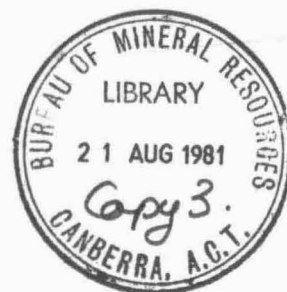


1981/27
43

085729⁺

BMR PUBLICATIONS COMPACTUS
(LENDING SECTION)



BUREAU OF MINERAL RESOURCES, GEOLOGY AND GEOPHYSICS

RECORD

1981/27

A GUIDE TO DIGITISING
STRIPCHART PROFILES
AND
SEISMIC HORIZONS

by

L.A. TILBURY & R. WHITWORTH

1981/27

A GUIDE TO DIGITISING
STRIPCHART PROFILES
AND
SEISMIC HORIZONS

by

L.A. TILBURY & R. WHITWORTH

CHAPTERS

1. INTRODUCTION
2. AN OUTLINE OF THE PROCEDURES INVOLVED
3. OPTIONAL FLAGS FOR DIGITISING
4. INSTRUCTIONS FOR DIGITISING SEISMIC SECTIONS
5. INSTRUCTIONS FOR DIGITISING MAGNETOGRAMS
6. INSTRUCTIONS FOR DIGITISING STRIPCHARTS
7. Program DIGDATA
8. Program SAMPLE
9. Program SEGPACK
10. Program MERGPOS
11. APPLICATIONS
12. REFERENCES

APPENDICES

1. Digitiser questions and answers
2. Sample digitiser output
3. DIGDATA sample output printout
4. SAMPLE output printout
5. The use and editing of card image files
6. GRADICON - HP2100 Digitising System
7. Source program listings DIGDATA, SAMPLE, SEGPACK, MERGPOS (microfiche)

TABLES

1. Optional seismic flags

FIGURES

1. Generalised flowchart for digitising seismic sections, Part 1: at BMR
 2. Generalised flowchart for digitising seismic sections, Part 2: at CSIRO
 3. Explanation of seismic flags
 4. Seismic Section: Original, Overlay Plot and Final Seismic Profile.
 5. Magnetogram; Original, Overlay Plot and Final Diurnal
 6. Stripchart: Original, Overlay plot and Final Magnetics
 7. Parameters solved for in DIGDATA
 8. Example of posting for seismic horizons (depth)
- CENOMANIAN horizon on Ceduna Terrace.
 9. Example of posting for isopachs - UPPER CRETACEOUS isopach on Ceduna Terrace.
- 2

10. Example of posting of faults - CENOMANIAN horizon on Ceduna Terrace.
11. Stacked seismic profiles.
12. Gradicon: Front Panel Controls.

1. INTRODUCTION

In the production of structural contour and isopach maps from seismic records, considerable effort is needed to convert two-way reflection times into depths. In particular, if the velocity function requires modification, the process must be repeated bit by bit over the map area. Further work is required if the scale and projection of the maps have to be changed.

As the amount of data involved in interpretation has increased, particularly in marine work, the necessity of using automated techniques to reduce the time spent on data manipulation has become more pressing. To do this, shot-point location and seismic data must be made available in digital form as a principal facts file containing station number, latitude, longitude and two-way times of interpreted seismic horizons. This report shows how this principal facts file can be created, using equipment available in BMR.

An earlier report by the authors (Tilbury and Whitworth, 1976) outlined a procedure for creating from any seismic shot-point location map, a file of digital shot-point positions in the standard format used for BMR marine surveys. This provides the station number, latitude and longitude portion of the principal facts file.

To obtain the digitised seismic two-way times, an earlier procedure for digitising seismic sections was extended and the associated program was expanded to provide further facilities needed in more detailed interpretation. New programs were developed for the resampling and formatting of the data, and for the automatic posting and contouring of horizon and isopach values in both time and depth.

The above conversion programs will process any digitised profiles and chapters have been included on their use in digitising of stripcharts and magnetograms. However, these are incidental to the main report, which is primarily concerned with the digitisation of seismic horizons, and a step by step description of the programs necessary to convert these digitised data into a seismic principal-facts file.

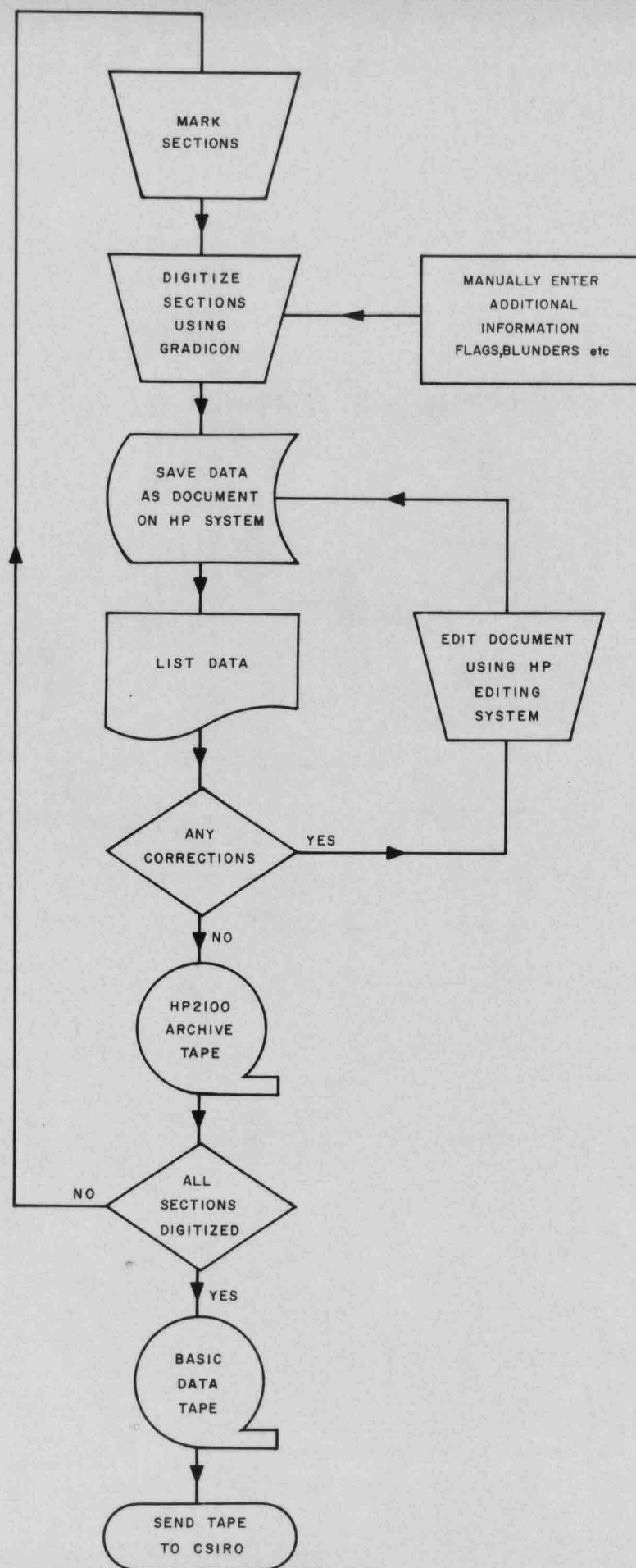


Fig.1 GENERALISED FLOWCHART FOR DIGITISING SEISMIC SECTIONS
PART1: PRODUCTION OF BASIC DIGITISED DATA TAPE (AT BMR)

2. AN OUTLINE OF THE PROCEDURES INVOLVED.

The work involved in the production of a principal-facts file containing station number, location, and seismic two-way times can be conveniently divided into two parts. The first is the digitising of the interpreted seismic records and shot-point location maps on a digitising table, and the second is the running of a series of computer programs to convert the output from the table into the principal facts required by the user.

While the techniques can be applied using any suitable devices, they will be presented here to suit the equipment available to the Marine Geophysics Group of BMR: namely a GRADICON-HP2100 digitising system at BMR (Appendix 7) and a Control Data Corporation CYBER 76 computer at CSIRO.

As already mentioned, the digitising of shot-point location maps is described in an earlier report in this series (Tilbury and Whitworth, 1976), and will not be repeated here.

Procedure at BMR - Production of basic digitised data tape

Before commencement of digitising, several routine steps should be completed and the digitising requirements should be clearly laid out so that the task on the digitising table is as simple and mechanical as possible, to minimise mistakes.

- 1) The seismic sections must be fully interpreted, with all horizons clearly marked with a distinguishing colour code.
- 2) All horizons must be tied at line intersections.
- 3) If flagging of data is required, as described in Chapter 3, the flag types are annotated on the sections.
- 4) The time marks, timing lines and delays are checked for consistency.

A flowchart of the procedure for digitising seismic sections is shown in Figure 1.

The section is digitised using the procedures detailed in Chapter 4 and the data file is saved as a document in the HP system. The digitised output consists of blocks of data of four main types. The first three types are the scales and rotation data blocks used to control the conversion from digitiser coordinates to shot-point numbers and seismic two-way times, while the fourth type consists of the digitised seismic horizons. Information is input manually to identify the data type and to control the later processing.

The resulting document is then output listed for editing purposes. The data that were input manually are checked for logical order or missing information, and any obvious errors made during digitising are corrected. The person digitising should keep note of blunders by manually entering suitable messages to aid in the editing, e.g. DELETE ABOVE BLOCK, DELETE 10 LINES. ETC. In this way mistakes are more readily identified when checking through the listing.

The edited data file can then be saved on an archival tape to await further digitisation for the desired region and avoid accidental flushing of the document from the HP disc system. Archiving is carried out on request by the HP2100 computer operator. When all required sections are digitised, the archival tape is converted by the operator into a card-image magnetic tape in a format compatible with the Cyber 76. This tape is then sent to CSIRO.

Procedure at CSIRO - Conversion of digitised data into principal-facts file

The magnetic tape containing perhaps several files of basic digitised data, one per seismic line, is read by the Cyber 76 computer at CSIRO and saved within the system as a document. The principal-facts file is generated from this document via a series of FORTRAN programs.

A flowchart showing the procedure for conversion of the digitised seismic data is given in Figure 2.

Two main programs are used: DIGDATA, which converts the digitised data into shot-point number and seismic two-way time; and SAMPLE, which resamples the data into regular shot-points. Other programs

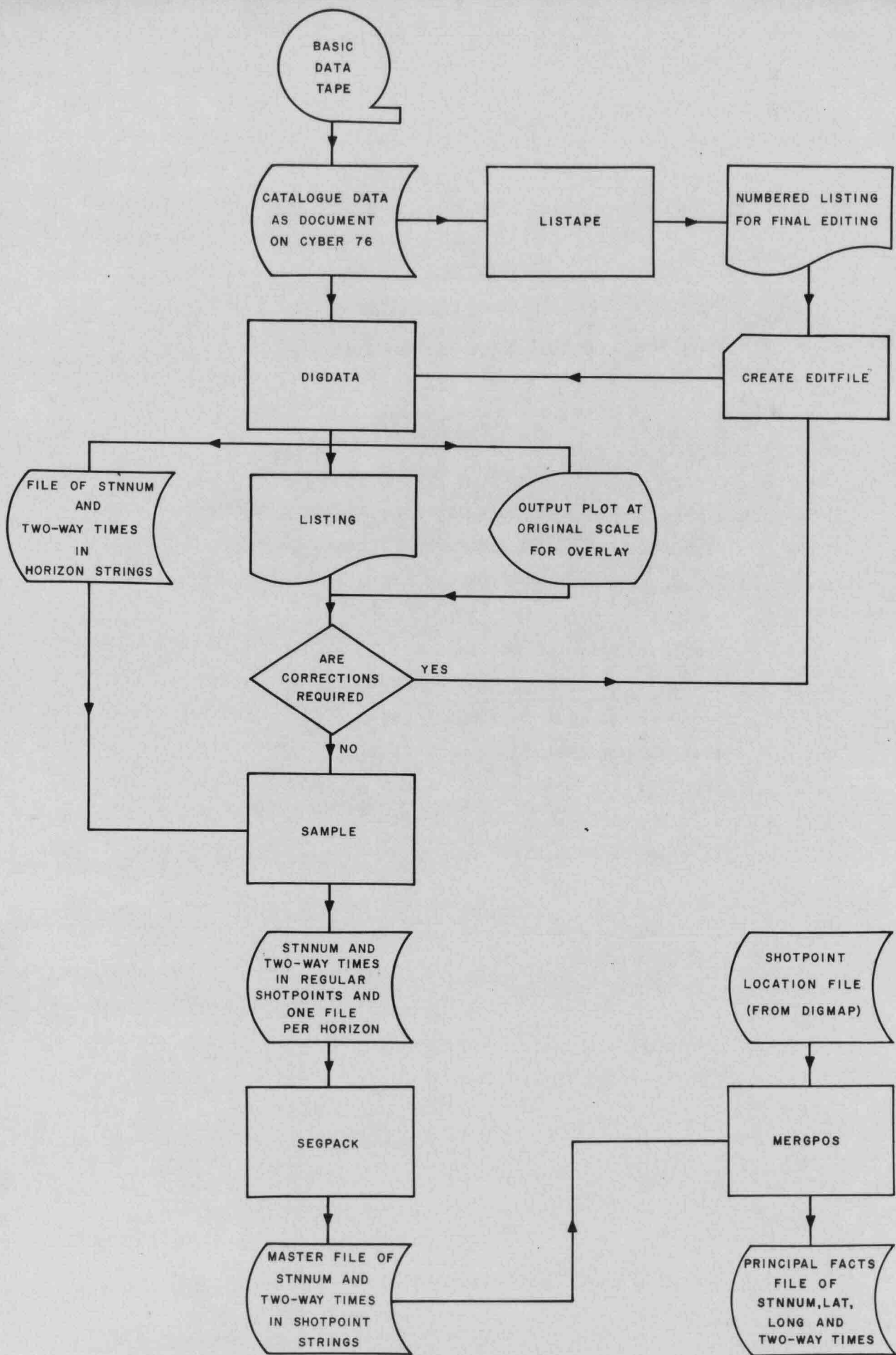


Fig.2 GENERALISED FLOWCHART FOR DIGITISING SEISMIC SECTIONS
PART 1: CONVERSION OF DIGITISED DATA INTO PRINCIPAL FACTS FILE

used are for file manipulation, merging and editing: SEGPACK formats the resampled data into a standard BMR Marine survey file; MERGPOS merges the location data file with the seismic data file; LISTAPE produces a numbered listing of every card image in a data file; NEXFILE merges editing corrections with the main file to produce an updated file, of the digitiser output in this case; see Whitworth (in prep.) for a more detailed explanation of the handling of card-image files (Appendix 6). These programs are available in the program libraries maintained by the Marine Geophysics Group on its discpack (at present, Serial Number DMR1345).

Because mistakes, however minor, are almost inevitable the first stop on the Cyber 76 is the production using program LISTAPE of a numbered listing of the card images in each file that is to be processed. As it is generally found that several passes through the processing system are needed to eliminate the mistakes, on-line editing is used; that is, corrections are made to the input file as it is read into a program but the file itself remains unaltered. This means the user needs only to produce the numbered listing at the beginning of the process and work out corrections relative to the original file.

Program DIGDATA is then run to convert the digitiser output into shot-point number and seismic two-way time. The program printout is inspected for irregularities, and the cause found by reference back to the numbered card-image listing. Major errors are usually associated with mistakes in one of the manual entries or in digitising of the scales. A plot at original scale is also output to aid in the editing stage. Edit additions and modifications are punched onto cards and added to the existing cards in the edit file. When the user is satisfied that all corrections have been made, program DIGDATA is rerun and the converted data file saved.

Program SAMPLE is then executed using the converted data file as input, and the seismic data is resampled according to parameters specified by the user. The output data has the station number and two-way times in regular shot-point numbers, and has one file per horizon. The resampled data is then processed through program SEGPACK to obtain a standard Marine Group Survey file containing station number and two-way reflection times.

The final step is carried out with program MERGPOS and involves the merging of the standard seismic data file with the location file produced from program DIGMAP (Tilbury and Whitworth, 1976). This merged file now contains the principal facts of station number, latitude, longitude and seismic two-way reflection times and is saved for future applications, mainly the production of structure and isopach maps and stacked seismic profiles.

3. OPTIONAL FLAGS FOR DIGITISING

While digitising, the user may wish to input more information than just the digitised data for each horizon. For example, the quality of reflectors, the type of geological feature, areas of interpolated data etc would be useful if annotated on the final seismic maps.

Several flags are available which can be input during digitising to indicate such information. These flags are incorporated by program DIGDATA into the data word containing the horizon time-depth, and can be extracted during later processing to post information onto the seismic maps (see Figure 8).

Flags available at present are given in Table 1. The flags have been chosen so that the first letter of the meaning is the mnemonic for the flag. Originally this led to inconsistencies because the normal quality names included FAIR instead of MODERATE and this conflicted with FAULT. For consistency the meanings were changed to arrive at the unique set of flag names listed.

TABLE 1: OPTIONAL SEISMIC FLAGS

FLAG	VALUE x 10 ⁵	MEANING		
+G	0	GOOD	) Quality	)
			)	)
+M	1	MODERATE	) of	)
			)	) Values
+P	2	POOR	) Reflector	) Posted
				) in
+F	3	FAULT		) SEISMAP
				)
+N	4	NO REFLECTION		)
				)
+E	5	ERODED		)
				)
+O	6	OUTCROP		)
				) Values
+B	7	BASEMENT		) not
				) Posted
+A	8	ABSENT		)
				)

The first three flags in the table are used to describe the quality of the reflector. The program assumes all horizons are GOOD unless flags are input. When posting data, MODERATE or POOR will follow the values if data has been flagged as moderate or poor, but if data is GOOD only the data value will be posted.

Apart from the quality flags, there are flags which describe specific features on the seismic records. Figure 3 shows these features and indicate how they are flagged while digitising. In general, the user digitises up to the fault, outcrop etc, manually inserts a flag to indicate the type of 'geological' feature, continues digitising to the end of the feature, and then resets flag to the quality of reflector (usually +G). When the data is posted, the flag meanings such as FAULT, ERODED, OUTCROP etc, will be annotated on the map.

Most options are straight forward and only result in the flagging of a particular horizon. However the ERODED and OUTCROP options result in the programs replacing the values of the digitised horizon with values from the horizon above or below it. If an horizon truncates against an horizon

above it as for example top lap, erosional truncation etc, the ERODED option is used. Similarly if the horizon truncates against a lower horizon as for example downlap, onlap etc, the OUTCROP option is used.

In the ERODED case (Figure 3(d)), horizon B is digitised continuously across the record. Horizon C is digitised up to the intersection with B and the flag +E is entered. A few points along the common horizon must be digitised. At the right intersection of the two horizons the flag +G is entered and digitising continues. In this case the program will insert values from the upper horizon into horizon C between the +E and +G flags.

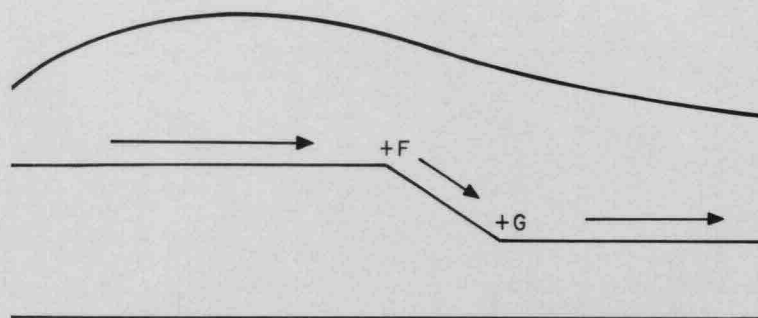
The OUTCROP case (Figure 3(c)) is similar, except in this case horizon B is discontinuous and is flagged, whereas C is the continuous horizon. The program will insert values from the lower horizon C into horizon B between the +O and +G flags.

Where two horizons are coincident, the common horizon is digitised only once, and the values are used for both horizons to ensure that isopachs would truly give zero thickness rather than small thicknesses which would arise if each horizon were digitised separately. Furthermore, when stacked profiles are plotted the coincident horizons would lead to a thicker ink line which would be aesthetically displeasing.

The NO REFLECTION flag (Figure 3(b)) is used where a time horizon is known to exist, but cannot be identified as a reflector on the seismic record. In this case a +N flag is inserted at the end of the reflector, and the interpolated horizon is digitised up to the reappearance of the reflector where another flag is inserted. The +N flag is particularly useful if interval velocities between horizons are to be used for the conversion of seismic two-way times to depth. In this case each horizon must be defined at every point, and where an horizon cannot be identified as a reflector, the user-interpolated horizon is better than any automatic interpolation over the gap. Any interpolated values used will be flagged when maps are being prepared.

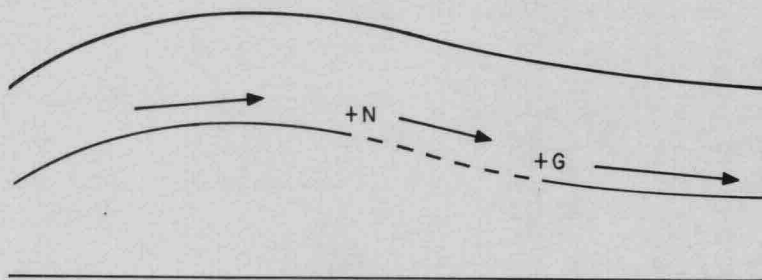
Whenever a flag is input, it remains in force, until reset by another flag. Usually the flag is reset to the quality of the reflector.

12

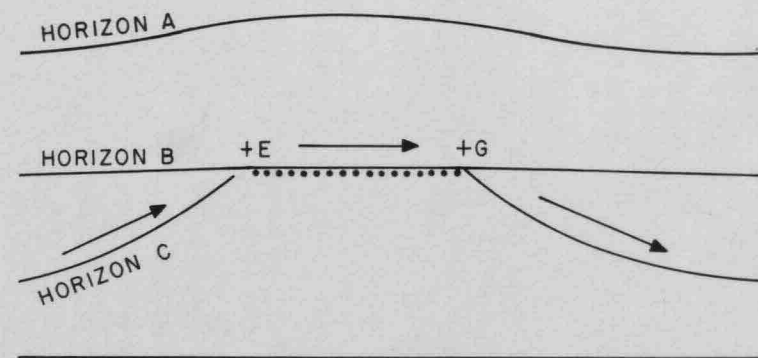


a) FAULT

----- USER INTERPOLATED HORIZON

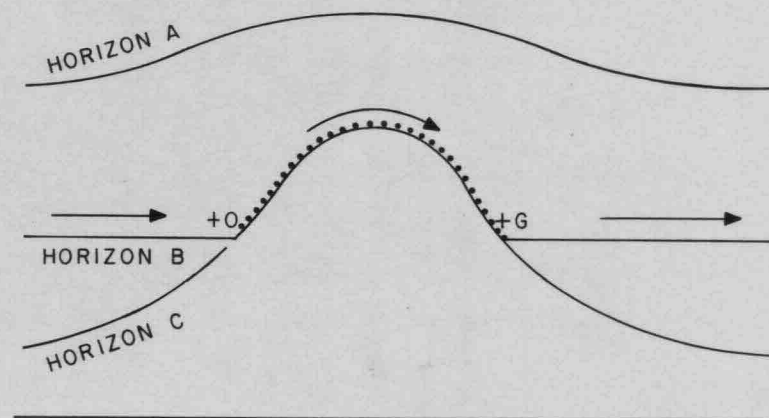


b) NO REFLECTION



d) ERODED

—— DIGITISED HORIZON
 INSERTED BY PROGRAM



c) OUTCROP

Fig.3 EXPLANATION OF SEISMIC FLAGS

Also whenever a flag is input there must be at least one data value to the right of it, as the program ascribes the flag to this position on the seismic record. In most cases this is no problem as digitisation commences immediately after the flag. However with the +E and +O options a few points must be digitised along the common horizon.

4: INSTRUCTIONS FOR DIGITISING SEISMIC SECTIONS

Interpreted seismic sections are digitised to provide the principal facts of station number versus two-way reflection times to each horizon. An example of a digitised seismic section from the Ceduna Terrace (Fraser and Tilbury, 1979) is given in Figure 4. This shows the original section, the overlay plot with annotated flags produced for editing purposes, and the final seismic profile. Note that the seismic section and overlay plot forms only about half of the final seismic profile shown.

Preparation for digitising:

Before digitising begins, all horizons in the project area should be clearly marked and annotated with the appropriate flags. Extensive horizons should be defined at all points, even in areas where they are truncated by an erosion surface or outcrop, or are not visible on the seismic record. All horizons should be tied at line intersections.

The seismic section is laid down roughly parallel to the x-axis of the digitising table, positioned so that the limits of digitisation lie within the operating area of the table, and taped down securely. Unless convenient breaks occur in the record, the limits of digitisation should be set at 10 minute marks.

The power switches for the table and console are turned on and the variable-increment switch is set to the desired output interval. The user then logs into the digitising program and answers a series of questions asked by the computer to establish the mode of operation and the format required for the digitised data. An outline of the GRADICON-HP2100 digitising system is given in Appendix 7.

Digitisation of a section is carried out in two steps.

14

Firstly its orientation, and horizontal and vertical scales are digitised using POINT mode digitisation. This information is required by the computer programs used subsequently to convert the digitised horizons into a string of station numbers versus seismic two-way times. Secondly, each horizon is digitised in turn using TIME mode (or INCREMENTAL if appropriate) on the digitiser. The rate of digitisation must be set appropriately to avoid paralysis of the hardware.

Orientation of record

The user starts by inserting two messages on the GRADICON keyboard. The first is the file name, an 80 character free-format label, by which the file will be identified and located during the processing to follow. The next card image identifies the following data as the orientation block and is:

SLOPE.

Following this the user digitises two points on the same horizontal timing line; at the first and last time mark respectively. The block is then terminated by manual entry of an End-of-File (.EOF).

Horizontal Scale

Following the orientation block is the horizontal scale data block. To reduce manual input, only the starting value and increment between regular time marks is inserted. The program automatically assigns station numbers to the digitised points. The horizontal scale is digitised from left to right, with new station numbers inserted if breaks or jumps occur in the time marks on the record. The horizontal scale must be digitised along a single timing line as the program corrects for curvature of the section on the table using the digitised scale values (Fig. 7).

The major differences between digitising seismic records with shot-point numbers, and digitising ones with survey times, is the construction of the station number. For records in shot-points it is of the form SSSS.NNNNNNN where SSSS is an assigned survey number and NNNNNNN is the shot-point number. In contrast a marine station number is of the form SS.DDDHHMM which gives the survey number, and time in yeardays, hours, and minutes for a station. To minimise confusion,

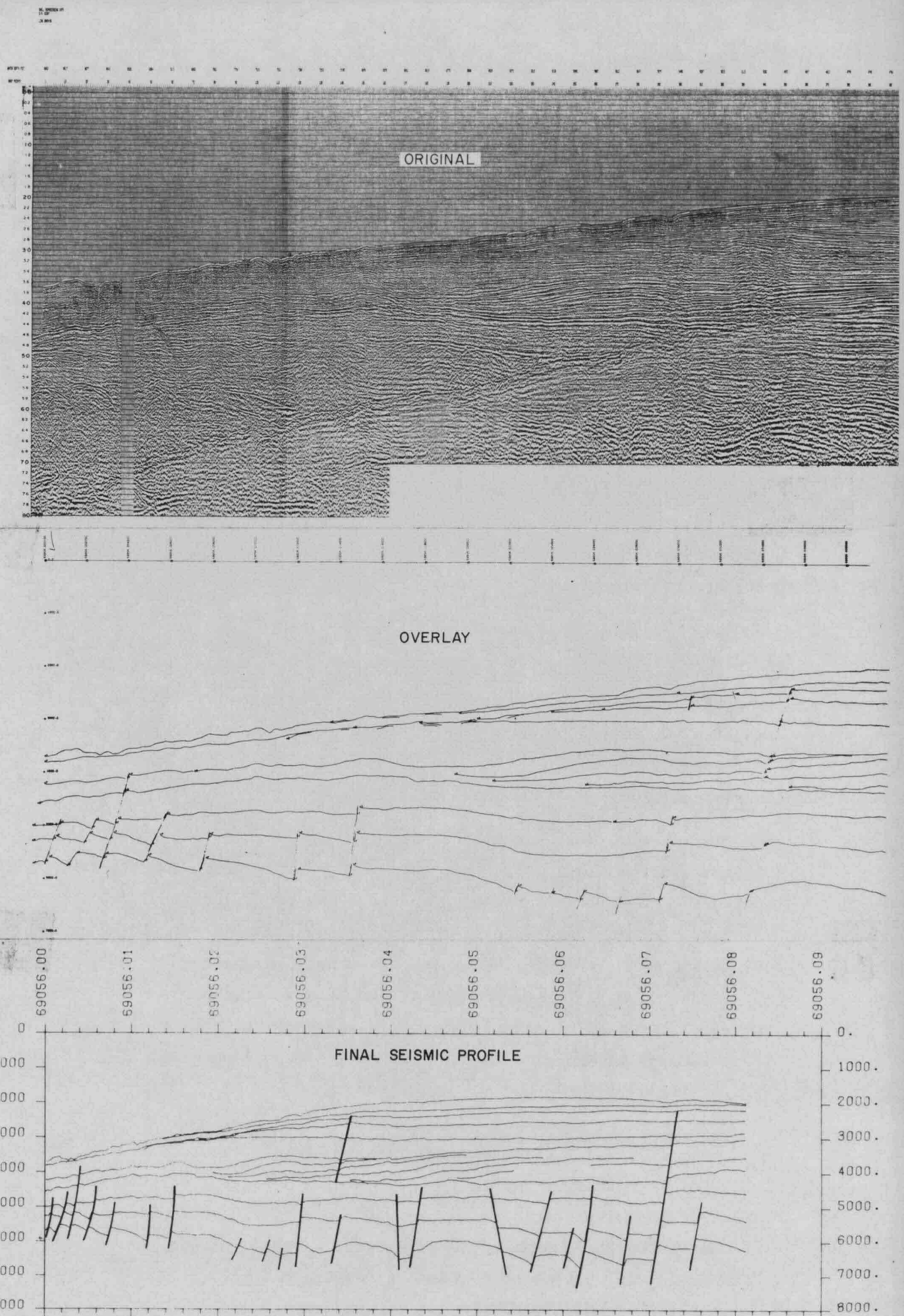


Fig.4 SEISMIC SECTION: ORIGINAL OVERLAY PLOT AND FINAL SEISMIC PROFILE

all descriptions hereafter will relate to a marine seismic section with the horizontal scale in survey time.

In digitising the scales on the record it is important to realise the significance of the term 'regular interval'. The interval between successive time-marks (or timing lines) is not simply added to the previous value, but is added to the rounded value. With survey time digitisation, an interval of 10 refers to the ten-minute marks and 60 to the hour marks, i.e. if the first point is a 5-minute mark and an interval of 10 is used, the next number automatically assigned is 10 (not 15). However, for shot-point numbers the interval is a straight addition, that is, if an interval of 100 is used, and the previous shot-point is 106, the next number automatically assigned is 206.

The horizontal data block begins with three typed messages; the first being the identifier for the block:

HORIZONTAL

The second the interval in minutes between the time marks on the record to be digitised; this can be from 1 to 7 characters, positive or negative, but must not include blanks:

Inn

and the third the station number of the starting point of digitisation which does not necessarily have to be a time mark.

TSS.DDDHHMM (remember TSSSS.NNNNNNN for sections in shot-point number)

Following these manually input messages, the user digitises the point corresponding to the above T-value and then continues digitising, from left to right along the same timing line, all the regular time marks until either the end of the section to be digitised is reached or some non-regular event occurs. A non-regular event may be:

- i) a missing time mark
- ii) a record break
- iii) a paper advance
- iv) the end of the record, if not a time mark

In this case another T-value is inserted and the corresponding point digitised. Digitisation then continues with the regular time marks. In some cases, e.g. a paper advance, a second T-value needs to be inserted before continuing with the regular time marks. The block is terminated with an End-of-File. (.EOF).

Vertical Scale.

The vertical scale must be digitised at the first (i.e. left-most) horizontal scale point. This is essential for curvature of the section to be properly corrected. If non-linear scales are digitised a T-value must be inserted to ensure correct time-depth is associated with the digitised point. To make best use of visible information, the user may wish to digitise only those timing lines that can be identified. As long as all are preceded with a T-value, the vertical scale will be adequately defined. Note that at least three points must be digitised.

The vertical scale data block is similar in form to the horizontal data block.

The first typed message is the identifier for the block:

VERTICAL

followed by the interval in milliseconds of the timing lines to be digitised. Again this can be from 1 to 7 characters, positive or negative, but must not include blanks.

Innn

and the third the true value in milliseconds of the topmost timing line that can be accurately digitised. Note that if time delays are present on the record, these must be added to the value.

Tmmmm

The user digitises the point corresponding to the above T-value and then continues digitising the timing lines at regular intervals, working from the top of the seismic section to the bottom along the same vertical time mark. If any non-regular timing lines occur, the user can insert a new T-value and the sequence can be restarted. Again the block is terminated by an End-of-File (.EOF).

Seismic Horizon Digitisation

Once the rotation and scale data blocks are digitised, the user is ready to commence digitising the interpreted seismic horizons. Each horizon must be assigned a unique label (up to 30 characters) by which the horizon will be identified in the processing to follow.

The user changes the digitisation mode to TIME (OR INCREMENTAL) by changing the switches on the GRADICON console, typing .ALTER on the GRADICON keyboard and answering the resulting questions to establish time mode digitisation. (See Appendix 7).

The first message is the horizon label, prefixed by "R" to identify a run of seismic data: i.e.

RHORIZONONE

The user then digitises the horizon from left to right ensuring the cross-hair follower is not moved too fast, otherwise points will be missed. The user can at any time, type in any of the optional flags outlined in Chapter 3, remembering that there must be at least one digitised point to the right of every flag entered.

If a time delay is encountered when digitising a horizon, the time of delay must be inserted prefixed with a 'D' i.e.

D+nnnn for upward shift of horizon

D-nnnn for downward shift

The value of the time delay, positive or negative, is added to the base value of the vertical scale to produce a new base value. The data block for the first horizon is terminated by an End-of-File (.EOF).

When digitisation of the horizon is complete, the user inserts a new R identifier for the next horizon and digitises it as before.

When all horizons are digitised, the user removes the seismic section from the table and repositions it in readiness for digitisation of the next part of the section. The user then continues digitising beginning with: SLOPE.

These four sequences, viz. SLOPE, HORIZONTAL, VERTICAL and RHORIZONn are repeated until the whole line is digitised. An extra end-of-file (.EOF) is entered to close the file for that line.

The GRADICON run is then ended by .END (Appendix 7) and the data saved as a HP document on disc.

The user than restarts the whole sequence of events for the next seismic section to be digitised.

Summary of digitising procedure:

The orientation and scale data blocks are digitised using POINT mode digitisation and is of the form;

file label	Label of file for identifying and locating
SLOPE	Identifier for rotation block
X Y X Y	Two points to determine slope of record
.EOF	Terminate block with End-of-File.

HORIZONTAL	Identifier for horizontal scale
Inn	Interval between time marks to be digitised.
TSS.DDDHHMM	Station number of start of section
X Y X Y	(TSSSS.NNNNNNN for surveys in shot-point numbers)
X Y X Y	Time marks digitised at regular intervals
TSS.DDDHHMM	New Station number where irregular event occurs
X Y X Y	Continue digitising time marks at regular intervals
.EOF	Terminate block with End-of-File.

VERTICAL	Identifier for vertical scale
Inn	Interval between timing lines to be digitised
Tmmmm	Two-way time (milliseconds) of uppermost Timing Line
X Y X Y	
X Y X Y	Timing lines digitised at regular intervals
.EOF	Terminate block with End-of-File.

Following the orientation and scale blocks is the record data block. Each horizon is digitised in turn using INCREMENTAL (or TIME) mode digitisation and the data block is of the form.

RHORIZONONE	Descriptor for first horizon
X Y X Y	Digitisation of horizon one
+M	Flag inserted for moderate (fair) quality reflector
X Y X Y	
X Y X Y	
D+1000	time delay of 1 sec (1000 msecs) inserted
X Y X Y	
X Y X Y	
+G	Reflector quality now good
X Y X Y	
X Y X Y	
.EOF	Terminate data block with End-of-File.

RHORIZONTWO	Descriptor for second horizon
X Y X Y	
X Y X Y	
etc	
.EOF	Terminate Second Horizon.

Continue digitising until all horizons are complete. The data is then saved as a document, the seismic section moved to a new part, and the procedure repeated starting with SLOPE.

5. INSTRUCTIONS FOR DIGITISING MAGNETOGRAMS

At times geophysical observatory magnetograms are used to correct shipboard magnetometer records for the daily variation in the Earth's magnetic field. As instantaneous values are required at the same time as those used for the onboard sampling, the usual observatory reduction method which supplies hourly mean values is not applicable. The digitising program (DIGDATA) has therefore been extended to handle this specialised requirement.

Magnetograms are photographic paper records which display three components of the Earth's magnetic field, usually H, D and Z (Figure 5). Each magnetogram has the horizontal axis in Universal Time and covers one day's recording. Each component has a base line whose value is constant (at a given temperature) for each Geophysical Observatory. The departure in millimetres of the H and Z components from their base lines on the magnetogram, is related (though not always linearly) to the number of nanoteslas departure from the base-line value. Once the user obtains these departures in millimetres, the components can be converted to nanoteslas using the standard reduction formulae for magnetograms. The diurnal variation in F can then be calculated directly by using H and Z. (See Figure 5).

Shrinkage points on the magnetogram which are set 100 mm apart, allow correction for distortion of the photographic paper caused by the developing process, and by changes in humidity.

Magnetograms are digitised in a similar fashion to seismic sections, except provision is made for the digitisation of the base lines for the H and Z components of the diurnal, and for the digitisation of the shrinkage points on the magnetogram (Figure 5).

Orientation of Record:

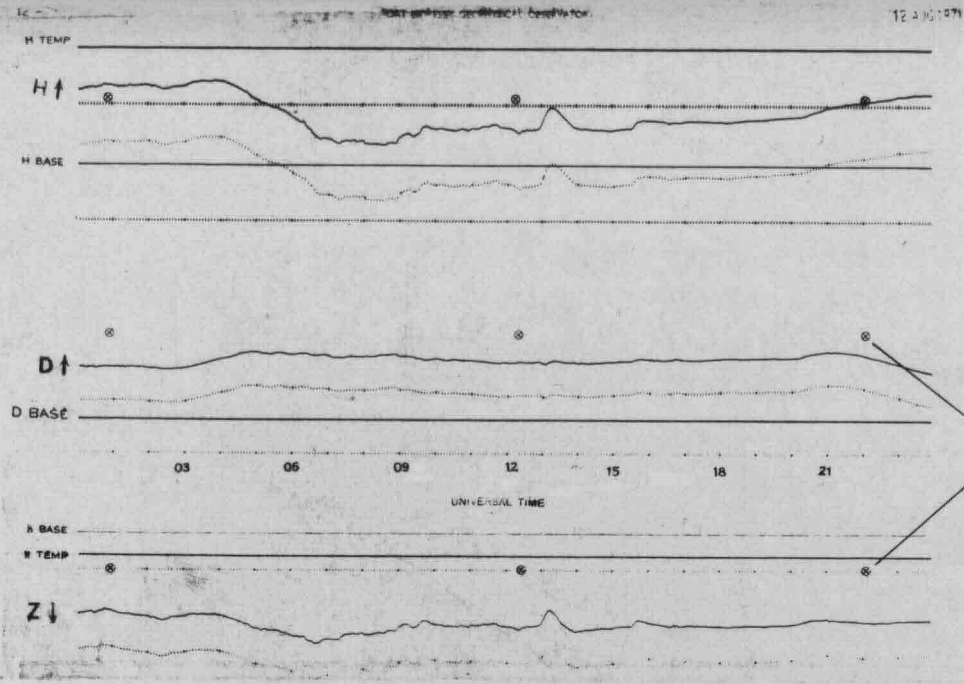
This data block is identical to that for digitising seismic sections. The two slope points are digitised on any one of the base lines on the magnetogram.

Horizontal Scale:

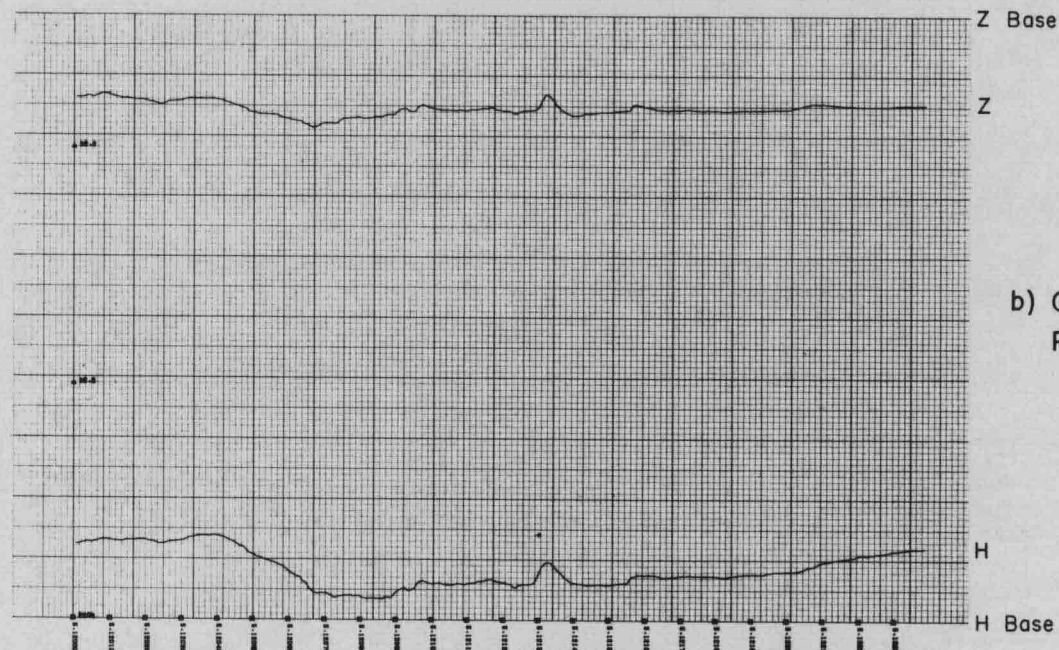
The horizontal scale is in time (0 to 24 hours) and is digitised using survey time numbers in the form SS.DDDHHMM. That is, if the diurnals are required for a BMR marine survey, the survey and day number will be meaningful, otherwise they can be assigned any values by the user. The day number is usually set to a three-digit year day number.

As the first and last hour points, ie for 00 and 24 hours, are

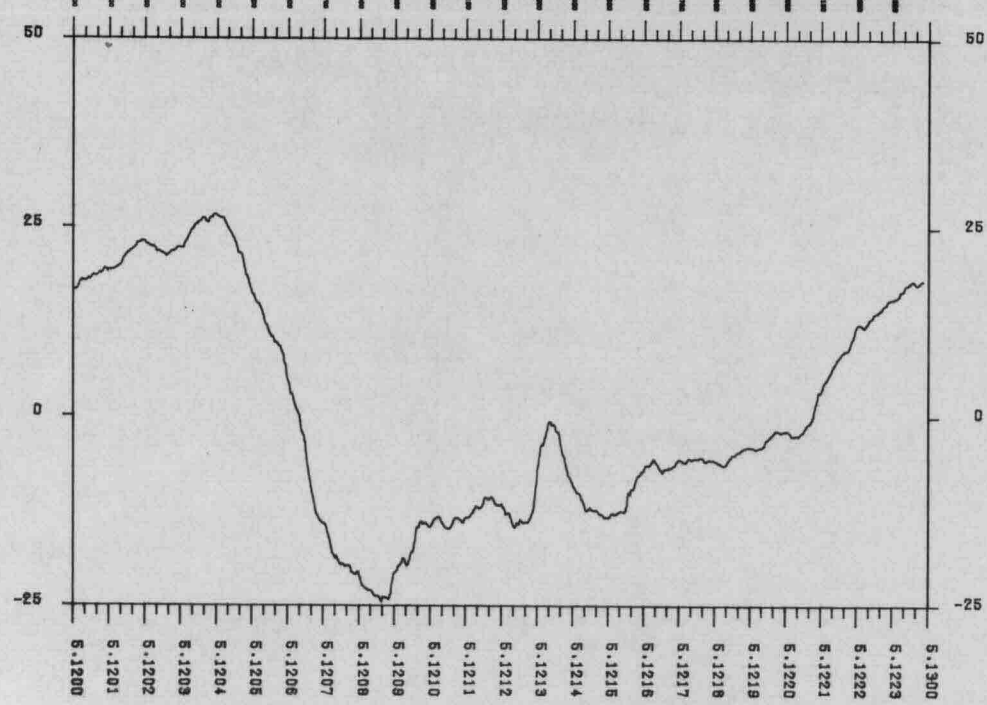
an



a) ORIGINAL
MAGNETOGRAM



b) OVERLAY
PLOT



c) FINAL
DIURNAL
(TOTAL FIELD)

Fig.5 MAGNETOGRAM: ORIGINAL, OVERLAY PLOT AND FINAL DIURNAL

usually missing or ill-defined, it is best not to digitise these points, and instead to allow the program to extrapolate the start and end positions.

Vertical Scale:

As the user requires the vertical scale in millimetres, there is no conversion of the vertical co-ordinate from the table output. Therefore the I and T values inserted in this data block are meaningless. However, they must be present to keep the program in step, and thus these values can be set to any values the user desires.

To allow for shrinkage of the magnetogram caused by the developing process, the shrinkage points are digitised and the vertical scale is corrected to true millimetre distances. The nine shrinkage points are POINT digitised from top to bottom, left to right, in the order shown in Figure 5.

Origin Data Block:

This data block can be used whenever the origin for the vertical scale changes, and it is necessary to digitise the corresponding point. For magnetogram digitisation, it is used to reset the origin on the base lines for the H or Z components of the diurnal. The first message is the identifier:

ORIGIN

followed by the new T-value for the base line. The number nn is the ordinate value of the origin. For magnetograms this value is always zero.

Tnn

The user then digitises the point at the extreme left hand side of the base line of the trace to be digitised (on the first pass this is the H component).

An End-of-File (.EOF) is inserted to terminate the data block.

Diurnal Trace Digitisation:

At this stage the user is ready to digitise the H - component of the

the diurnal field. As with seismic horizon digitising, the user changes the digitising mode to INCREMENTAL (or TIME).

The first message is the horizon label, prefixed by R to identify a run of diurnal data : i.e.

RHTRACE

The user then digitises the trace from left to right ensuring the digitising speed is adequate to determine the frequency of variations on the trace.

Following the digitisation of the H - component, the user repeats the Origin and Diurnal Digitisation Blocks for the Z-component of the diurnal.

Summary of Digitising Procedure for Magnetograms:

The orientation and scale data blocks are digitised using POINT mode digitisation and is of the form:

File label	- Label of file for identifying and locating
SLOPE	- Identifier for orientation block
X - Y - X - Y -	Two points on base line to determine slope of record
.EOF	- Terminate block with End-of-File

HORIZONTAL	- Identifier for horizontal scale
Inn	- Interval between time marks to be digitised
TSS.DDDHHMM	- Time of first visible time mark
X - Y - X - Y -	
X - Y - X - Y -	- Time marks digitised at regular intervals
X - Y - X - Y -	
.EOF	- Terminate block with End-of-File

VERTICAL	- Identifier for vertical scale
I1	Interval and T - value meaningless - set to one
T1	
X - Y - X - Y -	Digitise the nine shrinkage
X - Y - X - Y -	points on the magnetogram
.EOF	- Terminate block with End-of-File

25

ORIGIN	- Identifier for origin block
TO	- T-value of H-base (always zero)
X - Y -	Digitise base line for H component at Left Hand end
.EOF	- Terminate block with End-of-File

Following the orientation and scale data blocks is the record data block. The trace of the H - component of the diurnal is digitised using INCREMENTAL (or TIME) mode digitisation.

RDIURNAL 14/1 H	- Descriptor for H-component
X - Y - X - Y -	
X - Y - X - Y -	Digitise trace
X - Y - X - T -	
.EOF	- Terminate block

The digitising mode is then changed back to POINT digitisation, to digitise the Z-base.

ORIGIN	- Identifier for origin block
TO	- T-value of Z base (always zero)
X- Y -	Digitise base line for Z component at left hand end
.EOF	- Terminate block

Again the digitising mode is changed to INCREMENTAL(or TIME) to digitise the Z component of the diurnal

R DIURNAL 14/1 Z	- Descriptor for ^Z - component
X - Y - X - Y -	
X - Y - X - Y -	Digitise trace
X - Y - X - Y -	
.EOF	- Terminate block

The digitised data is then saved in the HP document region and listed for editing purposes. The user may then repeat all steps for the next magnetogram to be digitised.

6. INSTRUCTIONS FOR DIGITISING STRIPCHARTS

Stripchart records are digitised in almost identical fashion to seismic sections. In the example shown in Figure 6 a magnetic stripchart has been digitised. Also shown is the overlay plot produced by the conversion program DIGDATA and the final magnetic trace.

Orientation of Record

Again two points on the same horizontal lines are digitised to allow calculation of the orientation of the record on the table.

Horizontal Scale

If the horizontal scale represents time it is digitised using survey time numbers in the form SS.DDDHHMM. That is, if the strip-chart data are required for a BMR marine survey, the survey and day number will be meaningful, otherwise they can be assigned any values by the user. The day number is usually set to the yearday number.

If the horizontal scale is in decimal units, for example distance or shot-point number, it is digitised using survey numbers in the form SSSS.NNNNNNN where SSSS is an assigned survey number and NNNNNNN is the shotpoint number or unit of distance etc. No integer rounding is used for 4-digit survey numbers.

Vertical Scale

This is digitised in a similar fashion to that for seismic sections. Again the vertical scale must be digitised at the first horizontal scale point. The increment value and starting value in the specified data units are inserted, and the vertical scale is digitised.

Stripchart Trace Digitisation

This is digitised as if it were a single seismic horizon. Again the user changes the digitisation mode to INCREMENTAL (or TIME) and after inserting the run label the trace is digitised from left to right. No flagging of data is allowed.

If the trace steps due to origin shifts (See Figure 6) a delay value, equal to the step value, must be inserted.

When the trace is digitised, the stripchart is moved onto the next section and the complete digitisation sequence is repeated.

Summary of Digitising Procedure for Stripcharts

The orientation and scale data blocks are digitised using POINT mode digitisation and is of the form:

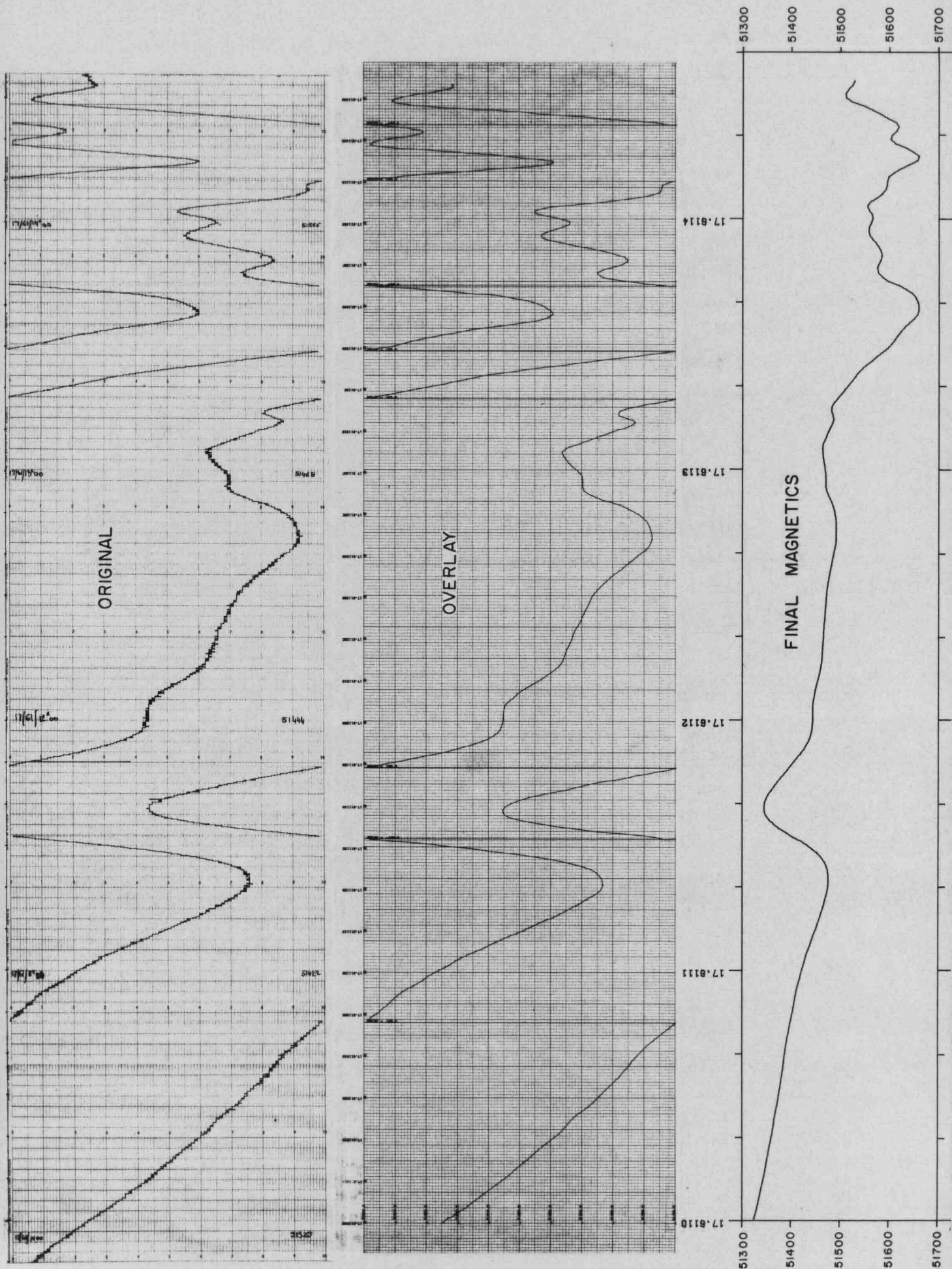


Fig.6 STRIPCHART:ORIGINAL,OVERLAY PLOT AND FINAL MAGNETICS

File label	- Label of file for identifying and locating
SLOPE	- Identifier for orientation block
X-Y-X-Y-	- Two points to determine slope of record
.EOF	- Terminate block

HORIZONTAL	- Identifier for horizontal scale
Inn	- Interval between time marks to be digitised
TSS.DDDHHMM	- Time of first visible time mark
X-Y-X-Y-	- Time marks digitised at regular intervals
X-Y-X-Y-	
.EOF	- Terminate block

VERTICAL	- Identifier for vertical scale
Inn	- Interval between data points to be digitised
Tmmmm	- Base value of chart
X-Y-X-Y-	
X-Y-X-Y-	- Scale digitised at regular intervals
X-Y-X-Y-	
.EOF	- Terminate block

Following the orientation and scale data blocks is the record data block. The stripchart trace is digitised using INCREMENTAL(or TIME) mode digitisation.

RTRACENAME	- Descriptor for trace
X-Y-X-Y-	
X-Y-X-Y-	- Digitise trace
X-Y-X-Y-	
D+ww	- Change base value by step in trace (usually a chart width)
X-Y-X-Y-	
X-Y-X-Y-	- Continue digitising
.EOF	- Terminate block

The digitised data is then saved in the HP document region and listed for editing purposes. The user may then repeat all steps for the next portion of the stripchart to be digitised.

7: Program DIGDATA

The primary aim of this program is to convert the GRADICON output, as digitised using the procedures outlined in Chapter 4, into station number versus seismic two-way time for each horizon. The same program can be used to process stripcharts or magnetograms. A listing of the source program and a sample output printout are given in Appendices 7 and 3.

Before the digitiser co-ordinates can be converted to the required values, four parameters must be determined. These are the orientation of the record on the table; the horizontal and vertical scales; and the curvature of the section on the digitiser (Fig. 7).

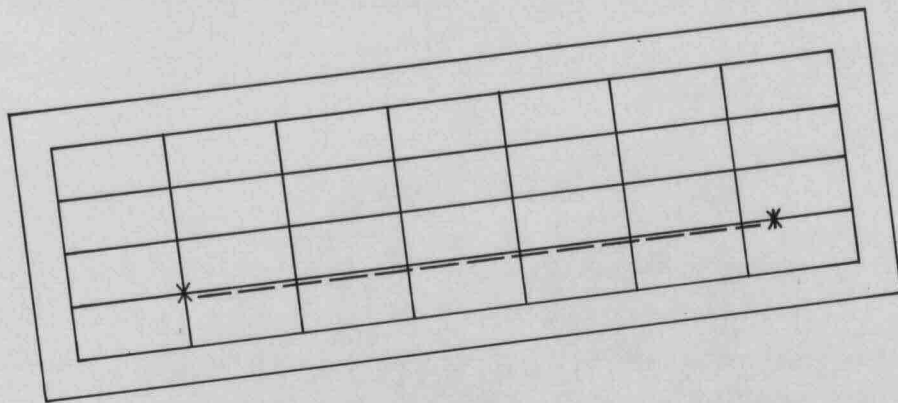
The calculation of the orientation of the section on the table bypasses the tedious and time consuming task of accurately positioning the section horizontally on the table.

For both the horizontal and vertical scales, the program uses integer rounding of the inserted T-values to obtain a regular time value. Consecutive time values are assigned automatically by the addition of the supplied time interval to this rounded value. Although the horizontal and vertical scales are treated in much the same way there is a significant difference in the way the program interpolates intermediate values.

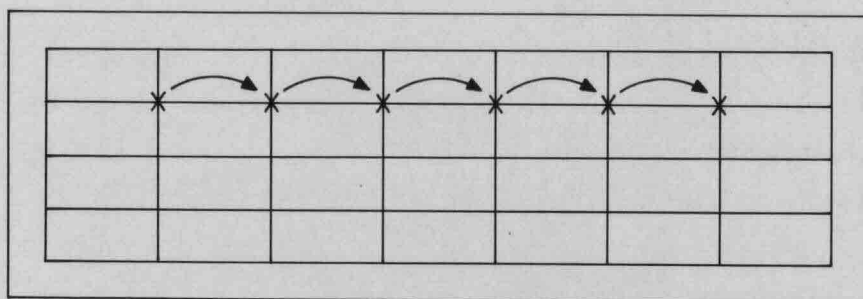
The horizontal scale is considered to be irregular, for example paper advances, breaks in shooting, or changes in shot interval, may cause a variation in the horizontal scale of the seismic record. Owing to this possible irregularity, horizontal positions are linearly interpolated between the digitised time marks. Note that there is no rounding of the T-values for shot-point data.

The vertical scale however is assumed to be regular and as such a better estimate of the vertical scale can be found by placing a least-squares straight line through the digitised points. This least squares technique minimises the digitising errors, and the residual errors are printed for assessment of accuracy.

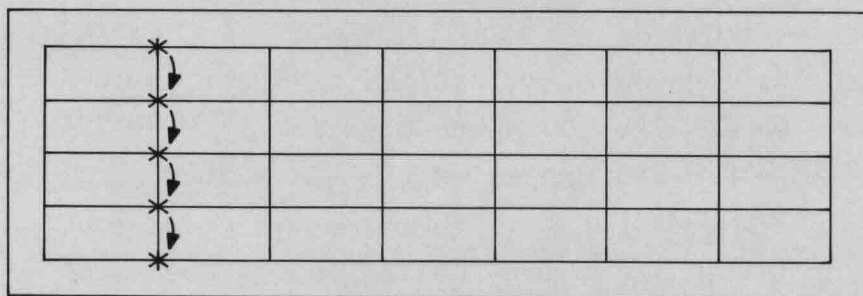
30



a) SLOPE

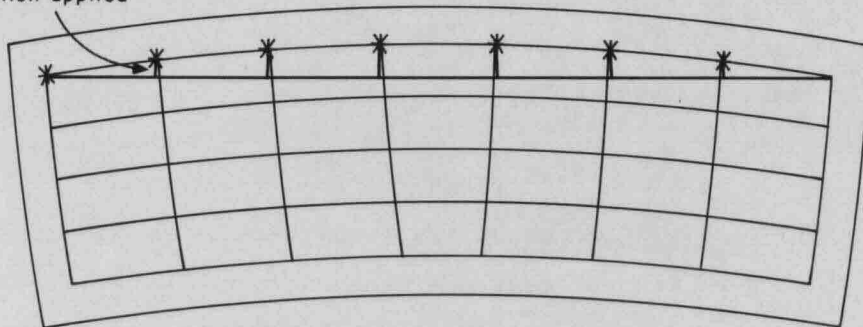


b) HORIZONTAL SCALE



c) VERTICAL SCALE

ΔY correction applied



d) CURVATURE OF SECTION

(Computed from slope and horizontal scale digitization)

Fig.7 PARAMETERS SOLVED FOR DIGDATA

The vertical scale origin can be shifted either positively or negatively by the insertion of a delay value. Time delays, used to display the appropriate part of the section on the recorder, are thus easily allowed for by changing the vertical origin appropriately.

The curvature correction minimises the effects of folds in the record, joins in taped Xerox sections and the like. The vertical corrections are determined from the differences between the digitised horizontal T-values and the values computed using the slope between the first and last points (Fig. 7). The curvature correction at each point is calculated by piece-wise linear interpolation between these values.

The seismic flags input during digitising, and outlined in Chapter 3, are decoded and the corresponding flag value (Table 1) is incorporated into the data word containing the horizon time-depth value. The flag value can be easily extracted from the data word later in the processing and posted on maps or profiles.

An overlay plot can be produced to aid in the editing of the seismic data. This plot is a true scale plot produced using the actual digitised co-ordinates from the table. Later in the processing, other programs can be used to produce plots at reduced scales. The output plot is fully annotated with horizontal and vertical scales, all delays are indicated as such and the delay value is plotted, and all flags input are plotted on the section in their correct position (See Overlay plots in Figures 4, 5 and 6).

The DIGDATA program is preset for the processing of seismic data. However options exist to process stripchart profiles and magnetograms. Stripchart digitisation is readily converted as it is essentially equivalent to a single seismic horizon. However, there is no flagging of the converted stripchart data.

Magnetograms have provision for the decoding of shrinkage points and base lines, which are used to control the conversion of the digitised data for each field component.

Data Cards for Digdata

<u>Card</u>	<u>Type</u>	<u>Format</u>
1.	OUTPUT FILE LABEL	(8A10)
2. (ET SEQ)	OPTIONAL PROCESSING CARDS	(A10)
3. (ET SEQ)	INPUT FILES LABELS	(8A10)
4.	- - END OF PARTITION FOR CYBER 76 - - -	
5-8	Repeat of cards 1-4 if the user wishes to process another type of input file.	

Explanation of Cards

1. OUTPUT FILE LABEL

This card is a free format label up to 80 characters in length.

2. OPTIONAL PROCESSING CARDS

The program presets processing for

- a) SEISMIC DATA
- b) FULL LISTING
- c) NO PLOT OUTPUT

If other processing is required the optional cards must be present and are:

- *OBSERVATORY - For GNANGARA and PORT MORESBY observatory magnetograms
- *TOOLANGI - For TOOLANGI observatory magnetograms
- *STRIPCHART - For stripchart records
- *SUPPRESS - Reduces output listing to minimum necessary to check processing
- *PLOT - To produce an output plot file.

3. INPUT FILE LABELS

These cards are 80 character labels which are used to locate the input files. The label on the card must be identical to the header card on the input file; otherwise the file will not be located. These cards must be in the same order as the files on the input document as the search is strictly sequential.

If editing of the input file is required the INPUT FILE LABEL is replaced by a sequence of cards:

* EDITFILE

INPUT FILE LABEL

Edit Instructions

- - - END OF PARTITION FOR CYBER 76 - - -

Edit instructions and explanation are given in Appendix 6.

4. END-OF-PARTITION CARD

This terminates search of the input document.

8: Program SAMPLE

This program is designed to resample the output from DIGDATA into regular times (or shotpoints). Each horizon is extracted and resampled in order using parameters specified by the user. An output file is created for each horizon.

The sampling parameters specified are the name of the horizon, starting time of line, the sampling interval, the maximum interpolation interval, the size of the output blocks and an origin shift (optional) to be added to the data values.

The program extracts all data for the horizon specified, begins at the starting time and resamples the data by linear interpolation between the converted digitised values. Following each interpolated value the survey time is incremented by the sampling interval and the process is repeated. Note if the time axis on the digitised record decreases to the right the sampling interval must be negative.

Data gaps are spanned by linear interpolation if the gap is less than the maximum interpolation interval specified; otherwise the values within the gap are set to a large positive number (1.0E10) for ease of detection.

When the resampling for a horizon is complete, the program blocks the data into units of the size specified, and outputs the file. The header is automatically assigned to the output file by combining the horizon name and input file header.

In similar fashion to DIGDATA, program SAMPLE is preset for seismic resampling. Provision is made for the resampling of stripchart records and magnetograms. The optional origin shift which is added to the data values during resampling is most useful when reducing shore magnetics to a diurnal value. The optional chart width parameter (used for strip-charts) will check for uncorrected steps equal to the chart width, and will apply the necessary correction.

Data Cards for Sample:

<u>Card</u>	<u>Type</u>	<u>Format</u>
1	Optional processing card	(A10)
2	Input file header	(8A10)
3 ET SEQ	Sampling parameter cards	(2A10, F20.7, 2F10.1, I10, 2F5.1)
4	_____ END OF PARTITION _____	
5-7	Repeat of cards 2 to 4 for another input file of the same type	

Explanation of Cards:

1. OPTIONAL PROCESSING CARD

As the program is preset for SEISMIC resampling, this card is necessary if the user wishes to override this preset option. Two alternative options are available:

- *STRIPCHART - to resample stripchart records
- *OBSERVATORY - to resample observatory magnetograms

2. INPUT FILE HEADER

This is the header label of the input file and must be identical to the label used in program DIGDATA when producing the output file.

3. SAMPLING PARAMETER CARDS

Each horizon to be resampled must have one of these cards. The card contains the sampling parameters required by the user and has the form:

Description	Format	Columns	Example
Horizon name	2A10	1-20	Blue
Time of left hand end of digitisation	F20.7	21-40	17.0480030
Sampling Interval (in minutes)	F10.0	41-50	1.0
Maximum Interpolation Interval (in minutes)	F10.0	51-60	30.0
Number of samples per output block	I10	61-70	60
Origin Shift (if applicable)	F5.1	71-75	
Chart width (if applicable)	F5.1	76-80	

The Sampling Interval is negative if the survey time decreases to the right on the digitised record.

4. END-OF-PARTITION CARD

This terminates search of the input document for horizons to be resampled.

9: Program SEGPACK

This program produces a seismic data file of station unumber versus two-way times in the format standard for BMR marine surveys. Each of the resampled horizon data files are extracted, assigned a user specified horiaon number and packed in order. Special flags inserted during digitising are checked and the appropriate data value inserted.

The horizon number is assigned by the user and may be any number between 1 and 1000. This number is incorporated into the horizon data word for identification of the data in the following processing. Numbers must be assigned to horizons so that they increase down the seismic record. This is important as the program rearranges the horizons into number order.

The special flags checked for within the program are the ERODED, OUTCROP and BASEMENT flags. (See Chapter 3). These flags require the insertion of data values from either the horizon above (ERODED and BASEMENT) or the horizon below (OUTCROP).

At this stage in the processing, the seismic two-way time (milliseconds), the flag number (Table 1, Chapter 3), and the horizon number are incorporated into the horizon data word as shown in Table 2. In the example shown, the number 10203456 represents a reflection time of 3456 milliseconds to a poor quality reflection (flag = 2), labelled horizon 10.

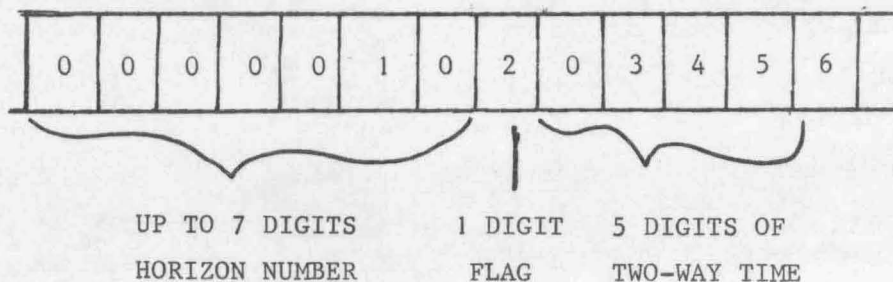


TABLE 2: HORIZON DATA WORD.

36

Data Cards for Segpack:

<u>Card</u>	<u>Type</u>	<u>Format</u>
1	OUTPUT FILE LABEL	(8A10)
2	BLOCK SIZE	(I10)
3	HORIZON FILE LABEL	(8A10)
4	HORIZON NUMBER	(I10)
5-6 ET SEQ	Repeat of cards 3 and 4 for next horizon	

Explanation of Cards

1. OUTPUT FILE LABEL

This card is a free format label up to 80 characters in length. It must contain a hyphen within it, as the program searches for one and uses the preceding descriptor as the first part of each segment label.

2. BLOCK SIZE

This is the size of the blocks specified in the resampling program and is necessary to enable the program to input the data.

3. HORIZON FILE LABEL

This card is the header produced by sample for each horizon. The horizon labels must be input in the same order as they appear on the data file, otherwise they will not be located as search is strictly sequential.

4. HORIZON NUMBER

This is a user supplied number which is assigned to each horizon. Any numbers between 1 and 1000 can be used, but numbers must increase down the seismic section, otherwise the program will rearrange the horizons. Each horizon of the same type must be allocated the same number or the data will not be located when drawing maps.

10: Program MERGPOS

This program is used to provide location information for the seismic data file. A position is found by interpolation for every station number on the seismic data file.

The location information can be provided from any source as long as the file is reformatted into the standard BMR marine data format. If no information is available a location data file can be produced from the digitisation of a shot-point location map using the procedures outlined in Tilbury and Whitworth (1976).

Location data can be supplied at any interval, as the program will interpolate between consecutive values to provide the positions for the seismic data file. Station numbers in both time, for example BMR marine surveys, and shot-points, for example land seismic surveys, can be handled. At present the program will interpolate up to 100 minutes for time surveys and up to 5000 shot-points for other surveys.

The output from this program is the principal-facts file containing station number, latitude, longitude, and seismic two-way times. This data file comprises an index summarising the data-file contents, followed by the basic data.

The data file can be edited and sorted using a variety of marine programs. Possibly the most useful is program SEG SORT (see Whitworth, in prep) which sorts the data into map areas, and sorts the data into increasing station number order within each specified map area. This is useful for concatenating traverse lines which cross map areas, and which were originally digitised from separate maps.

Data Cards for Mergpos:

<u>Card</u>	<u>Type</u>	<u>Format</u>
1.	OUTPUT FILE LABEL	(8A10)
2.	SEISMIC DATA INPUT LABEL	(8A10)
3.	LOCATION DATA INPUT LABEL	(8A10)

Explanation of Cards

1. OUTPUT FILE LABEL

This card is a free format label up to 80 characters in length which describes the contents of the file. It is used to locate and identify the file in later processing.

2. SEISMIC DATA INPUT LABEL

Again this card is on 80-character free-format label and is used to locate the seismic data file for which positions are required.

3. LOCATION DATA INPUT LABEL

An 80-character free-format label which is used to locate the location data file.

11. APPLICATIONS

Once the principal facts file is created, the user can apply automated techniques to produce structure contour and isopach maps. Both posting or contouring of data can be carried out in either reflection time or converted to depth using a user-supplied velocity function. Profiles along traverses can also be produced in both formats.

Firstly a suite of seismic maps of all interpreted horizons can be produced in reflection time (milliseconds), using option *HORTIME of program SEISMAP, and their associated intervals using option *ISOTIME. Data values are posted together with any supplementary information flagged during digitising e.g. quality of reflectors, type of geological feature etc. (see Figures 8 and 9). The irregularly spaced data values can be automatically contoured using the contouring package of programs DATAMAP/SURFACE/PENPUSH and the contours labelled either automatically or by manual intervention following a preliminary plot. In complex areas the user may choose to only post the data and then manually contour the structure inserting faults and trends that cannot presently be mapped satisfactorily by computer.

With the addition of a few data the entire suite of maps can be repeated in depth (metres) using specified constant interval velocities or velocity function (options *HORDEPTH and *ISODEPTH in program SEISMAP). If a more complex velocity conversion process is required, the user may integrate his velocity routine with the system, and the maps can be produced as before. Generally, if a new velocity function is required, only a few cards within a program deck need to be changed to produce a suite of maps that would previously have required repetition of a slow, tedious manual process to produce.

29

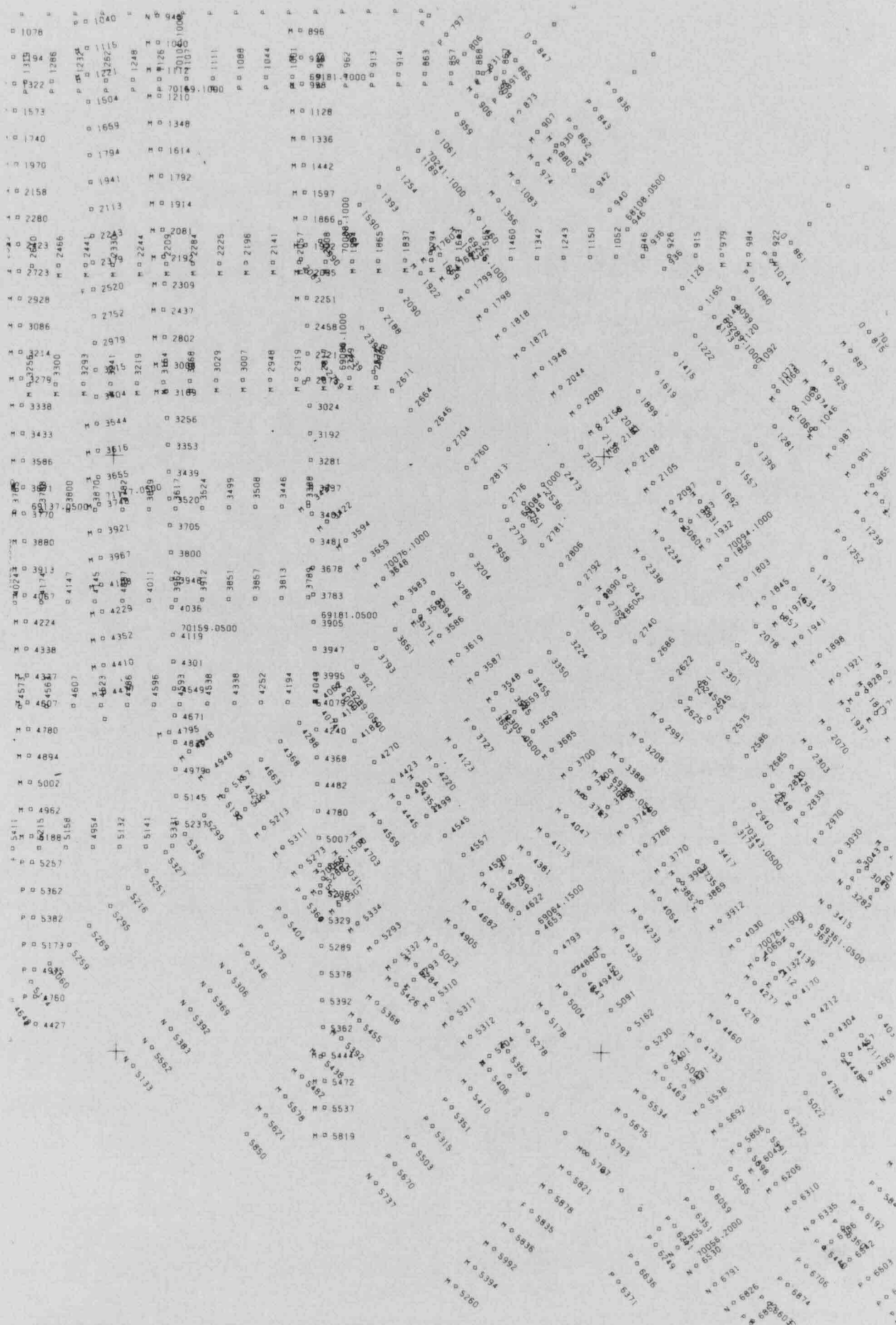


Fig.8 EXAMPLE OF POSTING FOR SEISMIC HORIZONS(DEPTH)-CENOMANIAN HORIZON ON CEDUNA TERRACE

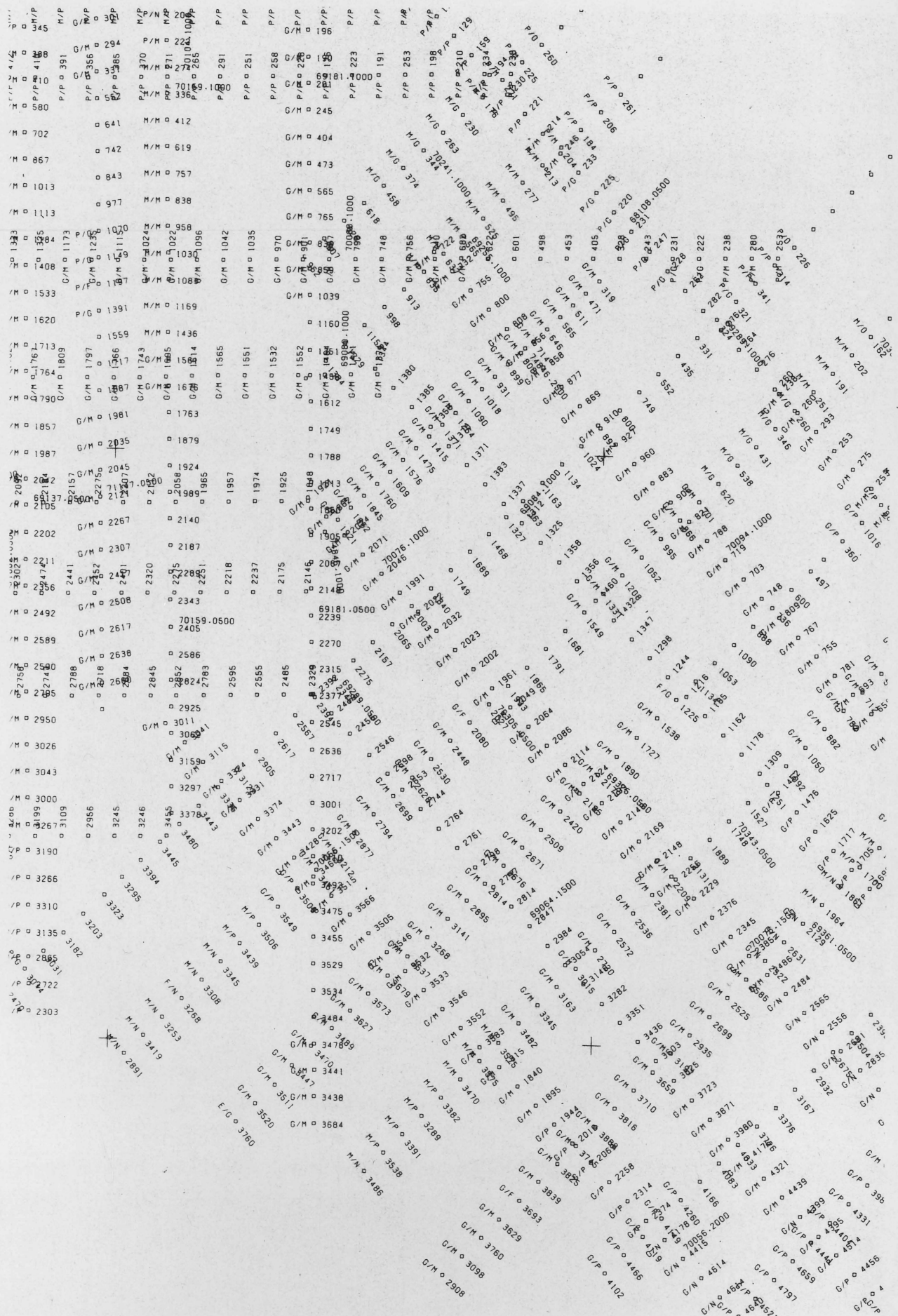


Fig.9 EXAMPLE OF POSTING FOR ISOPACHS-UPPER CRETACEOUS ISOPACH ON CEDUNA TERRACE

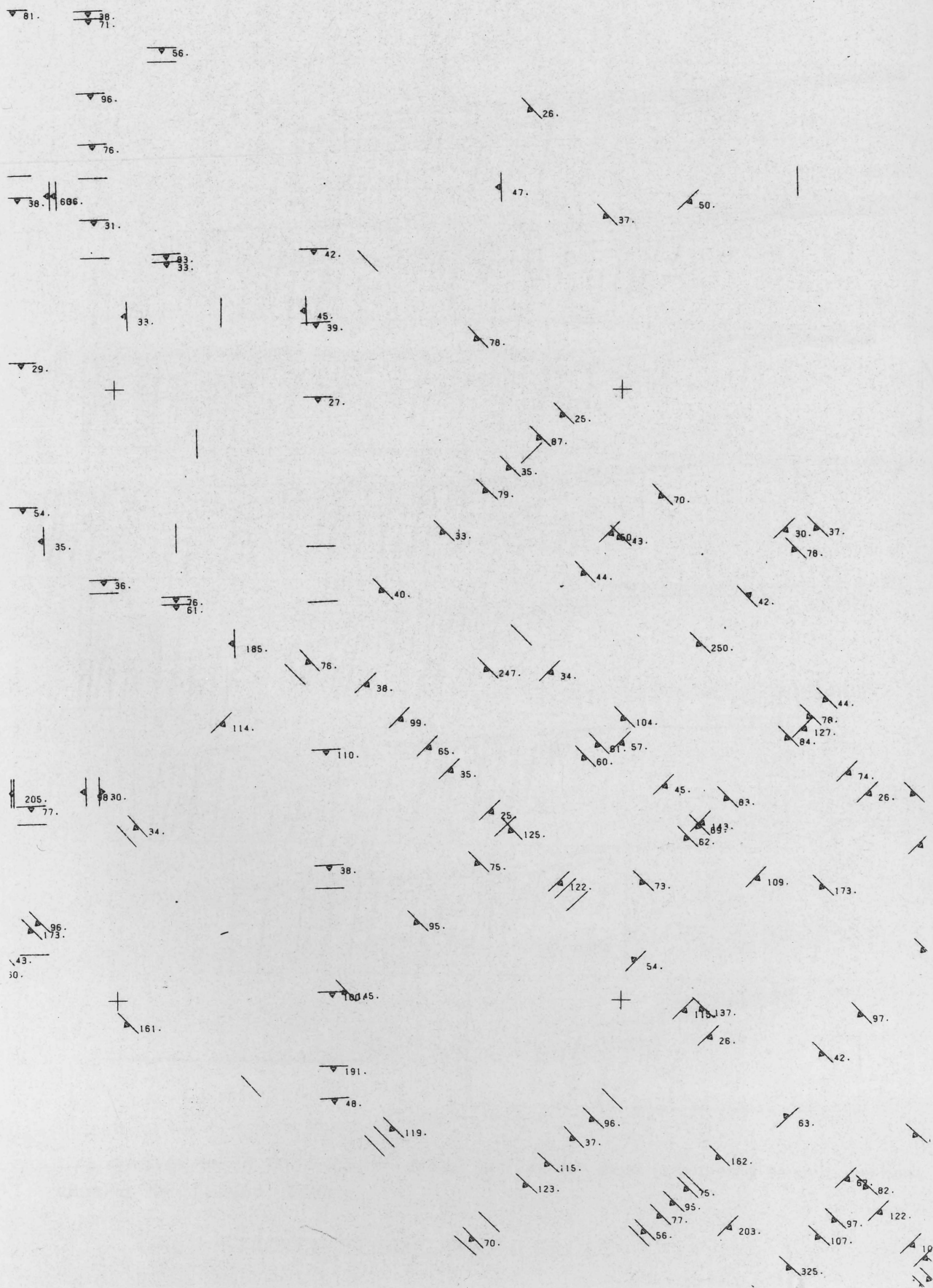
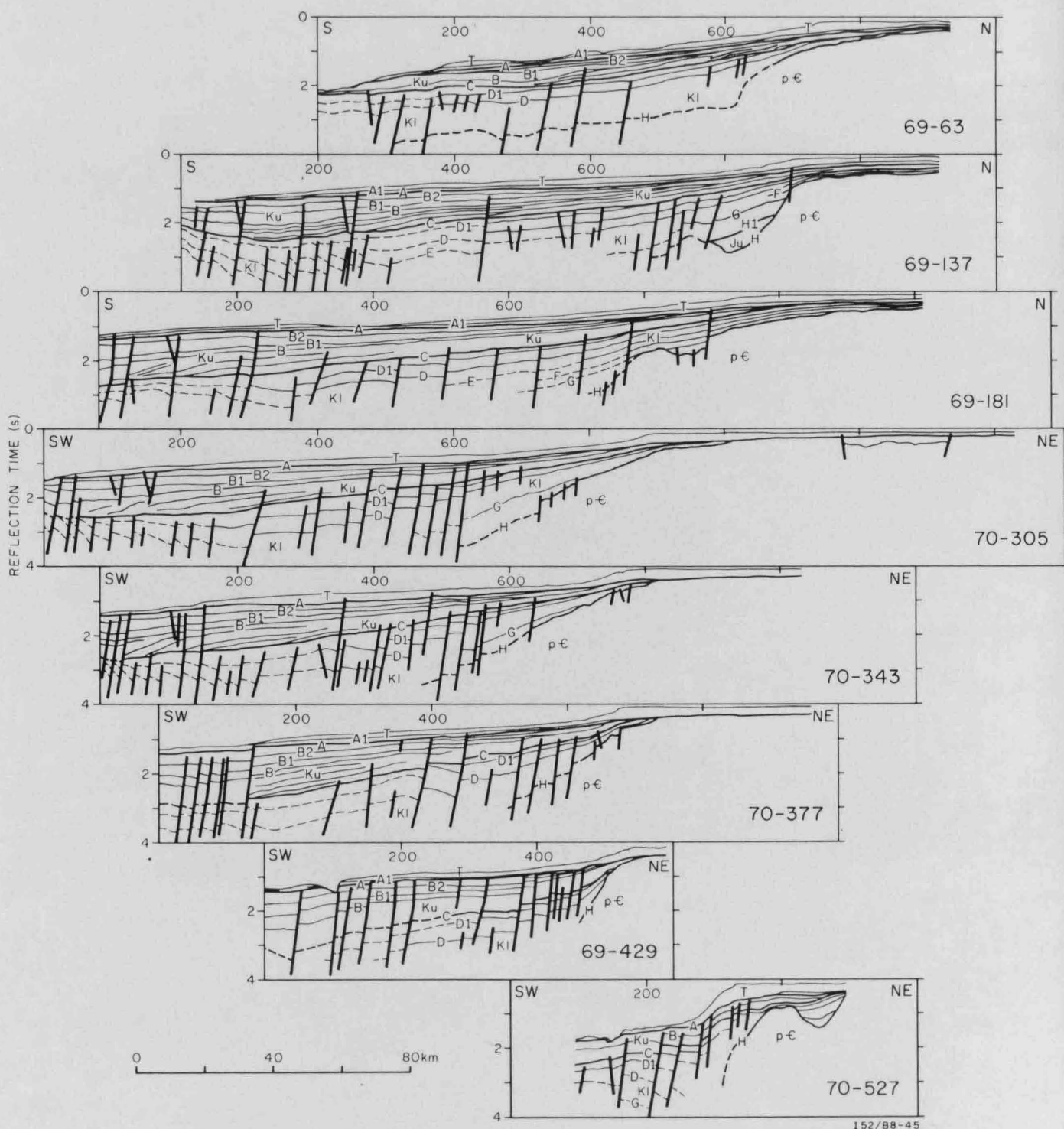


Fig.10 EXAMPLE OF POSTING OF FAULTS - CENOMANIAN HORIZON ON CEDUNA TERRACE



Line drawings based on computer-drawn profiles for eight interpreted seismic sections crossing the Ceduna Terrace

Fig II: STACKED SEISMIC PROFILES

Examples of the types of maps and profiles which can be produced are shown in Figures 8 to 11. These examples have been produced from the principal facts file created for the Ceduna Terrace project (Fraser and Tilbury, 1979).

The posting of depths to a Cenomanian horizon on the Terrace is shown in Figure 8. The depth in metres is posted to the right of the shot-point location. Also, at a user-defined interval, shot-point number is posted above the depth value. If any data is flagged, the mnemonic for the flag is posted to the left of the shot-point location. Note that the quality of the reflecting horizon can be seen to decrease towards the southern end of the survey lines.

An isopach map of the Upper Cretaceous is illustrated in Figure 9. Posting of values is similar to that for horizons except that two flag mnemonics are posted to the left of the shot-point number. These represent the quality of reflectors above and below the isopach respectively.

All faults can be plotted for a particular horizon showing the down-thrown side and the amount of throw on the fault (Fig. 10). At present the posting of faults is done only in reflection time (milliseconds).

Stacked seismic profiles can be produced in both time and depth using similar options in program LINEMAP, as an aid to interpretation and for display in publications. Profiles from eight interpreted seismic sections across the Ceduna Terrace (Fig. 11) show the structural variation across the Terrace, from the most westerly line at the top of the figure to the most easterly line at the bottom.

These techniques and programs already exist within the Marine Geophysics Group to carry out the afore-mentioned plotting of seismic data maps and profiles. Further development is needed for velocity conversion routines in the application of spatially distributed velocity data rather than constant velocity functions or functions which are line distributed.

Contouring on extensively faulted horizons does not, as yet,

give acceptable results. However as the contouring system is modular in design, and the only step affecting the acceptability of the contours is smoothing technique applied, it requires only the finding or development of an acceptable smoothing technique and the automatic contouring will be feasible. In any case the automatic contouring with the present system is a good guide for the manual contouring of the data.

12. REFERENCES

*A.D.P. Group, 1974a - Gradicon-HP2100 Digitising System. Computer System Documentation Report.

*A.D.P. Group, 1980 - HP1000 Minicomputer System: User's Manual. Computer System Documentation Report.

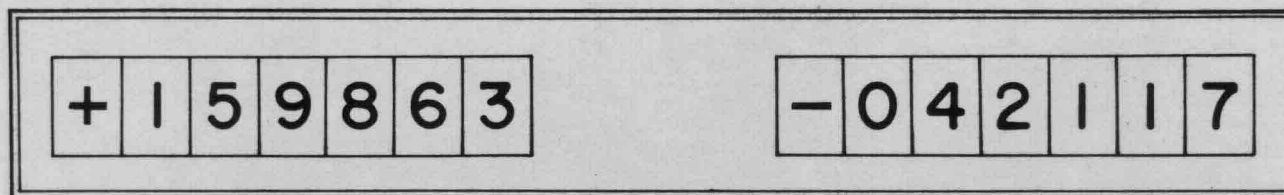
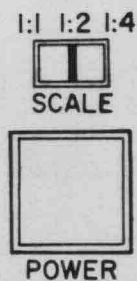
FRASER, A.R., & TILBURY, L.A., 1979 - Structure and stratigraphy of the Ceduna Terrace region, Great Australian Bight Basin. APEA Journal, 19(1), 53-65.

ISTRONICS LTD., 1975 - Gradicon: Graphic to Digital Converter. Operator's Manual.

TILBURY, L.A. & WHITWORTH, R., 1976 - A guide to the production of shot-point location maps using a digitising table and computer. Bureau of Mineral Resources, Australia, Record 1976/65.

*WHITWORTH, R. (in prep.) - A guide to the Marine Group processing system. Computer Program Documentation Report.

* References with asterisks are not published reports but are available in computer printout form on request.



X RESET

Y RESET

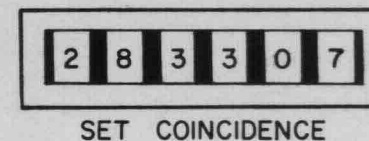
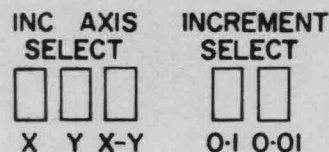
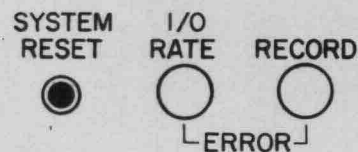
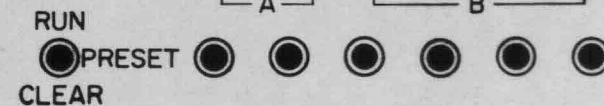
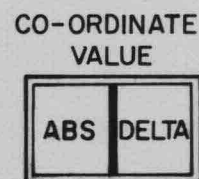
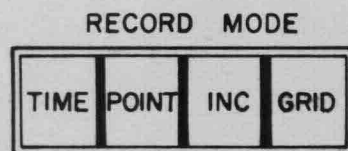
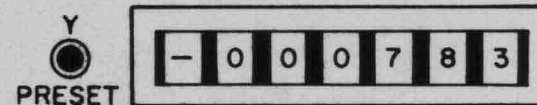
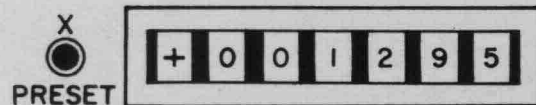


Fig.12 GRADICON : FRONT PANEL CONTROLS

APPENDIX 1: DIGITISER QUESTIONS AND ANSWERS

LG,GRADI

GRADI ON

** GRADICON-TEROS DIGITISING SYSTEM **

REV 5-11/4/1979

DIGITISER DATA ID?

!!LT

XY OR X OR Y CO-ORDINATES?

!!XY

F2 OPTION?

!!O

POINT MODE OR INCREMENTAL MODE?

!!POINT

PRINT DATA?

!!YES

READY - START DIGITISING

! LT

After typing .END on Gradicon keyboard computer replies with:

REMEMBER TO SAVE OR OUTPUT YOUR DATA!

*SF,800,NAME

NAME SAVED 1 RECORDS, 100 WORDS

EX - log off terminal

FILE *LINE 69471 SHELL

OF ORDER 2

```
1 *SLNPE
2 *X+0185.7A Y+0363.03 X+1311.92 Y+0374.57
3 * - - - - - END - OF - FILE - - - - -
4 *HOR
5 *I2A
6 *T69471.0001
7 *X+0185.66 Y+0363.04
8 *T69471.002A
9 *X+0223.96 Y+0363.56 X+0264.33 Y+0364.23 X+0304.71 Y+0364.73 X+0344.99 Y+0365.18
10 *X+0385.23 Y+0365.58 X+0425.69 Y+0366.30 X+0465.88 Y+0366.82 X+0506.32 Y+0367.41
11 *X+0546.58 Y+0367.81 X+0586.71 Y+0368.22 X+0627.10 Y+0368.63 X+0667.50 Y+0369.17
12 *X+0707.66 Y+0369.51 X+0748.23 Y+0370.16 X+0788.41 Y+0370.56 X+0828.71 Y+0371.24
13 *X+0869.10 Y+0371.59 X+0908.50 Y+0371.95 X+0949.12 Y+0372.32 X+0989.33 Y+0372.53
14 *X+1029.43 Y+0372.56 X+1070.20 Y+0372.99 X+1110.70 Y+0373.42 X+1150.70 Y+0373.43
15 *X+1191.21 Y+0374.01 X+1231.50 Y+0374.14 X+1271.49 Y+0374.34 X+1311.98 Y+0374.38
16 * - - - - - END - OF - FILE - - - - -
17 *VERT
18 *I1900
19 *T0
20 *X+0173.22 Y+0363.03 X+0173.56 Y+0312.87 X+0174.13 Y+0262.74 X+0174.62 Y+0212.53
21 *X+0175.43 Y+0162.76 X+0175.89 Y+0112.63 X+0176.08 Y+0062.48 X+0176.72 Y+0012.55
22 * - - - - - END - OF - FILE - - - - -
23 *RWATER
24 *X+0185.40 Y+0241.42 X+0187.40 Y+0242.06 X+0189.40 Y+0242.39 X+0191.40 Y+0242.79
25 *X+0193.40 Y+0242.82 X+0195.40 Y+0242.55 X+0197.40 Y+0241.92 X+0199.40 Y+0241.73
26 *X+0201.40 Y+0241.77 X+0203.40 Y+0241.87 X+0205.40 Y+0242.10 X+0207.40 Y+0242.22
27 *X+0209.40 Y+0242.36 X+0211.40 Y+0242.42 X+0213.40 Y+0242.36 X+0215.40 Y+0241.85
308 *X+1303.40 Y+0346.66 X+1305.40 Y+0346.70 X+1307.40 Y+0346.84 X+1309.40 Y+0346.95
309 *X+1311.40 Y+0347.06
310 * - - - - - END - OF - FILE - - - - -
311 *ORANGE
312 *X+0187.40 Y+0224.42 X+0189.40 Y+0224.51 X+0191.40 Y+0224.74 X+0193.40 Y+0224.96
313 *X+0195.40 Y+0225.12 X+0197.40 Y+0225.12 X+0199.40 Y+0225.19 X+0201.40 Y+0225.27
314 *X+0203.40 Y+0225.40 X+0205.40 Y+0225.69 X+0207.40 Y+0226.10 X+0209.40 Y+0226.36
315 *X+0211.40 Y+0226.36 X+0213.40 Y+0226.36 X+0215.40 Y+0226.33 X+0217.40 Y+0226.33
316 *X+0219.40 Y+0226.32 X+0221.40 Y+0226.33 X+0223.40 Y+0226.35 X+0225.40 Y+0226.37
317 *X+0227.40 Y+0226.38 X+0229.40 Y+0226.32 X+0231.40 Y+0226.25 X+0233.40 Y+0226.1A
318 *X+0235.40 Y+0226.19 X+0237.40 Y+0226.27 X+0239.40 Y+0226.66 X+0241.40 Y+0226.93
319 *X+0243.40 Y+0227.12 X+0245.40 Y+0227.14 X+0247.40 Y+0227.14 X+0249.40 Y+0227.14
320 *X+0251.40 Y+0227.16 X+0253.40 Y+0227.26 X+0255.40 Y+0227.49 X+0257.40 Y+0227.73
321 *X+0259.40 Y+0227.79 X+0261.40 Y+0227.95 X+0263.40 Y+0228.1A
322 *+F
323 *X+0265.40 Y+0228.66
324 *+G
325 *X+0267.40 Y+0229.80 X+0269.40 Y+0229.81 X+0271.40 Y+0229.81 X+0273.40 Y+0229.93
326 *X+0275.40 Y+0230.30 X+0277.40 Y+0230.55 X+0279.40 Y+0231.05 X+0281.40 Y+0231.63
327 *X+0283.40 Y+0231.72 X+0285.40 Y+0231.94
328 *+F
329 *X+0287.40 Y+0234.09
330 *+G
331 *X+0289.40 Y+0237.15 X+0291.40 Y+0237.39 X+0293.40 Y+0237.40 X+0295.40 Y+0237.18
332 *X+0297.40 Y+0237.12 X+0299.40 Y+0237.13 X+0301.40 Y+0237.12 X+0303.40 Y+0237.13
333 *X+0305.40 Y+0237.24 X+0307.40 Y+0237.35 X+0309.40 Y+0237.42 X+0311.40 Y+0237.72
334 *+F
335 *X+0313.40 Y+0239.59
336 *+G
337 *X+0315.40 Y+0240.96 X+0317.40 Y+0240.81 X+0319.40 Y+0240.79 X+0321.40 Y+0240.80
1314 *X+1145.40 Y+0292.68 X+1147.40 Y+0292.66 X+1149.40 Y+0292.66 X+1151.40 Y+0292.68
1315 *X+1153.40 Y+0292.84 X+1155.40 Y+0293.10 X+1157.40 Y+0293.43 X+1159.40 Y+0293.66
1316 *X+1161.40 Y+0293.82 X+1163.40 Y+0293.98 X+1165.40 Y+0294.18 X+1167.40 Y+0294.36
1317 *X+1169.40 Y+0294.50 X+1171.40 Y+0294.58 X+1173.40 Y+0294.60 X+1175.40 Y+0294.65
1318 *X+1177.40 Y+0294.83 X+1179.40 Y+0294.90 X+1181.40 Y+0294.97 X+1183.40 Y+0295.06
1319 * - - - - - END - OF - FILE - - - - -
1320 *BASEMENT
1321 *+H
1322 *X+1011.40 Y+0182.10 X+1013.40 Y+0182.39 X+1015.40 Y+0183.08 X+1017.40 Y+0183.84
1323 *X+1019.40 Y+0184.29 X+1021.40 Y+0184.26 X+1023.40 Y+0183.96 X+1025.40 Y+0183.62
1324 *X+1027.40 Y+0183.24 X+1029.40 Y+0182.61 X+1031.40 Y+0182.00 X+1033.40 Y+0181.40
1325 *X+1035.40 Y+0180.76 X+1037.40 Y+0180.09 X+1039.40 Y+0179.54 X+1041.40 Y+0179.00
1326 *X+1043.40 Y+0178.45 X+1045.40 Y+0178.04 X+1047.40 Y+0177.88 X+1049.40 Y+0177.69
1327 *X+1051.40 Y+0177.41 X+1053.40 Y+0177.22 X+1055.40 Y+0177.20 X+1057.40 Y+0177.13
1328 *X+1059.40 Y+0176.85 X+1061.40 Y+0176.33 X+1063.40 Y+0175.98 X+1065.40 Y+0175.73
1329 *X+1067.40 Y+0175.37 X+1069.40 Y+0174.99 X+1071.40 Y+0174.68 X+1073.40 Y+0174.43
1330 *X+1075.40 Y+0174.33 X+1077.40 Y+0174.43 X+1079.40 Y+0174.86 X+1081.40 Y+0175.38
1331 *X+1083.40 Y+0175.59 X+1085.40 Y+0175.85 X+1087.40 Y+0176.17 X+1089.40 Y+0176.66
1332 *X+1091.40 Y+0177.1A X+1093.40 Y+0177.85 X+1095.40 Y+0178.61 X+1097.40 Y+0179.33
1333 *X+1099.40 Y+0179.86 X+1101.40 Y+0180.59 X+1103.40 Y+0181.36 X+1105.40 Y+0182.36
1334 *X+1107.40 Y+0183.53 X+1109.40 Y+0184.77 X+1111.40 Y+0186.14 X+1113.40 Y+0187.57
1335 * - - - - - END - OF - FILE - - - - -
```

NUMBER OF CARDS IN FILE 1335

Appendix 2: EXAMPLE OF DIGITISER OUTPUT

SSSSSSSSSS

FILE LISTING COMPLETED AT 8.168 SECS

SSSSSSSSSS

END OF MULTIPLE FILE LISTING

Record I981/27

23/OSA/7

```

-----
1 INPUT DATA TYPE = SEISMIC 1
1 LISTING REQUIRED = COMPRESSED 1
1 TEST PLOT FILE = NOT MADE 1
-----

```

```

SSSSS FILE *LINE 69084 SHEL  * SKIPPED ON LUN 40
SSSSS FILE *LINE 69181 SHEL  * SKIPPED ON LUN 40
SSSSS FILE *LINE 69255 SHEL  * SKIPPED ON LUN 40
SSSSS FILE *LINE 69471 SHEL  * FOUND ON LUN 40
EDITFILE READ FROM CARD INPUT

```

DIGITISATION OF SLOPE DATA = COMMENCED 1.859 SECS

```

SINROT = .010247
COSROT = .999947

```

DIGITISATION OF HORIZONTAL SCALE DATA = COMMENCED 1.862 SECS

```

AUTOMATIC HORIZONTAL SCALE INTERVAL IS 20.00
HORIZONTAL SCALE ORIGIN IS 69471.000100
1 69471.0001 0.00 0.00
HORIZONTAL SCALE ORIGIN IS 69471.002000
2 69471.0020 3.83 3.83 .01
3 69471.0040 7.87 4.04 .04
4 69471.0060 11.91 4.04 .05
5 69471.0080 15.93 4.03 .05
6 69471.0100 19.96 4.02 .05
7 69471.0120 24.01 4.05 .08
8 69471.0140 28.02 4.02 .09
9 69471.0160 32.07 4.04 .11
10 69471.0180 36.09 4.03 .11
11 69471.0200 40.11 4.01 .11
12 69471.0220 44.15 4.04 .11
13 69471.0240 48.19 4.04 .12
14 69471.0260 52.20 4.02 .11
15 69471.0280 56.26 4.06 .14
16 69471.0300 60.28 4.02 .13
17 69471.0320 64.31 4.03 .16
18 69471.0340 68.35 4.04 .15
19 69471.0360 72.39 4.44 .15
20 69471.0380 76.43 4.06 .15
21 69471.0400 80.37 4.02 .13
22 69471.0420 84.36 4.01 .04
23 69471.0440 88.07 4.09 .04
24 69471.0460 92.51 4.04 .04
25 69471.0480 96.51 4.00 .05
26 69471.0500 100.56 4.05 .07
27 69471.0520 104.54 4.03 .04
28 69471.0540 108.59 4.00 .02
29 69471.0560 112.64 4.05 -.02

```

DIGITISATION OF VERTICAL SCALE DATA = COMMENCED 1.864 SECS

```

AUTOMATIC VERTICAL SCALE INTERVAL IS 1000.00
VERTICAL SCALE ORIGIN IS 0.00
1 0.00 0.00
2 1000.00 -5.02 -5.02
3 2000.00 -10.03 -5.01
4 3000.00 -15.05 -5.02
5 4000.00 -20.03 -4.98
6 5000.00 -25.04 -5.01
7 6000.00 -30.06 -5.01
8 7000.00 -35.05 -4.99

```

LEAST-SQUARES FIT OF LINE TO DATA

```

NO. OF VALUES = 8
INTERCEPT, A0 = -2.048353
SLOPE, A1 = -199.732642
CORR. COEFF. = -1.000000
STD. ERROR = 1.899202

```

DIGITISATION OF RECORD DATA = COMMENCED 1.893 SECS

RECORD LABEL *WATER 1978/09/14*

```

-----
SEISMIC FLAG WAS PRESET TO +6(U0U1)
CHANGE IN SEISMIC FLAG TO +6(U0U1) AT 69471.000025
-----
Record 1981/27 END OF LEVEL NUMBER 1 STARTS 69471.000025 E405 69471.056164 562 VALUES

```

Appendix 3: DIGDATA OUTPUT PRINTOUT (Part 1)

23/OSA/8

DIGITISATION OF RECORD DATA - COMMENCED 2.432 SECS

RECORD LABEL *TUPPINK

1978/09/14*

SEISMIC FLAG WAS PRESET TO +G(UUU) AT 69471.000021
CHANGE IN SEISMIC FLAG TO +G(UUU) AT 69471.011432
CHANGE IN SEISMIC FLAG TO +F(AULT) AT 69471.011531
CHANGE IN SEISMIC FLAG TO +G(UUU) AT 69471.013818
CHANGE IN SEISMIC FLAG TO +F(AULT) AT 69471.013917
CHANGE IN SEISMIC FLAG TO +G(UUU) AT 69471.024540
CHANGE IN SEISMIC FLAG TO +P(UUH) AT 69471.030401

END OF LEVEL NUMBER 2 STARTS 69471.000021

ENDS 69471.036503 368 VALUES

END OF LEVEL NUMBER 3 STARTS 69471.039468

ENDS 69471.055958 167 VALUES

DIGITISATION OF RECORD DATA - COMMENCED 2.170 SECS

RECORD LABEL *DNGNE

1978/09/14*

SEISMIC FLAG WAS PRESET TO +G(UUU) AT 69471.000116
CHANGE IN SEISMIC FLAG TO +G(UUU) AT 69471.003904
CHANGE IN SEISMIC FLAG TO +F(AULT) AT 69471.004004
CHANGE IN SEISMIC FLAG TO +G(UUU) AT 69471.005076
CHANGE IN SEISMIC FLAG TO +F(AULT) AT 69471.005177
CHANGE IN SEISMIC FLAG TO +G(UUU) AT 69471.006368
CHANGE IN SEISMIC FLAG TO +F(AULT) AT 69471.006408
CHANGE IN SEISMIC FLAG TO +F(AULT) AT 69471.009647
CHANGE IN SEISMIC FLAG TO +G(UUU) AT 69471.009746
CHANGE IN SEISMIC FLAG TO +F(AULT) AT 69471.011350
CHANGE IN SEISMIC FLAG TO +G(UUU) AT 69471.011528
CHANGE IN SEISMIC FLAG TO +F(AULT) AT 69471.013816
CHANGE IN SEISMIC FLAG TO +G(UUU) AT 69471.015915
CHANGE IN SEISMIC FLAG TO +F(AULT) AT 69471.020173
CHANGE IN SEISMIC FLAG TO +G(UUU) AT 69471.020272
CHANGE IN SEISMIC FLAG TO +F(AULT) AT 69471.022120
CHANGE IN SEISMIC FLAG TO +G(UUU) AT 69471.022129
CHANGE IN SEISMIC FLAG TO +F(AULT) AT 69471.032000
CHANGE IN SEISMIC FLAG TO +G(UUU) AT 69471.032704

END OF LEVEL NUMBER 4 STARTS 69471.000116

ENDS 69471.036205 361 VALUES

CHANGE IN SEISMIC FLAG TO +M(UDERAT) AT 69471.039468

END OF LEVEL NUMBER 5 STARTS 69471.039468

ENDS 69471.055954 160 VALUES

CHANGE IN SEISMIC FLAG TO +F(AULT) AT 69471.029669
CHANGE IN SEISMIC FLAG TO +M(UDERAT) AT 69471.029770
CHANGE IN SEISMIC FLAG TO +F(AULT) AT 69471.030514
CHANGE IN SEISMIC FLAG TO +M(UDERAT) AT 69471.030514
CHANGE IN SEISMIC FLAG TO +F(AULT) AT 69471.032156
CHANGE IN SEISMIC FLAG TO +M(UDERAT) AT 69471.032300

END OF LEVEL NUMBER 14 STARTS 69471.000113

ENDS 69471.035473 353 VALUES

DIGITISATION OF RECORD DATA - COMMENCED 2.940 SECS

RECORD LABEL *RED

1978/09/14*

SEISMIC FLAG WAS PRESET TO +G(UUU) AT 69471.010424
CHANGE IN SEISMIC FLAG TO +M(UDERAT) AT 69471.012113
CHANGE IN SEISMIC FLAG TO +F(AULT) AT 69471.012318
CHANGE IN SEISMIC FLAG TO +M(UDERAT) AT 69471.018592
CHANGE IN SEISMIC FLAG TO +F(AULT) AT 69471.019003
CHANGE IN SEISMIC FLAG TO +M(UDERAT) AT 69471.022200
CHANGE IN SEISMIC FLAG TO +F(AULT) AT 69471.022363
CHANGE IN SEISMIC FLAG TO +M(UDERAT) AT 69471.025201
CHANGE IN SEISMIC FLAG TO +F(AULT) AT 69471.025707
CHANGE IN SEISMIC FLAG TO +M(UDERAT) AT 69471.027048
CHANGE IN SEISMIC FLAG TO +F(AULT) AT 69471.027053
CHANGE IN SEISMIC FLAG TO +F(AULT) AT 69471.029444
CHANGE IN SEISMIC FLAG TO +M(UDERAT) AT 69471.029546
CHANGE IN SEISMIC FLAG TO +F(AULT) AT 69471.031729
CHANGE IN SEISMIC FLAG TO +M(UDERAT) AT 69471.032149

END OF LEVEL NUMBER 15 STARTS 69471.010424

ENDS 69471.034948 243 VALUES

CHANGE IN SEISMIC FLAG TO +M(UDERAT) AT 69471.041347

END OF LEVEL NUMBER 16 STARTS 69471.041347

ENDS 69471.049575 84 VALUES

DIGITISATION OF RECORD DATA - COMMENCED 3.055 SECS

RECORD LABEL *BASEMENT

1978/09/14*

SEISMIC FLAG WAS PRESET TO +G(UUU) AT 69471.041003
CHANGE IN SEISMIC FLAG TO +M(UDERAT) AT 69471.041003

END OF LEVEL NUMBER 17 STARTS 69471.041003

ENDS 69471.046040 52 VALUES

***** COMPLETED INPUT OF DIGITISER FILE

***** PROCESSING COMPLETED AT 3.052 SECS

Appendix 3: DIGDATA OUTPUT PRINTOUT (Part 2)

Record 1981/27

23/OSA/9

50

EXTRACTION AND RESAMPLING OF RECORD DATA - COMMENCED AT .225 SECS

INPUT DATA TYPE - SEISMIC

RECORDS TO BE EXTRACTED LABELLED *WATER

STARTING TIME OF OUTPUT FILE - 69471.000000
DATA SAMPLING INTERVAL WILL BE - 1.0 MINUTES
MAXIMUM INTERPOLATION INTERVAL - 20.0 MINUTES
OUTPUT RECORDS WILL CONTAIN - 200 SAMPLES
ORIGIN SHIFT (IF APPLICABLE) - 0. DATA UNITS
CHART WIDTH (IF APPLICABLE) - 0. DATA UNITS

SSSSS FILE *DIGUATA 69471

1978/09/14* FOUND ON LUN 1

----- MEMORY SPACE RESET TO 460 WORDS FOR L.C.M. AT .229 SECS
----- MEMORY SPACE RESET TO 820 WORDS FOR L.C.M. AT .234 SECS
----- MEMORY SPACE RESET TO 700 WORDS FOR L.C.M. AT .241 SECS

OUTPUT OF FILE LABELLED *WATER - DIGUATA 69471

1978/09/15*

69471.000000*****
69471.020000 1870.26
69471.040000 281.23

DATA INTERVAL SAMPLED IS 69471.000000 TO 69471.059900

EXTRACTION AND RESAMPLING OF RECORD DATA - COMMENCED AT .270 SECS

INPUT DATA TYPE - SEISMIC

RECORDS TO BE EXTRACTED LABELLED *URANGE

STARTING TIME OF OUTPUT FILE - 69471.000000
DATA SAMPLING INTERVAL WILL BE - 1.0 MINUTES
MAXIMUM INTERPOLATION INTERVAL - 20.0 MINUTES
OUTPUT RECORDS WILL CONTAIN - 200 SAMPLES

SSSSS FILE *DIGUATA 69471

1978/09/14* FOUND ON LUN 1

OUTPUT OF FILE LABELLED *URANGE - DIGUATA 69471

1978/09/15*

69471.000000*****
69471.020000 2346.04
69471.040000 100948.45

DATA INTERVAL SAMPLED IS 69471.000000 TO 69471.059900

EXTRACTION AND RESAMPLING OF RECORD DATA - COMMENCED AT .413 SECS

INPUT DATA TYPE - SEISMIC

RECORDS TO BE EXTRACTED LABELLED *BASEMENT

STARTING TIME OF OUTPUT FILE - 69471.000000
DATA SAMPLING INTERVAL WILL BE - 1.0 MINUTES
MAXIMUM INTERPOLATION INTERVAL - 20.0 MINUTES
OUTPUT RECORDS WILL CONTAIN - 200 SAMPLES
ORIGIN SHIFT (IF APPLICABLE) - 0. DATA UNITS
CHART WIDTH (IF APPLICABLE) - 0. DATA UNITS

SSSSS FILE *DIGUATA 69471

1978/09/14* FOUND ON LUN 1

----- DATA GAP BETWEEN 69471.000000 AND 69471.041003
----- MEMORY SPACE RESET TO 460 WORDS FOR L.C.M. AT .422 SECS
----- MEMORY SPACE RESET TO 820 WORDS FOR L.C.M. AT .424 SECS
----- MEMORY SPACE RESET TO 700 WORDS FOR L.C.M. AT .426 SECS

OUTPUT OF FILE LABELLED *BASEMENT - DIGUATA 69471

1978/09/15*

69471.000000*****
69471.020000*****
69471.040000*****

DATA INTERVAL SAMPLED IS 69471.000000 TO 69471.059900

SSSSSSSSSS SAMPLE COMPLETED AT .432 SECS

Record 1981/27

Appendix 4: SAMPLE OUTPUT
PRINTOUT

23/OSA/10

APPENDIX 5: THE USE AND EDITING OF CARD IMAGE FILES (after R. Whitworth, in prep.).

One of the main problems with the CDC 7600 system is the time taken to input large card files through the terminal. Repeated input is inefficient and creates unnecessary delays in turn-round for other users.

The policy has been adopted of creating card image files as documents on disc, either by direct input through the terminal, or as output from a program in lieu of punch cards. Multiple files within one disc document are allowed, but then the internal structure of the document must follow a consistent pattern if individual files are to be identified.

Each file has a card image header describing the file contents in free format. This header is not considered as part of the file, and usually contains the date of creation of the file in columns 71-80 in the format YYYY/MM/DD. A particular file is found by comparing a card supplied by the user with the file headers in the disc document.

For the system to be of much value, it must be possible to correct for mistakes in punching and erroneous data, and to carry out simple editing procedures. Editing is done in terms of card images, i.e. a whole card is deleted, inserted or replaced, and consecutive groups of cards can be treated in the same fashion. There is an option to delete or replace individual characters; but the program still works in terms of the card image. A numbered list of each card image file is required to allow identification of any particular card: the header is not included in this count. All marine programs that produce card-image output also provide the necessary listing.

A program is informed as to which file is required by the user supplying a card identical to the header of the card image file wanted. The multiple file is searched sequentially until agreement is reached between card and header. Forward search only is allowed, so should a series of files be needed, they must be requested in the correct order.

When editing of this file is needed, the single card header is replaced by:

*EDITFILE

CARD WITH HEADER OF INPUT FILE REQUIRED

EDITING CARDS

***** CONTINUE

EDITING CARDS

.....END OF PARTITION FOR 7600.....

The editing cards include instructions to the program, all of which commence with an asterisk, and data cards to be inserted or to replace existing cards in the file. The edit list is terminated by a single E-O-S card. There are five basic edit instructions which can operate on a single card or sequence of cards.

To delete a single card, N

*DELETE N

To delete cards N1 to N2 inclusive

*DELETE N1 N2

To insert an end-of-file after card N

*ENDFILE N

To insert one or several cards after card N

*INSERT N

Followed by data cards - ended by next * card or E-O-S card

To replace a single card N by one or more data cards

*REPLACE N

Followed by data cards - ended by next * card or E-O-S card

To replace cards N1 to N2 inclusive by one or more data cards

*REPLACE N1 N2

Followed by data cards - ended by next * card or E-O-S card

To update certain characters in card N

*UPDATE N

Followed by a card containing the updated characters in their respective columns (other columns left blank). To insert a blank an up arrow ' ^ ' is placed in the appropriate column.

APPENDIX 6: GRADICON HP2100 DIGITISING SYSTEM (After Tilbury & Whitworth, 1976)

The GRADICON digitising table was designed as a stand-alone system capable of a significant amount of control and formatting by hardware (INSTRONICS, 1975). However, it had two short-comings, its output rate was low, and recording on digital magnetic tape was not possible. To overcome these limitations it has been integrated with the Hewlett-Packard HP2100 system operated by the ADP Group. There are four pieces of hardware of interest to the user: the digitising table itself, the console that controls accuracy and formatting, the keyboard for manual entry of information, and an NCR terminal for communication with the HP system.

The table has an operating area of 160 by 130 cm, and outputs digitised positions relative to a local origin, the ABS option, or incremental or relative to the preceding digitised point, the DELTA option. The values output may be scaled with ratios of 1:1, 1:2, and 1:4 by setting the scale switch on the console (Fig. 10); with it set to 1:1 the readings are in millimetres with a precision of 0.01 mm. At greater scales, the precision in millimetres falls. All work is done with a ratio of 1:1. The user may choose POINT mode digitisation where one value is output each time the cross-hair follower is pressed, TIME mode where values are continuously output with time at a frequency selected by the RATE control on the console, or INC mode where values are continuously output with distance along the axes at a frequency selected by the VARIABLE INC/INC AXIS SELECT controls.

The origin may be set wherever the user desires by pressing the X and Y reset buttons. However, it is important to establish the origin on the scribed cross-hair at the bottom left-hand corner of the table as this readily allows the user to check whether the origin has shifted. The origin should be checked at the end of the digitising run and at any convenient points during the work as the origin occasionally changes because of electrical faults with the GRADICON system or by accidental pressing of the origin reset buttons.

Information that the user wishes to insert manually is entered through the GRADICON keyboard. This is used in a variety of ways, depending

The numbers N and N1 are right adjusted in columns 11 to 20 while numbers N2 are right adjusted in columns 21 to 30.

The main file and editing instructions are read parallel by the program and the corrected file used within the program. This means that the main file is not altered in any way by the correction cards. The original numbered listing can be used to set up the edit cards for one or more runs until the user is confident that he has the information he wants.

output by the digitiser. The data file is edited using the HP interactive editor (ADP Group, 1980) which is similar to the ED editing system on the Cyber 76 at CSIRO but not as powerful.

To edit, the user types the following commands on any available terminal:

GF, filename	- get file called filename
ED	- turn on editor

Edit file using appropriate instructions

MF	- make file and terminate editor
RF, security code filename	- replaces old filename with edited version

When all editing is finished the computer operator will archive the data to await other data from surveys within the same area. After all interpreted sections have been digitised the operator will write the digitised data onto a magnetic tape in CYBER 76 display code. This tape can be sent to CSIRO and read into the computer.

The GRADICON digitising table at BMR is usually used to digitise the seismic sections, but data from the D-MAC digitising table at CSIRO can be handled equally as well.

upon the format chosen by the user as described in more detail elsewhere (ADP Group, 1974). A limited degree of communication with the Hewlett-Packard system is allowed via the keyboard by entry of special messages that are recognised as commands. These commands are all prefixed with a period (.) and include:

- .ALTER - change the format and printing options (in our case suppress data printout after digitising the map graticule)
- .EOF - to insert an END-OF-FILE in the output file
- .END - to terminate run when digitising is complete

The NCR terminal allows the user to access the digitiser data acquisition program GRADI on the HP2100 system. To commence digitising the user logs into the program by typing into the terminal LG, GRADI and then answering a series of questions asked by the computer to establish the mode of operation and the format required for the digitised data. Appendix 1 show the questions asked by the computer and the answers necessary for seismic digitisation.

The user defines the format he needs via the F2 option query. The co-ordinate data are packed into 80-character card image records (four X-Y readings per card image) if option 0 (zero) is chosen. This is the most efficient packing possible and is used to minimise the length of the output file. It also automatically converts each manual entry into a separate card image which makes identification straightforward.

Point mode digitisation is used for seismic shot-point locations. This also requires the POINT and ABS buttons to be depressed on the digitiser console. In this mode a single X-Y value is sent from the table to the HP system each time the button on the cross-hair follower is pressed.

When the digitising is complete the user finishes by typing .END and saves the document on the HP disc (see Appendix 1). The user can then obtain a numbered listing of the card-image digitised data for checking purposes, by typing

LI, document header

The listing is output on the electrostatic or impact printer in the HP computer room. The printout is scanned for errors or omissions in the manually entered data, and also for any spurious X-Y values that have been

APPENDIX 7: Source program listings

DIGDATA, SAMPLE, SEGPACK, MERGPOS

Microfiche being prepared. (MF165)

Master copy will be with Paul Shelley.

```

1      PROGRAM DIGDATA(TAPE99,TAPE10,TAPE40,INPUT,TAPE60=INPUT,OUTPUT)
C***      DIGDATA
C**      THIS PROGRAM PROCESSES DATA FILES PRODUCED FROM THE GRADICON
5 C**      DIGITISER AT B.M.R. AND SAVES THE CONVERTED RESULTS AS SCALED
C**      DATA PAIRS SUITABLE FOR INPUT TO THE RESAMPLING PROGRAM.
C**      DATA TYPE MAY BE OBSERVATORY RECORDS FROM (1) TOOLANGI,
C**      (2) ELSEWHERE, (3) SEISMIC RECORDS, AND (4) STRIP CHARTS.
10 C**      THE OUTPUT LISTING MAY BE COMPRESSED, IF REQUIRED.
C**      A PLOT AT REAL SCALE MAY BE OUTPUT FOR CHECKING PURPOSES.

      DIMENSION LABEL(8),ITEM1(4),ITEM2(2),ITEM3(2)
      COMMON/DIGDATA/ NWORD(8),NERROR,MASK,IFORM
      COMMON/LIMITS/ XLEFT,XRIGHT,YCHANGE,WIDTH,TSAVE,XSAVE,XMAX,LEVEL
15 COMMON/LUN/ MTIN,MTOUT,MTSCR,LUNSCR,LUNDOC,LUNSAVE,LUNPLOT
      DATA (ITEM1=8HTOOLANGI,10HOBSEVATOR,7HSEISMIC,10HSTRIPCHART)
      DATA (ITEM2=10HCOMPRESSED,7HIN FULL),(ITEM3=8HNOT MADE,8HPRODUCED)
      DATA (MTOUT=10),(LUNDOC=40),(LUNPLOT=99)
20 DATA (MASK=77B),(IFORM=6H(8A10) )

      REWIND LUNDOC      $ REWIND MTOUT

10 READ 20, LABEL
20 FORMAT(8A10)
25 IF (EOF(60)) 280,30
30 CALL TODAY(LABEL(8)) $ PRINT 40, LABEL
40 FORMAT(1H1,/,/,10X,*CREATION OF OUTPUT FILE LABELLED *,
1 1H*,8A10,1H*,/,10X,117(1H- ) )
30 BUFFER OUT(MTOUT,1) (LABEL(1),LABEL(8))
IF (UNIT(MTOUT)) 45,777,777
777 PRINT 778
778 FORMAT(/,10X,10(1H*),10X,*PARITY ERROR WRITING FILE LAB L*)
CALL ABORT

35 45 NERROR = 0 $ WIDTH = -20.0
XLEFT = -0.15 $ XRIGHT = 1.15 $ YCHANGE = 4.0
ITYPE = 3 $ IPRINT = 1 $ IPLOT = 0

50 READ 20, LABEL
IF (LABEL(1).NE.10H*SUPPRESS ) GO TO 31
IPRINT = 0 $ GO TO 50
40 31 IF (LABEL(1).NE.10H*PLOT ) GO TO 32
CALL DIGPLOT(0.0,WIDTH,1)
IPLOT = 1 $ GO TO 50
45 32 IF (LABEL(1).NE.10H*TOOLANGI ) GO TO 33
WIDTH = 10.0 $ ITYPE = 1 $ GO TO 50
33 IF (LABEL(1).NE.10H*OBSERVATO) GO TO 34
WIDTH = 10.0 $ ITYPE = 2 $ GO TO 50
50 34 IF (LABEL(1).NE.10H*STRIPCHAR) GO TO 90
XLEFT = -10.0 $ XRIGHT = 10.0 $ YCHANGE = 15.0
WIDTH = -10.0
ITYPE = 4 $ GO TO 50

60 READ 20, LABEL
IF (EOF(60)) 70,90
55 70 PRINT 80
80 FORMAT(/,10X,10(1H*),10X,*COMPLETED INPUT OF DIGITISER FILE*)

```

```

DIGDATA 2
DIGDATA 3
DIGDATA 4
DIGDATA 5
DIGDATA 6
DIGDATA 7
DIGDATA 8
DIGDATA 9
DIGDATA 10
DIGDATA 11
DIGDATA 12
DIGDATA 13
DIGDATA 14
DIGDATA 15
DIGDATA 16
DIGDATA 17
DIGDATA 18
DIGDATA 19
DIGDATA 20
DIGDATA 21
DIGDATA 22
DIGDATA 23
DIGDATA 24
DIGDATA 25
DIGDATA 26
DIGDATA 27
DIGDATA 28
DIGDATA 29
DIGDATA 30
DIGDATA 31
DIGDATA 32
DIGDATA 33
DIGDATA 34
DIGDATA 35
DIGDATA 36
DIGDATA 37
DIGDATA 38
DIGDATA 39
DIGDATA 40
DIGDATA 41
DIGDATA 42
DIGDATA 43
DIGDATA 44
DIGDATA 45
DIGDATA 46
DIGDATA 47
DIGDATA 48
DIGDATA 49
DIGDATA 50
DIGDATA 51
DIGDATA 52
DIGDATA 53
DIGDATA 54
DIGDATA 55
DIGDATA 56
DIGDATA 57
DIGDATA 58

```

	ENDFILE MTOUT \$ GO TO 10	DIGDATA 59
60	90 PRINT 95, ITEM1(ITYPE),ITEM2(IPRINT+1),ITEM3(IPL0T+1)	DIGDATA 60
	95 FORMAT(//,20X,35(1H-),/,20X,*I*,33X,*I*,	DIGDATA 61
	1 /,20X,*I INPUT DATA TYPE - *,A10,* I*,/,20X,*I*,33X,*I*,	DIGDATA 62
	2 /,20X,*I LISTING REQUIRED - *,A10,* I*,/,20X,*I*,33X,*I*,	DIGDATA 63
65	3 /,20X,*I TEST PLOT FILE - *,A10,* I*,/,20X,*I*,33X,*I*,	DIGDATA 64
	4 /,20X,35(1H-))	DIGDATA 65
	CALL GETFILE(LUNDOC,LABEL,2,NFLAG,60)	DIGDATA 66
	IF (NFLAG.EQ.1) CALL ABORT	DIGDATA 67
70	100 CALL DIGTYPE(ITYPE,IPRINT,IPL0T)	DIGDATA 68
	GO TO 60	DIGDATA 69
	280 TIMEINT = SECOND(UU) \$ PRINT 290, TIMEINT	DIGDATA 70
75	290 FORMAT(//,10X,10(1H\$),10X,*PROCESSING COMPLETED AT*,F9.3,* SECS*)	DIGDATA 71
	IF (IPL0T.EQ.1) CALL DIGPLOT(0.0,7HDIGDATA,9)	DIGDATA 72
	END	DIGDATA 73
		DIGDATA 74
		DIGDATA 75
		DIGDATA 76
		DIGDATA 77
		DIGDATA 78

1	SUBROUTINE DIGPLOT(X,Y,IDENT)	DIGPLOT	2
		DIGPLOT	3
	C*** DIGPLOT	DIGPLOT	4
	C** THIS ROUTINE PLOTS DIGITISER OUTPUT AT REAL SCALE FOR CHECKING	DIGPLOT	5
5	C** PURPOSES. INCLUDED IS POSTING OF INFORMATION THAT CONTROLS	DIGPLOT	6
	C** THE CONVERSION OF THE DIGITISER OUTPUT.	DIGPLOT	7
	C** X,Y - LOCAL ROTATED DIGITISER COORDS IN CENTIMETRES	DIGPLOT	8
	C** IDENT - TYPE FLAG, 1=OPEN PLOT FILE, 2=PLOT WITH PEN DOWN,	DIGPLOT	9
	C** 3=PLOT WITH PEN UP, 4=POST HORIZONTAL SCALE,	DIGPLOT	10
10	C** 5=POST VERTICAL SCALE, 6=POST DELAY OR VERTICAL SHIFT,	DIGPLOT	11
	C** 7=POST SEISMIC FLAGS, 8=PLOT BLOCK, 9=CLOSE PLOT FILE	DIGPLOT	12
		DIGPLOT	13
	DIMENSION IBUF(64)	DIGPLOT	14
	COMMON/LUN/ MTIN,MTOUT,MTSCR,LUNSCR,LUNDOC,LUNSAVE,LUNPLOT	DIGPLOT	15
15	DATA (DUBIOUS=1.0E9),(SIZE=0.07),(ISTART=0)	DIGPLOT	16
		DIGPLOT	17
	GO TO (10,20,20,40,50,60,70,80,90) IDENT	DIGPLOT	18
		DIGPLOT	19
20	10 IF (ISTART.EQ.0) CALL PLOTS(IBUF,64,LUNPLOT)	DIGPLOT	20
	WIDTH = Y \$ AWIDTH = ABS(WIDTH)	DIGPLOT	21
	YMIN = AMIN1(0.0,WIDTH)-1.0	DIGPLOT	22
	YMAX = AMAX1(0.0,WIDTH)+1.0	DIGPLOT	23
	ISIGN = -1 \$ IF (WIDTH.GT.0.0) GO TO 15	DIGPLOT	24
	ISIGN = 1 \$ CALL PLOTBLK(0.0,-WIDTH,-3.6HOFFSET)	DIGPLOT	25
25	15 CALL SYMBOL(0.0,0.0,SIZE,3,0.0,-1)	DIGPLOT	26
	CALL PLOTBLK(1.0,0.0,-3.6HORIGIN)	DIGPLOT	27
	XX = YY = 0.0 \$ AWIDTH = ABS(WIDTH)	DIGPLOT	28
	ISTART = 1	DIGPLOT	29
	RETURN	DIGPLOT	30
30		DIGPLOT	31
	20 KPEN = IDENT	DIGPLOT	32
	IF (X.GT.DUBIOUS) GO TO 35	DIGPLOT	33
	XX = X/2.54 \$ YY = Y/2.54	DIGPLOT	34
	IF (YY.GT.YMIN) GO TO 25	DIGPLOT	35
35	YY = YY+AWIDTH \$ GO TO 30	DIGPLOT	36
	25 IF (YY.LT.YMAX) GO TO 30	DIGPLOT	37
	YY = YY-AWIDTH	DIGPLOT	38
	30 CALL PLOT(XX,YY,KPEN)	DIGPLOT	39
	RETURN	DIGPLOT	40
40		DIGPLOT	41
	35 CALL SYMBOL(XX,YY,SIZE,11,0.0, 1)	DIGPLOT	42
	RETURN	DIGPLOT	43
		DIGPLOT	44
45	40 XX = X/2.54	DIGPLOT	45
	CALL SYMBOL(XX,0.0,SIZE,0,0.0,-1)	DIGPLOT	46
	NDEC = 7	DIGPLOT	47
	CALL NUMBER(XX,0.1*ISIGN,SIZE,Y,90.0*ISIGN,NDEC)	DIGPLOT	48
	RETURN	DIGPLOT	49
		DIGPLOT	50
50	50 XX = X/2.54	DIGPLOT	51
	CALL SYMBOL(0.0,XX,SIZE,2,0.0,-1)	DIGPLOT	52
	CALL NUMBER(0.1,XX,SIZE,Y,0.0,1)	DIGPLOT	53
	RETURN	DIGPLOT	54
		DIGPLOT	55
55	60 XX = X/2.54	DIGPLOT	56
	CALL SYMBOL(XX,-0.1*ISIGN,SIZE,7HDELAY =, -90.0*ISIGN,7)	DIGPLOT	57
	CALL NUMBER(XX,-0.7*ISIGN,SIZE,Y,-90.0*ISIGN,1)	DIGPLOT	58

	CALL PLOT(XX,0.0,3) \$ CALL PLOT(XX,WIDTH,2)	DIGPLOT	59
	RETURN	DIGPLOT	60
60		DIGPLOT	61
	70 CALL SYMBOL(XX,YY,SIZE,Y,0.0,2)	DIGPLOT	62
	CALL PLOT(XX,YY,3)	DIGPLOT	63
	RETURN	DIGPLOT	64
		DIGPLOT	65
65	80 XX = 0.0 \$ IF (ABS(X).GT.0.1) XX = INT(X/2.54)+3.0	DIGPLOT	66
	CALL PLOTBLK(XX,0.0,-3,Y)	DIGPLOT	67
	RETURN	DIGPLOT	68
		DIGPLOT	69
	90 CALL PLOTBLK(0.0,0.0,999,Y)	DIGPLOT	70
70	RETURN	DIGPLOT	71
		DIGPLOT	72
	END	DIGPLOT	73

1	SUBROUTINE DIGTYPE(ITYPE,IPRINT,IPLLOT)	DIGTYPE	2
		DIGTYPE	3
	C*** DIGTYPE	DIGTYPE	4
	C** THIS ROUTINE DECIDES WHAT FORM OF DATA IS TO BE DECODED	DIGTYPE	5
5	C** AND SWITCHES TO THE APPROPRIATE SUBROUTINE.	DIGTYPE	6
	C** ITYPE - DATA TYPE, 1,2=OBSERVATORY, 3=SEISMIC, 4=STRIP CHART	DIGTYPE	7
	C** IPRINT - PRINT FLAG, 0=SUPPRESS MESSAGE, 1=PRINT MESSAGE	DIGTYPE	8
	C** IPLLOT - PLOT FLAG, 0=NO PLOT OUTPUT, 1=PRODUCE PLOT FILE	DIGTYPE	9
10	COMMON/DIGDATA/ NWORD(8),NERROR,MASK,IFORM	DIGTYPE	10
	COMMON/LUN/ MTIN,MTOUT,MTSCR,LUNSCR,LUNDOC,LUNSAVE,LUNPLOT	DIGTYPE	11
	COMMON/LIMITS/ XLEFT,XRIGHT,YCHANGE,WIDTH,TSAVE,XSAVE,XMAX,LEVEL	DIGTYPE	12
		DIGTYPE	13
	CALL READIN (LUNDOC,IFORM,8,NWORD,IFEOF)	DIGTYPE	14
15	GO TO (100,777) IFEOF	DIGTYPE	15
	100 NCHAR = SHIFT(NWORD(1),-54).AND.MASK	DIGTYPE	16
	IF (NCHAR.NE.1RS) GO TO 777	DIGTYPE	17
	CALL SLOPE	DIGTYPE	18
		DIGTYPE	19
20	CALL READIN (LUNDOC,IFORM,8,NWORD,IFEOF)	DIGTYPE	20
	GO TO (160,777) IFEOF	DIGTYPE	21
	160 NCHAR = SHIFT(NWORD(1),-54).AND.MASK	DIGTYPE	22
	IF (NCHAR.NE.1RH) GO TO 777	DIGTYPE	23
	CALL HSCALE(ITYPE,IPRINT,IPLLOT)	DIGTYPE	24
25	IF (IPLLOT.EQ.1) CALL DIGPLOT(0.0,10HHORIZONTAL,8)	DIGTYPE	25
		DIGTYPE	26
	CALL READIN (LUNDOC,IFORM,8,NWORD,IFEOF)	DIGTYPE	27
	GO TO (200,777) IFEOF	DIGTYPE	28
	200 NCHAR = SHIFT(NWORD(1),-54).AND.MASK	DIGTYPE	29
30	IF (NCHAR.NE.1RV) GO TO 777	DIGTYPE	30
	CALL VSCALE(ITYPE,IPRINT,IPLLOT)	DIGTYPE	31
	IF (IPLLOT.EQ.1) CALL DIGPLOT(0.0,8HVERTICAL,8)	DIGTYPE	32
		DIGTYPE	33
	LEVEL = 0 \$ XMAX = 0.0	DIGTYPE	34
35	KOUNT = 0 \$ IF (ITYPE.GT.2) KOUNT = 3	DIGTYPE	35
		DIGTYPE	36
	230 KOUNT = KOUNT+1	DIGTYPE	37
	CALL READIN (LUNDOC,IFORM,8,NWORD,IFEOF)	DIGTYPE	38
40	GO TO (240,500) IFEOF	DIGTYPE	39
	240 NCHAR = SHIFT(NWORD(1),-54).AND.MASK	DIGTYPE	40
	IF (ITYPE.GT.2) GO TO 260	DIGTYPE	41
	GO TO (250,260,250,260,510) KOUNT	DIGTYPE	42
	250 IF (NCHAR.NE.1RO) GO TO 777	DIGTYPE	43
	CALL ORIGIN \$ GO TO 230	DIGTYPE	44
45	260 IF (NCHAR.NE.1RR) GO TO 500	DIGTYPE	45
	CALL RECORD(ITYPE,IPRINT,IPLLOT)	DIGTYPE	46
	IF (IPLLOT.EQ.1) CALL DIGPLOT(0,6HRECORD,8)	DIGTYPE	47
	GO TO 230	DIGTYPE	48
		DIGTYPE	49
50	500 IF (KOUNT.LT.5) GO TO 777	DIGTYPE	50
	510 IF (IPLLOT.EQ.1) CALL DIGPLOT(XMAX,10HNEW ORIGIN,8)	DIGTYPE	51
	IF (IFEOF.EQ.1) GO TO 100	DIGTYPE	52
	RETURN	DIGTYPE	53
		DIGTYPE	54
55	777 PRINT 778, NWORD	DIGTYPE	55
	778 FORMAT(/,10X,10(1H*),10X,*UNRECOGNISED INITIAL CHARACTER AT BEGINN	DIGTYPE	56
	ING OF BLOCK OF DATA - SHOULD BE 'S,H,V,R, OR O'*,	DIGTYPE	57
		DIGTYPE	58

2 /,30X,1H*,8A10,1H*)
CALL ABORT

60

END

DIGTYPE 59
DIGTYPE 60
DIGTYPE 61
DIGTYPE 62

CARD NR. SEVERITY DETAILS DIAGNOSIS OF PROBLEM

15	I	AN IF STATEMENT MAY BE MORE EFFICIENT THAN A Z OR 3 BRANCH COMPUTED GO TO STATEMENT.
21	I	AN IF STATEMENT MAY BE MORE EFFICIENT THAN A Z OR 3 BRANCH COMPUTED GO TO STATEMENT.
28	I	AN IF STATEMENT MAY BE MORE EFFICIENT THAN A Z OR 3 BRANCH COMPUTED GO TO STATEMENT.
39	I	AN IF STATEMENT MAY BE MORE EFFICIENT THAN A Z OR 3 BRANCH COMPUTED GO TO STATEMENT.

1	SUBROUTINE SLOPE	SLOPE	2
		SLOPE	3
	C*** SLOPE	SLOPE	4
	C** THIS SUBROUTINE IS USED BY DIGDATA TO CALCULATE THE	SLOPE	5
5	C** ORIENTATION OF THE RECORD ON THE DIGITISING TABLE.	SLOPE	6
	C** IF AN ERROR OCCURS IN CONVERSION THE SLOPE IS SET TO ZERO.	SLOPE	7
		SLOPE	8
	DIMENSION X(2),Y(2)	SLOPE	9
	COMMON/LUN/ MTIN,MTOUT,MTSCR,LUNSCR,LUNDOC,LUNSAVE,LUNPLOT	SLOPE	10
10	COMMON/DIGDATA/ NWORD(8),NERROR,MASK,IFORM	SLOPE	11
	COMMON/SLOPE/SINROT,COSROT	SLOPE	12
	TIMEINT = SECOND(UU) \$ PRINT 10, TIMEINT	SLOPE	13
	10 FORMAT(/,10X,*DIGITISATION OF SLOPE DATA - COMMENCED*,	SLOPE	14
15	1 F9.3,* SECS*,/,10X,55(1H-))	SLOPE	15
		SLOPE	16
	K = 0	SLOPE	17
	200 CALL READIN(LUNDOC,IFORM,8,NWORD,IFEOF)	SLOPE	18
	GO TO (210,250) IFEOF	SLOPE	19
20		SLOPE	20
	210 CALL XYCODE(XX,YY,NWORD,IFERROR,IFEND)	SLOPE	21
	GO TO (230,220) IFERROR	SLOPE	22
	220 PRINT 60, NWORD	SLOPE	23
	60 FORMAT(/,10X,10(1H*),10X,*NON-NUMERIC INFORMATION IN SLOPE DATA*,	SLOPE	24
25	1 /,30X,1H*,8A10,1H*)	SLOPE	25
	NERROR = NERROR+1 \$ GO TO 240	SLOPE	26
		SLOPE	27
	230 K = K+1 \$ IF (K.GT.2) GO TO 240	SLOPE	28
	X(K) = XX \$ Y(K) = YY	SLOPE	29
30	240 GO TO (210,200) IFEND	SLOPE	30
		SLOPE	31
	250 IF (K-2) 20,100,80	SLOPE	32
		SLOPE	33
	20 PRINT 30 \$ GO TO 120	SLOPE	34
35	30 FORMAT(/,10X,10(1H*),10X,*EOF ENCOUNTERED IN FIRST TWO VALUES OF S	SLOPE	35
	1LOPE DATA - SHOULD BE NUMERIC DATA*)	SLOPE	36
	80 PRINT 90 \$ GO TO 100	SLOPE	37
	90 FORMAT(/,10X,10(1H*),10X,*SLOPE DATA HAS MORE THAN TWO VALUES*)	SLOPE	38
		SLOPE	39
40	100 DX = X(2)-X(1) \$ DY = Y(2)-Y(1)	SLOPE	40
	DENOM = SQRT(DX**2+DY**2)	SLOPE	41
	SINROT = DY/DENOM \$ COSROT = DX/DENOM	SLOPE	42
	PRINT 110, SINROT,COSROT	SLOPE	43
	110 FORMAT(/,20X,*SINROT =*,F10.6,/,20X,*COSROT =*,F10.6)	SLOPE	44
45	RETURN	SLOPE	45
		SLOPE	46
	120 PRINT 130	SLOPE	47
	130 FORMAT(/,10X,10(1H-),10X,*ERROR IN SLOPE - ROTATION SET TO ZERO*)	SLOPE	48
	SINROT = 0.0 \$ COSROT = 1.0	SLOPE	49
50		SLOPE	50
	RETURN	SLOPE	51
		SLOPE	52
	END	SLOPE	53
		SLOPE	54

CARD NR. SEVERITY DETAILS

DIAGNOSIS OF PROBLEM

19	I	AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
22	I	AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
30	I	AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.

```

1      SUBROUTINE HSCALE(ITYPE,IPRINT,IPILOT)
C***      HSCALE
C**      THIS ROUTINE DECODES THE DIGITISER DATA THAT DEFINE
5      C**      HORIZONTAL SCALE AND PROCESSES IT TO OBTAIN A PAIRED SET
C**      OF TIME AND DISTANCE VALUES
C**      ITYPE - DATA TYPE, 1,2=OBSERVATORY, 3=SEISMIC, 4=STRIP CHART
C**      IPRINT - PRINT FLAG, 0=SUPPRESS MESSAGE, 1=PRINT MESSAGE
C**      IPILOT - PLOT FLAG, 0=NO PLOT OUTPUT, 1=PRODUCE PLOT FILE
10     COMMON/SLOPE/SINROT,COSROT
COMMON/DIGDATA/ NWORD(8),NERROR,MASK,IFORM
COMMON/LUN/ MTIN,MTOUT,MTSCR,LUNSCR,LUNDOC,LUNSAVE,LUNPLOT
15     COMMON/HSCALE/ KOUNT,T(100),X(100),Y(100),XH,YH,ISWAP,TFIRST,TLAST
COMMON/RECORD/ NAME(3),TBEGIN,TEND,NPOINTS,TDATA(2,1500)
TIMEINT = SECC - JUU) $ PRINT 10, TIMEINT
10     FORMAT(/,10X,-DIGITISATION OF HORIZONTAL SCALE DATA - COMMENCED*,
1     F9.3,* SECS*,/,10X,64(1H-))
20     CALL READIN(LUNDOC,IFORM,8,NWORD,IFEOF)
GO TO (20,30) IFEOF
20     NCHAR = SHIFT(NWORD(1),-54).AND.MASK
IF (NCHAR.EQ.1R1) GO TO 50
25     30 PRINT 40, NWORD
40     FORMAT(/,10X,10(1H*),10X,*UNRECOGNISED INITIAL CHARACTER AFTER HOR
11ZONTAL LABEL - SHOULD BE SCALE INTERVAL STARTING WITH 'I'*,
2     /,30X,1H*,8A10,1H*)
NERROR = NERROR+1
30     50 CALL DIGINT(HINT,ITYPE,2)
ISWAP = 0 $ IF (HINT.LT.0.0) ISWAP = 1
35     CALL READIN(LUNDOC,IFORM,8,NWORD,IFEOF)
GO TO (100,110) IFEOF
100    NCHAR = SHIFT(NWORD(1),-54).AND.MASK
IF (NCHAR.EQ.1RT) GO TO 130
40     110 PRINT 120, NWORD
120    FORMAT(/,10X,10(1H*),10X,*UNRECOGNISED INITIAL CHARACTER AFTER INT
1ERVAL LABEL - SHOULD BE HORIZONTAL ORIGIN STARTING WITH 'T'*,
2     /,30X,1H*,8A10,1H*)
NERROR = NERROR+1
45     130 CALL DIGTIME(TVALUE,ITYPE,2)
TFIRST = TVALUE
50     CALL HDATA(TVALUE,HINT,ITYPE,IPRINT,IPILOT)
RETURN
END

```

```

HSCALE 2
HSCALE 3
HSCALE 4
HSCALE 5
HSCALE 6
HSCALE 7
HSCALE 8
HSCALE 9
HSCALE 10
HSCALE 11
HSCALE 12
HSCALE 13
HSCALE 14
HSCALE 15
HSCALE 16
HSCALE 17
HSCALE 18
HSCALE 19
HSCALE 20
HSCALE 21
HSCALE 22
HSCALE 23
HSCALE 24
HSCALE 25
HSCALE 26
HSCALE 27
HSCALE 28
HSCALE 29
HSCALE 30
HSCALE 31
HSCALE 32
HSCALE 33
HSCALE 34
HSCALE 35
HSCALE 36
HSCALE 37
HSCALE 38
HSCALE 39
HSCALE 40
HSCALE 41
HSCALE 42
HSCALE 43
HSCALE 44
HSCALE 45
HSCALE 46
HSCALE 47
HSCALE 48
HSCALE 49
HSCALE 50
HSCALE 51
HSCALE 52
HSCALE 53
HSCALE 54
HSCALE 55

```


SUBROUTINE HSCALE

76/76 OPT=1

FTN 4.8+528

11/06/81 14.18.29

PAGE

2

CARD NR. SEVERITY DETAILS

DIAGNOSIS OF PROBLEM

22 1
36 1

AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.

1	SUBROUTINE DIGINT(TINT,ITYPE,NTYPE)	DIGINT	2
		DIGINT	3
	C*** DIGINT	DIGINT	4
	C** THIS ROUTINE DECODES THE SCALE INTERVAL TO BE USED IN	DIGINT	5
5	C** ROUTINES 'VSCALE' AND 'HSCALE' WHEN CONSTRUCTING THE SCALING	DIGINT	6
	C** CONSTANTS AND ARRAYS RESPECTIVELY.	DIGINT	7
	C** TINT - SCALE INTERVAL IN DATA UNITS USED LATER IN PROCESSING	DIGINT	8
	C** ITYPE - DATA TYPE, 1,2=OBSERVATORY, 3=SEISMIC, 4=STRIP CHART	DIGINT	9
	C** NTYPE - FLAG IDENTIFYING CALLING ROUTINE, 1=VSCALE, 2=HSCALE	DIGINT	10
10	COMMON/DIGDATA/ NWORD(8),NERROR,MASK,IFORM	DIGINT	11
	IF (ITYPE.LT.3.AND.NTYPE.EQ.1) RETURN	DIGINT	12
		DIGINT	13
15	CALL FPCODE(NWORD(1),2,TINT,IFLAG)	DIGINT	14
	IF (IFLAG.EQ.0) GO TO 160	DIGINT	15
		DIGINT	16
	IF (NTYPE.EQ.1) PRINT 100, NWORD	DIGINT	17
	IF (NTYPE.EQ.2) PRINT 110, NWORD	DIGINT	18
20	100 FORMAT(/,10X,10(1H*),10X,*ERROR IN DECODING VERTICAL INTERVAL*,	DIGINT	19
	1 5X,*- VALUE SET TO 100 UNITS*,/,30X,1H*,8A10,1H*)	DIGINT	20
	110 FORMAT(/,10X,10(1H*),10X,*ERROR IN DECODING HORIZONTAL INTERVAL*,	DIGINT	21
	1 5X,*- VALUE SET TO 10 UNITS*,/,30X,1H*,8A10,1H*)	DIGINT	22
	TINT = 100.0 \$ IF (NTYPE.EQ.2) TINT = 10.0	DIGINT	23
25	NERROR = NERROR+1	DIGINT	24
		DIGINT	25
	160 IF (NTYPE.EQ.1) PRINT 200, TINT	DIGINT	26
	IF (NTYPE.EQ.2) PRINT 210, TINT	DIGINT	27
	200 FORMAT(/,20X,*AUTOMATIC VERTICAL SCALE INTERVAL IS *,F10.2)	DIGINT	28
30	210 FORMAT(/,20X,*AUTOMATIC HORIZONTAL SCALE INTERVAL IS *,F10.2)	DIGINT	29
		DIGINT	30
	RETURN	DIGINT	31
		DIGINT	32
	END	DIGINT	33
		DIGINT	34
		DIGINT	35

1	SUBROUTINE DIGTIME(TVALUE, ITYPE, NTYPE)	DIGTIME	2
		DIGTIME	3
	C*** DIGTIME	DIGTIME	4
	C** THIS ROUTINE DECODES THE STARTING TIME OR ORIGIN VALUES TO	DIGTIME	5
5	C** BE USED IN ROUTINES 'VSCALE' AND 'HSCALE' WHEN CONSTRUCTING	DIGTIME	6
	C** THE SCALING CONSTANTS AND ARRAYS RESPECTIVELY.	DIGTIME	7
	C** TVALUE - STARTING VALUE IN UNITS USED LATER IN PROCESSING	DIGTIME	8
	C** ITYPE - DATA TYPE, 1,2=OBSERVATORY, 3=SEISMIC, 4=STRIP CHART	DIGTIME	9
	C** NTYPE - FLAG IDENTIFYING CALLING ROUTINE, 1=VSCALE, 2=HSCALE	DIGTIME	10
10	COMMON/DIGDATA/ NWORD(8), NERROR, MASK, IFORM	DIGTIME	11
	IF (ITYPE.LT.3.AND.NTYPE.EQ.1) RETURN	DIGTIME	12
		DIGTIME	13
15	CALL FPCODE(NWORD(1), 2, TVALUE, IFLAG)	DIGTIME	14
	IF (IFLAG.EQ.0) GO TO 160	DIGTIME	15
		DIGTIME	16
	IF (NTYPE.EQ.1) PRINT 100, NWORD	DIGTIME	17
	IF (NTYPE.EQ.2) PRINT 110, NWORD	DIGTIME	18
20	100 FORMAT(/, 10X, 10(1H*), 10X, *ERROR IN DECODING VERTICAL ORIGIN*,	DIGTIME	19
	1 5X, *- VALUE SET TO ZERO*, /, 30X, 1H*, 8A10, 1H*)	DIGTIME	20
	110 FORMAT(/, 10X, 10(1H*), 10X, *ERROR IN DECODING HORIZONTAL ORIGIN*,	DIGTIME	21
	1 5X, *- VALUE SET TO ZERO*, /, 30X, 1H*, 8A10, 1H*)	DIGTIME	22
25	NERROR = NERROR+1 \$ TVALUE = 0.0	DIGTIME	23
		DIGTIME	24
	160 IF (NTYPE.EQ.1) PRINT 200, TVALUE	DIGTIME	25
	IF (NTYPE.EQ.2) PRINT 210, TVALUE	DIGTIME	26
	200 FORMAT(/, 20X, *VERTICAL SCALE ORIGIN IS *, F10.2, /)	DIGTIME	27
30	210 FORMAT(/, 20X, *HORIZONTAL SCALE ORIGIN IS *, F15.7, /)	DIGTIME	28
		DIGTIME	29
	RETURN	DIGTIME	30
		DIGTIME	31
	END	DIGTIME	32
		DIGTIME	33
		DIGTIME	34

```

1      SUBROUTINE HDATA(TVALUE,HINT,ITYPE,IPRINT,IPLT)
      HDATA 2
      HDATA 3
      C***  HDATA 4
      C**   THIS ROUTINE READS IN THE X-Y COORDINATES FOR THE HORIZONTAL 5
      C**   SCALE, APPLIES AN ORIGIN SHIFT AND ROTATION TO THE COORDS 6
      C**   AND COMPUTES A CORRESPONDING TIME VALUE. 7
      C**   THE HORIZONTAL TIMES CAN BE RESET AT IRREGULAR INTERVALS 8
      C**   TO ALLOW FOR BREAKS IN THE RECORD. 9
      C**   THIS INFORMATION IS SAVED IN ARRAYS 'T' AND 'X' 10
      C**   TVALUE - HORIZONTAL SCALE ORIGIN, INCREMENTED AT EACH SAMPLE 11
      C**   HINT - HORIZONTAL SCALE INTERVAL 12
      C**   ITYPE - DATA TYPE, 1,2=OBSERVATORY, 3=SEISMIC, 4=STRIP CHART 13
      C**   IPRINT - PRINT FLAG, 0=SUPPRESS MESSAGE, 1=PRINT MESSAGE 14
      C**   IPLT - PLOT FLAG, 0=NO PLOT OUTPUT, 1=PRODUCE PLOT FILE 15
      HDATA 16
      DIMENSION JFORM(4) 17
      COMMON/SLOPE/SINROT,COSROT 18
      COMMON/HSCALE/ KOUNT,T(100),X(100),Y(100),XH,YH,ISWAP,TFIRST,TLAST 19
      COMMON/DIGDATA/ NWORD(8),NERROR,MASK,IFORM 20
      COMMON/LUN/ MTIN,MTOUT,MTSCR,LUNSCR,LUNDOC,LUNSAVE,LUNPLOT 21
      DATA (JFORM=31H(20X,110,F15.7,F10.2,10X,F10.2) ) 22
      HDATA 23
      KOUNT = 0 24
      250 TBEFORE = TMINS(TVALUE) 25
      TNEXT = INT((TBEFORE+0.5-ISWAP)/ABS(HINT)+ISWAP)*ABS(HINT) 26
      IF (TVALUE.GT.1000.0) TNEXT = TBEFORE 27
      HDATA 28
      260 CALL READIN(LUNDOC,IFORM,8,NWORD,IFEOF) 29
      GO TO (270,330) IFEOF 30
      270 NCHAR = SHIFT(NWORD(1),-54).AND.MASK 31
      IF (NCHAR.NE.1RT) GO TO 280 32
      HDATA 33
      CALL DIGTIME(TVALUE,ITYPE,2) 34
      GO TO 250 35
      HDATA 36
      280 CALL XYCODE (XX,YY,NWORD,IFERROR,IFEND) 37
      GO TO (310,290) IFERROR 38
      HDATA 39
      290 PRINT 300,NWORD 40
      300 FORMAT(/,10X,10(1H*),10X,*NON-NUMERIC INFORMATION IN AUTOMATICALLY 41
      1 INCREMENTED HORIZONTAL SCALE DATA*,/,30X,1H*,8A10,1H*) 42
      NERROR = NERROR+1 43
      GO TO (280,260) IFEND 44
      HDATA 45
      310 IF (KOUNT.GT.0) GO TO 320 46
      XH = XX $ YH = YY 47
      HDATA 48
      320 KOUNT = KOUNT+1 49
      XX = XX-XH $ YY = YY-YH 50
      X(KOUNT) = SINROT*YY+COSROT*XX 51
      Y(KOUNT) = COSROT*YY-SINROT*XX 52
      T(KOUNT) = TBEFORE 53
      HDATA 54
      IF (IPLT.NE.1) GO TO 400 55
      CALL DIGPLOT(X(KOUNT),Y(KOUNT),2) 56
      CALL DIGPLOT(Y(KOUNT),TVALUE,4) 57
      CALL DIGPLOT(X(KOUNT),Y(KOUNT),5) 58

```

	400 PRINT JFORM, KOUNT, TVALUE, X(KOUNT), Y(KOUNT)	HDATA	59
	IF (KOUNT.EQ.1) GO TO 420	HDATA	60
60	DX = X(KOUNT) - X(KOUNT-1)	HDATA	61
	PRINT 410, DX	HDATA	62
	410 FORMAT(1H+, 54X, F10.2)	HDATA	63
		HDATA	64
65	420 TBEFORE = TNEXT = TNEXT+HINT	HDATA	65
	TLAST = TVALUE	HDATA	66
	TVALUE = TSURVEY(TBEFORE)	HDATA	67
	GO TO (280, 260) IFEND	HDATA	68
		HDATA	69
70	330 RETURN	HDATA	70
		HDATA	71
	END	HDATA	72

CARD NR. SEVERITY DETAILS DIAGNOSIS OF PROBLEM

21	I	(20X, 110	HOLLERITH CONSTANT .GT. 10 CHARACTERS, EXCESS CHARACTERS INITIALIZED INTO SUCCEEDING WORDS.
29	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
37	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
43	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
67	I		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.

1	SUBROUTINE VSCALE(ITYPE,IPRINT,IPL0T)	VSCALE	2
		VSCALE	3
	C*** VSCALE	VSCALE	4
	C** THIS ROUTINE DECODES THE DIGITISER DATA THAT DEFINE THE	VSCALE	5
5	C** VERTICAL SCALE, AND PROCESSES IT TO OBTAIN ORIGIN AND SCALE	VSCALE	6
	C** ITYPE - DATA TYPE, 1,2=OBSERVATORY, 3=SEISMIC, 4=STRIP CHART	VSCALE	7
	C** IPRINT - PRINT FLAG, 0=SUPPRESS MESSAGE, 1=PRINT MESSAGE	VSCALE	8
	C** IPL0T - PLOT FLAG, 0=NO PLOT OUTPUT, 1=PRODUCE PLOT FILE	VSCALE	9
		VSCALE	10
10	DIMENSION Y(100),T(100)	VSCALE	11
	COMMON/DIGDATA/ NWORD(8),NERROR,MASK,IFORM	VSCALE	12
	COMMON/LUN/ MTIN,MTOUT,MTSCR,LUNSCR,LUNDOC,LUNSAVE,LUNPLOT	VSCALE	13
	COMMON/VSCALE/ T0,A0,A1,XV,YV	VSCALE	14
		VSCALE	15
15	TIMEINT = SECOND(UU) \$ PRINT 10, TIMEINT	VSCALE	16
	10 FORMAT(/,10X,*DIGITISATION OF VERTICAL SCALE DATA - COMMENCED*,	VSCALE	17
	1 F9.3,* SECS*,/,10X,64(1H-))	VSCALE	18
		VSCALE	19
	CALL READIN (LUNDOC,IFORM,8,NWORD,IFEOF)	VSCALE	20
20	GO TO (20,30) IFEOF	VSCALE	21
	20 NCHAR = SHIFT(NWORD(1),-54).AND.M*ASK	VSCALE	22
	IF (NCHAR.EQ.1R1) GO TO 50	VSCALE	23
		VSCALE	24
	30 PRINT 40,NWORD	VSCALE	25
25	40 FORMAT(/,10X,10(1H*),10X,*UNRECOGNISED INITIAL CHARACTER AFTER VER	VSCALE	26
	1TICAL LABEL - SHOULD BE SCALE INTERVAL STARTING WITH 'I'*,	VSCALE	27
	2 /,30X,1H*,8A10,1H*)	VSCALE	28
	NERROR = NERROR+1	VSCALE	29
		VSCALE	30
30	50 CALL DIGINT(VINT,ITYPE,1)	VSCALE	31
		VSCALE	32
	CALL READIN (LUNDOC,IFORM,8,NWORD,IFEOF)	VSCALE	33
	GO TO (100,110) IFEOF	VSCALE	34
35	100 NCHAR = SHIFT(NWORD(1),-54).AND.M*ASK	VSCALE	35
	IF (NCHAR.EQ.1RT) GO TO 130	VSCALE	36
		VSCALE	37
	110 PRINT 120, NWORD	VSCALE	38
	120 FORMAT(/,10X,10(1H*),10X,*UNRECOGNISED INITIAL CHARACTER AFTER INT	VSCALE	39
40	1ERVAL LABEL - SHOULD BE VERTICAL ORIGIN STARTING WITH 'T'*,	VSCALE	40
	2 /,30X,1H*,8A10,1H*)	VSCALE	41
	NERROR = NERROR+1	VSCALE	42
		VSCALE	43
	130 CALL DIGTIME(TVALUE,ITYPE,1)	VSCALE	44
		VSCALE	45
45	CALL VDATA(TVALUE,VINT,KOUNT,Y,T,ITYPE,IPRINT,IPL0T)	VSCALE	46
		VSCALE	47
	IF (ITYPE.GT.2) GO TO 360	VSCALE	48
	CALL OBSERVE(Y,T,A0,A1,ITYPE)	VSCALE	49
	GO TO 400	VSCALE	50
50	360 CALL DIGLSA(Y,T,KOUNT,A0,A1,IPRINT)	VSCALE	51
		VSCALE	52
	400 T0 = A0 \$ RETURN	VSCALE	53
		VSCALE	54
	END	VSCALE	55

SUBROUTINE VSCALE

76/76 OPT=1

FTN 4.8+528

11/06/81 14.18.29

PAGE 2

CARD NR. SEVERITY DETAILS

DIAGNOSIS OF PROBLEM

20 1
33 1

AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.

1	SUBROUTINE VDATA(TVALUE,VINT,KOUNT,Y,T,ITYPE,IPRINT,IPLLOT)	VDATA	2
		VDATA	3
	C*** VDATA	VDATA	4
	C** THIS ROUTINE READS IN THE X-Y COORDINATES FOR THE VERTICAL	VDATA	5
5	C** SCALE, APPLIES AN ORIGIN SHIFT AND ROTATION TO THE COORDS	VDATA	6
	C** AND COMPUTES A CORRESPONDING VALUE FOR THE VERTICAL PARAMETER	VDATA	7
	C** THIS INFORMATION IS SAVED IN ARRAYS 'T' AND 'Y'	VDATA	8
	C** TVALUE - VERTICAL SCALE ORIGIN, INCREMENTED WITH EACH SAMPLE	VDATA	9
	C** VINT - VERTICAL SCALE INTERVAL	VDATA	10
10	C** Y,T,KOUNT - ARRAYS OF Y,T DATA AND NUMBER OF VALUES	VDATA	11
	C** ITYPE - DATA TYPE, 1,2=OBSERVATORY, 3=SEISMIC, 4=STRIP CHART	VDATA	12
	C** IPRINT - PRINT FLAG, 0=SUPPRESS MESSAGE, 1=PRINT MESSAGE	VDATA	13
	C** IPLLOT - PLOT FLAG, 0=NO PLOT OUTPUT, 1=PRODUCE PLOT FILE	VDATA	14
		VDATA	15
15	DIMENSION Y(100),T(100)	VDATA	16
	COMMON/SLOPE/SINROT,COSROT	VDATA	17
	COMMON/VSCALE/ TO,A0,A1,XV,YV	VDATA	18
	COMMON/DIGDATA/ NWORD(8),NERROR,MASK,IFORM	VDATA	19
	COMMON/LUN/ MTIN,MTOUT,MTSCR,LUNSCR,LUNDOC,LUNSAVE,LUNPLOT	VDATA	20
20	KOUNT = 0	VDATA	21
	250 IF (ITYPE.LT.3) GO TO 260	VDATA	22
	TLAST = (INT((TVALUE+0.1)/VINT+1000000.0)-1000000)*VINT	VDATA	23
		VDATA	24
25	260 CALL READIN (LUNDOC,IFORM,8,NWORD,IFEOF)	VDATA	25
	GO TO (270,370) IFEOF	VDATA	26
	270 NCHAR = SHIFT(NWORD(1),-54).AND.MASK	VDATA	27
	IF (NCHAR.NE.1RT) GO TO 280	VDATA	28
		VDATA	29
30	CALL DIGTIME(TVALUE,ITYPE,1)	VDATA	30
	GO TO 250	VDATA	31
		VDATA	32
	280 CALL XYCODE (XX,YY,NWORD,IFERROR,IFEND)	VDATA	33
	GO TO (310,290) IFERROR	VDATA	34
35		VDATA	35
	290 PRINT 300, NWORD	VDATA	36
	300 FORMAT(/,10X,10(1H*),10X,*NON-NUMERIC INFORMATION IN AUTOMATICALLY	VDATA	37
	1 INCREMENTED VERTICAL SCALE DATA*,/,30X,1H*,8A10,1H*)	VDATA	38
	NERROR = NERROR+1	VDATA	39
40	GO TO (280,260) IFEND	VDATA	40
		VDATA	41
		VDATA	42
	310 IF (KOUNT.GT.0) GO TO 320	VDATA	43
	XV = XX \$ YV = YY	VDATA	44
		VDATA	45
45	320 KOUNT = KOUNT+1	VDATA	46
	XX = XX-XV \$ YY = YY-YV	VDATA	47
	Y(KOUNT) = COSROT*YY-SINROT*XX	VDATA	48
	IF (ITYPE.GT.2) GO TO 340	VDATA	49
	T(KOUNT) = SINROT*YY+COSROT*XX \$ GO TO 350	VDATA	50
50	340 T(KOUNT) = TVALUE	VDATA	51
		VDATA	52
	350 PRINT 360, KOUNT,T(KOUNT),Y(KOUNT)	VDATA	53
	360 FORMAT(20X,I10,F15.2,F10.2)	VDATA	54
	IF (KOUNT.EQ.1) GO TO 380	VDATA	55
55	DY = Y(KOUNT) - Y(KOUNT-1)	VDATA	56
	PRINT 365, DY	VDATA	57
	365 FORMAT(1H*,54X,F10.2)	VDATA	58

60 380 IF (IPLOT.EQ.1) CALL DIGPLOT(Y(KOUNT),T(KOUNT),5)
TVALUE = TLAST = TLAST+VINT
GO TO (280,260) IFEND
370 RETURN
END

VDATA 59
VDATA 60
VDATA 61
VDATA 62
VDATA 63
VDATA 64
VDATA 65

CARD NR. SEVERITY DETAILS DIAGNOSIS OF PROBLEM

26	1	AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
34	1	AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
40	1	AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
60	1	AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.

1	SUBROUTINE OBSERVE(X,Y,A0,A1,ITYPE)	OBSERVE 2
		OBSERVE 3
	C** OBSERVE	OBSERVE 4
	C** THIS ROUTINE DETERMINES THE SHRINKAGE FACTOR FOR OBSERVATORY	OBSERVE 5
5	C** MAGNETOGRAMS FROM THE DATA SUPPLIED BY 'VSCALE'. THE FACTOR	OBSERVE 6
	C** IS MEASURED OVER A NUMBER OF MARKED 100 MM INTERVALS	OBSERVE 7
	C** X,Y - ARRAYS CONTAINING COORDS OF SHRINKAGE POINTS	OBSERVE 8
	C** A0,A1 - ORIGIN AND SLOPE OF VERTICAL SCALE COMPUTED	OBSERVE 9
	C** ITYPE - DATA FLAG, 1=TOOLANGI, 2=OTHER OBSERVATORIES	OBSERVE 10
10	DIMENSION X(1),Y(1)	OBSERVE 11
	PRINT 10	OBSERVE 12
15	10 FORMAT(//,10X,*DETERMINATION OF SHRINKAGE FACTOR FOR MAGNETOGRAM*,	OBSERVE 13
	1 /,10X,49(1H-))	OBSERVE 14
	SUMDIFF = 0.0 \$ KOUNT = 0	OBSERVE 15
		OBSERVE 16
	K = 0	OBSERVE 17
20	DO 50 J=1,3	OBSERVE 18
	K = K+1	OBSERVE 19
	DO 40 I=1,ITYPE	OBSERVE 20
	K = K+1	OBSERVE 21
25	DIFF = SQRT((X(K)-X(K-1))**2+(Y(K)-Y(K-1))**2)	OBSERVE 22
		OBSERVE 23
	KOUNT = KOUNT+1	OBSERVE 24
	SUMDIFF = SUMDIFF+DIFF	OBSERVE 25
		OBSERVE 26
	40 CONTINUE	OBSERVE 27
30	50 CONTINUE	OBSERVE 28
		OBSERVE 29
	AVDIFF = SUMDIFF/KOUNT	OBSERVE 30
	A0 = 0.0 \$ A1 = 100.0/AVDIFF	OBSERVE 31
	AVDIFF = AVDIFF*10.0	OBSERVE 32
35		OBSERVE 33
	PRINT 60,AVDIFF	OBSERVE 34
	60 FORMAT (/,20X,*SHRINKAGE FACTOR - 100 UNITS = *,F6.2,* MMS*)	OBSERVE 35
		OBSERVE 36
		OBSERVE 37
		OBSERVE 38
		OBSERVE 39
40	RETURN	OBSERVE 40
	END	OBSERVE 41
		OBSERVE 42

```

1      SUBROUTINE DIGLSA(X,Y,N,A0,A1,IPRINT)
C***      DIGLSA
C**      THIS ROUTINE COMPUTES THE LINE THAT BEST FITS THE VERTICAL
5      C**      SCALE DATA IN THE LEAST-SQUARES SENSE.
C**      X,Y,N - ARRAYS CONTAINING X,Y DATA, AND NUMBER OF VALUES
C**      A0,A1 - ORIGIN AND SLOPE OF LEAST-SQUARES LINE FOUND
C**      IPRINT - PRINT FLAG, 0=SUPPRESS MESSAGE, 1=PRINT MESSAGE

10     DIMENSION X(1),Y(1)

      SUMX = SUMY = SUMXSQ = SUMYSQ = SUMXY = 0.0

      DO 10 K = 1,N
15     SUMX = SUMX + X(K)
      SUMY = SUMY + Y(K)
      SUMXSQ = SUMXSQ + X(K)**2
      SUMYSQ = SUMYSQ + Y(K)**2
      SUMXY = SUMXY + X(K)*Y(K)
20     10 CONTINUE

      COUNT = N
      XBAR = SUMX/COUNT
      YBAR = SUMY/COUNT
25     XSQBAR = SUMXSQ/COUNT
      YSQBAR = SUMYSQ/COUNT
      XYBAR = SUMXY/COUNT

      SDX = SQRT(XSQBAR - XBAR**2)
      SDY = SQRT(YSQBAR - YBAR**2)
      A0 = (YBAR*XSQBAR - XBAR*XYBAR)/SDX**2
      A1 = (XYBAR - XBAR*YBAR)/SDX**2
      R = (XYBAR - XBAR*YBAR)/(SDX*SDY)
      SEY = SDY * SQRT(1.0 - R**2)
35     PRINT 20, N,A0,A1,R,SEY
20     FORMAT(/,10X,*LEAST-SQUARES FIT OF LINE TO DATA*,/,10X,33(1H-),
1 //,20X,*NO. OF VALUES = *,I7,/,20X,*INTERCEPT, A0 = *,F13.6,
2 /,20X,*SLOPE, A1 = *,F13.6,/,20X,*CORR. COEFF. = *,F13.6,
40     3 /,20X,*STD. ERROR = *,F13.6)

      IF (IPRINT.EQ.0) RETURN

      PRINT 25
45     25 FORMAT(/,25X,*Y*,19X,*TIME*,16X,*DY*,18X,*DT*,/)

      DO 40 K=1,N
      DX = X(K) - (Y(K)-A0)/A1
      DY = Y(K) - (A0 + A1*X(K))
50     PRINT 30, X(K),Y(K),DX,DY
30     FORMAT(20X,F10.5,10X,F10.2,2(10X,F10.5) )
40     CONTINUE

      RETURN
55     END

```

```

DIGLSA 2
DIGLSA 3
DIGLSA 4
DIGLSA 5
DIGLSA 6
DIGLSA 7
DIGLSA 8
DIGLSA 9
DIGLSA 10
DIGLSA 11
DIGLSA 12
DIGLSA 13
DIGLSA 14
DIGLSA 15
DIGLSA 16
DIGLSA 17
DIGLSA 18
DIGLSA 19
DIGLSA 20
DIGLSA 21
DIGLSA 22
DIGLSA 23
DIGLSA 24
DIGLSA 25
DIGLSA 26
DIGLSA 27
DIGLSA 28
DIGLSA 29
DIGLSA 30
DIGLSA 31
DIGLSA 32
DIGLSA 33
DIGLSA 34
DIGLSA 35
DIGLSA 36
DIGLSA 37
DIGLSA 38
DIGLSA 39
DIGLSA 40
DIGLSA 41
DIGLSA 42
DIGLSA 43
DIGLSA 44
DIGLSA 45
DIGLSA 46
DIGLSA 47
DIGLSA 48
DIGLSA 49
DIGLSA 50
DIGLSA 51
DIGLSA 52
DIGLSA 53
DIGLSA 54
DIGLSA 55
DIGLSA 56
DIGLSA 57

```

1	SUBROUTINE ORIGIN	ORIGIN	2
		ORIGIN	3
	C*** ORIGIN	ORIGIN	4
	C** THIS ROUTINE RESETS THE VERTICAL ORIGIN 'A0' AND ITS DIGITISER	ORIGIN	5
5	C** COORDINATES 'XV,YV' IN THE REDUCTION OF DIGITISER DATA.	ORIGIN	6
		ORIGIN	7
	COMMON/LUN/ MTIN,MTOUT,MTSCR,LUNSCR,LUNDOC,LUNSAVE,LUNPLOT	ORIGIN	8
	COMMON/DIGDATA/ NWORD(8),NERROR,MASK,IFORM	ORIGIN	9
	COMMON/VSCALE/ TO,A0,A1,XV,YV	ORIGIN	10
10		ORIGIN	11
	PRINT 10	ORIGIN	12
	10 FORMAT(/,10X,*RESET OF VERTICAL SCALE ORIGIN*,/,10X,30(1H-))	ORIGIN	13
		ORIGIN	14
	CALL READIN(LUNDOC,IFORM,8,NWORD,IFEOF)	ORIGIN	15
15	GO TO (20,30) IFEOF	ORIGIN	16
	20 NCHAR=SHIFT(NWORD(1),-54).AND.MASK	ORIGIN	17
	IF (NCHAR.EQ.1RT) GO TO 50	ORIGIN	18
	30 PRINT 40,NWORD	ORIGIN	19
20	40 FORMAT(/,10X,10(1H*),10X,*UNRECOGNISED INITIAL CHARACTER AFTER ORI	ORIGIN	20
	GIN LABEL - SHOULD BE NEW VERTICAL ORIGIN STARTING WITH 'T',	ORIGIN	21
	2 /,30X,1H*,8A10,1H*)	ORIGIN	22
	NERROR = NERROR+1	ORIGIN	23
		ORIGIN	24
25	50 CALL FPCODE(NWORD(1),2,A0,IFLAG)	ORIGIN	25
	IF (IFLAG.EQ.0) GO TO 80	ORIGIN	26
		ORIGIN	27
	PRINT 70, NWORD	ORIGIN	28
	70 FORMAT(/,10X,10(1H*),10X,*ERROR IN DECODING NEW VERTICAL ORIGIN -	ORIGIN	29
30	1 VALUE SET TO ZERO*,/,30X,1H*,8A10,1H*)	ORIGIN	30
	NERROR = NERROR+1 \$ A0 = 0.0	ORIGIN	31
		ORIGIN	32
	80 PRINT 90, A0	ORIGIN	33
	90 FORMAT(/,20X,*ORIGIN RESET TO *,F10.2,* UNITS*)	ORIGIN	34
35		ORIGIN	35
	K = 0	ORIGIN	36
	95 CALL READIN(LUNDOC,IFORM,8,NWORD,IFEOF)	ORIGIN	37
	GO TO (100,190) IFEOF	ORIGIN	38
	100 CALL XYCODE (XX,YY,NWORD,IFERROR,IFEND)	ORIGIN	39
	GO TO (130,110) IFERROR	ORIGIN	40
40	110 IF (K.EQ.0) PRINT 120, NWORD	ORIGIN	41
	120 FORMAT(/,10X,10(1H*),10X,*NON-NUMERIC INFORMATION FOLLO ING NE V	ORIGIN	42
	ERTICAL ORIGIN*,/,30X,1H*,8A10,1H*)	ORIGIN	43
	GO TO (100,95) IFEND	ORIGIN	44
45	130 K = K+1 \$ IF (K.GT.1) GO TO 140	ORIGIN	45
	XV = XX \$ YV = YY	ORIGIN	46
	140 GO TO (100,95) IFEND	ORIGIN	47
		ORIGIN	48
	190 IF (K-1) 200,250,230	ORIGIN	49
	200 PRINT 210, NWORD	ORIGIN	50
50	210 FORMAT(/,10X,10(1H*),10X,*EOF FOUND INSTEAD OF NEW VERTICAL ORIGIN	ORIGIN	51
	1 COORDINATES - VALUE SET TO ZERO*,/,30X,1H*,8A10,1H*)	ORIGIN	52
	XV = YV = 0.0 \$ GO TO 250	ORIGIN	53
	230 PRINT 240, NWORD	ORIGIN	54
55	240 FORMAT(/,10X,10(1H-),10X,*EOF MISSING AFTER ORIGIN COORDINATES*)	ORIGIN	55
		ORIGIN	56
	250 TO = A0 \$ RETURN	ORIGIN	57
		ORIGIN	58

END

ORIGIN 59

CARD NR. SEVERITY DETAILS DIAGNOSIS OF PROBLEM

15	I	AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
37	I	AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
39	I	AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
43	I	AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
46	I	AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.

1	SUBROUTINE RECORD(ITYPE,IPRINT,IPLLOT)	RECORD 2
	C*** RECORD	RECORD 3
	C** THIS ROUTINE PROCESSES DIGITISED RECORD DATA, CONVERTING	RECORD 4
5	C** THE X-Y POINTS USING THE SCALING DATA SUPPLIED BY 'VSCALE'	RECORD 5
	C** AND 'HSCALE'.	RECORD 6
	C** ITYPE - DATA TYPE, 1,2=OBSERVATORY, 3=SEISMIC, 4=STRIP CHART	RECORD 7
	C** IPRINT - PRINT FLAG, 0=SUPPRESS MESSAGE, 1=PRINT MESSAGE	RECORD 8
	C** IPLLOT - PLOT FLAG, 0=NO PLOT OUTPUT, 1=PRODUCE PLOT FILE	RECORD 9
10		RECORD 10
	COMMON/DIGDATA/ NWORD(8),NERROR,MASK,IFORM	RECORD 11
	COMMON/LUN/ MTIN,MTOUT,MTSCR,LUNSCR,LUNDOC,LUNSAVE,LUNPLOT	RECORD 12
	COMMON/HSCALE/ KOUNT,T(100),X(100),Y(100),XH,YH,ISWAP,TFIRST,TLAST	RECORD 13
15	COMMON/LIMITS/ XLEFT,XR GHT,YCHANGE,WIDTH,TSAVE,XSAVE,XMAX,LEVEL	RECORD 14
	COMMON/RECORD/ NAME(3),I,EGIN,TEND,NPOINTS,TDATA(2,1500)	RECORD 15
	COMMON/FLAGS/ ISEIS,RNUM,NSEIS,LIST(9)	RECORD 16
	COMMON/VSCALE/ TO,A0,A1,XV,YV	RECORD 17
		RECORD 18
20	TSAVE = TFIRST \$ XSAVE = 0.0	RECORD 19
	ISEIS = NSEIS = 1 \$ RNUM = 0.0	RECORD 20
	NPOINTS = 0 \$ KPEN = 3 \$ A0 = TO	RECORD 21
	CALL EXTRACT(20,NWORD(1),2,NAME(1),1)	RECORD 22
	CALL TODAY(NAME(3))	RECORD 23
25		RECORD 24
	TIMEINT = SECOND(UU) \$ PRINT 40, TIMEINT,NAME	RECORD 25
	40 FORMAT(/,10X,*DIGITISATION OF RECORD DATA - COMMENCED*,	RECORD 26
	1 F9.3,* SECS*,/,10X,56(1H-),/,10X,*RECORD LABEL *,1H* 3A10,1H*)	RECORD 27
		RECORD 28
30	IF (IPRINT.EQ.1) PRINT 50, TO,TFIRST,TLAST	RECORD 29
	50 FORMAT(/,20X,*VERTICAL SCALE ORIGIN IS *,F10.2,	RECORD 30
	1 //,20X,*SECTION STARTS AT *,F15.7,5X,*ENDS AT *,F15.7)	RECORD 31
		RECORD 32
	IF (ITYPE.EQ.3) PRINT 70	RECORD 33
35	70 FORMAT(/,10X,10(1H-),10X,*SEISMIC FLAG WAS PRESET TO +G(OOD)*)	RECORD 34
		RECORD 35
	110 CALL READIN (LUNDOC,IFORM,8,NWORD,IFEOF)	RECORD 36
	GO TO (120,280) IFEOF	RECORD 37
	120 NCHAR = SHIFT(NWORD(1),-54).AND.MASK	RECORD 38
		RECORD 39
40	IF (NCHAR.NE.1RD) GO TO 130	RECORD 40
	CALL DELAY(A0,ITYPE,IPRINT,IPLLOT)	RECORD 41
	KPEN = 3 \$ GO TO 110	RECORD 42
		RECORD 43
45	130 IF (NCHAR.NE.1R*) GO TO 140	RECORD 44
	CALL FLAGS(NWORD(1))	RECORD 45
	KPEN = 3 \$ GO TO 110	RECORD 46
		RECORD 47
	140 CALL RECDATA(KPEN,ITYPE,IPRINT,IPLLOT)	RECORD 48
	GO TO 110	RECORD 49
50		RECORD 50
	280 CALL BUFCHK(ITYPE,IPRINT)	RECORD 51
	XMAX = AMAX1(XMAX,XSAVE)	RECORD 52
		RECORD 53
		RECORD 54
55	RETURN	RECORD 55
		RECORD 56
	END	RECORD 57

SUBROUTINE RECORD 76/76 OPT=1

FTN 4.8+528

11/06/81 14.18.29

PAGE 2

CARD NR. SEVERITY DETAILS DIAGNOSIS OF PROBLEM

37 I

AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.

1	SUBROUTINE DELAY(TO, ITYPE, IPRINT, IPLOT)	DELAY	2
		DELAY	3
	C*** DELAY	DELAY	4
	C** THIS ROUTINE DECODES DELAYS TO BE APPLIED TO SEISMIC RECORDS,	DELAY	5
5	C** OR THE EQUIVALENT OFFSETS IN VERTICAL SCALE OF STRIP-CHARTS.	DELAY	6
	C** IF REQUIRED, THE DELAY IS PLOTTED VIA 'DIGPLOT'	DELAY	7
	C** TO - VERTICAL ORIGIN, RESET TO NEW VALUE BEFORE RETURN	DELAY	8
	C** ITYPE - DATA TYPE, 1,2=OBSERVATORY, 3=SEISMIC, 4=STRIP CHART	DELAY	9
	C** IPRINT - PRINT FLAG, 0=SUPPRESS PRINT, 1=PRINT LIST	DELAY	10
10	C** IPLOT - PLOT FLAG, 0=NO PLOT OUTPUT, 1=PRODUCE PLOT FILE	DELAY	11
		DELAY	12
	DIMENSION JFORM(4)	DELAY	13
	COMMON/LIMITS/ XLEFT,XRIGHT,YCHANGE,WIDTH,TSAVE,XSAVE,XMAX,LEVEL	DELAY	14
	COMMON/DIGDATA/ NWORD(8),NERROR,MASK,IFORM	DELAY	15
15	COMMON/RECORD/ NAME(3),TBEGIN,TEND,NPOINTS,TDATA(2,1500)	DELAY	16
	DATA (JFORM=40H(1H+,53X,F15.6) (1H+,53X,F15.7))	DELAY	17
		DELAY	18
	IF (NPOINTS.EQ.0) GO TO 170	DELAY	19
	CALL BUFCHK(IATYPE,IPRINT)	DELAY	20
20		DELAY	21
	170 CALL FPCODE(NWORD(1),2,TDELAY,IFLAG)	DELAY	22
	IF (IFLAG.EQ.0) GO TO 200	DELAY	23
		DELAY	24
	PRINT 190, NWORD	DELAY	25
25	190 FORMAT(/,10X,10(1H+),10X,*ERROR IN DECODING CHANGE IN VERTICAL SCA	DELAY	26
	1LE ORIGIN - CHANGE SET TO ZERO*,/,30X,1H+,8A10,1H+)	DELAY	27
	NERROR = NERROR+1 \$ TDELAY = 0.0	DELAY	28
		DELAY	29
	200 TO = TO+TDELAY	DELAY	30
30	PRINT 210, TDELAY,TO	DELAY	31
	210 FORMAT(/,20X,*CHANGE IN VERTICAL SCALE ORIGIN AT *,20X,	DELAY	32
	1 *OF *,F8.1,* UNITS - ORIGIN NOW *,F10.1,* UNITS*)	DELAY	33
		DELAY	34
	LOUT = 1 \$ IF (TSAVE.GT.20.0) LOUT = 3	DELAY	35
35	PRINT JFORM, TSAVE	DELAY	36
		DELAY	37
	IF (IPLOT.EQ.1) CALL DIGPLOT(XSAVE,TDELAY,6)	DELAY	38
		DELAY	39
	RETURN	DELAY	40
40		DELAY	41
	END	DELAY	42

CARD NR. SEVERITY DETAILS DIAGNOSIS OF PROBLEM

16 1 (1H+,53X HOLLERITH CONSTANT .GT. 10 CHARACTERS, EXCESS CHARACTERS INITIALIZED INTO SUCCEEDING WORDS.

1	SUBROUTINE FLAGS(NWORD)	FLAGS	2
		FLAGS	3
	C*** FLAGS	FLAGS	4
	C** THIS ROUTINE DECODES THE FLAGS INDICATING THE TYPE OF	FLAGS	5
5	C** SEISMIC DATA BEING DIGITISED, AND RETURNS A NUMBER USED	FLAGS	6
	C** AS A SHIFT IN THE VERTICAL TIME TO MARK THE DATA TYPE	FLAGS	7
	C** NWORD - WORD CONTAINING FLAG TO BE DECODED	FLAGS	8
	C** ITYPE - DATA TYPE, 1,2=OBSERVATORY, 3=SEISMIC, 4=STRIP CHART	FLAGS	9
10	C** IPRINT - PRINT FLAG, 0=SUPPRESS MESSAGE, 1=PRINT MESSAGE	FLAGS	10
	C** RNUM - NUMBER RETURNED FOR USE IN MARKING DATA TYPE	FLAGS	11
		FLAGS	12
	COMMON/FLAGS/ ISEIS,RNUM,NSEIS,LIST(9)	FLAGS	13
	DATA (LIST=7H+G(00D),10H+M(00ERAT),7H+P(00R),8H+F(AULT),	FLAGS	14
15	1 10H+N(O REFL),9H+E(RODED),10H+O(UTCROP),9H+B('MENT),9H+A(BSENT))	FLAGS	15
	DATA (MASK=77B)	FLAGS	16
		FLAGS	17
	NCHAR = SHIFT(NWORD,-48).AND.MASK	FLAGS	18
		FLAGS	19
	IF (NCHAR.NE.1RG) GO TO 10	FLAGS	20
20	RNUM = 0.0 \$ NSEIS = 1 \$ GO TO 110	FLAGS	21
		FLAGS	22
	10 IF (NCHAR.NE.1RM) GO TO 20	FLAGS	23
	RNUM = 1.0E5 \$ NSEIS = 2 \$ GO TO 110	FLAGS	24
		FLAGS	25
25	20 IF (NCHAR.NE.1RP) GO TO 30	FLAGS	26
	RNUM = 2.0E5 \$ NSEIS = 3 \$ GO TO 110	FLAGS	27
		FLAGS	28
	30 IF (NCHAR.NE.1RF) GO TO 40	FLAGS	29
30	RNUM = 3.0E5 \$ NSEIS = 4 \$ GO TO 110	FLAGS	30
		FLAGS	31
	40 IF (NCHAR.NE.1RN) GO TO 50	FLAGS	32
	RNUM = 4.0E5 \$ NSEIS = 5 \$ GO TO 110	FLAGS	33
		FLAGS	34
	50 IF (NCHAR.NE.1RE) GO TO 60	FLAGS	35
35	RNUM = 5.0E5 \$ NSEIS = 6 \$ GO TO 110	FLAGS	36
		FLAGS	37
	60 IF (NCHAR.NE.1RO) GO TO 70	FLAGS	38
	RNUM = 6.0E5 \$ NSEIS = 7 \$ GO TO 110	FLAGS	39
		FLAGS	40
40	70 IF (NCHAR.NE.1RB) GO TO 80	FLAGS	41
	RNUM = 7.0E5 \$ NSEIS = 8 \$ GO TO 110	FLAGS	42
		FLAGS	43
	80 IF (NCHAR.NE.1RA) GO TO 90	FLAGS	44
45	RNUM = 1.0E10 \$ NSEIS = 9 \$ GO TO 110	FLAGS	45
		FLAGS	46
	90 PRINT 100, NWORD	FLAGS	47
	100 FORMAT(/,10X,10(1H-),10X,*ILLEGAL SEISMIC FLAG WAS *,1H*,A10,1H*)	FLAGS	48
	RNUM = 0.0 \$ NSEIS = 1	FLAGS	49
		FLAGS	50
50	110 ISEIS = 1 \$ RETURN	FLAGS	51
		FLAGS	52
	END	FLAGS	53

```

1      SUBROUTINE RECDATA(KPEN, ITYPE, IPRINT, IPLOT)
      RECDATA 2
      RECDATA 3
      C*** RECDATA
      C** THIS ROUTINE DECODES ALL THE X-Y DATA ON A CARD IMAGE, ONE
      C** VALUE AT A TIME. IF REQUIRED, THE X-Y POINT IS PLOTTED USING
      C** 'DIGPLOT' SO THAT IT CAN BE CHECKED AGAINST THE ORIGINAL
      C** 'DIGPLOT' SO THAT IT CAN BE CHECKED AGAINST THE ORIGINAL
      C** KPEN - PEN UP/DOWN COMMAND FOR PLOTTING, 2=DOWN, 3=UP
      C** ITYPE - DATA TYPE, 1,2=OBSERVATORY, 3=SEISMIC, 4=STRIP CHART
      C** IPRINT - PRINT FLAG, 0=SUPPRESS MESSAGE, 1=PRINT MESSAGE
      C** IPLOT - PLOT FLAG, 0=NO PLOT OUTPUT, 1=PRODUCE PLOT FILE
      RECDATA 11
      RECDATA 12
      COMMON/DIGDATA/ NWORD(8), NERROR, MASK, IFORM
      RECDATA 13
      COMMON/SLOPE/SINROT, COSROT
      RECDATA 14
      COMMON/HSCALE/ KOUNT, T(100), X(100), Y(100), XH, YH, ISWAP, TFIRST, TLAST
      RECDATA 15
      COMMON/VSCALE/ TO, AO, A1, XV, YV
      RECDATA 16
      COMMON/RECORD/ NAME(3), TBEGIN, TEND, NPOINTS, TDATA(2, 1500)
      RECDATA 17
      COMMON/LIMITS/ XLEFT, XRIGHT, YCHANGE, WIDTH, TSAVE, XSAVE, XMAX, LEVEL
      RECDATA 18
      COMMON/FLAGS/ ISEIS, RNUM, NSEIS, LIST(9)
      RECDATA 19
      DATA (UNKNOWN:-1.0E10)
      RECDATA 20
      220 CALL XYCODE (XX, YY, NWORD, IFERROR, IFEND)
      RECDATA 21
      GO TO (250, 230) IFERROR
      RECDATA 22
      230 PRINT 240, NWORD
      RECDATA 23
      240 FORMAT(/, 10X, 10(1H*), 10X, *ERROR IN DECODING RECORD X-Y DATA*,
      RECDATA 24
      2 /, 3(1H*, 8A10, 1H*)
      RECDATA 25
      IF (IPLOT.EQ.1) CALL DIGPLOT(UNKNOWN, UNKNOWN, 3)
      RECDATA 26
      NERROR = NERROR + 1 $ KPEN = 3
      RECDATA 27
      GO TO (220, 110) IFEND
      RECDATA 28
      30 XROT = SINROT*(YY-YH)+COSROT*(XX-XH)
      RECDATA 29
      YROT = COSROT*(YY-YV)-SINROT*(XX-XV)
      RECDATA 30
      IF (NPOINTS.LT.1500) GO TO 270
      RECDATA 31
      270 CALL BUFCHK(ITYPE, IPRINT)
      RECDATA 32
      KPEN = 3
      RECDATA 33
      270 IF (NPOINTS.GT.0) GO TO 280
      RECDATA 34
      J = 2 $ XLAST = XROT $ YLAST = YROT
      RECDATA 35
      TSTART = T(1) $ DT = T(2)-TSTART
      RECDATA 36
      XSTART = X(1) $ DX = X(2)-XSTART
      RECDATA 37
      YSTART = Y(1) $ DY = Y(2)-YSTART
      RECDATA 38
      40 280 IF ((XROT-XLAST).LT.XLEFT .OR. (XROT-XLAST).GT.XRIGHT) GO TO 260
      RECDATA 39
      IF (ABS(YROT-YLAST).GT.YCHANGE) GO TO 260
      RECDATA 40
      330 IF (XROT.LT.X(J)) GO TO 360
      RECDATA 41
      IF (J.GE.KOUNT) GO TO 360
      RECDATA 42
      J = J+1
      RECDATA 43
      TSTART = T(J-1) $ DT = T(J)-TSTART
      RECDATA 44
      XSTART = X(J-1) $ DX = X(J)-XSTART
      RECDATA 45
      YSTART = Y(J-1) $ DY = Y(J)-YSTART
      RECDATA 46
      50 GO TO 330
      RECDATA 47
      360 NPOINTS = NPOINTS+1
      RECDATA 48
      DELTA = (XROT-XSTART)/DX
      RECDATA 49
      TDATA(1, NPOINTS) = TSTART + DELTA*DT
      RECDATA 50
      YSHIFT = YROT-(YSTART+DELTA*DY)
      RECDATA 51
      TDATA(2, NPOINTS) = AO+A1*YSHIFT+RNUM
      RECDATA 52
      RECDATA 53
      RECDATA 54
      RECDATA 55
      RECDATA 56
      RECDATA 57
      RECDATA 58

```

```

        IF (IPL0T.EQ.1) CALL DIGPLOT(XROT,YROT,KPEN)
        IF (ISEIS.EQ.0) GO TO 370
60      ISEIS = 0 $ IF (ITYPE.NE.3) GO TO 370

        TVALUE = TSURVEY(TDATA(1,NPOINTS))

        PRINT 365, LIST(NSEIS),TVALUE
65      365 FORMAT(10X,10(1H-),10X,*CHANGE IN SEISMIC FLAG TO *,A10,4X,
1 *AT*,F15.7)
        IF (IPL0T.EQ.1) CALL DIGPLOT(0.0,LIST(NSEIS),7)
370     KPEN = 2 $ XLAST = XROT $ YLAST = YROT
70     XSAVE = AMAX1(XSAVE,XROT)
        GO TO (220,110) IFEND

110 RETURN

END

```

```

RECDATA 59
RECDATA 60
RECDATA 61
RECDATA 62
RECDATA 63
RECDATA 64
RECDATA 65
RECDATA 66
RECDATA 67
RECDATA 68
RECDATA 69
RECDATA 70
RECDATA 71
RECDATA 72
RECDATA 73
RECDATA 74
RECDATA 75

```

CARD NR. SEVERITY DETAILS DIAGNOSIS OF PROBLEM

```

22 1 AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
29 1 AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
70 1 AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.

```


1	SUBROUTINE BUFCHK(ITYPE,IPRINT)	BUFCHK	2
	C*** BUFCHK	BUFCHK	3
	C** THIS ROUTINE SAVES THE DATA IN THE /RECORD/ COMMON BLOCK	BUFCHK	4
5	C** ON THE SCRATCH FILE FOR PROCESSING THROUGH 'SAMPLE'.	BUFCHK	5
	C** A VARIETY OF MESSAGES ARE THEN PRINTED IF REQUIRED.	BUFCHK	6
	C** ITYPE - DATA TYPE, 1=OBSERVATORY, 2=SEISMIC, 3=STRIP CHART	BUFCHK	7
	C** IPRINT - PRINT FLAG, 0=SUPPRESS MESSAGE, 1=PRINT MESSAGE	BUFCHK	8
	C** TSAVE - THE TIME OF THE FURTHEST POINT REACHED TO THE RIGHT	BUFCHK	9
10	COMMON/HSCALE/ KOUNT,T(100),X(100),Y(100),XH,YH,ISWAP,TFIRST,TLAST	BUFCHK	10
	COMMON/RECORD/ NAME(3),TBEGIN,TEND,NPOINTS,TDATA(2,1500)	BUFCHK	11
	COMMON/LUN/ MTIN,MTOUT,MTSCR,LUNSCR,LUNDOC,LUNSAVE,LUNPLOT	BUFCHK	12
	COMMON/LIMITS/ XLEFT,XRIGHT,YCHANGE,WIDTH,TSAVE,XSAVE,XMAX,LEVEL	BUFCHK	13
15	TBEGIN = TSURVEY(TDATA(1,1))	BUFCHK	14
	TEND = TSURVEY(TDATA(1,NPOINTS))	BUFCHK	15
20	100 BUFFER OUT(MTOUT,1) (NAME(1),TDATA(2,NPOINTS))	BUFCHK	16
	IF(UNIT(MTOUT)) 130,110,110	BUFCHK	17
	110 PRINT 120, NAME,TBEGIN,TEND	BUFCHK	18
	120 FORMAT(/,10X,10(1H*),10X,*PARITY ERROR IN RECORD*,	BUFCHK	19
	1 1H*,3A10,1H*,5X,2F15.6)	BUFCHK	20
25	130 IF (ITYPE.EQ.3) GO TO 230	BUFCHK	21
	IF (TDATA(1,1).GT.TSAVE-1.5) GO TO 210	BUFCHK	22
	IF (IPRINT.EQ.1) PRINT 200	BUFCHK	23
30	200 FORMAT(/,10X,10(1H*),10X,*REVERSAL IN TIME SEQUENCE*)	BUFCHK	24
	210 IF (TDATA(1,1).GT.TSAVE+1.5) GO TO 230	BUFCHK	25
	IF (IPRINT.EQ.1) PRINT 220	BUFCHK	26
	220 FORMAT(/,10X,10(1H*),10X,*FORWARD JUMP OF LESS THAN TWO MINUTES*)	BUFCHK	27
35	230 LEVEL = LEVEL+1	BUFCHK	28
	PRINT 240, LEVEL,TBEGIN,TEND,NPOINTS	BUFCHK	29
	240 FORMAT(/,36X,*END OF LEVEL NUMBER*,15,5X,*STARTS *,F15.7,	BUFCHK	30
	1 5X,*ENDS *,F15.7,110,* VALUES*)	BUFCHK	31
40	IF (ISWAP.EQ.1) GO TO 300	BUFCHK	32
	TSAVE = AMAX1(TEND,TSAVE) \$ GO TO 310	BUFCHK	33
	300 TSAVE = AMIN1(TEND,TSAVE)	BUFCHK	34
	310 NPOINTS = 0 \$ RETURN	BUFCHK	35
	END	BUFCHK	36
		BUFCHK	37
		BUFCHK	38
		BUFCHK	39
		BUFCHK	40
		BUFCHK	41
		BUFCHK	42
		BUFCHK	43
		BUFCHK	44
		BUFCHK	45

```

1      SUBROUTINE NEWCHAR(NCHAR,NWORD,NC,NW,IFEND)
      C***      NEWCHAR
      DIMENSION NBITS(10),NWORD(8)
5      DATA (NBITS=-54,-48,-42,-36,-30,-24,-18,-12,-6,0),(MASK=77B)

      NC = NC+1
      IF (NC.GT.10) 10,30
10     NC = 1 $ NW = NW+1
      IF (NW.GT.8) 20,30

      20 NC = 10 $ NW = 8
      IFEND = 2 $ RETURN
15     30 NCHAR = SHIFT(NWORD(NW),NBITS(NC)).AND.MASK

      IFEND = 1 $ RETURN
20     END

```

```

NEWCHAR 2
NEWCHAR 3
NEWCHAR 4
NEWCHAR 5
NEWCHAR 6
NEWCHAR 7
NEWCHAR 8
NEWCHAR 9
NEWCHAR 10
NEWCHAR 11
NEWCHAR 12
NEWCHAR 13
NEWCHAR 14
NEWCHAR 15
NEWCHAR 16
NEWCHAR 17
NEWCHAR 18
NEWCHAR 19
NEWCHAR 20
NEWCHAR 21

```

1	SUBROUTINE XYCODE(X,Y,NWORD,IFERROR,IFEND)	XYCODE	2
	C*** XYCODE	XYCODE	3
	DIMENSION NWORD(8),LIST1(2),LIST2(2)	XYCODE	4
5	DATA (LIST1=1RA,1RX),(LIST2=1RB,1RY),(ISAVE=2)	XYCODE	5
	IFERROR = 1 \$ IFEND = ISAVE	XYCODE	6
	GO TO (80,10) IFEND	XYCODE	7
		XYCODE	8
10	10 NC = 0 \$ NW = IFEND = ISAVE = 1	XYCODE	9
	20 CALL NEWCHAR(NCHAR,NWORD,NC,NW,IFEND)	XYCODE	10
	GO TO (40,170) IFEND	XYCODE	11
	40 DO 50 NTYPE=1,2	XYCODE	12
	IF (NCHAR.EQ.LIST1(NTYPE)) 80,50	XYCODE	13
15	50 CONTINUE	XYCODE	14
	GO TO 20	XYCODE	15
		XYCODE	16
	80 CALL XYTABLE(X,NTYPE,NWORD,NC,NW,IFERROR)	XYCODE	17
	GO TO (90,110) IFERROR	XYCODE	18
20	90 CALL NEWCHAR(NCHAR,NWORD,NC,NW,IFEND)	XYCODE	19
	GO TO (290,170) IFEND	XYCODE	20
		XYCODE	21
	290 IF (NCHAR.EQ.LIST2(NTYPE)) 100,110	XYCODE	22
25	100 CALL XYTABLE(Y,NTYPE,NWORD,NC,NW,IFERROR)	XYCODE	23
	GO TO (120,110) IFERROR	XYCODE	24
		XYCODE	25
	110 X = Y = 0.0 \$ IFERROR = 2	XYCODE	26
		XYCODE	27
30	120 CALL NEWCHAR(NCHAR,NWORD,NC,NW,IFEND)	XYCODE	28
	GO TO (130,180) IFEND	XYCODE	29
	130 IF (NCHAR.EQ.LIST1(NTYPE)) 190,120	XYCODE	30
		XYCODE	31
	170 X = Y = 0.0 \$ IFERROR = 2	XYCODE	32
35	180 IFEND = ISAVE = 2	XYCODE	33
	190 RETURN	XYCODE	34
		XYCODE	35
	END	XYCODE	36
		XYCODE	37
		XYCODE	38
		XYCODE	39

CARD NR. SEVERITY DETAILS DIAGNOSIS OF PROBLEM

8	I	AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
12	I	AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
19	I	AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
22	I	AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
26	I	AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
31	I	AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.

1	SUBROUTINE XYTABLE(X, NTYPE, NWORD, NC, NW, IFERROR)	XYTABLE	2
	C*** XYTABLE	XYTABLE	3
	DIMENSION NWORD(8)	XYTABLE	4
5	IX = 0 \$ IFERROR = 1	XYTABLE	5
	GO TO (10,80) NTYPE	XYTABLE	6
		XYTABLE	7
		XYTABLE	8
		XYTABLE	9
10	10 DO 60 K=1,4	XYTABLE	10
	CALL NEWCHAR(NCHAR, NWORD, NC, NW, IFEND)	XYTABLE	11
	GO TO (40,70) IFEND	XYTABLE	12
	40 IF (NCHAR.GE.1R0.AND.NCHAR.LE.1R9) 50,70	XYTABLE	13
	50 IX = IX+10*(NCHAR-1R0)	XYTABLE	14
	60 CONTINUE	XYTABLE	15
15	X = IX*0.01 \$ RETURN	XYTABLE	16
		XYTABLE	17
	70 X = 0.0 \$ IFERROR = 2 \$ RETURN	XYTABLE	18
		XYTABLE	19
	80 DO 200 K=1,9	XYTABLE	20
20	CALL NEWCHAR(NCHAR, NWORD, NC, NW, IFEND)	XYTABLE	21
	GO TO (100,300) IFEND	XYTABLE	22
		XYTABLE	23
	100 GO TO (110,150,150,150,150,170,150,200) K	XYTABLE	24
		XYTABLE	25
25	110 IF (NCHAR.EQ.1R+) 120,130	XYTABLE	26
	120 SIGN = 1.0 \$ GO TO 200	XYTABLE	27
	130 IF (NCHAR.EQ.1R-) 140,300	XYTABLE	28
	140 SIGN = -1.0 \$ GO TO 200	XYTABLE	29
	150 IF (NCHAR.EQ.1R) NCHAR = 1R0	XYTABLE	30
30	IF (NCHAR.GE.1R0.AND.NCHAR.LE.1R9) 160,300	XYTABLE	31
	160 IX = IX+10*(NCHAR-1R0) \$ GO TO 200	XYTABLE	32
		XYTABLE	33
	170 IF (NCHAR.EQ.1R.) 200,300	XYTABLE	34
	200 CONTINUE	XYTABLE	35
35		XYTABLE	36
	X = SIGN*IX*0.001 \$ RETURN	XYTABLE	37
		XYTABLE	38
	300 X = 0.0 \$ IFERROR = 2 \$ RETURN	XYTABLE	39
	END	XYTABLE	40

CARD NR. SEVERITY DETAILS DIAGNOSIS OF PROBLEM

7	1		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
11	1		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
13	1	0	A HOLLERITH CONSTANT IS AN OPERAND OF AN ARITHMETIC OPERATOR.
21	1		AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
31	1	0	A HOLLERITH CONSTANT IS AN OPERAND OF AN ARITHMETIC OPERATOR.

1	SUBROUTINE EXTRACT(LCHARS, ICARD, ICHAR, JCARD, JCHAR)	EXTRACT 2
	C*** EXTRACT	EXTRACT 3
	DIMENSION ICARD(1), JCARD(1), NBITS(10)	EXTRACT 4
5	DATA(MASK=77B), (NBITS=-54, -48, -42, -36, -30, -24, -18, -12, -6, 0)	EXTRACT 5
	ICOUNT = 0	EXTRACT 6
	ITYPE = 1 \$ IF (ICAR.LT.JCHAR) ITYPE = 2	EXTRACT 7
	GO TO (10,210) ITYPE	EXTRACT 8
10	10 IW = JW = 1 \$ IC = ICHAR-1 \$ JC = JCHAR-1	EXTRACT 9
	IC=ICAR-1 \$ JC=JCHAR-1	EXTRACT 10
	20 ICOUNT = ICOUNT+1 \$ IF (ICOUNT.GT.LCHARS) 80,30	EXTRACT 11
15	30 IC = IC+1 \$ IF (IC.GT.10) 40,50	EXTRACT 12
	40 IC = 1 \$ IW = IW+1	EXTRACT 13
	50 JC = JC+1 \$ IF (JC.GT.10) 60,70	EXTRACT 14
	60 JC = 1 \$ JW = JW+1	EXTRACT 15
	GO TO 70	EXTRACT 16
20	210 ISTART = ICHAR+LCHARS-1 \$ JSTART = JCHAR+LCHARS-1	EXTRACT 17
	IW = (ISTART-1)/10+1 \$ IC = ISTART-(IW-1)*10+1	EXTRACT 18
	JW = (JSTART-1)/10+1 \$ JC = JSTART-(JW-1)*10+1	EXTRACT 19
	220 ICOUNT = ICOUNT+1 \$ IF (IC.GT.LCHARS) 80,230	EXTRACT 20
25	230 IC = IC-1 \$ IF (IC.LT.1) 240,250	EXTRACT 21
	240 IC = 10 \$ IW = IW-1	EXTRACT 22
	250 JC = JC-1 \$ IF (JC.LT.1) 260,70	EXTRACT 23
	260 JC = 10 \$ JW = JW-1	EXTRACT 24
30	70 NCHAR = SHIFT(ICARD(IW), NBITS(IC)).AND.MASK	EXTRACT 25
	MCHAR = SHIFT(NCHAR, -NBITS(JC))	EXTRACT 26
	NCLEAR = .NOT.SHIFT(MASK, -NBITS(JC))	EXTRACT 27
	JCARD(JW) = JCARD(JW).AND.NCLEAR	EXTRACT 28
	JCARD(JW) = JCARD(JW).OR.MCHAR	EXTRACT 29
35	GO TO (20,220) ITYPE	EXTRACT 30
	80 RETURN	EXTRACT 31
40	END	EXTRACT 32
		EXTRACT 33
		EXTRACT 34
		EXTRACT 35
		EXTRACT 36
		EXTRACT 37
		EXTRACT 38
		EXTRACT 39
		EXTRACT 40
		EXTRACT 41

CARD NR. SEVERITY DETAILS DIAGNOSIS OF PROBLEM

9	I	AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.
34	I	AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.

1	SUBROUTINE FPCODE(NWORD,IC,VALUE,IFLAG)	FPCODE	2
		FPCODE	3
	C*** FPCODE	FPCODE	4
	C** THIS ROUTINE DECODES UP TO 30 CHARACTERS AS A F.P. NUMBER	FPCODE	5
5	C** ANY ERRORS RETURN A VALUE OF 1.0E10	FPCODE	6
	C** NWORD,IC - WORD AND CHARACTER COUNT WHERE DECODING STARTS	FPCODE	7
	C** VALUE - F.P. NUMBER RETURNED TO CALLING ROUTINE	FPCODE	8
	C** IFLAG - ERROR FLAG, 0=OK, 1=ERROR	FPCODE	9
10	DIMENSION NBITS(10),NWORD(3)	FPCODE	10
	DATA (NBITS=-54,-48,-42,-36,-30,-24,-18,-12,-6,0),(MASK=77B)	FPCODE	11
	NC = IC-1 \$ NW = 1 \$ ISTART = 0	FPCODE	12
	NDEC = IDEC = INUM = 0 \$ ISIGN = +1	FPCODE	13
15		FPCODE	14
	10 NC = NC+1 \$ IF (NC.LE.10) GO TO 20	FPCODE	15
	NC = 1 \$ NW = NW+1 \$ IF (NW.GT.3) GO TO 80	FPCODE	16
	20 NCHAR = SHIFT(NWORD(NW),NBITS(NC)).AND.MASK	FPCODE	17
	IF (NCHAR.LT.1R0.OR.NCHAR.GT.1R9) GO TO 30	FPCODE	18
20	INUM = INUM*10+(NCHAR-1R0)	FPCODE	19
	IF (IDEC.EQ.1) NDEC = NDEC-1	FPCODE	20
	ISTART = 1 \$ GO TO 10	FPCODE	21
		FPCODE	22
	30 IF (ISTART.EQ.1) GO TO 40	FPCODE	23
25	IF (NCHAR.EQ.1R) GO TO 10	FPCODE	24
	IF (NCHAR.EQ.1R,) GO TO 80	FPCODE	25
	ISTART = 1	FPCODE	26
	IF (NCHAR.EQ.1R+) GO TO 10	FPCODE	27
	IF (NCHAR.NE.1R-) GO TO 35	FPCODE	28
30	ISIGN = -1 \$ GO TO 10	FPCODE	29
	35 IF (NCHAR.NE.1R.) GO TO 60	FPCODE	30
	IDEC = 1 \$ GO TO 10	FPCODE	31
		FPCODE	32
	40 IF (NCHAR.EQ.1R) GO TO 50	FPCODE	33
35	IF (NCHAR.EQ.1R,) GO TO 50	FPCODE	34
	IF (NCHAR.NE.1R.) GO TO 60	FPCODE	35
	IDEC = 1 \$ GO TO 10	FPCODE	36
		FPCODE	37
	50 VALUE = FLOAT(ISIGN*INUM)*10.0**NDEC	FPCODE	38
40	IFLAG = 0 \$ RETURN	FPCODE	39
		FPCODE	40
	60 VALUE = 1.0E10	FPCODE	41
	PRINT 70, NWORD	FPCODE	42
	70 FORMAT(/,10X,10(1H+),*ERROR IN DECODING F.P. NUMBER - *,3A10)	FPCODE	43
45	IFLAG = 1 \$ RETURN	FPCODE	44
		FPCODE	45
	80 VALUE = 0.0	FPCODE	46
	IFLAG = -1 \$ RETURN	FPCODE	47
		FPCODE	48
50	END	FPCODE	49
		FPCODE	50
		FPCODE	51

CARD NR. SEVERITY DETAILS DIAGNOSIS OF PROBLEM

20 I 0 A HOLLERITH CONSTANT IS AN OPERAND OF AN ARITHMETIC OPERATOR.

1981/06/05


```

1      PROGRAM SAMPLE(TAPE10,TAPE1,INPUT,TAPE60=INPUT,OUTPUT)
C***      SAMPLE
C**      THIS PROGRAM RESAMPLES THE OUTPUT FROM DIGDATA INTO REGULAR
5      C**      INTERVALS. THE PROGRAM IS PRESET FOR SEISMIC DATA.
C**      HORIZONS TO BE RESAMPLED, SAMPLING INTERVAL AND INTERPOLATION
C**      INTERVAL ARE SPECIFIED.
C*      OPTIONAL CARDS
10      C*      *STRIPCHART - FOR STRIPCHARTS
C*      *OBSERVATORY - FOR MAGNETOGRAMS
C*      LABEL - FILE LABEL OF DATA TO BE RESAMPLED
C*      NEXT - HORIZON TO BE EXTRACTED AND RESAMPLED
15      C*      TSTART - TIME OF LEFT HAND END OF DIGITISATION
C*      MUST BE AN INTEGRAL NUMBER OF THE SAMPLE INTERVAL
C*      TINT - SAMPLING INTERVAL : NEGATIVE IF TIME DECREASES TO
C*      THE RIGHT
C*      TGAP - MAXIMUM WIDTH OVER WHICH DATA IS INTERPOLATED
20      C*      M - OUTPUT DATA BLOCK SIZE IE.(2XM)
C*      VSHIFT - ORIGIN SHIFT (OPTIONAL) TO BE ADDED TO DATA VALUE
C*      WIDTH - STRIPCHART PAPER WIDTH
C*      DIMENSION LABEL(8),NEXT(2),ITEM(3)
25      COMMON/RECORD/ NAME(3),TBEGIN,TEND,NPOINTS,TDATA(2,1500)
COMMON/LUN/ MTIN,MTOUT,MTSCR,LUNSCR,LUNDOC,LUNSAVE,LUNPLOT
COMMON VALUE(100)
LEVEL 2, VALUE
DATA (ITEM=7HSEISMIC,10HSTRIPCHART,10HOBSERVATOR)
30      DATA (MTIN=1),(MTOUT=10)
C*      DETERMINE TYPE OF DIGITISATION
REWIND MTOUT
ITYPE = 1
35      READ 10, LABEL
10      FORMAT(&A10)
IF (LABEL(1).NE.10H*STRIPCHAR) GO TO 30
ITYPE = 2
40      C*      READ INPUT LABEL - IF EOF TERMINATE PROGRAM
30      READ 10, LABEL
IF (EOF(60)) 999,40
C*      READ SAMPLING PARAMETERS
45      40      READ 50,NEXT,TSTART,TINT,TGAP,M,VSHIFT,WIDTH
50      FORMAT(2A10,F20.7,2F10.1,I10,2F5.1)
IF (EOF(60)) 30,60
C*      PRINT SAMPLING INFORMATION
50      60      REWIND MTIN $ TIMEINT = SECOND(UU)
PRINT 70, TIMEINT,ITEM(ITYPE),NEXT,TSTART,TINT,TGAP,M,VSHIFT,WIDTH
70      FORMAT(1H1,/,/,10X,*EXTRACTION AND RESAMPLING OF RECORD DATA - COMM
1      ENCED AT*,F9.3,* SECS*,/,10X,69(1H-),/,/,20X,*INPUT DATA TYPE - *
2      A10,/,/,20X,*RECORDS TO BE EXTRACTED LABELLED *,1H*,2A10,1H*,
3      /,/,20X,*STARTING TIME OF OUTPUT FILE - *,F20.7,
55      4      /,/,20X,*DATA SAMPLING INTERVAL WILL BE - *,F6.1,* MINUTES*,
5      /,/,20X,*MAXIMUM INTERPOLATION INTERVAL - *,F6.1,* MINUTES*,

```

```

2      SAMPLE
3      SAMPLE
4      SAMPLE
5      SAMPLE
6      SAMPLE
7      SAMPLE
8      SAMPLE
9      SAMPLE
10     SAMPLE
11     SAMPLE
12     SAMPLE
13     SAMPLE
14     SAMPLE
15     SAMPLE
16     SAMPLE
17     SAMPLE
18     SAMPLE
19     SAMPLE
20     SAMPLE
21     SAMPLE
22     SAMPLE
23     SAMPLE
24     SAMPLE
25     SAMPLE
26     SAMPLE
27     SAMPLE
28     SAMPLE
29     SAMPLE
30     SAMPLE
31     SAMPLE
32     SAMPLE
33     SAMPLE
34     SAMPLE
35     SAMPLE
36     SAMPLE
37     SAMPLE
38     SAMPLE
39     SAMPLE
40     SAMPLE
41     SAMPLE
42     SAMPLE
43     SAMPLE
44     SAMPLE
45     SAMPLE
46     SAMPLE
MAR25A
1      SAMPLE
48     SAMPLE
49     SAMPLE
50     SAMPLE
51     SAMPLE
52     SAMPLE
53     SAMPLE
54     SAMPLE
MAR26A
1      SAMPLE
2      SAMPLE
57     SAMPLE
58     SAMPLE

```

60	6 ///	20X,*OUTPUT RECORDS WILL CONTAIN	- *,16,* SAMPLES*,	SAMPLE	59
	7 ///	20X,*ORIGIN SHIFT (IF APPLICABLE)	- *,F5.0,* DATA UNITS*,	MAR26A	3
	8 ///	20X,*CHART WIDTH (IF APPLICABLE)	- *,F5.0,* DATA UNITS*)	MAR26A	4
		CALL LOOKUP(LABEL,MTIN,1,1,NFLAG)		SAMPLE	62
		IF (NFLAG.EQ.1) CALL ABORT		SAMPLE	63
65	C*	BUFFER IN A BLOCK OF DATA FROM THE DIGDATA FILE		SAMPLE	64
		KOUNT = 0 \$ TVALUE = TMIN(TSTART)		SAMPLE	65
		IF (TINT.LT.0.0) TVALUE = TVALUE+TINT		SAMPLE	66
	80	BUFFER IN(MTIN,1) (NAME(1),TDATA(2,1500))		SAMPLE	67
		IF (UNIT(MTIN)) 110,160,90		SAMPLE	68
70	90	PRINT 100, NAME,TBEGIN,TEND		SAMPLE	69
	100	FORMAT(/,10X,10(1H*),10X,*PARITY ERROR IN - *,3A10,2F15.7)		SAMPLE	70
				MAR24AA	71
				SAMPLE	8
	C*	EXTRACT CORRECT HORIZON		SAMPLE	73
75	110	DO 120 K=1,2		SAMPLE	74
		IF (NAME(K).NE.NEXT(K)) GO TO 80		SAMPLE	75
	120	CONTINUE		SAMPLE	76
				SAMPLE	77
	C*	JUMP TO APPROPRIATE ROUTINE TO RESAMPLE DATA AND STORE IN		SAMPLE	78
	C*	L.C.M. TO AWAIT FURTHER DATA		SAMPLE	79
80		GO TO (130,140) ITYPE		SAMPLE	80
	130	CALL INFILL(VALUE,KOUNT,TVALUE,TINT,TGAP,VSHIFT)		SAMPLE	81
		GO TO 80		SAMPLE	82
	140	CALL JUMPER(VALUE,KOUNT,TVALUE,TINT,TGAP,VSHIFT,WIDTH)		SAMPLE	83
		GO TO 80		SAMPLE	84
85				SAMPLE	85
	C*	AT EOF - FILL LAST BLOCK WITH UNKNOWN DATA		SAMPLE	86
	160	IF ((KOUNT/M)*M.EQ.KOUNT) GO TO 180		SAMPLE	87
		NCOUNT = (KOUNT/M+1)*M		SAMPLE	88
	170	KOUNT = KOUNT+1		SAMPLE	89
90		VALUE(KOUNT) = 1.0E10		SAMPLE	90
		IF (KOUNT.LT.NCOUNT) GO TO 170		SAMPLE	91
				SAMPLE	92
	C*	EXTENDED L.C.M., ASSIGN LABEL TO RESAMPLED DATA, THEN CHOP DAT		SAMPLE	93
	C*	INTO BLOCK SIZES REQUIRED		SAMPLE	94
95	180	CALL SETMEM(KOUNT+100,2)		SAMPLE	95
		CALL LABNAME(LABEL,NEXT)		SAMPLE	96
		CALL CHOPPER(VALUE,KOUNT,TSTART,TINT,TDATA,M)		SAMPLE	97
		GO TO 40		SAMPLE	98
				SAMPLE	99
100	999	TIMEINT = SECOND(UU) \$ PRINT 1000, TIMEINT		SAMPLE	100
	1000	FORMAT(///,10X,10(1H\$),10X,*SAMPLE COMPLETED AT *,F9.3,* SECS*)		SAMPLE	101
				SAMPLE	102
				SAMPLE	103
		END		SAMPLE	104

CARD NR. SEVERITY DETAILS DIAGNOSIS OF PROBLEM

80 I AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRANCH COMPUTED GO TO STATEMENT.

1	SUBROUTINE LABNAME(LABEL,NAME)	LABNAME	2
		LABNAME	3
	C*** LABNAME	LABNAME	4
	C** THIS SUBROUTINE AUTOMATICALLY ASSIGNS FILE LABELS FOR THE	LABNAME	5
5	C** OUTPUT DATA.	LABNAME	6
	C** LABEL IS OF FORM : HORIZON LABEL - INPUT LABEL	LABNAME	7
		LABNAME	8
	C* LABEL - INPUT LABEL	LABNAME	9
	C* NAME - HORIZON LABEL	LABNAME	10
10		LABNAME	11
	DIMENSION LABEL(8),NAME(2),ICARD(8),LCHAR(70),NCHAR(23)	LABNAME	12
	COMMON/LUN/ MTIN,MTOUT,MTSCR,LUNSCR,LUNDOC,LUNSAVE,LUNPLOT	LABNAME	13
		LABNAME	14
	DECODE(70,10,LABEL(1)) LCHAR	LABNAME	15
15	10 FORMAT(70R1)	LABNAME	16
		LABNAME	17
	DECODE(20,20,NAME(1)) (NCHAR(K),K=1,20)	LABNAME	18
	20 FORMAT(20R1)	LABNAME	19
		LABNAME	20
20	DO 30 K=1,20	LABNAME	21
	K = 21-KK	LABNAME	22
	IF (NCHAR(K).NE.1R) GO TO 100	LABNAME	23
	30 CONTINUE	LABNAME	24
	PRINT 40	LABNAME	25
25	40 FORMAT(/,10X,10(1H-),10X,*BLANK RECORD NAME DETECTED*)	LABNAME	26
	CALL ABORT	LABNAME	27
		LABNAME	28
	100 NCHAR(K+1) = NCHAR(K+3) = 1R	LABNAME	29
	NCHAR(K+2) = 1R-	LABNAME	30
30	NSTOP = K+3 \$ LSTOP = 70-NSTOP	LABNAME	31
	ENCODE(70,10,ICARD(1)) (NCHAR(K),K=1,NSTOP),(LCHAR(K),K=1,LSTOP)	LABNAME	32
	CALL TODAY(ICARD(8))	LABNAME	33
		LABNAME	34
35	BACKOUT(MTOUT,1) (ICARD(1),ICARD(8))	LABNAME	35
	IF (UMT(MTOUT)) 130,110,110	LABNAME	36
	110 PRINT /20, ICARD	LABNAME	37
	120 FORMAT(/,10X,10(1H*),10X,*PARITY IN LABEL *,1H*,8A10,1H*)	LABNAME	38
		LABNAME	39
40	130 PRINT /40, ICARD	LABNAME	40
	140 FORMAT(/,10X,*OUTPUT OF FILE LABELLED *,1H*,8A10,1H*)	LABNAME	41
		LABNAME	42
	RETURN	LABNAME	43
		LABNAME	44
45	END	LABNAME	45
		LABNAME	46

1		SUBROUTINE JUMPFIX(KOUNT,WIDTH)	JUMPFIX	2
	C***	JUMPFIX	JUMPFIX	3
	C**	FIXES ERRORS IN STRIPCHART WIDTH	JUMPFIX	4
	C**	KOUNT - DATA BLOCK NO	JUMPFIX	5
5	C**	WIDTH - STRIPCHART WIDTH	JUMPFIX	6
	C**		JUMPFIX	7
		COMMON/RECORD/NAME(3),TBEGIN, IEND,NPOINTS,DATA(2,1500)	JUMPFIX	8
		DATA(UNKNOWN=1.0E10),(DUBIOUS=1.0E9),(IFLAG=0)	JUMPFIX	9
10		IF(KOUNT.NE.0) GO TO 5	JUMPFIX	10
		BAD=WIDTH*0.5	JUMPFIX	11
		VNEXT=DATA(2,1)	JUMPFIX	12
			JUMPFIX	13
15	5	DO 100 J=1,NPOINTS	JUMPFIX	14
		IF(DATA(2,J).GT.DUBIOUS)GO TO 100	JUMPFIX	15
		TEST=VNEXT-DATA(2,J)	JUMPFIX	16
			JUMPFIX	17
		IF(ABS(TEST).LT.BAD)GO TO 90	JUMPFIX	18
20		IFLAG=IFLAG+1	JUMPFIX	19
		IF(TBEGIN.GT.1000) GO TO 12	JUMPFIX	20
		TSHIFT=TSURVEY(DATA(1,J))	JUMPFIX	21
		GO TO 15	JUMPFIX	22
			JUMPFIX	23
25	12	TSHIFT=DATA(1,J)*1.0E-4	JUMPFIX	24
	15	IF(TEST) 30,10,50	JUMPFIX	25
			JUMPFIX	26
	10	IF(IFLAG.EQ.1) PRINT 20,TSHIFT,VNEXT,DATA(2,J)	JUMPFIX	27
30	20	FORMAT(/,10X,10(1H-),10X,*IMPOSSIBLE CONDITION AT TIME = *,	JUMPFIX	28
		F10.7,5X,*VALUES CONCERNED = *,F8.2,2X,F8.2)	JUMPFIX	29
		GO TO 95	JUMPFIX	30
			JUMPFIX	31
	30	IF(IFLAG.EQ.1) PRINT 40,TSHIFT,VNEXT,DATA(2,J)	JUMPFIX	32
35	40	FORMAT(/,10X,10(1H-),10X,*SHIFT LEVEL DOWN AT TIME = *,F10.7,5X,	JUMPFIX	33
		*VALUES CONCERNED = *,F8.2,2X,F8.2)	JUMPFIX	34
		DATA(2,J)=DATA(2,J)-WIDTH	JUMPFIX	35
		GO TO 95	JUMPFIX	36
			JUMPFIX	37
	50	IF(IFLAG.EQ.1) PRINT 60,TSHIFT,VNEXT,DATA(2,J)	JUMPFIX	38
40	60	FORMAT(/,10X,10(1H-),10X,*SHIFT LEVEL UP AT TIME = *,F10.7,5X,	JUMPFIX	39
		*VALUES CONCERNED = *,F8.2,2X,F8.2)	JUMPFIX	40
		DATA(2,J)=DATA(2,J)+WIDTH	JUMPFIX	41
		GO TO 95	JUMPFIX	42
			JUMPFIX	43
45	90	IF(IFLAG.NE.0) PRINT 110,TSHIFT,VNEXT	JUMPFIX	44
	110	FORMAT(/,30X,*LEVEL SHIFT COMPLETED AT TIME = *,	JUMPFIX	45
		F10.7,5X,*FINAL VALUE = *,F8.2)	JUMPFIX	46
		IFLAG=0	JUMPFIX	47
	95	VNEXT=DATA(2,J)	JUMPFIX	48
50	100	CONTINUE	JUMPFIX	49
		RETURN	JUMPFIX	50
		END	JUMPFIX	51
			JUMPFIX	52
			JUMPFIX	53

1	SUBROUTINE JUMPER(VALUE,KOUNT,TVALUE,TOELTA,TGAP,VSHIFT,WIDTH)	JUMPER	2
	C*** JUMPER	JUMPER	3
	C** THIS SUBROUTINE RESAMPLES DIGITISED STRIPCHART DATA	JUMPER	4
5		JUMPER	5
	C* VALUE - ARRAY CONTAINING HORIZON DATA TO BE RESAMPLED	JUMPER	6
	C* KOUNT - DATA BLOCK NUMBER	JUMPER	7
	C* TVALUE - TIME FOR WHICH A RESAMPLED VALUE IS REQUIRED	JUMPER	8
	C* TDELTA - SAMPLING INTERVAL	JUMPER	9
10	C* TGAP - MAXIMUM INTERPOLATION INTERVAL IN MINUTES	JUMPER	10
	C* VSHIFT - ORIGIN SHIFT (OPTIONAL) TO BE ADDED TO DATA VALUES	JUMPER	11
	C* WIDTH - STRIPCHART PAPER WIDTH	JUMPER	12
		JUMPER	13
	DIMENSION VALUE(100)	JUMPER	14
15	COMMON/RECORD/ NAME(3),TBEGIN,TEND,NPOINTS,TDATA(2,1500)	JUMPER	15
	LEVEL 2, VALUE	JUMPER	16
	DATA (UNKNOWN=1.0E10)	JUMPER	17
		JUMPER	18
	IF (KOUNT.GT.0) GO TO 330	JUMPER	19
20	TLAST = TVALUE	JUMPER	20
	VLAST = TDATA(2,1)	JUMPER	21
	NMAX = 0	JUMPER	22
		JUMPER	23
25	330 CALL JUMPFIX(KOUNT,WIDTH)	JUMPER	24
		JUMPER	25
	DO 400 J=1,NPOINTS	JUMPER	26
	TNEXT = TDATA(1,J) \$ VNEXT = TDATA(2,J)	JUMPER	27
	IF (ABS(TNEXT-TLAST).LT.TGAP) GO TO 370	JUMPER	28
		JUMPER	29
30	C* RESAMPLING LOGIC FOR LARGE GAPS IE DATA IS UNKNOWN	JUMPER	30
	IF (TBEGIN.GT.1000.0) GO TO 335	JUMPER	31
	TSTART = TSURVEY(TLAST) \$ TSTOP = TSURVEY(TNEXT)	JUMPER	32
	GO TO 337	JUMPER	33
	335 TSTART = (TLAST+1.0E-4) \$ TSTOP = TDATA(1,J)+1.0E-4	JUMPER	34
35	337 PRINT 340, TSTART,TSTOP	JUMPER	35
	340 FORMAT(/,10X,10(1H-),10X,'DATA GAP BETWEEN *,F15.6,* AND *,F15.6)	JUMPER	36
	350 IF (TVALUE.GT.TNEXT) GO TO 390	JUMPER	37
	KOUNT = KOUNT+1 \$ IF (KOUNT.LT.NMAX) GO TO 360	JUMPER	38
	NMAX = NMAX+360 \$ CALL SETMEM(NMAX+100,2)	JUMPER	39
40	360 VALUE(KOUNT) = UNKNOWN \$ TVALUE = TVALUE+TDELTA	JUMPER	40
	GO TO 350	JUMPER	41
		JUMPER	42
	C* NORMAL RESAMPLING LOGIC	JUMPER	43
	370 IF (TVALUE.GT.TNEXT) GO TO 390	JUMPER	44
45	KOUNT = KOUNT+1 \$ IF (KOUNT.LT.NMAX) GO TO 380	JUMPER	45
	NMAX = NMAX+360 \$ CALL SETMEM(NMAX+100,2)	JUMPER	46
	380 DT = TNEXT-TLAST \$ IF (ABS(DT).LT.1.0E-8) DT = 1.0E-8+TDELTA	JUMPER	47
	FRACTN = (TVALUE-TLAST)/DT	JUMPER	48
	VALUE(KOUNT) = VLAST+FRACTN*(VNEXT-VLAST)+VSHIFT	JUMPER	49
50	TVALUE = TVALUE+TDELTA \$ GO TO 370	JUMPER	50
		JUMPER	51
	390 TLAST = TNEXT \$ VLAST = VNEXT	JUMPER	52
	400 CONTINUE	JUMPER	53
		JUMPER	54
55	RETURN	JUMPER	55
		JUMPER	56
	END	JUMPER	57
		JUMPER	58

1	SUBROUTINE CHOPPER(VALUE,KOUNT,TSTART,TINT,STORE,M)	CHOPPER	2
		CHOPPER	3
	C*** CHOPPER	CHOPPER	4
5	C** THIS SUBROUTINE CHOPS THE RESAMPLED DATA INTO THE BLOCK SIZE	CHOPPER	5
	C** SPECIFIED BY THE USER	CHOPPER	6
		CHOPPER	7
	C* VALUE - ARRAY CONTAINING HORIZON DATA TO BE RESAMPLED	CHOPPER	8
	C* KOUNT - DATA BLOCK NUMBER	CHOPPER	9
	C* TSTART - TIME OF START OF RESAMPLING	CHOPPER	10
10	C* TINT - SAMPLING INTERVAL	CHOPPER	11
	C* STORE - OUTPUT ARRAY FOR RESAMPLED DATA	CHOPPER	12
	C* M - OUTPUT DATA BLOCK SIZE IE. (ZXM)	CHOPPER	13
		CHOPPER	14
15	DIMENSION VALUE(100),STORE(2,M)	CHOPPER	15
	COMMON/LUN/ MTIN,MTOUT,MTSCR,LUNSCR,LUNDOC,LUNSAVE,LUNPLOT	CHOPPER	16
	LEVEL 2, VALUE	CHOPPER	17
		CHOPPER	18
	TVALUE = TMIN(TSTART)	CHOPPER	19
	IF (TINT.LT.0.0) GO TO 100	CHOPPER	20
20	N = 1 \$ NSTEP = +1 \$ GO TO 110	CHOPPER	21
	100 N = KOUNT \$ NSTEP = -1	CHOPPER	22
	TVALUE = TVALUE+KOUNT*TINT	CHOPPER	23
	110 TDELTA = ABS(TINT) \$ ISTART = 0	CHOPPER	24
		CHOPPER	25
25	DO 320 K=1,KOUNT,M	CHOPPER	26
		CHOPPER	27
	DO 220 J=1,M	CHOPPER	28
	TNEXT = TSURVEY(TVALUE)	APR02	1
		CHOPPER	32
30	210 STORE(1,J) = TNEXT	CHOPPER	33
	STORE(2,J) = VALUE(N)	CHOPPER	34
		CHOPPER	35
	N = N+NSTEP \$ TVALUE = TVALUE+TDELTA	CHOPPER	36
	IF (ISTART.EQ.1) GO TO 220	CHOPPER	37
35	TLAST = TNEXT \$ ISTART = 1	CHOPPER	38
	220 CONTINUE	CHOPPER	39
		CHOPPER	40
	PRINT 500,STORE(1,1),STORE(2,1)	CHOPPER	41
40	500 FORMAT(10X,F20.7,F10.2)	MAR24AA	1
		CHOPPER	43
	BUFFER OUT(MTOUT,1) (STORE(1,1),STORE(2,M))	CHOPPER	44
	IF (UNIT(MTOUT)) 320,300,300	CHOPPER	45
	300 PRINT 310,STORE(1,1)	CHOPPER	46
	310 FORMAT(/,10X,10(1H*),10X,*PARITY ERROR IN RECORD STARTING *,F15.7)	MAR24AA	2
45		CHOPPER	48
	320 CONTINUE	CHOPPER	49
		CHOPPER	50
	ENDFILE MTOUT	CHOPPER	51
	PRINT 400,TLAST,TNEXT	CHOPPER	52
50	400 FORMAT(/,20X,*DATA INTERVAL SAMPLED IS *,F15.7,* TO*,F15.7)	MAR24AA	3
		CHOPPER	54
	RETURN	CHOPPER	55
		CHOPPER	56
	END	CHOPPER	57

```

1      SUBROUTINE INFILL(VALUE,KOUNT,TVALUE,TDELTA,TGAP,VSHIFT)
C***  INFILL
C**   THIS SUBROUTINE RESAMPLES DIGITISED SEISMIC HORIZONS
5  C**   NOTE - FLAGS ARE EXTRACTED FIRST. DATA RESAMPLED AND THEN
C**   FLAGS ARE ADDED BACK ONTO DATA
C*    VALUE - ARRAY CONTAINING HORIZON DATA TO BE RESAMPLED
C*    KOUNT - DATA BLOCK NUMBER
10 C*    TVALUE - TIME FOR WHICH A RESAMPLED VALUE IS REQUIRED
C*    TDELTA - SAMPLING INTERVAL
C*    TGAP - MAXIMUM INTERPOLATION INTERVAL IN MINUTES
C*    VSHIFT - ORIGIN SHIFT (OPTIONAL) TO BE ADDED TO DATA VALUES
15  DIMENSION VALUE(100)
COMMON/RECORD/ NAME(3),TBEGIN,TEND,NPOINTS,TDATA(2,1500)
LEVEL 2, VALUE
DATA (UNKNOWN=1.0E10)
20  IF (KOUNT.GT.0) GO TO 10
C*    PRESET INITIAL PARAMETERS TO ALLOW AUTOMATIC INTERPOLATION
C*    AT START
NMAX = 0 $ ISET = 1
25  TLAST = TVALUE $ VLAST = AMOD(TDATA(2,1),1.0E5)
IFLAG = TDATA(2,1)+1.0E-5 $ VFLAG = IFLAG+1.0E5
IF (ABS(TNEXT-TLAST).LT.TGAP) GO TO 10
IFLAG = 100000 $ VFLAG = UNKNOWN
30  DO 300 J=1,NPOINTS
TNEXT = TDATA(1,J) $ VNEXT = AMOD(TDATA(2,J),1.0E5)
IF (ABS(TNEXT-TLAST).LT.TGAP) GO TO 200
TSTART = TSURVEY(TLAST) $ TSTOP = TSURVEY(TNEXT)
60 PRINT 100, TSTART,TSTOP
35  100 FORMAT(/,10X,10(1H-),10X,*DATA GAP BETWEEN *,F15.7,* AND *,F15.7)
C*    RESAMPLING LOGIC FOR LARGE GAPS IE DATA IS UNKNOWN
110 IF (TDELTA) 120,130,130
120 IF (TVALUE.GT.TNEXT) 140,290
40 130 IF (TVALUE.LT.TNEXT) 140,290
140 KOUNT = KOUNT+1 $ IF (KOUNT.LT.NMAX) GO TO 150
NMAX = NMAX+360 $ CALL SETMEM(NMAX+100,2)
150 PRESET = UNKNOWN $ IF (ABS(IFLAG-6).LT.2) PRESET = VFLAG
45  VALUE(KOUNT) = PRESET $ TVALUE = TVALUE+TDELTA
ISET = 1 $ GO TO 110
C*    NORMAL RESAMPLING LOGIC
200 IF (TDELTA) 210,220,220
210 IF (TVALUE.GT.TNEXT) 230,290
50 220 IF (TVALUE.LT.TNEXT) 230,290
230 KOUNT = KOUNT+1 $ IF (KOUNT.LT.NMAX) GO TO 240
NMAX = NMAX+360 $ CALL SETMEM(NMAX+100,2)
240 DT = TNEXT-TLAST $ IF (ABS(DT).LT.1.0E-8) DT = 1.0E-8*TDELTA
55  FRACTN = (TVALUE-TLAST)/DT
VALUE(KOUNT) = VLAST+FRACTN*(VNEXT-VLAST)+VFLAG+VSHIFT
TVALUE = TVALUE+TDELTA
ISET = 1 $ GO TO 200

```

```

2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
35
38
4
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61

```

```
60      290 TLAST = TNEXT $ VLAST = VNEXT
        IF (ISET.EQ.0) GO TO 300
        ISET = 0
        IFLAG = TDATA(2,J)*1.0E-5 $ VFLAG = IFLAG*1.0E5
        300 CONTINUE
65      RETURN
        END
```

```
INFILL 62
INFILL 63
INFILL 64
INFILL 65
INFILL 66
INFILL 67
INFILL 68
INFILL 69
INFILL 70
INFILL 71
```

1981/06/05

1	PROGRAM SEGPACK(TAPE10,TAPE1,TAPE20,TAPE21,INPUT,TAPE60=INPUT, 1 OUTPUT)	SEGPACK 2
		SEGPACK 3
		SEGPACK 4
	C*** SEGPACK	SEGPACK 5
5	C** THIS PROGRAM PRODUCES A SEISMIC DATA FILE OF STATION NUMBER	SEGPACK 6
	C** VERSES TWO-WAY TIMES IN STANDARD PHASE3 FORMAT.	SEGPACK 7
	C** HORIZONS ARE EXTRACTED, ASSIGNED A USER SPECIFIED HORIZON	SEGPACK 8
	C** NUMBER AND THEN PACKED IN NUMBER ORDER	SEGPACK 9
		SEGPACK 10
10	C* LABEL - OUTPUT FILE LABEL	SEGPACK 11
	C* M - BLOCK SIZE OF RESAMPLED DATA	SEGPACK 12
	C* NAME - HORIZON LABEL	SEGPACK 13
	C* NHOR - HORIZON NUMBER	SEGPACK 14
		SEGPACK 15
15	DIMENSION NAME(8),MSINDEX(3000)	SEGPACK 16
	COMMON/STORE(100)	SEGPACK 17
	COMMON/INDEX/ LABEL(8),NSEGS,NSUM,ERRSUM(11),	SEGPACK 18
	1 INDEX(8,200),STNSUM(200),ST(3,200),NERROR(11,200)	SEGPACK 19
	COMMON/SEGMENT/SEGMENT(8),NSTNS,T(3),ERROR(11),STNDATA(12,1440)	SEGPACK 20
20	COMMON/SEGPACK/ NLIST,LIST(300),ITEM(300),TIME(300),TEND(300)	SEGPACK 21
	COMMON/LUN/ MTIN,MTOUT,MTSCR,LUNSCR,LUNDOC,LUNSAVE,LUNPLOT	SEGPACK 22
	DATA (MTIN=1),(MTOUT=10),(MTSCR=20),(LUNSCR=21),(UNKNOWN=1.0E10)	SEGPACK 23
		SEGPACK 24
	READ 10, LABEL,M	SEGPACK 25
25	10 FORMAT(8A10,/,I10)	SEGPACK 26
	TIMEINT = SECOND(UU) \$ PRINT 30, TIMEINT	SEGPACK 27
	30 FORMAT(1H1,/,10X,*TRANSFER OF SEISMIC HORIZON FILES TO MASS STORAG	SEGPACK 28
	1E - COMMENCED AT*,F9.3,* SECS*,/,10X,81(1H-),//)	SEGPACK 29
		SEGPACK 30
30	C* OPEN MASS STORAGE SCRATCH REGION	SEGPACK 31
	CALL OPENMS(LUNSCR,MSINDEX,3000,0)	SEGPACK 32
		SEGPACK 33
	C* READ HORIZON NAME AND NUMBER - LOCATE AND PRINT MESSAGE	SEGPACK 34
	L=2 \$ LM=L*M	SEGPACK 35
35	CALL SETMEM(LOC(STORE(1))+LM+100,1)	SEGPACK 36
	NRECORD = NLIST = 0	SEGPACK 37
	100 READ 10, NAME,NHOR	SEGPACK 38
	IF (EOF(60)) 300,120	SEGPACK 39
	120 CALL LOOKUP(NAME,MTIN,1,1,NFLAG)	SEGPACK 40
40	IF (NFLAG.EQ.1) CALL ABORT	SEGPACK 41
		SEGPACK 42
	NLIST = NLIST+1	SEGPACK 43
	LIST(NLIST) = NHOR	SEGPACK 44
	ITEM(NLIST) = NRECORD+1	SEGPACK 45
		SEGPACK 46
45	C* BUFFER IN DATA - WRITE TO MASS STORAGE	SEGPACK 47
	IFEOF = KOUNT = 0	SEGPACK 48
	200 KOUNT = KOUNT+1 \$ NRECORD = NRECORD+1	SEGPACK 49
	BUFFER IN(MTIN,1) (STORE(1),STORE(LM))	SEGPACK 50
50	IF (UNIT(MTIN)) 240,230,210	SEGPACK 51
	210 PRINT 220, KOUNT,STORE(1) \$ GO TO 240	SEGPACK 52
	220 FORMAT(/,10X,10(1H*),10X,*PARITY ERROR IN BLOCK*,16,	SEGPACK 53
	1 * STARTING AT TIME *,F13.6)	SEGPACK 54
	230 IFEOF = 1 \$ STORE(1) = UNKNOWN	SEGPACK 55
		SEGPACK 56
55	240 CALL WRITMS(LUNSCR,STORE(1),2*M,NRECORD)	SEGPACK 57
	IF (KOUNT.EQ.1) TIME(NLIST) = STORE(1)	SEGPACK 58

	TEND(NLIST) = STORE(LM-1)	SEGPACK	59
	IF(1FE0F.EQ.0) GO TO 200	SEGPACK	60
60		SEGPACK	61
	PRINT 250, NHOR,KOUNT	SEGPACK	62
250	FORMAT(/,20X,*SEISMIC HORIZON CODE IS*,16,* - NUMBER OF RECO	SEGPACK	63
	1RDS SAVED*,16,/,10X,80(1H-))	SEGPACK	64
	GO TO 100	SEGPACK	65
65		SEGPACK	66
	300 PRINT 310, NRECORD	SEGPACK	67
	310 FORMAT(/,10X,10(1H\$),10X,*TOTAL NUMBER OF BLOCKS SAVED*,16)	SEGPACK	68
		SEGPACK	69
70	C* CALL SEGSEIS TO SORT AND PACK DATA	SEGPACK	70
	CALL SEGSEIS(STORE,M)	SEGPACK	71
		SEGPACK	72
	TIMEINT = SECOND(UU) \$ PRINT 1000, TIMEINT	SEGPACK	73
1000	FORMAT(/,10X,10(1H\$),10X,*END OF SEGPACK AT *,F9.3,* SECS*)	SEGPACK	74
75	END	SEGPACK	75
		SEGPACK	76


```

1      SUBROUTINE SEGSEIS(STORE,M)
C***   SEGSEIS
C**    THIS ROUTINE PACKS THE SEVERAL SEISMIC HORIZON FILES KEPT ON
5      C**    MASS STORAGE INTO A MULTI-CHANNEL OUTPUT FILE IN SEQUENTIAL
C**    TIME ORDER. THE NEXT RECORD FROM EACH FILE IS INSPECTED TO
C**    FIND THAT WITH THE EARLIEST STARTING TIME. SHOULD MORE THAN
C**    ONE FILE HAVE A RECORD WITH THE SAME START TIME, THE
C**    INFORMATION IS MERGED
10     C**    STORE - ARRAY FOR WORK SPACE
C**    M - BLOCK SIZE OF RESAMPLED DATA
15     DIMENSION STORE(2,M),DAY(3)
COMMON/SEGPACK/ NLIST,LIST(300),ITEM(300),TIME(300),TEND(300)
COMMON/LUN/ MTIN,MTOUT,MTSCR,LUNSCR,LUNDOC,LUNSAVE,LUNPLOT
COMMON/INDEX/ LABEL(8),NSEGS,NSUM,ERRSUM(11),
20     1 INDEX(8,200),STNSUM(200),ST(3,200),NERROR(11,200)
COMMON/SEGMENT/SEGMENT(8),NSTNS,T(3),ERROR(11),STNDATA(12,1440)
DATA (DUBIOUS=1.0E9),(FIVESEC=5.0E-8),(DAY=100.0,1000.0,1.0)
C*     CALL SORTER TO SORT HORIZONS INTO NUMBER ORDER
CALL ARRANGE(LIST,300,4,1,1,NLIST)
25     C*     FIND EARLIEST RECORD FOR STARTING UP PROCEDURE
CALL TODAY(LABEL(8))
PRINT 20,LABEL
20     FORMAT(1H1,/,10X,*PACKING OF SEISMIC DATA INTO SURVEY DATA FILE*,
30     1 /,10X,55(1H-),//,10X,*OUTPUT FILE LABELLED *,1H*,8A10,1H*)
TMIN = DUBIOUS
DO 400 K=1,NLIST
CALL READMS(LUNSCR,TIME(K),1,ITEM(K))
TMIN = AMIN1(TMIN,TIME(K))
PRINT 390, K,LIST(K),TIME(K),TEND(K)
35     390 FORMAT(/,10X,2I10,2(5X,F13.7) )
400 CONTINUE
C*     SET TYPE FLAG FOR CORRECT MULTIPLIER TO OBTAIN DAYS
ITYPE = 1
40     IF (TMIN.GT.20.0) ITYPE = 2
IF (TMIN.GT.1000.0) ITYPE = 3
LASTDAY = TMIN*DAY(ITYPE) + 0.00005
C*     SEARCH REMAINING RECORDS FOR THAT WITH EARLIEST START TIME
45     C*     IF NO VALID TIMES - ALL RECORDS HAVE THEN BEEN PROCESSED
410 KMIN = 0 $ TMIN = DUBIOUS
DO 420 K=1,NLIST
IF (TIME(K).GT.TMIN-FIVESEC) GO TO 420
KMIN = K $ TMIN = TIME(K)
50     420 CONTINUE
IF (KMIN.EQ.0) GO TO 600
C*     IF START OF NEWDAY,OUTPUT PREVIOUS DAY'S DATA
55     NEWDAY = TIME(KMIN)*DAY(ITYPE) + 0.00005
IF (NEWDAY.NE.LASTDAY) GO TO 500
IF (NSTNS+M.LE.1440) GO TO 520

```

```

SEGSEIS 2
SEGSEIS 3
SEGSEIS 4
SEGSEIS 5
SEGSEIS 6
SEGSEIS 7
SEGSEIS 8
SEGSEIS 9
SEGSEIS 10
SEGSEIS 11
SEGSEIS 12
SEGSEIS 13
SEGSEIS 14
SEGSEIS 15
SEGSEIS 16
SEGSEIS 17
SEGSEIS 18
SEGSEIS 19
SEGSEIS 20
SEGSEIS 21
SEGSEIS 22
SEGSEIS 23
SEGSEIS 24
SEGSEIS 25
SEGSEIS 26
SEGSEIS 27
SEGSEIS 28
SEGSEIS 29
SEGSEIS 30
SEGSEIS 31
SEGSEIS 32
SEGSEIS 33
SEGSEIS 34
SEGSEIS 35
SEGSEIS 36
SEGSEIS 37
SEGSEIS 38
SEGSEIS 39
SEGSEIS 40
SEGSEIS 41
SEGSEIS 42
SEGSEIS 43
SEGSEIS 44
SEGSEIS 45
SEGSEIS 46
SEGSEIS 47
SEGSEIS 48
SEGSEIS 49
SEGSEIS 50
SEGSEIS 51
SEGSEIS 52
SEGSEIS 53
SEGSEIS 54
SEGSEIS 55
SEGSEIS 56
SEGSEIS 57
SEGSEIS 58

```

	500 CALL SEGNAME(1,LABEL)	APR06A	1
	CALL INDEXER(MTSCR)	SEGSEIS	60
60	LASTDAY = NEWDAY	SEGSEIS	61
		SEGSEIS	62
	C* CHECK FOR RECORDS WITH CORRECT STARTING TIME	SEGSEIS	63
	C* IF FOUND RETRIEVE FROM MASS STORAGE AND PACK INTO OUTPUT SUBRE	SEGSEIS	64
	520 NCHAN = 0	SEGSEIS	65
65	DO 530 K=KMIN,NLIST	SEGSEIS	66
	IF (TIME(K).GT.TMIN+FIVESEC) GO TO 530	SEGSEIS	67
	NCHAN = NCHAN+1	SEGSEIS	68
	CALL READMS(LUNSCR,STORE(1),2*M,ITEM(K))	SEGSEIS	69
70	CALL STNPACK(NCHAN,STORE,STNDATA(1,NSTNS+1),M,LIST(K))	SEGSEIS	70
	ITEM(K) = ITEM(K)+1	SEGSEIS	71
	CALL READMS(LUNSCR,TIME(K),1,ITEM(K))	SEGSEIS	72
	530 CONTINUE	SEGSEIS	73
		SEGSEIS	74
75	C* CALL STNSEIS TO CHECK DATA FOR SPECIAL FLAGS	SEGSEIS	75
	CALL STNSEIS(STNDATA(1,NSTNS+1),M)	SEGSEIS	76
		SEGSEIS	77
	NSTNS = NSTNS+M	SEGSEIS	78
	GO TO 410	SEGSEIS	79
80	C* OUTPUT LAST DATA BLOCK, THEN PRODUCE OUTPUT FILE	SEGSEIS	80
	600 CALL SEGNAME(1,LABEL)	SEGSEIS	81
	CALL INDEXER(MTSCR)	APR06A	2
	CALL OUTAPE(MTOUT)	SEGSEIS	83
		SEGSEIS	84
85	RETURN	SEGSEIS	85
		SEGSEIS	86
	END	SEGSEIS	87
		SEGSEIS	88

1	SUBROUTINE STNSEIS(STNDATA,M)	STNSEIS 2
		STNSEIS 3
	C*** STNSEIS	STNSEIS 4
	C** THIS ROUTINE CHECKS THROUGH A SEISMIC DATA RECORD FOR SPECIAL	STNSEIS 5
5	C** FLAGS AND IF FOUND INSERTS APPROPRIATE TIME VALUES IN THESE	STNSEIS 6
	C** SLOTS	STNSEIS 7
	C** IT ALSO CHECKS FOR DUPLICATE SEISMIC HORIZONS	STNSEIS 8
	C** STNDATA - ARRAY CONTAINING DATA TO BE CHECKED	STNSEIS 9
	C** M - BLOCK LENGTH OF ARRAY	STNSEIS 10
10	DIMENSION STNDATA(12,M)	STNSEIS 11
	DATA (IEROD = 5),(IOUTCR = 6),(IBASE = 7)	STNSEIS 12
	DATA (DUBIOUS = 1.0E9),(UNKNOWN=1.0E10)	STNSEIS 13
15	IEROD = 0	STNSEIS 14
		STNSEIS 15
	DO 100 J=1,M	STNSEIS 16
		STNSEIS 17
	C* CHECK FOR DUPLICATE HORIZONS	STNSEIS 18
20	LASCHAN = (STNDATA(4,J)+0.5)*1.0E-6	STNSEIS 19
	DO 280 I=5,12	STNSEIS 20
	IF (STNDATA(I,J).GT.DUBIOUS) GO TO 10	STNSEIS 21
	NEXCHAN = (STNDATA(I,J)+0.5)*1.0E-6	STNSEIS 22
	IF (NEXCHAN.GT.LASCHAN) GO TO 240	STNSEIS 23
25	DO 220 K = 1,11	STNSEIS 24
	STNDATA(K,J) = STNDATA(K+1,J)	STNSEIS 25
	220 CONTINUE	STNSEIS 26
	STNDATA(12,J) = UNKNOWN	STNSEIS 27
	IEROD = IEROD + 1	STNSEIS 28
30	240 LASCHAN = NEXCHAN	STNSEIS 29
	280 CONTINUE	STNSEIS 30
		STNSEIS 31
	C* CHECK FOR ERODED OR BASEMENT FLAGS - IF FOUND INSERT TOP VALUE	STNSEIS 32
35	10 TLAST = AMOD(STNDATA(4,J),1.0E5)	STNSEIS 33
	DO 40 I=5,12	STNSEIS 34
	IF (STNDATA(I,J).GT.DUBIOUS) GO TO 50	STNSEIS 35
	IDATA = (STNDATA(I,J)+0.5)*1.0E-5	STNSEIS 36
	IFLAG = MOD(IDATA,10)	STNSEIS 37
	IF (IFLAG.EQ.IEROD.OR.IFLAG.EQ.IBASE) GO TO 20	STNSEIS 38
40	TLAST = STNDATA(I,J) - IDATA*1.0E5	STNSEIS 39
	GO TO 40	STNSEIS 40
	20 STNDATA(I,J) = IDATA*1.0E5 + TLAST	STNSEIS 41
	40 CONTINUE	STNSEIS 42
		STNSEIS 43
		STNSEIS 44
45	C* CHECK FOR OUTCROP FLAG - IF FOUND INSERT BOTTOM VALUE	STNSEIS 45
	50 TLAST = STNDATA(12,J)	STNSEIS 46
	DO 80 I1=4,11	STNSEIS 47
	I = 15-I1	STNSEIS 48
	IF (STNDATA(I,J).GT.DUBIOUS) GO TO 80	STNSEIS 49
50	IDATA = (STNDATA(I,J)+0.5)*1.0E-5	STNSEIS 50
	IFLAG = MOD(IDATA,10)	STNSEIS 51
	IF (IFLAG.EQ.IOUTCR) GO TO 60	STNSEIS 52
	TLAST = STNDATA(I,J) - IDATA*1.0E5	STNSEIS 53
	GO TO 80	STNSEIS 54
55	60 STNDATA(I,J) = IDATA*1.0E5 + TLAST	STNSEIS 55
	80 CONTINUE	STNSEIS 56
		STNSEIS 57
		STNSEIS 58

```
100 CONTINUE
60 C* IF ANY DUPLICATE HORIZONS FOUND WITHIN BLOCK PRINT MESSAGE
    IF (IERROR.GT.0) PRINT 110, STNDATA(1,1), IERROR
    110 FORMAT(/,10X,10(1H*),10X,*DUPLICATE VALUES IN DATA BLOCK AT *,
1 F13.7,* - TOTAL NUMBER *,110)
65 RETURN
END
```

```
STNSEIS 59
STNSEIS 60
STNSEIS 61
STNSEIS 62
STNSEIS 63
STNSEIS 64
STNSEIS 65
STNSEIS 66
STNSEIS 67
STNSEIS 68
```

```

SUBROUTINE STNPACK(NCHAN,STORE,STNDATA,M,NHOR)
C***      STNPACK
C**      THIS ROUTINE PACKS EACH INPUT VALUE FROM A SEISMIC HORIZON
C**      FILE INTO THE FIRST VACANT COLUMN IN THE OUTPUT RECORD, JNE
C**      RECORD AT A TIME
C*      NCHAN - CHANNEL NUMBER
C*      STORE - WORK SPACE
10 C*      STNDATA - OUTPUT ARRAY
C*      M - BLOCK SIZE
C*      NHOR - HORIZON NUMBER

DIMENSION STORE(2,M),STNDATA(12,M)
COMMON/INDEX/ LABEL(8),NSEGS,NSUM,ERRSUM(11),
1 INDEX(8,200),STNSUM(200),ST(3,200),NERROR(11,200)
DATA (DUBIOUS=1.0E9),(UNKNOWN=1.0E10)

IF (NCHAN.GT.1) GO TO 100

C*      IF FIRST CHANNEL IN OUTPUT SUB-RECORD, TRANSFER TIMES AND SET
C*      DATA VALUES TO UNKNOWN
DO 20 J=1,M
STNDATA(1,J) = STORE(1,J)
25 DO 10 I=2,12
STNDATA(I,J) = UNKNOWN
10 CONTINUE
20 CONTINUE

30 100 IOUT = 0
SHIFT = NHOR*1000000

C*      IF VALID CHANNEL VALUE, PACK INTO OUTPUT SUB-RECORD IN FIRST
C*      VACANT COLUMN
35 DO 200 J=1,M
IF (STORE(2,J).GT.DUBIOUS) GO TO 200

DO 190 I=4,12
IF (STNDATA(I,J).LT.DUBIOUS) GO TO 190
STNDATA(I,J) = STORE(2,J) + SHIFT
40 GO TO 200
190 CONTINUE

IOUT = IOUT+1
45 200 CONTINUE

IF (IOUT.GT.0) PRINT 210, NHOR,IOUT,STORE(1,1)
210 FORMAT(/,10X,10(1H-),10X,+HORIZON+,16,+ HAS+,15,
1 * EXCESS DATA VALUES IN BLOCK STARTING AT *,F13.6)

50 RETURN

END

```

STNPACK 2
STNPACK 3
STNPACK 4
STNPACK 5
STNPACK 6
STNPACK 7
STNPACK 8
STNPACK 9
STNPACK 10
STNPACK 11
STNPACK 12
STNPACK 13
STNPACK 14
STNPACK 15
STNPACK 16
STNPACK 17
STNPACK 18
STNPACK 19
STNPACK 20
STNPACK 21
STNPACK 22
STNPACK 23
STNPACK 24
STNPACK 25
STNPACK 26
STNPACK 27
STNPACK 28
STNPACK 29
STNPACK 30
STNPACK 31
STNPACK 32
STNPACK 33
STNPACK 34
STNPACK 35
STNPACK 36
STNPACK 37
STNPACK 38
STNPACK 39
STNPACK 40
STNPACK 41
STNPACK 42
STNPACK 43
STNPACK 44
STNPACK 45
STNPACK 46
STNPACK 47
STNPACK 48
STNPACK 49
STNPACK 50
STNPACK 51
STNPACK 52
STNPACK 53
STNPACK 54

1981/06/05

1	PROGRAM MERGPOS(TAPE10,TAPE1,TAPE2,TAPE20,INPUT,TAPE60=INPUT, 1 OUTPUT)	MERGPOS 2
		MERGPOS 3
	C*** MERGPOS	MERGPOS 4
5	C** THIS PROGRAM MERGES POSITIONS FROM THE LOCATION FILE PRODUCED	MERGPOS 5
	C** FROM DIGMAP INTO THE SEISMIC DATA FILE PRODUCED FROM DIGDATA,	MERGPOS 6
	C** SAMPLE AND SEGPACK. POSITIONS ARE INTERPOLATED IF NECESSARY.	MERGPOS 7
	C** THE SEISMIC FILE IS THE MASTER FILE. POSITIONS ARE FOUND	MERGPOS 8
	C** ONLY FOR TIMES SPECIFIED ON THIS FILE	MERGPOS 9
10		MERGPOS 10
	C** TAPE1 - SEISMIC DATA INPUT	MERGPOS 11
	C** TAPE10 - OUTPUT UNIT FOR INDEXER	MERGPOS 12
	C** TAPE2 - POSITION DATA INPUT	MERGPOS 13
	C** TAPE20 - SCRATCH UNIT FOR INDEXER	MERGPOS 14
15		MERGPOS 15
	DIMENSION LABSEIS(8),LABPOS(8)	MERGPOS 16
	COMMON/INDEX/ LABEL(8),NSEGS,NSUM,ERRSUM(11),	MERGPOS 17
	1 INDEX(8,200),STNSUM(200),ST(3,200),NERROR(11,200)	MERGPOS 18
	COMMON/SEGMENT/SEGMENT(8),NSTNS,T(3),ERROR(11),STNDATA(12,1440)	MERGPOS 19
20	COMMON/LUN/ MTIN,MTOUT,MTSCR,LUNSCR,LUNDOC,LUNSAVE,LUNPLOT	MERGPOS 20
	DATA (MTIN = 1),(LUNSCR = 2),(MTOUT = 10),(MTSCR = 20)	MERGPOS 21
	DATA (NCOUNT = 0)	MERGPOS 22
		MERGPOS 23
	REWIND MTIN \$ REWIND LUNSCR	MERGPOS 24
25	REWIND MTSCR \$ REWIND MTOUT	MERGPOS 25
		MERGPOS 26
	NOPOS = 0	MERGPOS 27
	READ 10, LABEL	MERGPOS 28
	10 FORMAT(8A10)	MERGPOS 29
30	CALL TODAY(LABEL(8))	MERGPOS 30
	PRINT 20, LABEL	MERGPOS 31
	20 FORMAT(1H1,/,10X,*CREATION OF SEISMIC PRINCIPAL FACTS FILE *,	MERGPOS 32
	1 /,10X,40(1H-),/,10X,*OUTPUT FILE LABELLED *,1H*,8A10,1H*)	MERGPOS 33
35		MERGPOS 34
	C* READ SEISMIC AND POSITION LABELS	MERGPOS 35
	C* LOCATE FILES - IF NOT FOUND ABORT JOB	MERGPOS 36
	READ 10, LABSEIS,LABPOS	MERGPOS 37
	IF (EOF(60)) 900,30	MERGPOS 38
	30 CALL LOOKUP(LABSEIS,MTIN,1,1,NFLAG)	MERGPOS 39
40	IF (NFLAG.EQ.1) CALL ABORT	MERGPOS 40
	CALL LOOKUP(LABPOS,LUNSCR,1,1,NFLAG)	MERGPOS 41
	IF (NFLAG.EQ.1) CALL ABORT	MERGPOS 42
		MERGPOS 43
	C* BUFFER IN SEGMENT OF SEISMIC DATA	MERGPOS 44
45	60 BUFFER IN(MTIN,1) (SEGMENT(1),STNDATA(12,1440))	MERGPOS 45
	IF (UNIT(MTIN)) 90,900,70	MERGPOS 46
	70 PRINT 80, SEGMENT	MERGPOS 47
	80 FORMAT(/,10X,10(1H*),10X,*PARITY ERROR IN *,1H*,8A10,1H*)	MERGPOS 48
		MERGPOS 49
50		MERGPOS 50
	C* LOOP THROUGH SEISMIC FILE CALL STNPOS TO FIND POSITION	MERGPOS 51
	C* IF NO POSITIONS JUMP LOOP	MERGPOS 52
	90 IF (NOPOS.GT.0) GO TO 210	MERGPOS 53
	DO 200 J=1,NSTNS	MERGPOS 54
	IF (STNDATA(1,1).GT.1000.0) GO TO 180	MERGPOS 55
55	CALL STNPOS(STNDATA(1,J),STNDATA(2,J),STNDATA(3,J),NOPOS,NCOUNT)	MERGPOS 56
	GO TO 200	MERGPOS 57
	180 CALL SHOTPOS(STNDATA(1,J),STNDATA(2,J),STNDATA(3,J),NOPOS,NCOUNT)	MERGPOS 58

	200 CONTINUE	MERGPOS	59
		MERGPOS	60
60	C* CALL INDEXER TO SAVE DATA BLOCK	MERGPOS	61
	210 CALL INDEXER(MTSCR)	MERGPOS	62
	GO TO 60	MERGPOS	63
		MERGPOS	64
	C* AT END OF SEISMIC DATA, OUTPUT PRINCIPAL FACTS FILE	MERGPOS	65
65	900 CALL OUTAPE(MTOUT)	MERGPOS	66
	PRINT 910, NCOUNT	MERGPOS	67
	910 FORMAT(10X,10(1H\$),10X,*POSITIONS MERGED INTO SEISMIC DATA FILE*,	MERGPOS	68
	1 //,30X,*A TOTAL OF*,15,* POSITIONS NOT FOUND *)	MERGPOS	69
	END	MERGPOS	70

1	SUBROUTINE STNPOS(STNNUM,RLAT,RLONG,NOPOS,NCOUNT)	STNPOS 2
		STNPOS 3
	C*** STNPOS	STNPOS 4
	C** THIS ROUTINE READS A LOCATION DATA FILE AND RETURNS A POSITION	STNPOS 5
5	C** FOR A SPECIFIED STATION NUMBER	STNPOS 6
		STNPOS 7
	C** STNNUM - STATION NUMBER FOR WHICH POSITION IS REQUIRED	STNPOS 8
	C** RLAT - LATITUDE OF STNNUM (RETURNED)	STNPOS 9
	C** RLONG - LONGITUDE OF STNNUM (RETURNED)	STNPOS 10
10	C** NOPOS - FLAG TO INDICATE END OF POSITION FILE (=1 AT END)	STNPOS 11
		STNPOS 12
	COMMON/STNPOS/ SEGMENT(8),NSTNS,T(3),ERROR(11),STNDATA(12,1440)	STNPOS 13
	COMMON/LUN/ MTIN,MTOUT,MTSCR,LUNSCR,LUNDOC,LUNSAVE,LUNPLOT	STNPOS 14
	DATA (NBUF = 0),(FIVESEC = 0.08),(UNKNOWN = 1.0E10)	STNPOS 15
15	DATA (NPRINT=0)	STNPOS 16
	TIME = TMIN(SSTNNUM)	STNPOS 17
		STNPOS 18
	IF (NBUF.GT.0) GO TO 100	STNPOS 19
20	XNEXT = YNEXT = TNEXT = 0.0	STNPOS 20
		STNPOS 21
	C* SET TIME INTERPOLATION LIMIT	DEC19 1
	TINTRPL = 720.0	DEC19 2
		DEC19 3
25	C* BUFFER IN BLOCK OF POSITION DATA, SET POSITIONS AND TIMES FOR	STNPOS 26
	C* LAST AND NEXT VALUES	STNPOS 27
	10 BUFFER IN(LUNSCR,1) (SEGMENT(1),STNDATA(12,1440))	STNPOS 28
	IF (UNIT(LUNSCR)) 50,900,30	STNPOS 29
	30 PRINT 40, SEGMENT	STNPOS 30
30	40 FORMAT(/,10X,10(1H*),10X,*PARITY ERROR IN *,1H*,8A10,1H*)	STNPOS 31
	50 NBUF = 0 \$ GO TO 110	STNPOS 32
		STNPOS 33
	C* LOOP THROUGH POSITION FILE UNTIL REQUIRED TIME IS STRADDLED	STNPOS 34
	C* BY LAST AND NEXT VALUES - THEN JUMP TO INTERPOLATE VALUES	STNPOS 35
35	100 IF (TIME.LT.TNEXT+FIVESEC) GO TO 120	STNPOS 36
	IF (NBUF.GE.NSTNS) GO TO 10	STNPOS 37
	110 NBUF = NBUF+1	STNPOS 38
	XLAST = XNEXT \$ XNEXT = STNDATA(2,NBUF)	STNPOS 39
	YLAST = YNEXT \$ YNEXT = STNDATA(3,NBUF)	STNPOS 40
40	TLAST = TNEXT \$ TNEXT = TMIN(STNDATA(1,NBUF))	STNPOS 41
	DX = XNEXT - XLAST \$ DY = YNEXT - YLAST	STNPOS 42
	DT = TNEXT-TLAST \$ IF (ABS(DT).LT.1.0E-8) DT = 1.0E-8	STNPOS 43
	COEFFX = DX/DT \$ COEFFY = DY/DT	STNPOS 44
45	GO TO 100	STNPOS 45
		STNPOS 46
	C* INTERPOLATE LATITUDE AND LONGITUDE,IF LAST AND NEXT WITHIN	STNPOS 47
	C* TINTRPL MINUTES OF EACH OTHER.	STNPOS 48
120	IF(TNEXT - TLAST.LT.TINTRPL) GO TO 800	STNPOS 49
	IF (TIME.GT.TNEXT-FIVESEC) GO TO 800	STNPOS 50
50		STNPOS 51
	C* RETURN POSITIONS AS UNKNOWN IF INTERPOLATION TIME TOO GREAT	STNPOS 52
	STNEXT=TSURVEY(TNEXT) \$ IF(NPRINT.EQ.0) PRINT 130,STNNUM,STNEXT	STNPOS 53
130	FORMAT(/,30X,*POSITION BREAK BETWEEN *,F10.7,* AND *,F10.7,/) \$	STNPOS 54
	NPRINT=1	STNPOS 55
55	RLAT = RLONG = UNKNOWN	STNPOS 56
	NCOUNT = NCOUNT + 1	STNPOS 57
	RETURN	STNPOS 58
		STNPOS 59

		STNPOS	60
	C* INTERPOLATE LATITUDE AND LONGITUDE, IF LAST AND NEXT WITHIN	STNPOS	61
60	C* TINTRPL MINUTES OF EACH OTHER.	STNPOS	62
	800 FRACTN = (TIME - TLAST)	STNPOS	63
	RLAT = XLAST + FRACTN*COEFFX	STNPOS	64
	RLONG = YLAST + FRACTN*COEFFY	STNPOS	65
	NPRINT=0	STNPOS	66
65		STNPOS	67
	RETURN	STNPOS	68
		STNPOS	69
	C* PRINT MESSAGE AND RETURN NOPOS EQ 1 IF END OF POSITION FILE	STNPOS	70
	C* REACHED BEFORE END OF SEISMIC DATA	STNPOS	71
70	900 PRINT 910, STNNUM	STNPOS	72
	910 FORMAT(10X,10(1H*),10X,*NO MORE POSITIONS ON LOCATION FILE AFTER T	STNPOS	73
	IME *,F10.6)	STNPOS	74
	NOPOS = 1	STNPOS	75
75		STNPOS	76
	RETURN	STNPOS	77
		STNPOS	78
	END	STNPOS	79

	900 PRINT 910, STNNUM	SHOTPOS	59
	910 FORMAT(10X,10(1H*),10X,*NO MORE POSITIONS ON LOCATION FILE AFTER T	SHOTPOS	60
60	TIME *,F10.6)	SHOTPOS	61
	NOPOS = 1	SHOTPOS	62
		SHOTPOS	63
	RETURN	SHOTPOS	64
65	END	SHOTPOS	65
		SHOTPOS	66

1981/06/05


```

1      PROGRAM SAMPLOT(TAPE99,TAPE1,INPUT,TAPE60=INPUT,OUTPUT)
      C***      SAMPLOT
      C**      THIS PROGRAM PLOTS DIGITISED PROFILE DATA AFTER IT HAS
5      C**      PASSED THROUGH PROGRAM 'SAMPLE'. THE TIME OR STATION
      C**      NUMBER INTERVAL, HORIZONTAL AND VERTICAL PLOTTING SCALES
      C**      AND FRAME SCALE ARE UNDER USER CONTROL. A DECREASING TIME
      C**      ORDER IS INTERPRETED AS A REVERSED PROFILE.
10     C**      MULTIPLE PLOTS ARE POSSIBLE, BUT CARE MUST BE TAKEN WHEN
      C**      PLOTTING STRIP-CHART DATA AS DATA CARD STRUCTURE IS
      C**      ORIENTED TO SEISMIC SECTION PLOTS.

      DIMENSION LABEL(8),IBUF(64)
15     COMMON/LUN/ MTIN,MTOUT,MTSCR,LUNSCR,LUNDOC,LUNSAVE,LUNPLOT
      DATA(LUNPLOT=99),(IBLANK=1H),(MTIN=1)

      REWIND MTIN
      CALL PLOTS(IBUF,64,LUNPLOT)
20     CALL PLOTBLK(0.0,10.0,-3,10HSEISMIC OR)

      10 READ 20, TSTART,TSTOP,XSCALE,YSCALE,VO,VMAX,VSCALE,NDEC
      20 FORMAT(2F10.7,5F10.2,110)
      IF (EOF(60)) 100,30
25     30 XSCALE = ABS(XSCALE) $ IF (TSTART.GT.TSTOP) XSCALE = -ABS(XSCALE)
      IF (VMAX.LT.1.0E-6) VMAX = 10000.0
      IF (VSCALE.LT.1.0E-6) VSCALE = 1000.0
      IF (NDEC.EQ.0) NDEC = 6
      TIMEINT = SECOND(UU)
30     PRINT 40, TIMEINT,TSTART,TSTOP,NDEC,VO,VMAX,VSCALE,XSCALE,YSCALE
      40 FORMAT(1H1,/,10X,*EDIT PLOT OF RE-SAMPLED DATA FILE - COMMENCED*,
1     1 F9.3,*, SECS,/,10X,60(1H-),
2     2 ///,20X,*INTERVAL *,F15.7,*, TO*,F15.7,
3     3 ///,20X,*STATION NUMBER PRECISION -,13,*, DECIMAL PLACES*,
35     4 ///,20X,*VERTICAL ORIGIN -,F12.1,*, UNITS*,
      5 ///,20X,*VERTICAL LIMIT -,F12.1,*, UNITS*,
      6 ///,20X,*VERTICAL MARKS -,F12.1,*, UNITS*,
      7 ///,20X,*TIME SCALE -,F12.1,*, MINS/INCH*,
40     8 ///,20X,*VERTICAL SCALE -,F12.1,*, UNITS/INCH*)

      C*      READ NEXT FILE HEADER - IF NONE DRAW FRAME ROUND PLOT
      50 READ 60, LABEL
      60 FORMAT(8A10)
45     IF (EOF(60)) 70,80
      70 CALL TFRAME(TSTART,TSTOP,NDEC,VO,VMAX,VSCALE,XSCALE,YSCALE)
      GO TO 10

      80 CALL LOOKUP(LABEL,MTIN,1,1,NFLAG)
      IF (NFLAG.EQ.1) CALL ABORT
50     C*      GO AND FIND DATA WITHIN LIMITS GIVEN
      CALL SAMDATA(TSTART,TSTOP,NDEC,VO,VMAX,XSCALE,YSCALF)
      GO TO 50

55     100 CALL PLOTBLK(0.0,0.0,999,7HSAMPLOT)

```

```

2      SAMPLOT
3      SAMPLOT
4      SAMPLOT
5      SAMPLOT
6      SAMPLOT
7      SAMPLOT
8      SAMPLOT
9      SAMPLOT
10     SAMPLOT
11     SAMPLOT
12     SAMPLOT
13     SAMPLOT
14     SAMPLOT
15     SAMPLOT
16     SAMPLOT
17     SAMPLOT
18     SAMPLOT
19     SAMPLOT
20     SAMPLOT
21     SAMPLOT
22     SAMPLOT
23     SAMPLOT
24     SAMPLOT
25     SAMPLOT
26     SAMPLOT
27     SAMPLOT
28     SAMPLOT
29     SAMPLOT
30     SAMPLOT
31     SAMPLOT
32     SAMPLOT
33     SAMPLOT
34     SAMPLOT
35     SAMPLOT
36     SAMPLOT
37     SAMPLOT
38     SAMPLOT
39     SAMPLOT
40     SAMPLOT
41     SAMPLOT
42     SAMPLOT
43     SAMPLOT
44     SAMPLOT
45     SAMPLOT
46     SAMPLOT
47     SAMPLOT
48     SAMPLOT
49     SAMPLOT
50     SAMPLOT
51     SAMPLOT
52     SAMPLOT
53     SAMPLOT
54     SAMPLOT
55     SAMPLOT
56     SAMPLOT
57     SAMPLOT
58     SAMPLOT

```

PROGRAM SAMLOT 76/76 OPT=1

FTN 4.8+528

11/06/81 14.20.32

PAGE 2

END

SAMLOT 59

1	SUBROUTINE SAMDATA(TSTART,TSTOP,NDEC,V0,VMAX,XSCALE,YSCALE)	SAMDATA	2
		SAMDATA	3
	C*** SAMDATA	SAMDATA	4
	C** THIS ROUTINE ACTUALLY PLOTS THE SECTION OF DATA REQUIRED AS	SAMDATA	5
5	C** SPECIFIED BY TIME (OR STATION NUMBER) LIMITS. THE BLOCK	SAMDATA	6
	C** DIMENSIONS ARE SET AUTOMATICALLY.	SAMDATA	7
	C**	SAMDATA	8
	C** TSTART,TSTOP - TIME LIMITS OF PLOT REQUIRED	SAMDATA	9
	C** NDEC - NUMBER OF DECIMAL PLACES NEEDED IN POSITION POSTING	SAMDATA	10
10	C** V0,VMAX - MINIMUM AND MAXIMUM VALUES TO BE PLOTTED	SAMDATA	11
	C** XSCALE,YSCALE - HORIZONTAL AND VERTICAL PLOTTING SCALE	SAMDATA	12
		SAMDATA	13
	DIMENSION STORE(2,1500)	SAMDATA	14
	COMMON/LUN/ MTIN,MTOUT,MTSCR,LUNSCR,LUNDOC,LUNSAVE,LUNPLOT	SAMDATA	15
15	DATA (MLAST=0),(FIVESEC=5.0E-8),(DUBIOUS=1.0E9),(UNKNOWN=1.0E10)	SAMDATA	16
	KPEN = 3	SAMDATA	17
	TMIN = AMIN1(TSTART,TSTOP) \$ TMAX = AMAX1(TSTART,TSTOP)	SAMDATA	18
	IF (TSTART.GT.1000.0) GO TO 5	SAMDATA	19
20	TO = TMINS(TSTART) \$ GO TO 10	SAMDATA	20
	5 TSHIFT = 10.0*NDEC \$ TO = TSTART+TSHIFT	SAMDATA	21
		SAMDATA	22
	C* BUFFER IN DATA FROM 'SAMPLE' OUTPUT FILE	SAMDATA	23
25	10 BUFFER IN (MTIN,1) (STORE(1,1),STORE(2,1500))	SAMDATA	24
	IF (UNIT(MTIN)) 40,200,20	SAMDATA	25
	20 PRINT 30, STORE(1,1)	SAMDATA	26
	30 FORMAT(/,10X,10(1H+),10X,*PARITY ERROR IN RECORD STARTING *,F12.6)	SAMDATA	27
		SAMDATA	28
	40 M = LENGTH(MTIN)/2	SAMDATA	29
30	IF (M.NE.MLAST) PRINT *,5, M,STORE(1,1)	SAMDATA	30
	45 FORMAT(/,10X,10(1H+),*,*ARRAY LENGTH IS*,I6,* AT *,F15.6)	SAMDATA	31
	MLAST = M	SAMDATA	32
		SAMDATA	33
	C* FIND DATA BETWEEN TSTART AND TSTOP	SAMDATA	34
35	IF (STORE(1,M).LT.TMIN-FIVESEC) GO TO 10	SAMDATA	35
		SAMDATA	36
	DO 100 J=1,M	SAMDATA	37
	IF (STORE(1,J).LT.TMIN-FIVESEC) GO TO 100	SAMDATA	38
	IF (STORE(1,J).GT.TMAX+FIVESEC) GO TO 190	SAMDATA	39
40	IF (STORE(2,J).LT.DUBIOUS) GO TO 50	SAMDATA	40
	KPEN = 3 \$ GO TO 100	SAMDATA	41
		SAMDATA	42
	50 VALUE = AMOD(STORE(2,J),1.0E5)	SAMDATA	43
45	IF (VALUE.GT.V0.AND.VALUE.LT.VMAX) GO TO 55	SAMDATA	44
	KPEN = 3 \$ GO TO 100	SAMDATA	45
		SAMDATA	46
	55 IF (STORE(1,J).LT.1000.0) GO TO 60	SAMDATA	47
	TNEXT = STORE(1,J)*TSHIFT \$ GO TO 70	SAMDATA	48
50	60 TNEXT = TMINS(STORE(1,J))	SAMDATA	49
		SAMDATA	50
	70 X = (TNEXT-TO)/XSCALE	SAMDATA	51
	Y = (VALUE-V0)/YSCALE	SAMDATA	52
	CALL PLOT(X,Y,KPEN) \$ KPEN = 2	SAMDATA	53
55	100 CONTINUE	SAMDATA	54
		SAMDATA	55
	GO TO 10	SAMDATA	56
		SAMDATA	57
		SAMDATA	58

60 190 CALL SKIP66(MTIN,1)
200 CALL PLOTBLK(0.0,0.0,-3,7HHORIZON)
RETURN
END

SAMDATA 59
SAMDATA 60
SAMDATA 61
SAMDATA 62
SAMDATA 63
SAMDATA 64

1	SUBROUTINE TFRAME(TSTART,TSTOP,NDEC,V0,VMAX,VSCALE,XSCALE,YSCALE)	TFRAME	2
		TFRAME	3
	C*** TFRAME	TFRAME	4
	C** THIS ROUTINE PLOTS A FRAME AROUND THE PLOTTED SECTION OR	TFRAME	5
5	C** PROFILE, WITH A SUITABLE TIME AND DATA SCALE	TFRAME	6
	C**	TFRAME	7
	C** TSTART,TSTOP - TIME LIMITS OF PLOT REQUIRED	TFRAME	8
	C** NDEC - NUMBER OF DECIMAL PLACES NEEDED IN POSITION POSTING	TFRAME	9
	C** V0,VMAX - MINIMUM AND MAXIMUM VALUES TO BE PLOTTED	TFRAME	10
10	C** VSCALE - INTERVAL AT WHICH FRAME SCALE IS TO BE POSTED	TFRAME	11
	C** XSCALE,YSCALE - HORIZONTAL AND VERTICAL PLOTTING SCALE	TFRAME	12
		TFRAME	13
	TSHIFT = 10.0*NDEC \$ IF (TSTART.LT.1000.0) GO TO 10	TFRAME	14
	T0 = TSTART*TSHIFT \$ TMAX = TSTOP*TSHIFT \$ GO TO 15	TFRAME	15
15	10 T0 = TMINS(TSTART) \$ TMAX = TMINS(TSTOP)	TFRAME	16
	15 TSTEP = 20.0 \$ IF (XSCALE.LT.0.0) TSTEP = -20.0	TFRAME	17
	JUMP = 1 \$ IF (YSCALE.GT.0.0) JUMP = -1	TFRAME	18
	INUM = (TMAX-T0)/TSTEP+1.5	TFRAME	19
	M = (VMAX-V0)/VSCALE+1.5	TFRAME	20
20		TFRAME	21
	C* DRAW RIGHT HAND EDGE OF FRAME	TFRAME	22
	KPEN = 3 \$ X = (TMAX-T0)/XSCALE	TFRAME	23
	DO 20 J = 1,M	TFRAME	24
	Y = (J-1)*VSCALE/YSCALE	TFRAME	25
25	CALL PLOT(X,Y,KPEN)	TFRAME	26
	CALL PLOT(X+0.1,Y,2)	TFRAME	27
	VALUE = V0+(J-1)*VSCALE	TFRAME	28
	CALL NUMBER(X+0.2,Y,0.1,VALUE,0.0,0)	TFRAME	29
	CALL PLOT(X,Y,3) \$ KPEN = 2	TFRAME	30
30	20 CONTINUE	TFRAME	31
		TFRAME	32
	C* DRAW TOP EDGE (IF SCALE +VE) OR BOTTOM EDGE (IF -VE)	TFRAME	33
	KPEN = 3 \$ Y = (M-1)*VSCALE/YSCALE	TFRAME	34
	DO 30 I=1,INUM	TFRAME	35
35	DT = (INUM-1)*TSTEP \$ X = DT/XSCALE	TFRAME	36
	CALL PLOT(X,Y,KPEN)	TFRAME	37
	CALL PLOT(X,Y-0.05*JUMP,2)	TFRAME	38
	IF (TSTART.LT.1000.0) GO TO 25	TFRAME	39
	ITIME = T0+DT \$ GO TO 26	TFRAME	40
40	25 ITIME = TSURVEY(T0+DT)*TSHIFT	TFRAME	41
	26 IF (MOD(ITIME,100).NE.0) GO TO 27	TFRAME	42
	CALL PLOT(X,Y-0.1*JUMP,2)	TFRAME	43
	27 CALL PLOT(X,Y,3) \$ KPEN = 2	TFRAME	44
	30 CONTINUE	TFRAME	45
45		TFRAME	46
	C* DRAW LEFT EDGE OF FRAME	TFRAME	47
	KPEN = 3 \$ X = 0.0	TFRAME	48
	DO 50 J = 1,M	TFRAME	49
	Y = (M-J)*VSCALE/YSCALE	TFRAME	50
50	CALL PLOT(X,Y,KPEN)	TFRAME	51
	CALL PLOT(X-0.1,Y,2)	TFRAME	52
	VALUE = V0+(M-J)*VSCALE	TFRAME	53
	ENCODE(10,40,IVALUE) VALUE	TFRAME	54
	40 FORMAT(F10.1)	TFRAME	55
55	CALL SYMBOL(X-1.0,Y,0.1,IVALUE,0.0,8)	TFRAME	56
	CALL PLOT(X,Y,3) \$ KPEN = 2	TFRAME	57
	50 CONTINUE	TFRAME	58

	C*	DRAW BOTTOM EDGE (IF SCALE +VE) OR TOP EDGE (IF -VE)	TFRAME	59
60		KPEN = 3 \$ Y = 0.0	TFRAME	60
		DO 90 I = 1, INUM	TFRAME	61
		DT = (I-1)*TSTEP \$ X = DT/XSCALE	TFRAME	62
		CALL PLOT(X,Y,KPEN)	TFRAME	63
		CALL PLOT(X,Y+0.05*JUMP,2)	TFRAME	64
65		IF (TSTART.LT.1000.0) GO TO 55	TFRAME	65
		ITIME = TO+DT \$ GO TO 56	TFRAME	66
	55	ITIME = TSURVEY(TO+DT)*TSHIFT	TFRAME	67
	56	IF (MOD(ITIME,100).NE.0) GO TO 80	TFRAME	68
		CALL PLOT(X,Y+0.1*JUMP,2)	TFRAME	69
70		IF (MOD(ITIME,10000).NE.0) GO TO 70	TFRAME	70
		CALL PLOT(X,(M-1)*VSCALE/YSCALE,2)	TFRAME	71
	70	TVALUE = ITIME/TSHIFT	TFRAME	72
		CALL NUMBER(X,Y+0.2*JUMP,0.1,TVALUE,90.0*JUMP,NDEC-2)	TFRAME	73
	80	CALL PLOT(X,Y,3) \$ KPEN = 2	TFRAME	74
75		90 CONTINUE	TFRAME	75
			TFRAME	76
		X5 = (INT((X+5.0)/5.0)+1)*5	TFRAME	77
		PRINT 100, X5	TFRAME	78
80	100	FORMAT(/,20X,*LENGTH OF PLOT IS *,F8.0,* INCHES*)	TFRAME	79
		CALL PLOTBLK(X5,0.0,-3,6HTFRAME)	TFRAME	80
			TFRAME	81
		RETURN	TFRAME	82
			TFRAME	83
		END	TFRAME	84
			TFRAME	85

INDEX

LOC	CONTENT	LOC	CONTENT	LOC	CONTENT	LOC	CONTENT	LOC	CONTENT
C01	PROGRAM DIGDATA	D04	SUBROUTINE STNPACK						
D01	PROGRAM DIGDATA	P04							
E01	SUBROUTINE DIGPLOT	C05							
F01	SUBROUTINE DIGPLOT	D05							
G01	SUBROUTINE DIGTYPE	E05	PROGRAM MERGPOS						
H01	SUBROUTINE DIGTYPE	F05	PROGRAM MERGPOS						
I01	SUBROUTINE SLOPE	G05	SUBROUTINE STNPOS						
J01	SUBROUTINE SLOPE	H05	SUBROUTINE STNPOS						
K01	SUBROUTINE HSCALE	I05	SUBROUTINE SHOTPOS						
L01	SUBROUTINE HSCALE	J05	SUBROUTINE SHOTPOS						
M01	SUBROUTINE DIGINT	K05							
'01	SUBROUTINE DIGTIME	L05							
O01	SUBROUTINE HOATA	M05							
P01	SUBROUTINE HOATA	N05	PROGRAM .AMPLUT						
C02	SUBROUTINE VSCALE	D05	PROGRAM SARPLUT						
D02	SUBROUTINE VSCALE	P05	SUBROUTINE SAMDATA						
E02	SUBROUTINE VOATA	C06	SUBROUTINE SAMDATA						
F02	SUBROUTINE VOATA	D06	SUBROUTINE TFRAME						
G02	SUBROUTINE OBSERVE	E06	SUBROUTINE TFRAME						
H02	SUBROUTINE DIGLSA								
I02	SUBROUTINE ORIGIN								
J02	SUBROUTINE ORIGIN								
K02	SUBROUTINE RECORD								
L02	SUBROUTINE RECORD								
M02	SUBROUTINE DELAY								
N02	SUBROUTINE FLAGS								
O02	SUBROUTINE RECDATA								
P02	SUBROUTINE RECDATA								
C03	SUBROUTINE BUFCHK								
D03	SUBROUTINE NEWCHAR								
E03	SUBROUTINE XYCODE								
F03	SUBROUTINE XYTABLE								
G03	SUBROUTINE EXTRACT								
H03	SUBROUTINE FPCODE								
I03									
J03									
K03									
L03									
M03	PROGRAM SAMPLE								
N03	SUBROUTINE LABNAME								
O03	SUBROUTINE JUMPFIX								
P03	SUBROUTINE JUMPER								
C04	SUBROUTINE CHOPPER								
D04	SUBROUTINE INFILL								
E04	SUBROUTINE INFILL								
F04									
G04									
H04									
I04	PROGRAM SEGPACK								
J04	PROGRAM SEGPACK								
K04	SUBROUTINE SEGSEIS								
L04	SUBROUTINE SEGSEIS								
M04	SUBROUTINE STNSEIS								
N04	SUBROUTINE STNSEIS								