE Xplot(1),Yplot(1) !/CROSS PLOT BEGINS AT THE FRONT
16100             ELSE                   !\
16105               J=I-K                ! \ INTERPOLATE AT THE BEGINNING
16110               Yy1=Ynew(I-K-1)      !  \   OF THE CURRENT LINE TO
16115               GOSUB Anotherstart   !  /  REACH THE EXISTING LINE
16120               MOVE Xx,Yy           ! /
16125             END IF                 !/
16130             FOR N=1 TO K           !\
16135               DRAW Xplot(N),Yplot(N) ! >PLOT CURRENT LINE
16140             NEXT N                 !/
16145             J=I                    !\
16150             Xx1=Xplot(K)           ! \
16155             Xx2=Cophi*Yin+Cothe*X1 !  \INTERPOLATE AT THE END OF THE
16160             Yy1=Yplot(K)           !   >   CURRENT LINE TO REACH
16165             Yy2=Ynew(I)            !  /      THE EXISTING LINE
16170             GOSUB Anotherend       ! /
16175             DRAW Xx,Yy             !/
16180             PENUP
16185             K=0       ! RESET POINT COUNTER TO ZERO
16190             Ydummy(I)=Yold(I)  !KEEP OLD HEIGHT AS UPPERMOST
16195                                !HEIGHT ON THE SCREEN
16200           ELSE
```

```
16205             K=K+1                                !\
16210             Xplot(K)=Cophi*Yin+Cothe*X1 ! \STORE NEW POINT IN PLOTTING
16215             Yplot(K)=Ynew(I)            !  >ARRAYS AND KEEP THIS NEW
16220             Ydummy(I)=Ynew(I)           ! / HEIGHT AS THE UPPERMOST
16225           END IF                        !/   HEIGHT ON THE SCREEN
16230         ELSE                              !\ CROSS PLOT REMAINS HIDDEN
16235           IF Ynew(I)<Yold(I)-1.E-10 THEN  ! >KEEP POINT FROM OLD DRAWN LINE
S
16240             Ydummy(I)=Yold(I)               !/AS THE UPPERMOST POINTS SO FAR
16245           ELSE
16250             Flag=0                        !\
16255             K=1                           ! \CROSS PLOT IS NO LONGER HIDDEN
16260             Yplot(K)=Ynew(I)              !  >STORE FIRST POINT AND RESET
16265             Xplot(K)=Cophi*Yin+Cothe*X1   ! /FLAG TO SIGNIFY THAT THIS PART
16270             Ydummy(I)=Ynew(I)             !/OF THE CROSS PLOT IS TO BE DRAWN
16275           END IF
16280         END IF
16285       END IF
16290     NEXT Yin
16295     IF Vert=Rightbound THEN  !\
16300       Total=I                ! \   THE FIRST & LAST POINTS FROM THE
16305       Y1=Yplot(1)            ! /RIGHTMOST CROSS PLOT ARE STORED AS THE
16310       Y2=Yplot(K)            !/CENTRE AND RIGHT CORNERS OF THE 3-D BLOCK
16315     END IF
16320     IF K<>0 THEN    !THE END OF THE LONGITUDINAL CROSS PLOT IS SEEN
16325       H=H+1         !COUNT CURRENT NUMBER OF CROSS PLOTS WITH ENDS SHOWING
16330       Backflag=1    !FLAG SET SO THAT THE BACK CROSS PLOT WILL BE DRAWN
16335       IF H=1 THEN                       !\STORE THE 'Dif' RIGHTMOST POINTS
16340         FOR J=1 TO Dif+1                ! \ON THE 3-D VIEW WHICH WERE
16345           Ykeep(J)=Yold(Total+J-1)      !  >PLOTTED BEFORE THIS CURRENT
16350         NEXT J                          ! /CROSS PLOT WHICH HAS THE SPECIAL
16355       END IF                            !/FEATURE THAT ITS END IS FULLY SEEN
16360       IF K=I THEN                   !\
16365         MOVE Xplot(1),Yplot(1)      ! \BEGIN DRAWING LINE:
16370       ELSE                          !  \     IF THIS CROSS PLOT HAD BEEN
16375         J=I-K+1                     !   \HIDDEN, THEN LINEAR INTERPOLATION IS
16380         Yy1=Ynew(I-K)               !   /REQUIRED FOR IT TO INTERSECT WITH THE
16385         GOSUB Anotherstart          !  / EXISTING CROSS PLOT WHICH BLOCKED
16390         MOVE Xx,Yy                  ! /  THE VIEW OF THE CURRENT CROSS PLOT
16395       END IF                        !/
16400       FOR N=1 TO K                    !\
16405         DRAW Xplot(N),Yplot(N)        ! >DRAW FROM POINT TO POINT
16410       NEXT N                          !/ FORMING A CONTINUOUS LINE
16415       PENUP
16420     ELSE                        !\CURRENT ENDING OF CROSS PLOT IS HIDDEN, BUT
16425       IF Backflag=1 THEN        ! >IF THE PREVIOUS CROSS PLOT IS SHOWING THEN
16430 Back_cross_plot:                !/FILL IN THE BACK TRANSVERSE CROSS PLOT
16435         Backflag=0  !\ RESET FLAG
16440         M=1         !/ SET COUNTER
16445         Start=Xstep*(Vert)+Lx(1)    !BACK CROSS PLOT STARTS AT END OF THE
16450         IF H+Vert=Rightbound THEN      !\             !PREVIOUS CROSS PLOT
16455           Rflag=1                      ! \
16460           Stop=Lx(Imax)                !  \  IN ORDER TO FILL IN THE BACK
16465         ELSE                           !   > BACK TRANSVERSE CROSS PLOT,
16470           Rflag=0                      !  /  COMPUTE POINT AT WHICH IT
16475           Stop=Xstep*(Vert+H)+Lx(1)    ! /   WILL BECOME HIDDEN
16480         END IF                         !/
16485         J=1                                          !INTERPOLATE ALONG
16490         FOR Xin=Start TO Stop+Yystep/100 STEP Yystep  !THE BACK TRANSVERSE
16495           IF Xin>Stop+Yystep/100 THEN Transverseplot  ! CROSS PLOT
16500           FOR I=M+1 TO Imax-1           !\
16505             IF Lx(I)>Xin THEN 16515     ! \SELECT APPROPRIATE
16510           NEXT I                        ! /TRANSVERSE GRID INTERVAL
16515           M=I-1                         !/
16520           Xrel=Xin-Lx(M)
16525           Upper=Lx(M+1)-Lx(M)
16530           Yout=FNRasplint(Ahor(Jmax,M),Bhor(Jmax,M),Chor(Jmax,M),Dhor(Jmax,
M),Phor(M),Qhor(M),Xrel,Upper)
16535           K=K+1
16540           Yplot(K)=Yout*Factor+Sithe*(Lx(Imax)-Xin)+Siphi*Ly(Jmax)
16545           Xplot(K)=Cothe*Xin+Cophi*Ly(Jmax)
16550           IF Xin>Stop-Xstep+Yystep/100 AND Rflag=0 THEN
16555             J=J+1                             !\   IF THE BACK CROSS PLOT DOES
16560             IF Yplot(K)<Ykeep(J) THEN         ! \  NOT EXTEND UP TO THE RIGHT
16565               K=K-1                           !  \ BOUNDARY, THEN INTERPOLATE
16570               J=J-1                           !   >ALONG IT UNTIL IT FALLS
16575               GOTO Transverseplot             !  / BEHIND A PREVIOUSLY DRAWN
16580             END IF                            ! /  LONGITUDINAL CROSS PLOT
16585           END IF                              !/
16590         NEXT Xin
```

```
16595 Transverseplot:!
16600         MOVE Xplot(1),Yplot(1)      !\
16605         FOR N=1 TO K                ! \DRAW THE BACK TRANSVERSE CROSS PLOT
16610           DRAW Xplot(N),Yplot(N)    ! /
16615         NEXT N                      !/
16620         IF Rflag=0 THEN
16625           M1=(Yplot(K)-Yplot(K+1))/(Xplot(K)-Xplot(K+1)) ! INTERSECT THE
16630           M2=(Ykeep(J)-Ykeep(J+1))/(Xplot(K)-Xplot(K+1)) ! BACK CROSS
16635           Xx=(Yplot(K)-Ykeep(J))/(M2-M1)+Xplot(K)        ! PLOT WITH THE
16640           Yy=M1*(Xx-Xplot(K))+Yplot(K)                   ! APPROPRIATE
16645           DRAW Xx,Yy                                     ! LONGITUDINAL
16650         END IF                                           ! CROSS PLOT
16655         PENUP
16660         H=0       !RESET COUNTER
16665       END IF
16670     END IF
16675     IF Vert=1 AND Backflag=1 THEN   !\
16680       K=0                           ! \ IF THE ENDING OF THE LEFTMOST
16685       Vert=0                        !  >CROSS PLOT IS SHOWING THEN
16690       GOTO Back_cross_plot          ! / GO BACK TO FILL IN THE BACK
16695     END IF                          !/  TRANSVERSE CROSS PLOT
16700     IF Vert<=1 THEN 16780
16705     Next=Next-Dif       !
16710     FOR I=1 TO Dif              !\UPPERMOST POINTS TO TEST AT THE BEGINNING
16715       Yold(I)=Ytemp(Next+I-1)   ! >OF THE NEXT CROSS PLOT COMES FROM THE
16720     NEXT I                      !/POINTS DRAWN TO FORM THE FRONT CROSS PLOT
16725                                     !\KEEP TRACK OF ADDITIONAL POINTS
16730     FOR I=Total+Dif TO Total+1 STEP -1  ! \SO THAT THE BACK CROSS (WHEN
16735       Yold(I)=Yold(I-Dif)               ! /DRAWN) CAN BE INTERSECTED WITH
16740     NEXT I                              !/THE EXISTING LINES
16745     FOR I=Dif+1 TO Total      !\MOST OF THE UPPERMOST HEIGHTS COME
16750       Yold(I)=Ydummy(I-Dif)   ! >FROM THE SET OF ALL LONGITUDINAL
16755     NEXT I                    !/CROSS PLOTS DRAWN SO FAR
16760   NEXT Vert
16765         !\
16770         ! >DRAW THE TWO FRONT (flat) SIDES OF 3-D BLOCK
16775         !/
16780   MOVE Cothe*Lx(1)+Cophi*Ly(1),Y3
16785   DRAW Cothe*Lx(1)+Cophi*Ly(1),Sithe*Xrange+Siphi*Ly(1)+Mini*Factor
16790   DRAW Cothe*Lx(Imax)+Cophi*Ly(1),Siphi*Ly(1)+Mini*Factor
16795   DRAW Cothe*Lx(Imax)+Cophi*Ly(1),Y1
16800   MOVE Cothe*Lx(Imax)+Cophi*Ly(1),Siphi*Ly(1)+Mini*Factor
16805   DRAW Cothe*Lx(Imax)+Cophi*Ly(Jmax),Siphi*Ly(Jmax)+Mini*Factor
16810   DRAW Cothe*Lx(Imax)+Cophi*Ly(Jmax),Y2
16815   PENUP
16820   DEALLOCATE Xplot(*),Yplot(*),Ynew(*),Yold(*),Ydummy(*),Ytemp(*),Ykeep(*)
16825   IF Auto_flag=1 THEN SUBEXIT
16830   PRINTER IS 1
16835   PRINT CHR$(129)
16840   PRINT TABXY(50,1);"3-D plot is completed."
16845   PRINT TABXY(50,1);"Press CONTINUE when ready"
16850   PRINT CHR$(128)
16855   BEEP
16860   BEEP
16865   PAUSE
16870   SUBEXIT
16875 Anotherstart:!
16880   Xx1=Xplot(1)-Cophi*Yystep !\
16885   Xx2=Xplot(1)              ! >SETUP COEFFICIENTS FOR LINEAR INTERSECTION
16890   Yy2=Yplot(1)              !/ OF 'SEEN' LINES
16895 Anotherend:!
16900   M1=(Yy1-Yy2)/(Xx1-Xx2)          !\
16905   M2=(Yold(J-1)-Yold(J))/(Xx1-Xx2) ! \PERFORM LINEAR INTERSECTION
16910   Xx=(Yy1-Yold(J-1))/(M2-M1)+Xx1   ! /TO REMOVE ANY GAPS WHENEVER
16915   Yy=M1*(Xx-Xx1)+Yy1               !/ TWO LINES MEET
16920   RETURN
16925 Interupt:Interupt=1
16930 SUBEND
16935 !
16940 !
16945 !!*************************************************************************
16950 !!                                                                        *
16955 !!    NEWPOINTS: 3rd level subroutine giving additional contour points    *
16960 !!               along the transverse grid lines.                         *
16965 !!                                                                        *
16970 !!*************************************************************************
16975 SUB Newpoints(Y1,Y2,Xsave,Keepx(*),Keepy(*),Level,INTEGER H,V)
16980   COM /Gen/ Lx(*),Ly(*),Form$,INTEGER Imax,Jmax
16985   COM /Horiz/ Crosshor(*),Ahor(*),Bhor(*),Chor(*),Dhor(*),Xstep
16990   COM /Vertical/ Crossver(*),Aver(*),Bver(*),Cver(*),Dver(*),Ystep
```

```
16995   COM /Interupt/ INTEGER Interupt
17000   COM /Damping/ Phor(*),Qhor(*),Pver(*),Qver(*)
17005   REAL Wtemp(8),W(4),Test,Upper
17010   INTEGER G,I,J,K,L,M,N,V1,Good,Old,Flag,Testsave
17015   ON KEY 9 LABEL "INTERUPT",15 GOTO Interupt
17020   H=0
17025   V1=V-1
17030   FOR I=1 TO Imax                              !\
17035     IF Lx(I)>Xsave THEN 17045                  ! \SELECT APPROPRIATE GRID
17040   NEXT I                                       ! /INTERVAL ON THE CHOSEN
17045   M=I-1                                        !/ TRANSVERSE CROSS PLOTS
17050   IF Xsave-Xstep<Lx(M) THEN                 !\
17055     Flag=1                                  ! \ SET FLAG IF TWO GRID INTERVALS ARE
17060   ELSE                                      !  >NEEDED TO DETERMINE THE ADDITIONAL
17065     Flag=0                                  ! / CONTOUR POINT WHEN IT IS EXPECTED
17070   END IF                                    !/    TO BE FOUND NEAR A DATA POINT
17075   FOR L=2 TO Jmax-1                  !\
17080     IF Ly(L)>Y1 THEN 17090           ! \
17085   NEXT L                             !  \DETERMINE WHICH TRANSVERSE
17090   L1=L-1                             !   >CROSS PLOTS LIE BETWEEN THE
17095   FOR K=Jmax-1 TO 2                  !  /TWO EXISTING CONTOUR POINTS
17100     IF Ly(K)<Y2 THEN 17110           ! /
17105   NEXT K                             !/
17110   FOR Yin=Y1+Ystep TO Y2 STEP Ystep
17115     Upper_y=Ly(L)-Ly(L-1)
17120     IF Yin>Ly(L)-Ystep/1.5 AND L<=K THEN    !\
17125       Yin=Ly(L)                             ! \FIND ADDITIONAL POINTS
17130       GOSUB Raspl_int                       !  >ON THE TRANSVERSE
17135       L1=L                                  ! /    CROSS PLOTS
17140       L=L+1                                 !/
17145     ELSE
17150       IF Yin>Y2-Ystep/1.5 THEN SUBEXIT
17155       GOSUB Lin_int  !FIND ADDIT. POINTS BETWEEN THE TRANSVERSE GRID LINES
17160     END IF
17165   NEXT Yin
17170   IF L<=K THEN 17120
17175   SUBEXIT
17180 Raspl_int:!
17185   Old=0                !   SET NUMBER OF ADDITIONAL POINTS FOUND ON THE
17190   Testsave=0           !   TRANSVERSE CROSS PLOT TO ZERO
17195   FOR G=M-Flag TO M
17200     Upper=Lx(G+1)-Lx(G)
17205     CALL Interpnew(W(*),Lx(G),Ahor(L,G),Bhor(L,G),Chor(L,G),Dhor(L,G),Phor(
G),Qhor(G),Level,0,Upper,Good)
17210     FOR N=1 TO Good                    !\
17215       Wtemp(Old+N)=W(N)                ! \STORE COMPUTED CONTOUR
17220     NEXT N                             ! /POINTS AND COUNT THEM
17225     Old=Good+Old                       !/
17230   NEXT G
17235   IF Old>=1 THEN                                  !\
17240     Test=Lx(Imax)-Lx(1)                           ! \   SELECT THE CLOSEST
17245                                                   !  \  ADDITIONAL CONTOUR
17250     FOR N=1 TO Old                                !   \ POINT IF MORE THAN
17255       IF Wtemp(N)>Xsave-Xstep AND Wtemp(N)<Xsave THEN
17260         IF ABS(Wtemp(N)-Xsave)<Test THEN         !    >ONE WAS FOUND IN THE
17265           Test=ABS(Wtemp(N)-Xsave)               !   / GIVEN INTERVAL(S)
17270           Testsave=N                             !  /  ALONG THE GIVEN
17275         END IF                                   ! /  TRANSVERSE CROSS PLOT
17280       END IF                                     !/
17285     NEXT N
17290     IF Testsave>0 THEN
17295       H=H+1
17300       Keepx(H)=Wtemp(Testsave)
17305       Keepy(H)=Yin
17310     END IF
17315   END IF
17320   RETURN
17325 Lin_int:     !\ PERFORM LINEAR INTERPOLATION BETWEEN ADJACENT
17330              ! >LONGITUDINAL CROSS PLOTS AND BETWEEN THE TRANSVERSE
17335              !/ GRID LINES TO FIND ADDITIONAL CONTOUR POINTS
17340   Y_diff=Yin-Ly(L-1)
17345   Yy1=FNRasplint(Aver(V1,L1),Bver(V1,L1),Cver(V1,L1),Dver(V1,L1),Pver(L1),Q
ver(L1),Y_diff,Upper_y)
17350   Yy2=FNRasplint(Aver(V,L1),Bver(V,L1),Cver(V,L1),Dver(V,L1),Pver(L1),Qver(
L1),Y_diff,Upper_y)
17355   IF (Level-Yy1)*(Level-Yy2)<0 THEN
17360     H=H+1
17365     Keepx(H)=Xsave+Xstep*(Level-Yy2)/(Yy2-Yy1)
17370     Keepy(H)=Yin
17375   END IF
```

```
17380   RETURN
17385 Interupt:Interupt=1
17390 SUBEND
17395 !!*************************************************************************
17400 !!                                                                         *
17405 !!     INTERPNEW: SUBROUTINE TO EXTRACT LOCATIONS OF CONTOUR POINTS        *
17410 !!                BY COMPUTING THE ROOTS OF THE FITTED RATIONAL SPLINES    *
17415 !!                                                                         *
17420 !!*************************************************************************
17425 SUB Interpnew(W(*),Add_on,Z11,Z22,Z33,Z44,Pp,Qq,Level,Lower,Upper,INTEGER G
ood)
17430   COM /Interupt/ INTEGER Interupt
17435   REAL Wtemp(4),A,A0,A1,A2,A3,A4,Check,Coef_a,Coef_b,Coef_c,D,Discrim
17440   REAL Dummy,Dummy1,Dummy2,E,P,Q,R,Test,U,Y,Z,Z2,Z3,Z4
17445   INTEGER H,I,J,K,Isave,Counter,Quartic
17450   ON KEY 9 LABEL "INTERUPT",15 GOTO Interupt
17455   ON ERROR GOTO Error
17460   RAD
17465   Good=0                  !\
17470   W(1)=0                  ! \
17475   W(2)=0                  !  >INITIALIZE VARIABLES
17480   W(3)=0                  ! /
17485   W(4)=0                  !/
17490   !Equations from CRC Standard Mathematical Tables, 16th edition, page 96
17495      !     FROM THE RATIONAL SPLINE COEFFICIENTS, TERMS ARE GROUPED
17500      !UNDER A COMMON DENOMINATOR TO OBTAIN A POLYNOMIAL OF DEGREE FOUR.
17505      !     IF THE VALUE OF EITHER OF THE HIGHEST TWO SPLINE COEFFICIENTS
17510      !IS ZERO, DIFFERENT NORMALIZING EQUATIONS MUST BE USED AND A LOWER
17515      !ORDER POLYNOMIAL WILL BE GENERATED.
17520   A4=0
17525   IF Z44<>0 THEN
17530     A0=-(Qq+1)*(Level-Z11-Z33)
17535     IF Z33<>0 THEN
17540       A4=Qq*(Z44+Z33)
17545       A1=-(Qq*Qq*(Level-Z11)+(Qq+1)*(Z11-Z22)+(4*Qq+3)*Z33)
17550       A2=(Qq*Qq*(Level-2*Z11+Z22)+Z33*(6*Qq+3))
17555       A3=(Qq*Qq*(Z11-Z22)-Z33*(4*Qq+1)+Z44)
17560     ELSE
17565       A1=Qq*(Level-Z11)-(Qq+1)*(Z11-Z22)
17570       A2=-Qq*(Z11-Z22)
17575       A3=Z44
17580     END IF
17585   ELSE
17590     IF Z33<>0 THEN
17595       A0=(Level-Z11-Z33)
17600       A1=(Qq*(Level-Z11)+Z11-Z22+3*Z33)
17605       A2=(Qq*(Z11-Z22)-3*Z33)
17610       A3=Z33
17615     ELSE
17620       A3=0
17625       A2=0
17630       A1=Z11-Z22
17635       A0=Level-Z11
17640     END IF
17645   END IF
17650        !CHECK POLYNOMIAL COEFICIENTS TO SELECT APPROPRIATE ROOT FORMULAS
17655   IF A4<>0 THEN
17660     Quartic=1 !FLAG SET SO THAT PROGRAMME RETURNS BACK AFTER ANALYZING
17665               !GENERATED CUBIC EQUATION IN ORDER TO COMPUTE QUARTIC ROOTS
17670     A0=A0/A4        !\
17675     A1=A1/A4        ! \NORMALIZE QUARTIC EQUATION AND IS NOW IN THE FORM:
17680     A2=A2/A4        ! /  x^4 + A3x^3 + A2x^2 + A1x + A0 = 0
17685     A3=A3/A4        !/
17690     Coef_a=-A2                        !\
17695     Coef_b=A3*A1-4*A0                 ! \GENERATE SECONDARY CUBIC EQUATION
17700     Coef_c=-A3*A3*A0+4*A2*A0-A1*A1    ! /   AND SOLVE FOR ITS ROOTS
17705     GOSUB Cubic                       !/  FORM OF THE CUBIC IS :
17710                        !    y^3 + (Coef_a)y^2 + (Coef_b)y + (Coef_c) =0
17715     K=1                               !\
17720     R=A3*A3/4-A2+Wtemp(K)             ! \
17725     IF ABS(R)<1.E-10 THEN R=0         !  \
17730     IF R<0 THEN                       !   \ TAKE CUBIC ROOT AND TEST IF IT WILL
17735       K=K+1                           !   / WORK SATISFACTORIALLY (IT SHOULD)
17740       IF K<=Counter THEN 17720        !  /  IF NOT, TEST THE NEXT CUBIC ROOT
17745     END IF                            ! /
17750     Y=Wtemp(K)                        !/
17755     R=SQR(R)                    !\COMPUTE TERMS INVOLVED IN THE QUARTIC ROOTS
17760     Dummy1=.75*A3*A3-2*A2       !/
17765     IF R=0 THEN      !USE DIFFERENT EQUATIONS BASED ON THE VALUE OF 'R'
17770       Dummy=Y*Y-4*A0
```

```
17775       IF ABS(Dummy)<1.E-10 THEN Dummy=0  !LOOK OUT FOR ROUND-OFF ERROR
17780       IF Dummy<0 THEN    !\
17785         Counter=0        ! >  NO REAL ROOTS
17790         GOTO 18535       !/
17795       ELSE
17800         D=Dummy1+2*SQR(Dummy)
17805         E=Dummy1-2*SQR(Dummy)
17810       END IF
17815     ELSE
17820       Dummy=(4*A3*A2-8*A1-A3*A3*A3)/(4*R)
17825       IF ABS(Dummy)<1.E-10 THEN Dummy=0  !LOOK OUT FOR ROUND-OFF ERROR
17830       D=Dummy1-R*R+Dummy
17835       E=Dummy1-R*R-Dummy
17840     END IF
17845     Counter=0
17850     IF ABS(D)<1.E-10 THEN D=0    !\LOOKING OUT FOR ROUND-OFF ERRORS
17855     IF ABS(E)<1.E-10 THEN E=0    !/
17860     IF D=0 AND E=0 THEN     !\
17865       Counter=2             ! \ONE REAL DOUBLE ROOT
17870       Wtemp(1)=-A3/4+R/2    ! /
17875       Wtemp(2)=Wtemp(1)     !/
17880     ELSE
17885       IF D>=0 THEN                !\
17890         D=SQR(D)                  ! \
17895         Counter=2                 !  \TWO REAL ROOTS
17900         Wtemp(1)=-A3/4+R/2+D/2    !  /
17905         Wtemp(2)=-A3/4+R/2-D/2    ! /
17910       END IF                      !/
17915       IF E>=0 THEN                          !\
17920         E=SQR(E)                            ! \
17925         Wtemp(Counter+1)=-A3/4-R/2+E/2      !  \TWO REAL ROOTS
17930         Wtemp(Counter+2)=-A3/4-R/2-E/2      !  /
17935         Counter=Counter+2                   ! /
17940       END IF                                !/
17945     END IF
17950     FOR I=1 TO Counter-1                !\
17955       FOR J=I+1 TO Counter              ! \
17960         IF Wtemp(I)=Wtemp(J) THEN       !  \CHECK FOR DOUBLE ROOTS, SINCE
17965           Wtemp(I)=Wtemp(I)*.9999       !   \ ROOTS BASED INDEPENDENTLY
17970           Wtemp(J)=Wtemp(J)*1.0001      !   /   ON 'D' AND ON 'E' MAY
17975         END IF                          !  /     HAVE BEEN REPEATED
17980       NEXT J                            ! /
17985     NEXT I                              !/
17990   ELSE
17995     Quartic=0   !\
18000     A=A0        ! \SINCE THE COEFFICIENT OF THE HIGHEST DEGREE TERM IN
18005     Z2=A1       !  >THE QUARTIC EQUATION IS IN FACT ZERO, THE QUARTIC
18010     Z3=A2       ! /  DEGENERATES INTO A LOWER ORDER POLYNOMIAL, AND
18015     Z4=A3       !/  HENCE THIRD ORDER COEFFICIENTS MUST BE INITIALIZE
18020     IF Z4<>0 THEN    !\
18025                      ! > SOLVE CUBIC EQUATION FOR ROOTS
18030                      !/
18035       Coef_a=Z3/Z4       !\  AGAIN THE CUBIC IS IN THE FORM  :
18040       Coef_b=Z2/Z4       ! >  y^3 + (Coef_a)y^2 + (Coef_b)y + (Coef_c) = 0
18045       Coef_c=A/Z4        !/
18050 Cubic:     !\
18055            ! > IF QUARTIC, COMPUTE ROOTS OF GENERATED CUBIC STARTING HERE
18060            !/
18065       P=Coef_a*Coef_a/9-Coef_b/3
18070       Q=(Coef_b*Coef_a/3-Coef_c-2*(Coef_a/3)^3)/2
18075       Test=Q*Q-P^3
18080       IF Test>0 THEN  !\
18085         Counter=1     ! >------ONE REAL ROOT-------
18090                       !/
18095         Dummy1=Q+SQR(Test)
18100         Dummy2=Q-SQR(Test)
18105         Wtemp(1)=SGN(Dummy1)*ABS(Dummy1)^(1/3)+SGN(Dummy2)*ABS(Dummy2)^(1/3
)
18110       ELSE                !\
18115         Counter=3         ! >----TWO ROOTS; WHERE ONE IS A DOUBLE ROOT-----
18120         IF Test=0 THEN    !/
18125           Wtemp(1)=2*SGN(Q)*ABS(Q)^(1/3)
18130           Wtemp(2)=Wtemp(1)/(-2.001)      !\DUE TO ORDERING ALGORITM TWO
18135           Wtemp(3)=Wtemp(1)/(-1.999)      ! > CLOSELY SPACED ROOTS ARE
18140                                           !/ NEEDED FOR THE DOUBLE ROOT
18145         ELSE     !!----------------THREE DISTINCT REAL ROOTS-------------
18150           U=ACS(Q/(P^1.5))/3
18155           Z=2*SQR(P)
18160           Wtemp(1)=Z*COS(U)
18165           Wtemp(2)=Z*COS(U+2*PI/3)
```

```
18170           Wtemp(3)=Z*COS(U+4*PI/3)
18175         END IF
18180       END IF
18185             !\
18190             ! >ADJUST ROOT VALUES ACCORDING TO CUBIC FORMULA
18195             !/
18200       FOR H=1 TO Counter
18205         Wtemp(H)=Wtemp(H)-Coef_a/3
18210       NEXT H
18215     ELSE                !\
18220       IF Z3<>0 THEN     ! >SOLVE QUADRATIC EQUATION FOR ITS ROOTS
18225                         !/
18230         Discrim=Z2*Z2-4*A*Z3 !\
18235         IF Discrim=0 THEN    ! > ONE REAL DOUBLE ROOT
18240           Counter=2          !/
18245           Wtemp(1)=Z2/(-2.0001*Z3) !\DUE TO ORDERING ALGORITM TWO
18250           Wtemp(2)=Z2/(-1.9999*Z3) !/CLOSELY SPACED ROOTS ARE NEEDED
18255         ELSE                 !\
18260           IF Discrim>0 THEN  ! > TWO DISTINCT REAL ROOTS
18265             Counter=2        !/
18270             Wtemp(1)=(SQR(Discrim)-Z2)/(2*Z3)
18275             Wtemp(2)=(-1*SQR(Discrim)-Z2)/(2*Z3)
18280           END IF
18285         END IF
18290       ELSE                   !\
18295         IF Z2<>0 THEN        ! >SOLVE LINEAR EQUATION FOR ROOT
18300                              !/
18305           Counter=1
18310           Wtemp(1)=-1*A/Z2
18315         ELSE
18320           IF A=0 THEN
18325             Counter=2
18330             Wtemp(1)=Lower
18335             Wtemp(2)=1
18340           END IF
18345         END IF
18350       END IF
18355     END IF
18360     IF Quartic=1 THEN RETURN !\AFTER ROOTS FROM GENERATED CUBIC ARE
18365                              !/COMPUTED, RETURN TO CALCULATE QUARTIC ROOTS
18370   END IF
18375           !\
18380           ! > DETERMINE IF ROOTS LIE IN THE VALID INTERVAL
18385           !/
18390   FOR H=1 TO Counter
18395     IF PROUND(Wtemp(H)-Lower,-6)=0 THEN  !\
18400       Wtemp(H)=Lower                     ! \
18405     END IF                               !  \LOOK OUT FOR ROUND-OFF ERROR
18410     IF PROUND(Wtemp(H)-1,-6)=0 THEN      !  /AT HE INTERVAL BOUNDARIES
18415       Wtemp(H)=1                         ! /
18420     END IF                               !/
18425     IF Wtemp(H)>=Lower AND Wtemp(H)<=1 THEN !\  COMPUTE THE TRUE VALUE
18430       Wtemp(H)=Wtemp(H)*Upper               ! \ OF THE CONTOUR POINT BY
18435       Good=Good+1                           !  >EXPANDING AND SHIFTING THE
18440       W(Good)=Wtemp(H)+Add_on               ! / ROOT TO COMPENSATE FOR THE
18445     END IF                                  !/  COMPRESSED AND TRANSLATED
18450   NEXT H                                    !NATURE OF THE RATIONAL SPLINE
18455   FOR J=1 TO Good-1           !\
18460     Isave=0                   ! \
18465     Check=W(J)                !  \
18470     FOR I=J+1 TO Good         !   \
18475       IF W(I)<Check THEN      !    \
18480         Check=W(I)            !     \
18485         Isave=I               !      \RE-ORDER FOUND CONTOUR POINTS
18490       END IF                  !       >IN INCREASING VALUE OF THE
18495     NEXT I                    !      /LONGITUDINAL COORDINATE   'Y'
18500     IF Isave<>0 THEN          !     /
18505       Dummy=W(J)              !    /
18510       W(J)=W(Isave)           !   /
18515       W(Isave)=Dummy          !  /
18520     END IF                    ! /
18525   NEXT J                      !/
18530   OFF ERROR
18535   SUBEXIT
18540 Error:OFF ERROR
18545   Good=0
18550   SUBEXIT
18555 Interupt:Interupt=1
18560 SUBEND
18565 !!*************************************************************************
```

```
18570 !!                                                                      *
18575 !!    Raspl1 :  RATIONAL SPLINE INTERPOLATING ROUTINE                   *
18580 !!                                                                      *
18585 !!        AUTHOR:   H. SPATH                                            *
18590 !!        DATE:     1974                                                *
18595 !!                                                                      *
18600 !!**********************************************************************
18605 SUB Raspl1(X(*),Y(*),P(*),Q(*),Y1(*),A(*),B(*),C(*),D(*),INTEGER N)
18610   OPTION BASE 1
18615   COM /Interupt/ INTEGER Interupt
18620   REAL H,H1,H2,Hp,Hq,P2,P21,P22,Pp,Pp2,Q2,Q22,Qq,Qq1,Qq2,R1,R2,Z
18625   INTEGER J1,J2,K,N1,N2
18630   ON KEY 9 LABEL "INTERUPT",15 GOTO Interupt
18635   N1=N-1
18640   C(1)=0
18645   D(1)=0
18650   FOR K=1 TO N1
18655     J2=K+1
18660     Pp=P(K)
18665     Qq=Q(K)
18670     Pp2=Pp*(Pp+3)+3
18675     Qq2=Qq*(Qq+3)+3
18680     P22=2+Pp
18685     Q22=2+Qq
18690     A(K)=X(J2)-X(K)
18695     H=1/A(K)
18700     B(K)=1/(P22*Q22-1)
18705     H2=H*B(K)
18710     R2=H*H2*(Y(J2)-Y(K))
18715     IF K=1 THEN 18760
18720     Hq=H1*Qq1
18725     Hp=H2*Pp2
18730     Z=1/(Hq*(P21-C(J1))+Hp*Q22)
18735     C(K)=Z*Hp
18740     H=R1*Qq1*(1+P21)+R2*Pp2*(1+Q22)
18745     IF K=2 THEN H=H-Hq*Y1(1)
18750     IF K=N1 THEN H=H-Hp*Y1(N)
18755     D(K)=Z*(H-Hq*D(J1))
18760     J1=K
18765     P21=P22
18770     Qq1=Qq2
18775     H1=H2
18780     R1=R2
18785   NEXT K
18790   Y1(N1)=D(N1)
18795   IF N1<=2 THEN 18825
18800   N2=N1-1
18805   FOR J1=2 TO N2
18810     K=N-J1
18815     Y1(K)=D(K)-C(K)*Y1(K+1)
18820   NEXT J1
18825   FOR K=1 TO N1
18830     J2=K+1
18835     H=B(K)*(Y(J2)-Y(K))
18840     Z=B(K)*A(K)
18845     P2=2+P(K)
18850     Q2=2+Q(K)
18855     C(K)=(1+Q2)*H-Z*(Y1(J2)+Q2*Y1(K))
18860     D(K)=-(1+P2)*H+Z*(P2*Y1(J2)+Y1(K))
18865     A(K)=Y(K)-C(K)
18870     B(K)=Y(J2)-D(K)
18875   NEXT K
18880   SUBEXIT
18885 Interupt:Interupt=1
18890 SUBEND
18895 !!**********************************************************************
18900 !!                                                                      *
18905 !!      INTERPOLATING RATIONAL SPLINE FUNCTION                          *
18910 !!                                                                      *
18915 !!**********************************************************************
18920 DEF FNRasplint(A,B,C,D,P,Q,X,Upper)
18925   REAL Splint,T,U
18930   T=X/Upper
18935   U=1-T
18940   Splint=A*U+B*T+C*U^3/(P*T+1)+D*T^3/(Q*U+1)
18945   RETURN Splint
18950 FNEND
18955 !!**********************************************************************
18960 !!                                                                      *
18965 !!      RATIONAL SPLINE INTEGRATION                                     *
```

```
18970 !!                                                                      *
18975 !!*****************************************************************
18980 DEF FNArea(A,B,C,D,P,Q,Upper)
18985   REAL A1,A2,A3,Area,Z
18990   Z=Q/(Q+1)
18995   A1=0
19000   A2=0
19005   A3=0
19010   A1=(A+B)/2
19015   IF C<>0 THEN A2=(C/P^3*((-11*P*P-15*P-6)/6+(P+1)^3*LOG(P+1)/P))
19020   IF D<>0 THEN A3=-(D*(Z^3/3+Z*Z/2+Z+LOG(1-Z))/((Q+1)*Z^4))
19025   Area=(A1+A2+A3)*Upper
19030   RETURN Area
19035 FNEND
19040 !!*****************************************************************
19045 !!                                                                      *
19050 !!     Datetime : Routine displays & changes current time & date        *
19055 !!                                                                      *
19060 !!*****************************************************************
19065 SUB Datetime
19070   INTEGER Flag
19075   DIM A$[80],Zdate$[80],Ztime$[80]
19080   ON ERROR GOTO 19085
19085   BEEP
19090   PRINT CHR$(12);TABXY(8,1);">>>>   Date and Time  verification/change rout
ine   <<<<"
19095   Flag=1
19100   A$="enter"
19105   IF DATE(DATE$(TIMEDATE))<DATE("2 AUG 1985") THEN
19110     GOSUB Datechange
19115     GOSUB Timechange
19120   END IF
19125   A$="change"
19130   Flag=0
19135   OFF KEY
19140   ON KEY 5 GOTO 19180
19145   ON KEY 6 GOTO 19180
19150   ON KEY 7 GOTO 19180
19155   ON KEY 8 GOTO 19180
19160   ON KEY 9 GOTO 19180
19165   ON KEY 0 LABEL "CHANGE DATE" GOTO Datechange
19170   ON KEY 1 LABEL "CHANGE TIME" GOTO Timechange
19175   ON KEY 2 LABEL "CONTINUE" GOTO 19290
19180   PRINT CHR$(12);TABXY(8,1);">>>>   Date and Time  verification/change rout
ine   <<<<"
19185   PRINT TABXY(15,6);"Here is the current TIME & DATE in the 9836."
19190   PRINT TABXY(10,12);">>>  Select the appropriate special function key.  <<
<"
19195   PRINT TABXY(25,8);"1-    ";DATE$(TIMEDATE)
19200   PRINT TABXY(25,9);"2-    ";TIME$(TIMEDATE)
19205   GOTO 19195
19210 Datechange:!
19215   PRINT TABXY(1,15);"The syntax to ";A$;" the DATE is:          "
19220   PRINT TABXY(1,16);"Day: two digits,  Month:  first three letters,   Year:
  four digits    "
19225   PRINT TABXY(10,17);"For example:    31 Oct 1962        "
19230   INPUT "Now enter the appropriate date",Zdate$
19235   SET TIMEDATE DATE(Zdate$)+TIME(TIME$(TIMEDATE))
19240   IF Flag=1 THEN RETURN
19245   GOTO 19180
19250 Timechange:!
19255   PRINT TABXY(1,15);"The syntax to ";A$;" the TIME is:          "
19260   PRINT TABXY(1,16);"Hour,Minutes,Seconds:  each two digits separated by co
lons            "
19265   PRINT TABXY(10,17);"For example:    15:38:19          "
19270   INPUT "Now enter the appropriate time",Ztime$
19275   SET TIMEDATE TIME(Ztime$)+DATE(DATE$(TIMEDATE))
19280   IF Flag=1 THEN RETURN
19285   GOTO 19180
19290   OFF KEY
19295 SUBEND
19300 !!*****************************************************************
19305 !!                                                                      *
19310 !!    Automatic :   Selection of data files for automatic plotting      *
19315 !!                                                                      *
19320 !!*****************************************************************
19325 SUB Automatic(Name$(*),INTEGER Total,Line)
19330   DIM Aflag$(2)[3],Ans$[160]
19335   INTEGER I
19340   Total=0
```

```
19345   Aflag$(1)="YES"
19350   Aflag$(2)="NO"
19355 Auto_menu:!
19360   ON ERROR GOTO Err
19365   PRINT CHR$(12);TABXY(30,1);"AUTOMATIC MODE MENU"
19370   PRINT TABXY(1,4);"1- CONTOUR PLOT?                          ";Aflag$(1)
19375   PRINT TABXY(1,5);"2- THREE DIMENSIONAL PLOT?                ";Aflag$(2)
19380   PRINT TABXY(1,7);"3- ENTER / ADD THE NAMES OF THE FILES TO BE ACCESSED"
19385   PRINT TABXY(1,8);"4- REVIEW / EDIT THE ENTERED FILE NAMES"
19390   PRINT TABXY(1,10);"5- COMMENCE PLOTTING"
19395   PRINT TABXY(1,11);"6- RETURN TO THE MAIN KEY MENU"
19400   Line=-1
19405   INPUT "ENTER THE APPROPRIATE LINE NUMBER",Line
19410   IF Line>6 OR Line<1 THEN
19415 Err:!
19420     PRINT TABXY(10,10);"PLEASE TRY AGAIN"
19425     GOTO Auto_menu
19430   END IF
19435   OFF ERROR
19440   IF Line=1 OR Line=2 THEN
19445     IF Aflag$(Line)="YES" THEN
19450       Aflag$(Line)="NO"
19455       Aflag$(((Line) MOD 2)+1)="YES"
19460     ELSE
19465       Aflag$(Line)="YES"
19470       Aflag$(((Line) MOD 2)+1)="NO"
19475     END IF
19480   END IF
19485   IF Line=3 THEN GOSUB Enter
19490   IF Line=4 THEN GOSUB Review
19495   IF Line=5 THEN SUBEXIT
19500   IF Line=6 THEN SUBEXIT
19505   GOTO Auto_menu
19510 Enter:!
19515   PRINT CHR$(12)
19520   Col=1
19525   Row=1
19530   FOR I=1 TO Total
19535     GOSUB Printing
19540   NEXT I
19545   PRINT TABXY(1,16);"To stop entering file names, simply enter the word:
QUIT"
19550   I=1+Total
19555 Again:!
19560   Ans$=""
19565   IF I=1 THEN
19570     INPUT "Enter the first file name",Ans$
19575   ELSE
19580     INPUT "Enter the next file name",Ans$
19585   END IF
19590   IF Ans$="" THEN Again
19595   IF UPC$(Ans$)="QUIT" THEN
19600     Total=I-1
19605     RETURN
19610   END IF
19615   IF LEN(Ans$)>10 THEN
19620     PRINT TABXY(1,17);"Invalid file name, too many characters - try again"
19625     BEEP
19630     GOTO Again
19635   ELSE
19640     PRINT TABXY(1,17);"                                                    "
19645   END IF
19650   Name$(I)=Ans$
19655   GOSUB Printing
19660   IF I=75 THEN
19665     PRINT TABXY(1,18);"Reached the maximum of 75 data files -  press CONTIN
UE"
19670     BEEP
19675     PAUSE
19680     RETURN
19685   END IF
19690   I=I+1
19695   GOTO Again
19700 Review:!
19705   PRINT CHR$(12)
19710   Col=1
19715   Row=1
19720   FOR I=1 TO Total
19725     GOSUB Printing
19730   NEXT I
```

```
19735 Change:!
19740   I=-1
19745   INPUT "Enter the number of the file name to be changed  (enter '0' to ret
urn to menu)",I
19750   IF I=0 THEN RETURN
19755   IF I<0 OR I>Total THEN Change
19760   Ans$=""
19765   INPUT "Now enter the desired file name",Ans$
19770   IF LEN(Ans$)>10 THEN
19775     PRINT TABXY(1,17);"Invalid file name, too many characters - try again"
19780     BEEP
19785     GOTO Change
19790   ELSE
19795     PRINT TABXY(1,17);"                                                    "
19800   END IF
19805   IF Ans$="" THEN
19810     FOR J=I TO Total-1
19815       Name$(J)=Name$(J+1)
19820     NEXT J
19825     Total=Total-1
19830     GOTO Review
19835   ELSE
19840     Name$(I)=Ans$
19845     Col=((I-1) MOD 5)*16+1
19850     Row=INT((I-1)/5)+1
19855     GOSUB Printing
19860     GOTO Change
19865   END IF
19870 Printing:!
19875   IF Col>65 THEN
19880     Col=1
19885     Row=Row+1
19890   END IF
19895   PRINT TABXY(Col,Row);
19900   PRINT USING "DD,""-"",1X,AAAAAAAAAA,2X";I,Name$(I)
19905   Col=Col+16
19910   RETURN
19915 SUBEND
19920 !!*************************************************************************
19925 !!                                                                        *
19930 !!     Arcplot : Routine which draws circular frames for polar plots      *
19935 !!                                                                        *
19940 !!*************************************************************************
19945 SUB Arcplot(R,Ang1,Ang2,Step)
19950   OPTION BASE 1
19955   REAL Ang
19960   MOVE R*COS(Ang1),R*SIN(Ang1)
19965   FOR Ang=Ang1 TO Ang2+Step/2 STEP Step
19970     IF Ang*SGN(Step)>Ang2*SGN(Step) THEN
19975       DRAW R*COS(Ang2),R*SIN(Ang2)
19980       SUBEXIT
19985     ELSE
19990       DRAW R*COS(Ang),R*SIN(Ang)
19995     END IF
20000   NEXT Ang
20005 SUBEND
20010 !!*************************************************************************
20015 !!                                                                        *
20020 !!    Data_set : Routine which writes a list of the data to the printer   *
20025 !!                                                                        *
20030 !!*************************************************************************
20035 SUB Data_set(A(*),File$)
20040   OPTION BASE 1
20045   COM /Gen/ Lx(*),Ly(*),Form$,INTEGER Imax,Jmax
20050   INTEGER I,J,K,I1,I2,Numlines
20055   Numlines=10
20060   ON TIMEOUT 7,.5 GOSUB Timeout
20065   PRINTER IS 701
20070   GOTO 20095
20075 Timeout:!
20080   CALL Printerset
20085   ON TIMEOUT 7,.5 GOSUB Timeout
20090   RETURN
20095   ON ERROR GOSUB 20285
20100   PRINT USING "@"
20105   PRINT DATE$(TIMEDATE);"    ";TIME$(TIMEDATE)
20110   PRINT ""
20115   PRINT ""
20120   PRINT ">>>>    HERE IS THE DATA FROM FILE: ";File$;"    <<<<"
20125   I2=0
```

```
20130   I1=1+I2
20135   I2=I1+Numlines-1
20140   IF I2>=Imax THEN
20145     I2=Imax
20150   ELSE
20155     IF I2>Imax-Numlines THEN
20160       I2=INT((Imax+I1)/2)
20165     END IF
20170   END IF
20175   PRINT ""
20180   PRINT ""
20185   PRINT ""
20190   PRINT USING "16(X),#"
20195   ON ERROR GOTO 20215
20200   FOR I=I1 TO I2
20205     PRINT USING "DDDD.DDDD,2(X),#";Lx(I)
20210     GOTO 20220
20215     PRINT USING "K,2(X),#";"*********"
20220   NEXT I
20225   PRINT ""
20230   PRINT ""
20235   FOR J=1 TO Jmax
20240     ON ERROR GOTO 20255
20245     PRINT USING "DDDD.DDDD,7(X),#";Ly(Jmax-J+1)
20250     GOTO 20260
20255     PRINT USING "K,7(x),#";"*********"
20260     ON ERROR GOTO 20285
20265     FOR I=I1 TO I2
20270       K=(I-1)*Jmax+J
20275       PRINT USING "DDDD.DDDD,2(X),#";A(K)
20280       GOTO 20290
20285       PRINT USING "K,2(X),#";"*********"
20290     NEXT I
20295     PRINT ""
20300   NEXT J
20305   IF I2<Imax THEN 20130
20310   PRINTER IS 1
20315   OFF TIMEOUT
20320 SUBEND
20325 !!*************************************************************************
20330 !!                                                                        *
20335 !!    Initialize:   Routine which resets the variables to their           *
20340 !!                  original from the temporary values required to        *
20345 !!                  perform an accurate plot                              *
20350 !!                                                                        *
20355 !!*************************************************************************
20360 SUB Initialize(Flag$(*))
20365   OPTION BASE 1
20370   COM /Gen/ Lx(*),Ly(*),Form$,INTEGER Imax,Jmax
20375   COM /Change/ Factor,Distort,Gamma
20380   INTEGER I,J
20385   IF Factor<>1 THEN          !\
20390     FOR J=1 TO Jmax          ! \
20395       Ly(J)=Ly(J)/Factor     !  \RETURN Y COORDINATES TO THEIR ORIGINAL SPECS
20400     NEXT J                   !  /
20405     Factor=1                 ! /
20410   END IF                     !/
20415   IF Form$="POLARSEC" THEN
20420     FOR I=1 TO Imax          !\
20425       Lx(I)=Lx(I)-Gamma      ! \ BRING BACK THE POLAR SECTOR
20430     NEXT I                   ! / PLOT TO ITS ORIGINAL ANGLES
20435     Gamma=0                  !/
20440   END IF
20445   IF Flag$(9)="NO" THEN                              !\
20450     FOR J=Jmax-1 TO 1 STEP -1                        ! \
20455       Ly(J)=Ly(Jmax)-(Ly(Jmax)-Ly(J))*Distort        !  \RETURN COORDINATES TO
20460     NEXT J                                           !  /THEIR NON-DISTORTED
20465     Distort=1                                        ! /       SPECS
20470   END IF                                             !/
20475 SUBEND
20480 !!*************************************************************************
20485 !!                                                                        *
20490 !!    Printerset:   Routine informing the user printer is not on-line.    *
20495 !!                                                                        *
20500 !!*************************************************************************
20505 SUB Printerset
20510   PRINTER IS 1
20515   PRINT CHR$(129)
20520   PRINT TABXY(10,17);"Printer has not been powered up or is not on line."
20525   PRINT TABXY(10,18);"Press CONTINUE when the printer is ready.          "
```

```
20530    PRINT CHR$(128)
20535    PAUSE
20540    BEEP
20545    PRINT TABXY(10,17);RPT$(" ",70)
20550    PRINT TABXY(10,18);RPT$(" ",70)
20555    PRINTER IS 701
20560 SUBEND
```

REPORT DOCUMENTATION PAGE/PAGE DE DOCUMENTATION DE RAPPORT

REPORT/RAPPORT LM–ENG–003 1a	REPORT/RAPPORT 1b
SECURITY CLASSIFICATION/ CLASSIFICATION DE SÉCURITÉ UNCLASSIFIED/NON CLASSIFIÉ 2	DISTRIBUTION/DIFFUSION ☒ Controlled/Contrôlé ☐ Unlimited/Illimité 3

TITLE/SUBTITLE/TITRE/SOUS-TITRE

4 Reference Guide for "CONTOURS": A Contour Plotting Program for the Hewlett Packard HP 9836.

AUTHOR(S)/AUTEUR(S)

B. McCrea

5

SERIES/SÉRIE

6

CORPORATE AUTHOR/PERFORMING AGENCY/AUTEUR D'ENTREPRISE/AGENCE D'EXÉCUTION

* B. McCrea

* Under contract to NRC

7

SPONSORING AGENCY/AGENCE DE SUBVENTION

National Research Council of Canada

8

DATE	FILE/DOSSIER	LAB. ORDER COMMANDE DU LAB.	PAGES	DIAGS	REFS
17 Jun 86 9	L–3641–3 10	11 22107–C	12a	12b	12c

NOTES

13 The computer coding of this program is also available on HP flexible mini disc.

DESCRIPTORS(KEY WORDS)/MOTS-CLÉS

Contour(s), Plotting, Machine, Computer, Lines, Curve(s) Fitting.

14

SUMMARY/SOMMAIRE

A program and user guide are presented for machine plotting report–quality cartesion and polar contour maps and three–dimensional views of data from input data sets 15(arrays) of a minimum of 3x3 to a maximum of 2000 for use with the HP 9836 in BASIC coding.

ADDRESS/ADRESSE

Engine Laboratory
Division of Mechanical Engineering
National Research Council of Canada
Montreal Road
16 Ottawa, Canada K1A 0R6