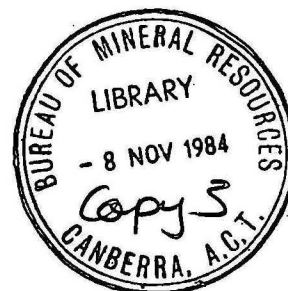
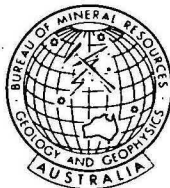


1984/27

03

BMR PUBLICATIONS COMPACTUS
(LENDING SECTION)



~~LIMITED DISTRIBUTION~~

BUREAU OF MINERAL RESOURCES, GEOLOGY AND GEOPHYSICS

~~LIMITED DISTRIBUTION~~ O/F

~~LIMITED DISTRIBUTION~~

RECORD

RECORD 1984/27

LOG PLOT

(COMPUTER PROGRAMS FOR

WIRELINE LOG DATA

MANIPULATION AND PLOTTING)

~~LIMITED DISTRIBUTION~~

BY

G.R. MORRISON AND R.W. DE NARDI

~~LIMITED DISTRIBUTION~~

RECORD 1984/27

LOG PLOT

(COMPUTER PROGRAMS FOR
WIRELINE LOG DATA
MANIPULATION AND PLOTTING)

BY

G.R. MORRISON AND R.W. DE NARDI

TABLE OF CONTENTS

	PAGE
I Introduction to the Record	1
II Computer Program TVD	4
(a) Introduction	4
(b) Operating Instructions	4
(c) Computer Program TVD Listing	9
III Computer Program LGPLT	14
(a) Introduction	14
(b) Operating Instructions	15
(c) Computer Program LGPLT Listing	22
IV Computer Program PLOG	27
(a) Introduction	27
(b) Operating Instructions	27
(c) Source Listing of Program 'PLOG'	33
(d) Sample plots	54
V Bibliography	57
VI Acknowledgment	57

A LIST OF FIGURES APPEARING IN THE RECORD

Figure 1 - Flowchart of the Current Computer Programs used for Wireline Log Data Analysis	2
---	---

I Introduction to the Record

This record is an instruction manual for the three computer programs TVD, LGPLT and PLOG used for the manipulation and plotting of wireline log data. Figure 1 shows how each of these programs fits into the overall wireline log interpretation process. The wireline log data processed by these programs is originally digitised (program SIMUL is the program which operates the digitiser) and then converted from raw X-Y coordinates to scaled log data by programs DIGFX and XYPLT (Figure 1).

A definition of the function of each of the programs appearing on Figure 1 is given below:

- (1) SIMUL: a program used to convert digitised data being entered into the computer from the digitiser into X-Y coordinates.
- (2) DIGFX: a program used to locate and correct digitising errors.
(refer to BMR record 1983/15)
- (3) XYPLT: converts the X-Y coordinates recorded by the digitiser into scaled wireline log data. (refer to BMR record 1983/15)
- (4) RW: computes wireline log interpretation parameters from a digitised data file. (refer to BMR record 1984/26)
- (5) LOG4: a shaly-sand wireline log interpretation model which uses the simandoux equation to calculate water saturation (refer to BMR record 1978/3).
- (6) DW2: a shaly-sand wireline log interpretation model which uses the dual water model equation to calculate water saturation. (This program is the modified version of program DW detailed in BMR record 1984/8.)
- (7) PLOG: a program used to plot a complete suite of digitised logs multiplied by a preselected vertical and horizontal scaling factor for the purposes of correlating log character across a field.

Symbolizes the points at which program TVD can be used, ie

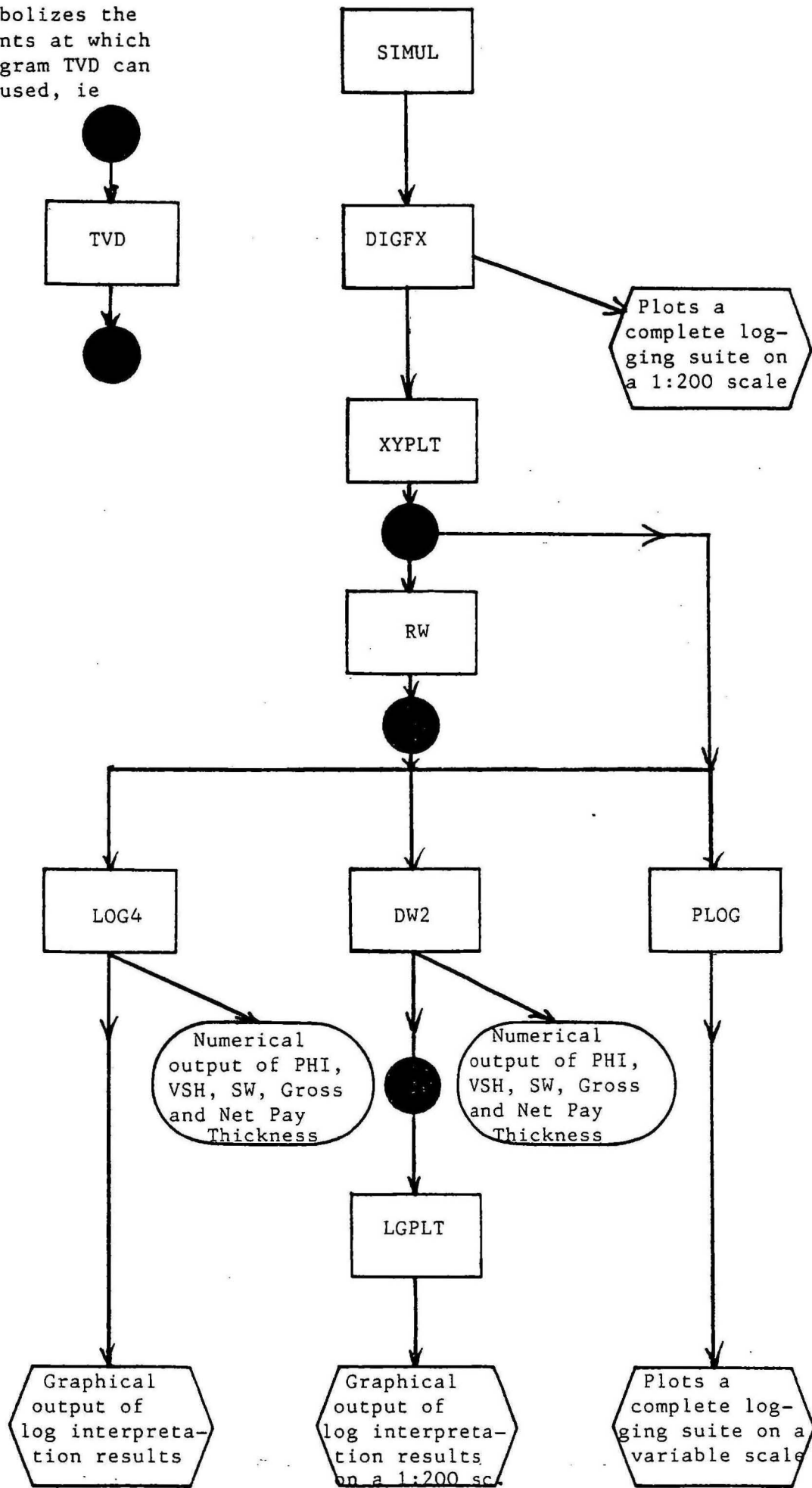


Figure 1 - Flowchart of all the Computer Programs Currently used for Wireline Log Analyses

- (8) TVD: a program used to convert measured depths in a wireline log data file to true vertical depths at any given datum.
- (9) LGPLT: a program which plots pre-interpreted wireline log data on a 1:200 scale. This program plots volume of shale, effective porosity and effective water saturation against depth for the purpose of visual identification of hydrocarbon bearing reservoirs.

Figure 1 indicates that program PLOG can access two different types of data files. These two types are:

- (i) a data file generated by program XYPLT;
- (ii) a data file containing the log interpretation parameters determined by program RW. These parameters are inserted into the top seven lines of the data file and consequently makes the data file compatible with programs LOG4, DW2 as well as PLOG.

Figure 1 also indicates that program TVD can access three different types of data files. These three types are:

- (i) a data file generated by program XYPLT;
- (ii) a data file containing the log interpretation parameters determined by program RW;
- (iii) a data file generated by program DW2 and accessed by program LGPLT. This data file contains the pre-interpreted information which is plotted out by program LGPLT.

Therefore, program TVD can be used to modify the datafiles accessed by programs PLOG and LGPLT so that the plots generated can be on true vertical depth at any given datum. Similarly, the outputs from programs LOG4 and DW2 can also be on true vertical depth at any given datum.

The three computer programs detailed in this record are written in Fortran 77 language. The two plotting programs PLOG and LGPLT use BMR's Hewlett-Packard plotter; HP 2631G. All of the programs mentioned in this record may be accessed via any of the on-line terminals connected to BMR's Hewlett-Packard computer (HP1000F series).

This record has been split into 3 parts, one part for each computer program. Each part contains an introduction to the program, operating instructions, a program listing and some examples of the type of output which can be generated.

II TVD

(a) Introduction

The object of program TVD is to convert measured depths in a wireline log data file to true vertical depths at any given datum. The program requires a measured depth log data file and a conversion data file containing information which relates measured depth to true vertical depth from a directional survey. Program TVD requires the data to be on any one of three possible levels of compatibility (Figure 1). These three compatibility levels or data file types are:

- (i) the data file generated by program XYPLT
- (ii) the data file generated by program RW
- (iii) the data file generated by program DW2 and accessed only by program LGPLT.

The format of the measured depth to true vertical depth conversion data file is given on the top half of page 9 on lines 19 to 27 of the program listing. The program operates by interpolating each depth value read from the measured depth data file with the data contained in the conversion data file. The number of data points required in the conversion data file depends on the amount of well deviation over the log interval and the required accuracy of the conversion. Generally the deviation over the logged interval is constant and consequently the conversion data file does not require many data points. Sufficient accuracy can be obtained by entering one data point for each change in deviation by one degree. In the special case where the directional survey specifies decimal fractions of degrees and not degrees and minutes, the deviation should be entered into the degree column of the data file while the minute column must then be set to zero.

(b) Operating Instructions

An example of a program compilation, loading and run is given on page 5. In this example the file WELLMD is the existing measured depth log data file while WELLCN is the conversion data file containing the directional survey information. The data file WELLTV is the true vertical depth data file being created by program TVD. These 3 data files (WELLMD, WELLCN and WELLTV) are shown on pages 6 to 8.

A special case when running program TVD occurs when converting a log data file which has no heading information except the title of the file (a type (3) data file which has not been processed by program RW). In this case the number of sample points in the measured depth data file has to be calculated externally and entered manually at the computer terminal. The computer will printout a message to the user when this information is required.

FT,&TVD,,&TVD ← compile the program
\$END FTN7X: No disasters, No errors, No warnings.
!

RU,LOADR ← load the program
/LOADR: REL,&TVD
/LOADR: EN
/LOADR:TVD READY AT 4:53 PM TUE., 3 JULY, 1984
/LOADR:\$END
!

RU,TVD ← run the program

INPUT THE TYPE OF DATA FILE TO BE PROCESSED BY ENTERING ONE OF THE BRACKETED NUMBERS BELOW:

- (1) PROGRAM LOG4 OR DW COMPATABLE DATA FILE
- (2) PROGRAM LGPLT COMPATABLE DATA FILE
- (3) A LOG DATA FILE WITHOUT HEADING INFORMATION

1 ← select the type of data file

INPUT EXISTING MEASURED DEPTH LOG DATA FILE NAME (I.E. name:SC:CRT)
WELLMD:919:9

INPUT NAME OF TVD LOG DATA FILE TO BE CREATED (I.E. name:SC:CRT)
WELLTV:919:9

INPUT MD TO TVD CONVERSION DATA FILE NAME (I.E. name:SC:CRT)
WELLON:919:9
NORMAL TERMINATION
!

5
Program TVD compilation, loading and run
(NOTE: the underlined type indicates user input)

FILE NAME : WELLM

.TOP.

WELL A1(2900.0M TO 2924.4MHDKB)

000112-04-84

1,1,1,2,1,1,1,11,2,6,0

50, 88.0, .209, .018, 1.0, 620.0, 0.0000

50.1, 2.650, 0.0, 0.0

197.5, 10.0, 2.900, 0.0, .160, 0.0

.0819, .800, 9.875

2900.0, 101.1, 2.436, .285, 0.0, 0.0, 1.2, 1.2, 1.2

2900.6, 100.6, 2.611, .260, 0.0, 0.0, 2.2, 1.4, 1.5

2901.1, 101.8, 2.535, .215, 0.0, 0.0, 2.0, 1.5, 1.6

2901.6, 95.9, 2.467, .188, 0.0, 0.0, 1.9, 1.5, 1.5

2902.1, 99.6, 2.463, .222, 0.0, 0.0, 1.6, 1.4, 1.4

2902.6, 90.7, 2.476, .220, 0.0, 0.0, 1.6, 1.4, 1.5

2903.1, 93.6, 2.487, .212, 0.0, 0.0, 1.7, 1.4, 1.4

2903.6, 86.9, 2.505, .207, 0.0, 0.0, 1.8, 1.5, 1.6

2904.1, 90.4, 2.507, .188, 0.0, 0.0, 1.7, 1.3, 1.5

2904.6, 90.6, 2.514, .209, 0.0, 0.0, 1.4, 1.3, 1.3

2905.1, 86.8, 2.501, .202, 0.0, 0.0, 1.7, 1.4, 1.4

2905.6, 86.0, 2.493, .190, 0.0, 0.0, 1.8, 1.5, 1.6

2906.1, 92.4, 2.494, .204, 0.0, 0.0, 1.7, 1.6, 1.6

2906.6, 89.6, 2.494, .192, 0.0, 0.0, 2.5, 1.8, 1.8

2907.1, 84.3, 2.518, .153, 0.0, 0.0, 3.5, 2.0, 2.1

2907.6, 85.0, 2.533, .166, 0.0, 0.0, 2.7, 1.9, 2.0

2908.2, 92.7, 2.521, .173, 0.0, 0.0, 1.8, 1.8, 1.9

2908.7, 104.1, 2.497, .201, 0.0, 0.0, 1.6, 1.5, 1.4

2909.2, 93.3, 2.511, .213, 0.0, 0.0, 1.6, 1.4, 1.4

2909.7, 93.4, 2.511, .203, 0.0, 0.0, 1.8, 1.5, 1.6

2910.2, 103.6, 2.520, .205, 0.0, 0.0, 2.1, 1.5, 1.5

2910.7, 114.5, 2.530, .205, 0.0, 0.0, 2.0, 1.6, 1.6

2911.2, 134.6, 2.615, .228, 0.0, 0.0, 2.3, 1.7, 1.9

2911.7, 137.6, 2.738, .326, 0.0, 0.0, 2.2, 1.9, 2.0

2912.2, 121.4, 2.771, .371, 0.0, 0.0, 2.8, 2.4, 2.7

2912.7, 115.6, 2.790, .404, 0.0, 0.0, 2.7, 2.6, 2.8

2913.2, 106.8, 2.799, .400, 0.0, 0.0, 2.8, 2.7, 3.0

2913.7, 106.3, 2.779, .360, 0.0, 0.0, 2.3, 2.7, 2.9

2914.2, 112.1, 2.790, .350, 0.0, 0.0, 2.3, 2.4, 2.7

2914.7, 128.6, 2.790, .397, 0.0, 0.0, 2.5, 2.4, 2.6

2915.3, 121.0, 2.812, .420, 0.0, 0.0, 2.4, 2.6, 2.7

2915.8, 107.7, 2.847, .436, 0.0, 0.0, 2.6, 2.6, 3.0

2916.3, 103.9, 2.849, .447, 0.0, 0.0, 2.4, 2.6, 2.9

2916.8, 106.3, 2.824, .426, 0.0, 0.0, 2.8, 2.7, 3.1

2917.3, 116.7, 2.828, .378, 0.0, 0.0, 2.9, 2.9, 3.2

2917.8, 112.2, 2.809, .407, 0.0, 0.0, 3.2, 2.9, 3.2

2918.3, 111.7, 2.825, .407, 0.0, 0.0, 3.0, 3.0, 3.3

2918.8, 113.7, 2.816, .404, 0.0, 0.0, 3.0, 3.0, 3.4

2919.3, 113.9, 2.785, .389, 0.0, 0.0, 3.0, 3.0, 3.5

2919.8, 117.5, 2.756, .361, 0.0, 0.0, 3.0, 3.1, 3.5

2920.3, 110.5, 2.770, .364, 0.0, 0.0, 2.9, 2.9, 3.6

2920.8, 121.1, 2.758, .389, 0.0, 0.0, 2.8, 2.9, 3.3

2921.3, 121.5, 2.792, .388, 0.0, 0.0, 3.2, 3.1, 3.9

2921.8, 122.3, 2.799, .384, 0.0, 0.0, 3.2, 3.1, 4.0

2922.3, 122.1, 2.806, .376, 0.0, 0.0, 3.3, 3.1, 4.1

2922.8, 125.4, 2.801, .388, 0.0, 0.0, 3.4, 3.3, 4.2

2923.4, 126.0, 2.767, .356, 0.0, 0.0, 3.5, 3.4, 4.4

2923.9, 111.0, 2.756, .323, 0.0, 0.0, 2.8, 3.4, 4.5

2924.4, 106.0, 2.724, .325, 0.0, 0.0, 1.9, 3.2, 4.2

2924.9, 112.9, 2.714, .342, 0.0, 0.0, 2.2, 2.8, 3.5

-1

.BOTTOM.

?RF

WELLM Closed 58 lines

!

FILE NAME : WELLCN

.TOP.

WELL A1(2333.8m TO 2354.2m TVDSS)

000118-04-84

4,34.5

2896.0,2365.00,35.0,0.0

2900.0,2368.28,35.0,0.0

2920.0,2384.66,35.0,0.0

2940.0,2401.05,35.0,0.0

-1

.BOTTOM.

?RF

WELLCN Closed 8 lines

!

FILE NAME : WELLTV

.TOP.

WELL A1(2333.8M TO 2354.2M TVDSS)

000118-04-84

1,1,1,2,1,1,1,11,2,6,0

50, 88.0, .209, .018, 1.0, 620.0, 0.000

50.1, 2.650, 0.0, 0.0

197.5, 10.0, 2.900, 0.0, .160, 0.0

.0019, .800, 9.875

2333.8, 101.1, 2.436, .285, 0.0, 0.0, 1.2, 1.2, 1.2

2334.3, 100.6, 2.611, .260, 0.0, 0.0, 2.2, 1.4, 1.5

2334.7, 101.8, 2.535, .215, 0.0, 0.0, 2.0, 1.5, 1.6

2335.1, 95.9, 2.467, .188, 0.0, 0.0, 1.9, 1.5, 1.5

2335.5, 99.6, 2.463, .222, 0.0, 0.0, 1.6, 1.4, 1.4

2335.9, 90.7, 2.476, .220, 0.0, 0.0, 1.6, 1.4, 1.5

2336.3, 93.6, 2.487, .212, 0.0, 0.0, 1.7, 1.4, 1.4

2336.7, 86.9, 2.505, .207, 0.0, 0.0, 1.8, 1.5, 1.6

2337.1, 90.4, 2.507, .188, 0.0, 0.0, 1.7, 1.3, 1.5

2337.5, 90.6, 2.514, .209, 0.0, 0.0, 1.4, 1.3, 1.3

2338.0, 86.8, 2.501, .202, 0.0, 0.0, 1.7, 1.4, 1.4

2338.4, 86.0, 2.493, .190, 0.0, 0.0, 1.8, 1.5, 1.6

2338.8, 92.4, 2.494, .204, 0.0, 0.0, 1.7, 1.6, 1.6

2339.2, 89.6, 2.494, .192, 0.0, 0.0, 2.5, 1.8, 1.8

2339.6, 84.3, 2.518, .153, 0.0, 0.0, 3.5, 2.0, 2.1

2340.0, 85.0, 2.533, .166, 0.0, 0.0, 2.7, 1.9, 2.0

2340.5, 92.7, 2.521, .173, 0.0, 0.0, 1.8, 1.8, 1.9

2340.9, 104.1, 2.497, .201, 0.0, 0.0, 1.6, 1.5, 1.4

2341.3, 93.3, 2.511, .213, 0.0, 0.0, 1.6, 1.4, 1.4

2341.7, 93.4, 2.511, .203, 0.0, 0.0, 1.8, 1.5, 1.6

2342.1, 103.6, 2.520, .205, 0.0, 0.0, 2.1, 1.5, 1.5

2342.5, 114.5, 2.538, .205, 0.0, 0.0, 2.0, 1.6, 1.6

2343.0, 134.6, 2.615, .228, 0.0, 0.0, 2.3, 1.7, 1.9

2343.4, 137.6, 2.738, .326, 0.0, 0.0, 2.2, 1.9, 2.0

2343.8, 121.4, 2.771, .371, 0.0, 0.0, 2.8, 2.4, 2.7

2344.2, 115.6, 2.790, .404, 0.0, 0.0, 2.7, 2.6, 2.8

2344.6, 106.8, 2.799, .400, 0.0, 0.0, 2.8, 2.7, 3.0

2345.0, 106.3, 2.779, .360, 0.0, 0.0, 2.3, 2.7, 2.9

2345.4, 112.1, 2.790, .350, 0.0, 0.0, 2.3, 2.4, 2.7

2345.8, 128.6, 2.790, .397, 0.0, 0.0, 2.5, 2.4, 2.6

2346.3, 121.0, 2.812, .420, 0.0, 0.0, 2.4, 2.6, 2.7

2346.7, 107.7, 2.847, .436, 0.0, 0.0, 2.6, 2.6, 3.0

2347.1, 103.9, 2.849, .447, 0.0, 0.0, 2.4, 2.6, 2.9

2347.5, 106.3, 2.824, .426, 0.0, 0.0, 2.8, 2.7, 3.1

2348.0, 116.7, 2.828, .378, 0.0, 0.0, 2.9, 2.9, 3.2

2348.4, 112.2, 2.809, .407, 0.0, 0.0, 3.2, 2.9, 3.2

2348.8, 111.7, 2.825, .407, 0.0, 0.0, 3.0, 3.0, 3.3

2349.2, 113.7, 2.816, .404, 0.0, 0.0, 3.0, 3.0, 3.4

2349.6, 113.9, 2.785, .389, 0.0, 0.0, 3.0, 3.0, 3.5

2350.0, 117.5, 2.756, .361, 0.0, 0.0, 3.0, 3.1, 3.5

2350.4, 110.5, 2.770, .364, 0.0, 0.0, 2.9, 2.9, 3.6

2350.8, 121.1, 2.758, .389, 0.0, 0.0, 2.8, 2.9, 3.3

2351.2, 121.5, 2.792, .388, 0.0, 0.0, 3.2, 3.1, 3.9

2351.6, 122.3, 2.799, .384, 0.0, 0.0, 3.2, 3.1, 4.0

2352.0, 122.1, 2.806, .376, 0.0, 0.0, 3.3, 3.1, 4.1

2352.5, 125.4, 2.801, .388, 0.0, 0.0, 3.4, 3.3, 4.2

2352.9, 126.0, 2.767, .356, 0.0, 0.0, 3.5, 3.4, 4.4

2353.4, 111.0, 2.756, .323, 0.0, 0.0, 2.8, 3.4, 4.5

2353.8, 106.0, 2.724, .325, 0.0, 0.0, 1.9, 3.2, 4.2

2354.2, 112.9, 2.714, .342, 0.0, 0.0, 2.2, 2.8, 3.5

-1

.BOTTOM.

?RF

WELLTV Closed 58 lines

!

(c) Computer Program TVD Listing

FTN77,Q,L,C

\$FILES(0,3)

PROGRAM TVD

THIS PROGRAM WAS DEVELOPED BY G.R.MORRISON IN APRIL 1984.

THIS PROGRAM CONVERTS EITHER:

(1) A LOG4 OR DW COMPATABLE DATA FILE (REFER RECORDS 1978/3 AND 1984/8 RESPECTIVELY)

OR (2) A PREVIOUSLY INTERPRETED WIRELINE LOG DATA FILE GENERATED BY THE DW PROGRAM;

OR (3) A LOG DATA FILE WITHOUT HEADING INFORMATION.

FROM MEASURED DEPTH TO TRUE VERTICAL DEPTH. A CONVERSION DATA FILE IS REQUIRED WHICH CAN BE CREATED FROM THE INFORMATION SUPPLIED ON A DIRECTIONAL SURVEY. THIS PROGRAM IS CAPABLE OF PROCESSING 300 SAMPLE POINTS WHICH IS EQUIVILANT TO 149.5 METRES (OR FEET) OF WIRELINE LOG DATA DIGITIZED AT 0.5 METRE (OR FEET) INTERVALS. THE FORMAT OF THE CONVERSION DATA FILE IS AS FOLLOWS:

LINE	DATA
0001	TITLE e.g. WELL A1(1000M TO 2000M TVDSS)
0002	RUN AND DATE e.g. 000118-01-84
0003	NUMBER OF SAMPLE POINTS (=NSP), DRILLING FLOOR ELEVATION
0004	MD(1) ,TVD(1) ,DEVDEG(1) ,DEVMIN(1)
0005	MD(2) ,TVD(2) ,DEVDEG(2) ,DEVMIN(2)
:	:
:	:
NSP+4	MD(NSP) ,TVD(NSP) ,DEVDEG(NSP) ,DEVMIN(NSP)

SEE DEFINITIONS BELOW.

A DEFINITION OF EACH PARAMETER USED IN THIS PROGRAM APPEARS BELOW

BORE= BOREHOLE DIAMETER

DATMD= MEASURED DEPTH DATA FILE DATE

DATTVD= TVD DATA FILE DATE

DENA= DENSITY LOG READINGS

DENHYD= DENSITY OF THE HYDROCARBON

DENMF= DENSITY OF THE MUD FILTRATE

DENSH= DENSITY OF THE SHALE

DEV= DEVIATION OF THE WELL FOR A PARTICULAR LOG INTERVAL

DEVDEG(I)= DEGREE SEGMENT OF THE WELL DEVIATION

DEVMIN(I)= MINUTE SEGMENT OF THE WELL DEVIATION

DFELV= DRILLING FLOOR ELEVATION

DFT= DATA FILE TYPE

DPH= DEPTHS FROM THE MD LOG

EQUATION= LOG INTERPRETATION EQUATION E.G. DUAL WATER

GRY= GAMMA RAY LOG READINGS


```

C      GRYMAX= MAXIMUM GAMMA RAY READING IN A 100% SHALE
C      GRYMIN= MINIMUM GAMMA RAY READING IN THE RESERVOIR SAND
C      IHYDR= LOG4/DW SWITCH
C      IPL= LOG4/DW SWITCH
C      IPORF= LOG4/DW SWITCH
C      IPRINT= LOG4/DW SWITCH
C      ISP= LOG4/DW SWITCH
C      LUN= LOG4/DW SWITCH
C      MD(I)= MEASURED DEPTH READ FROM THE DIRECTIONAL SURVEY
C      MDDAT= NAME OF THE MD TO TVD CONVERSION DATA FILE
C      MDPH= LOG4/DW SWITCH
C      MODEL= LOG INTERPRETATION MODEL E.G. SHALY-SAND
C      MTAC= LOG4/DW SWITCH
C      MTEM= LOG4/DW SWITCH
C      NAMMD= NAME OF THE EXISTING MEASURED DEPTH DATA FILE
C      NAMTVD= NAME OF THE TVD DATA FILE TO BE CREATED
C      NLTYPE= LOG4/DW SWITCH
C      NSP= NUMBER OF SAMPLE POINTS IN THE MD LOG DATA FILE
C      NSPTVD= NUMBER OF SAMPLE POINTS IN THE MD TO TVD CONVERSION DATA
C      PHIE= EFFECTIVE POROSITY
C      PMF= SALINITY OF THE MUD FILTRATE EXPRESSED AS A FRACTION
C      PORNLI= NEUTRON LOG POROSITY READINGS
C      PORNSH= POROSITY OF THE SHALE
C      PSP= SP LOG READINGS
C      RD= DEEP RESISTIVITY LOG READINGS
C      RM= SHALLOW RESISTIVITY LOG REWADINGS
C      RMF= RESISTIVITY OF THE MUD FILTRATE
C      RSH= RESISTIVITY OF THE SHALE
C      RUNMD= MEASURED DEPTH DATA FILE RUN NUMBER
C      RUNTVD= TVD DATA FILE RUN NUMBER
C      RW= FREE WATER RESISTIVITY
C      RXO= MUDCAKE RESISTIVITY LOG READINGS
C      SPMIN= SP READING IN A 100% SHALE
C      SSP= SP READING IN THE RESERVOIR SAND
C      SWE= EFFECTIVE WATER SATURATION
C      TAC= SONIC LOG READINGS
C      TACMA= SONIC LOG TRANSIT TIME OF THE RESERVOIR SAND
C      TACSH= SONIC LOG TRANSIT TIME OF A 100% SHALE
C      TEM= FORMATION TEMPERATURE
C      TTLMD= TITLE OF THE MEASURED DEPTH DATA FILE
C      TTLTVD= TITLE OF THE TVD DATA FILE
C      TVD(I)= TRUE VERTICAL DEPTHS FROM THE DIRECTIONAL SURVEY
C      TVDDPH= CONVERTED TRUE VERTICAL DEPTHS
C      VSH= VOLUME OF SHALE
C      -----
C      SET UP ARRAYS AND CHARACTERS
C      REAL MD(100),TVD(100),DEVDEG(100),DEVMIN(100)
C      CHARACTER*20 NAMMD
C      CHARACTER*40 TTLMD
C      CHARACTER*4 RUNMD
C      CHARACTER*8 DATMD

```

```

CHARACTER*20 NAMTVD
CHARACTER*20 MDDAT
CHARACTER*40 TTLTVD
CHARACTER*4 RUNTV
CHARACTER*8 DATTV
CHARACTER*40 MODEL
CHARACTER*40 EQUATION

```

```

C -----
C   DETERMINE THE TYPE OF DATA FILE TO BE CONVERTED
WRITE(1,5)
5   FORMAT(/,1X,'INPUT THE TYPE OF DATA FILE TO BE PROCESSED BY',1X
2,'ENTERING ONE OF THE BRACKETED NUMBERS BELOW:')
WRITE(1,6)
6   FORMAT(1X,'(1) PROGRAM LOG4 OR DW COMPATABLE DATA FILE')
WRITE(1,7)
7   FORMAT(1X,'(2) PROGRAM LGPLT COMPATABLE DATA FILE')
WRITE(1,8)
8   FORMAT(1X,'(3) A LOG DATA FILE WITHOUT HEADING INFORMATION')
READ(1,*)DFT
C -----
C   OPEN MEASURED DEPTH LOG DATA FILE
WRITE(1,10)
10  FORMAT(/,1X,'INPUT EXISTING MEASURED DEPTH LOG DATA FILE NAME'
2,1X,'(I.E. name:SC:CRT)')
READ(1,20)NAMMD
20  FORMAT(A20)
OPEN(4,FILE=NAMMD,STATUS='OLD')
C -----
C   OPEN NEW TVD LOG DATA FILE
WRITE(1,30)
30  FORMAT(/,1X,'INPUT NAME OF TVD LOG DATA FILE TO BE CREATED'
2,1X,'(I.E. name:SC:CRT)')
READ(1,40)NAMTVD
40  FORMAT(A20)
OPEN(5,FILE=NAMTVD,STATUS='NEW')
C -----
C   OPEN MD TO TVD CONVERSION DATA FILE
WRITE(1,50)
50  FORMAT(/,1X,'INPUT MD TO TVD CONVERSION DATA FILE NAME'
2,1X,'(I.E. name:SC:CRT)')
READ(1,60)MDDAT
60  FORMAT(A20)
OPEN(7,FILE=MDDAT,STATUS='OLD')
C -----
C   READ THE HEADING FROM THE MEASURED DEPTH LOG DATA FILE
IF(DFT.EQ.1.OR.DFT.EQ.3) THEN
READ(4,'(A40)')TTLMD
IF(DFT.EQ.3) THEN
WRITE(1,62)
62  FORMAT(10X,'ENTER THE NUMBER OF SAMPLE POINTS IN THE MD DATA'
2,1X,'FILE')

```



```

WRITE(5,125)NSP,MDPH
125  FORMAT(1X,I3,', ',I1)
END IF
C -----
C   CONVERT THE MD DATA TO TVD
127  DO 150 J=1,NSP
C
C   IF(DFT.EQ.1.OR.DFT.EQ.3) THEN
READ(4,*)DPH,GRY,DENA,PORNLI,TAC,PSP,RXO,RM,RD
END IF
C
C   IF(DFT.EQ.2) THEN
READ(4,*)DPH,VSH,PHIE,SWE
END IF
C
C   K=1
130  DIF=DPH-MD(K)
IF(DIF.GT.0.0) THEN
K=K+1
IF(K.GT.NSPTVD) STOP'INSUFFICIENT MD TO TVD DATA AT END'
GOTO 130
ELSE
IF(DIF.EQ.0.0) N=K
IF(DIF.LT.0.0) N=K-1
IF(N.LT.1) STOP'INSUFFICIENT MD TO TVD DATA AT TOP'
DEL=DPH-MD(N)
DEV=(DEVDEG(N)+DEVMIN(N)/60.0)*0.017453293
TVDDPH=TVD(N)+DEL*COS(DEV)-DFELV
END IF
IF(DFT.EQ.1.OR.DFT.EQ.3) THEN
WRITE(5,140)TVDDPH,GRY,DENA,PORNLI,TAC,PSP,RXO,RM,RD
140  FORMAT(1X,F6.1,', ',F5.1,', ',F5.3,', ',F4.3,', ',F5.1,', ',F5.1
2,', ',F6.1,', ',F6.1,', ',F6.1)
END IF
C
C   IF(DFT.EQ.2) THEN
WRITE(5,145)TVDDPH,VSH,PHIE,SWE
145  FORMAT(1X,F6.1,', ',F5.3,', ',F5.3,', ',F5.3)
END IF
C
150  CONTINUE
WRITE(5,160)
160  FORMAT(1X,'-1')
C -----
CLOSE(4)
CLOSE(5)
CLOSE(7)
STOP'NORMAL TERMINATION'
END
END$

```

III LGPLT

(a) Introduction

The object of program LGPLT is to produce a plot of pre-interpreted wireline log data. LGPLT accomplishes this task by accessing a data file created by program DW2, the dual water wireline log interpretation model (note that DW2 is the modified version of program DW, BMR record 1984/8). An example of the type of data file generated by program DW2 is shown on page 16. An example of the plot generated by program LGPLT is shown on pages 17 to 20. Usually the plot is produced on a continuous length of computer paper by the HP2361G plotter but here it has been split over several A4 pages for presentation in this record.

The format of the plot is a standard 1:200 scale three track log which can be overlayed with the original digitised 1:200 scale wireline logs being analysed. The three tracks on the log are labelled (from left to right) formation analysis, porosity and fluid analysis, and pore fluid analysis. The first track gives an overview of the entire formation showing volume of shale, porosity, fluid saturation and coal seams. The second track is used to read porosity from an 0 to 50 percent scale. The second track still indicates the amount of fluid saturation in the pores and coal seams. Track three is used to show the proportion of water and liquid hydrocarbon residing in the pore spaces of the formation.

Above the three tracks is a block of information detailing the well name, run number and date of run, the equation and model used for the interpretation, the lithology and the pore fluid codes.

Formation characteristics of interest displayed by the plot shown on pages 17 to 20 are:

- (i) coal seams at 2940, 2942, 2957 and 3001 metres;
- (ii) a residual oil zone in the interval 2960 to 2985 metres;
- (iii) two hydrocarbon bearing reservoirs from 2933 to 2940 metres and 2947 to 2957 metres separated by a coal shale formation.

Much of this type of information is usually lost in the numerical output generated by both log interpretation programs LOG4 and DW2.

(b) Operating Instructions

The program is compiled, loaded and run in the way indicated on page 16.

To run the program, the only user input required is entry of the data file name at the computer terminal. Note that because of the complexity of this plot it may take the computer up to half an hour (run time) to plot 150 metres of data with sample points spaced at half metre intervals.

FT,&LGPLT,,ZLGPLT ← compile the program
\$END FTN7X: No disasters, No errors, No warnings.
!

RU,LOADR ← load the program
/LOADR: REL,ZLGPLT
/LOADR: EN
/LOADR:LGPLT READY AT 2:46 PM THU., 16 AUG., 1984
/LOADR:\$END
!

RU,LGPLT ← run the program

INPUT THE DATA FILE NAME
WELLPL
NORMAL TERMINATION
! ←

program LGPLT replies with "normal termination" when the job has been completed. The computer then replies with the cursor (!).

Program LGPLT compilation, loading and run.
(NOTE: the underlined type indicates user input.)

An example of the type of data file generated by DW2 and accessed by LGPLT

```

0001 WELL A1(2900.0m TO 2924.9mMDKB)
0002 SHALY-SAND
0003 DUAL WATER
0004 000112-04-84
0005 50,1
0006 2900.0, .346, .167,1.000
0007 2900.6,1.000,0.000,1.000
0008 2901.1, .351, .103,1.000
0009 2901.6, .311, .115,1.000
0010 2902.1, .336, .129,1.000
0011 2902.6, .275, .134,1.000
0012 2903.1, .295, .124,1.000
0013 2903.6, .250, .123,1.000
0014 2904.1, .273, .110,1.000
0015 2904.6, .275, .118,1.000
0016 2905.1, .249, .122,1.000
0017 2905.6, .244, .119,1.000
0018 2906.1, .287, .119,1.000
0019 2906.6, .268, .116,1.000
0020 2907.1, .232, .096,1.000
0021 2907.6, .237, .097,1.000
0022 2908.2, .289, .096,1.000
0023 2908.7, .366, .104,1.000
0024 2909.2, .293, .118,1.000
0025 2909.7, .294, .113,1.000
0026 2910.2, .363, .100,1.000
0027 2910.7, .437, .083,1.000
0028 2911.2, .573, .050,1.000
0029 2911.7,1.000,0.000,1.000
0030 2912.2,1.000, .008,1.000
0031 2912.7,1.000, .019,1.000
0032 2913.2,1.000, .014,1.000
0033 2913.7,1.000,0.000,1.000
0034 2914.2,1.000,0.000,1.000
0035 2914.7,1.000, .015,1.000
0036 2915.3,1.000, .020,1.000
0037 2915.8,1.000, .018,1.000
0038 2916.3,1.000, .023,1.000
0039 2916.8,1.000, .020,1.000
0040 2917.3,1.000,0.000,1.000
0041 2917.8,1.000, .015,1.000
0042 2918.3,1.000, .010,1.000
0043 2918.8,1.000, .011,1.000
0044 2919.3,1.000, .013,1.000
0045 2919.8,1.000, .007,1.000
0046 2920.3,1.000, .004,1.000
0047 2920.8,1.000, .020,1.000
0048 2921.3,1.000, .010,1.000
0049 2921.8,1.000, .006,1.000
0050 2922.3,1.000,0.000,1.000
0051 2922.8,1.000, .007,1.000
0052 2923.4,1.000, .001,1.000
0053 2923.9,1.000,0.000,1.000
0054 2924.4,1.000,0.000,1.000
0055 2924.9,1.000, .010,1.000
0056 -1

```

WELL: WELL A1(2900 TO 3010M MDKB)WELLMD

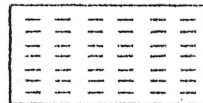
MODEL: SHALY-SAND

EQUATION: DUAL WATER

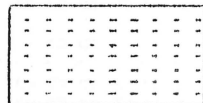
RUN NUMBER: 0001

DATE: 12-04-84

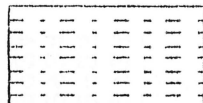
LITHOLOGY CODES:



SHALE



RESERVOIR SAND



MATRIX = 1 - POROSITY

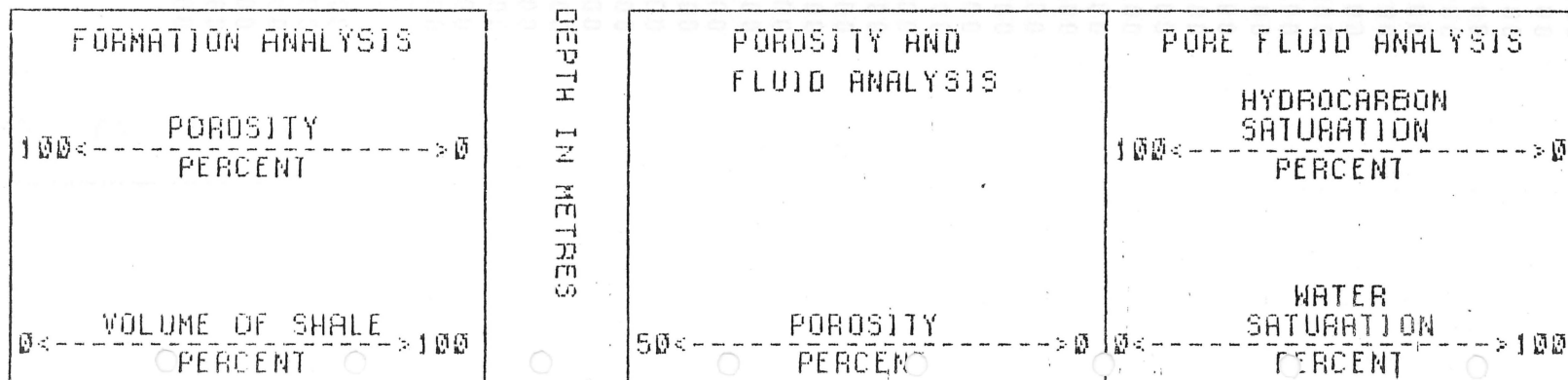
PORE FLUID CODES:



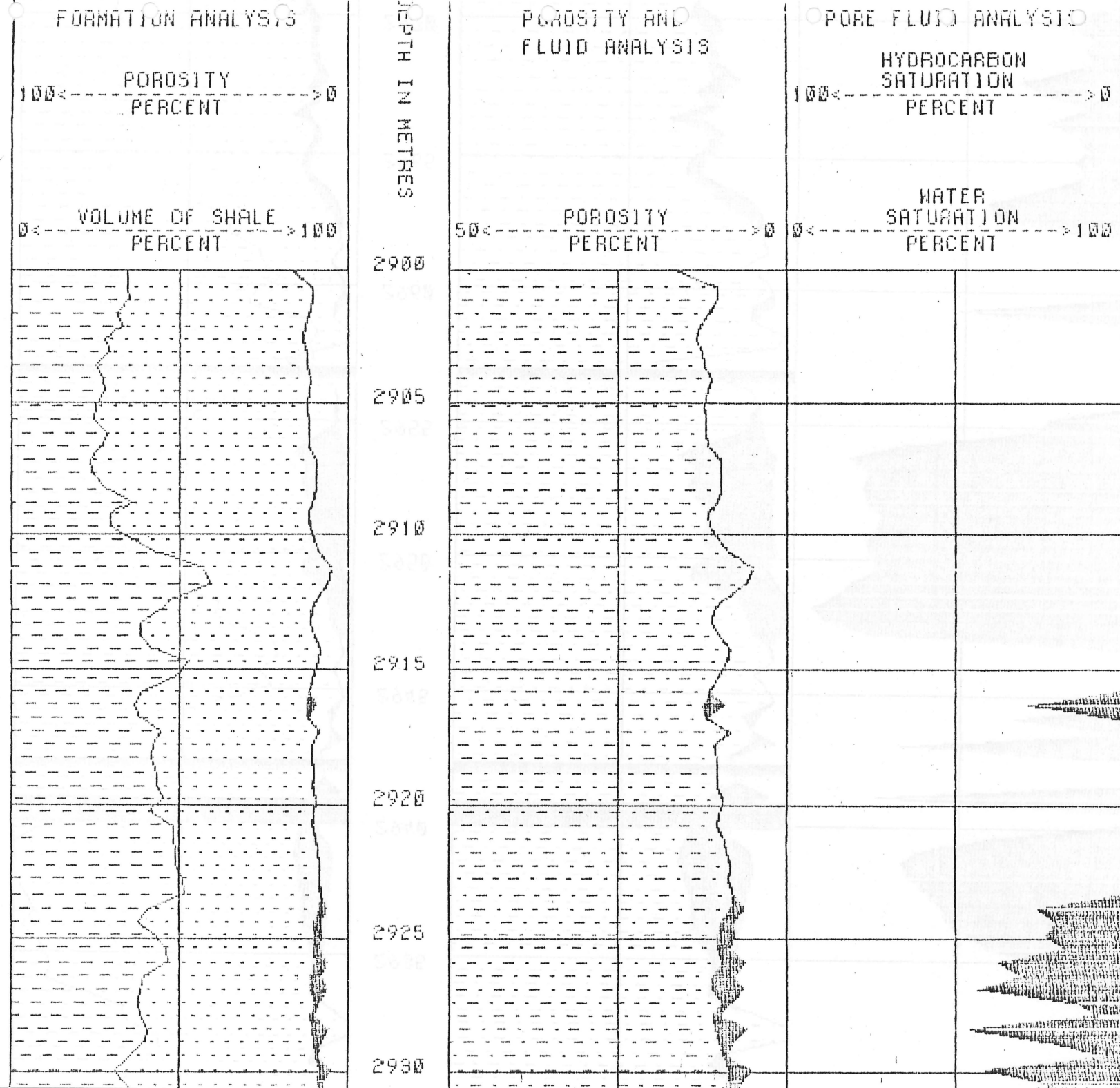
WATER

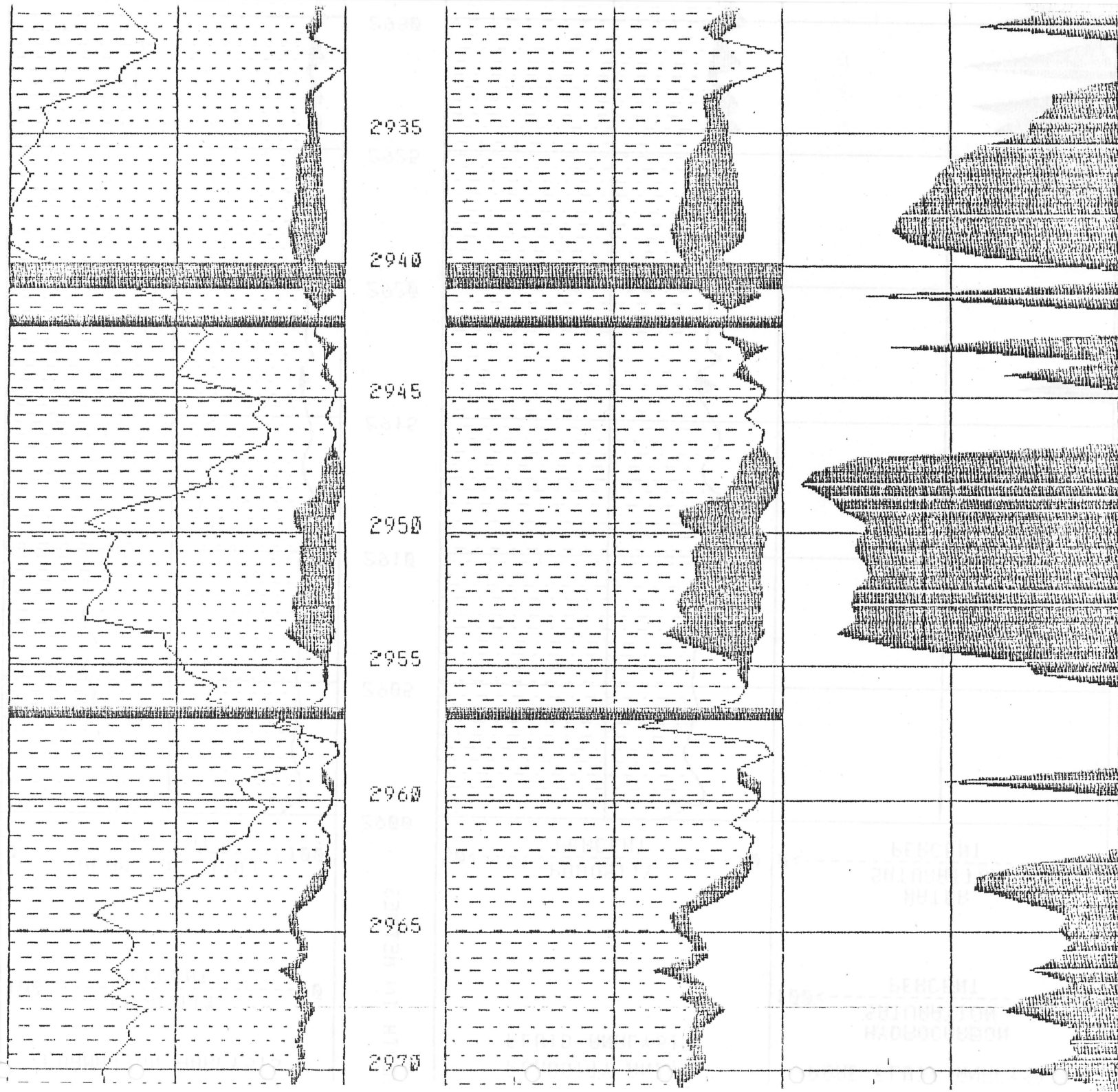


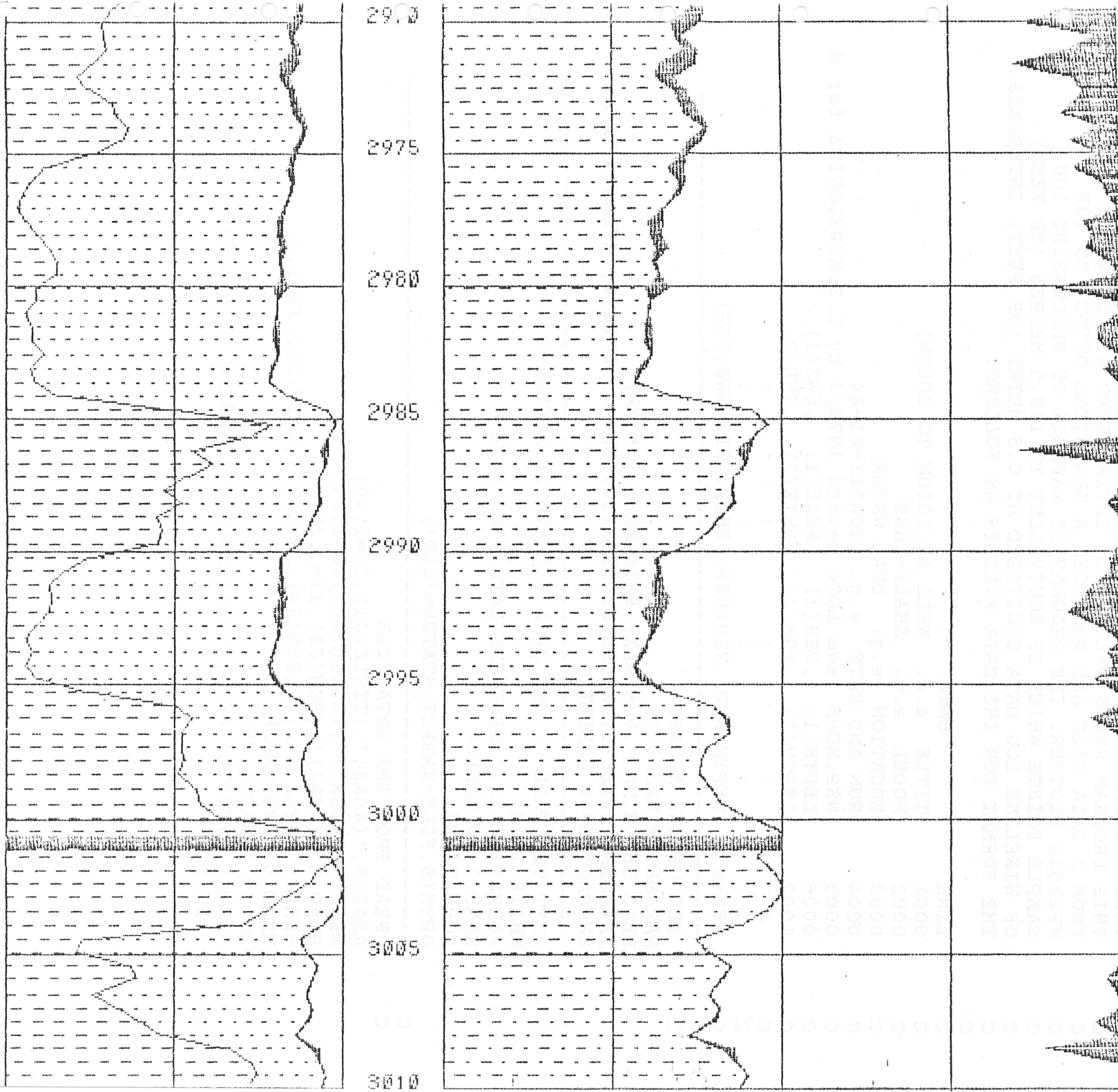
HYDROCARBON
(OR COAL)



An example of the type of plot generated by program LGPLT
(continued for the next three pages)







(c) Computer program LGPLT listing

\$FILES(0,1)

PROGRAM LGPLT

```

C
C   THIS PROGRAM WAS DEVELOPED BY G.R.MORRISON IN MAY 1984.
C   THIS PROGRAM READS PREVIOUSLY INTERPRETED WIRELINE LOG DATA
C   FROM A DATA FILE AND PRODUCES A GRAPHICAL OUTPUT ON THE
C   HP2361G PLOTTER. THE PROGRAM IS CAPABLE OF PROCESSING 300
C   SAMPLE POINTS WHICH IS EQUIVALENT TO 149.5 METRES (OR FEET)
C   OF WIRELINE LOG DATA DIGITIZED AT 0.5 METRE (OR FEET) INTERVALS.
C   THE FORMAT FOR THE DATA FILE IS AS FOLLOWS:
C
C   LINE          DATA
C   0001          TITLE e.g. WELL A1(1000M TO 2000M)
C   0002          MODEL e.g. SHALY-SAND
C   0003          EQUATION e.g. DUAL WATER
C   0004          RUN AND DATE e.g. 000218-01-84
C   0005          NSP,MDPH see LOG4 record 1978/3 by L.E.KURYLOWICZ for d
C   0006          DEPTH(1) ,VSH(1) ,PHIE(1) ,SWE(1)
C   0007          DEPTH(2) ,VSH(2) ,PHIE(2) ,SWE(2)
C   :             :             :             :             :
C   :             :             :             :             :
C   NSP+6         DEPTH(NSP) ,VSH(NSP) ,PHIE(NSP) ,SWE(NSP)
C   -----
C   SET UP THE ARRAYS
C   PARAMETER(MAX=300)
C   REAL DEPTH(MAX) ,VSH(MAX) ,PHIE(MAX) ,SWE(MAX)
C   INTEGER TITLE(40) ,MODEL(20) ,EQN(20) ,RUN(2) ,DATE(4) ,CFLAG ,SFLAG
C   CHARACTER*20 NAMPLT
C   -----
C   GET THE NAME OF THE DATA FILE TO BE ACCESSED
C   WRITE(1,10)
10  FORMAT(//,1X,'INPUT THE DATA FILE NAME')
C   READ(1,15)NAMPLT
15  FORMAT(A20)
C   OPEN(5,FILE=NAMPLT,STATUS='OLD')
C   -----
C   READ FROM THE DATA FILE
C   READ(5,'(40A2)')(TITLE(J1),J1=1,40)
C   READ(5,'(20A2)')(MODEL(J2),J2=1,20)
C   READ(5,'(20A2)')(EQN(J3),J3=1,20)
C   READ(5,'(2A2,4A2)')(RUN(J4),J4=1,2),(DATE(J5),J5=1,4)
C   READ(5,*)NSP,MDPH
C   DO 20 IREAD=1,NSP
C   READ(5,*)DEPTH(IREAD) ,VSH(IREAD) ,PHIE(IREAD) ,SWE(IREAD)
20  CONTINUE

```

C -----
C FIND THE TOP AND BOTTOM DEPTHS FOR TRACK SIZES

```

IDT=DEPTH(1)/10.0
DPHT=IDT*10.0
IDB=DEPTH(NSP)/10.0+1.0
DPHB=IDB*10.0
DPHCON=0.19765625
IF(MDPH.EQ.0)DPHCON=DPHCON*0.254
XMAX=(DPHB-DPHT)*DPHCON
CALL PLOTS(XMAX+20.0,10.0,10)

```

C -----
C PLOT OUT THE HEADINGS

```

CALL PLOT(0.6,2.0,3)
CALL ORIGI(0.6,2.0,0)
CALL SYMB(0.0,0.0,0.1,5HWELL:,90.0,5)
CALL SYMB(0.0,0.75,0.1,TITLE,90.0,80)
CALL SYMB(0.1,0.0,0.1,5H-----,90.0,5)
CALL SYMB(0.4,0.0,0.1,6HMODEL:,90.0,6)
CALL SYMB(0.4,0.75,0.1,MODEL,90.0,40)
CALL SYMB(0.5,0.0,0.1,6H-----,90.0,6)
CALL SYMB(0.8,0.0,0.1,9HEQUATION:,90.0,9)
CALL SYMB(0.8,1.0,0.1,EQN,90.0,40)
CALL SYMB(0.9,0.0,0.1,9H-----,90.0,9)
CALL SYMB(1.2,0.0,0.1,11HRUN NUMBER:,90.0,11)
CALL SYMB(1.2,1.25,0.1,RUN,90.0,4)
CALL SYMB(1.3,0.0,0.1,11H-----,90.0,11)
CALL SYMB(1.6,0.0,0.1,5HDATE:,90.0,5)
CALL SYMB(1.6,0.75,0.1,DATE,90.0,8)
CALL SYMB(1.7,0.0,0.1,5H-----,90.0,5)
CALL SYMB(2.2,0.0,0.1,16HLITHOLOGY CODES:,90.0,16)
CALL SYMB(2.3,0.0,0.1,16H-----,90.0,16)
CALL SYMB(2.2,4.5,0.1,17HPORE FLUID CODES:,90.0,17)
CALL SYMB(2.3,4.5,0.1,17H-----,90.0,17)

```

C -----
C PLOT OUT THE LITHOLOGY CODES

```

PLOT OUT SHALE I.D.
CALL ORIGI(3.0,2.0,0)
CALL PLOT(0.0,0.0,3)
CALL PLOT(0.5,0.0,2)
CALL PLOT(0.5,1.0,2)
CALL PLOT(0.0,1.0,2)
CALL PLOT(0.0,0.0,2)
CALL SYMB(0.25,1.1,0.1,5HSHALE,90.0,5)
XDASH=0.0
DO 21 IDASH=1,7
XDASH=XDASH+0.0625
YDASH=0.0
DO 22 JDASH=1,6
YDASH=YDASH+0.16
CALL PLOT(XDASH,YDASH-0.08,3)

```

```

22 CALL PLOT(XDASH,YDASH,2)
21 CONTINUE
C CONTINUE
C PLOT OUT RESERVOIR SAND I.D.
CALL PLOT(1.0,0.0,3)
CALL ORIGI(1.0,0.0,1)
CALL PLOT(0.5,0.0,2)
CALL PLOT(0.5,1.0,2)
CALL PLOT(0.0,1.0,2)
CALL PLOT(0.0,0.0,2)
CALL SYMB(0.25,1.1,0.1,14HRESERVOIR SAND,90.0,14)
XDOT=0.0
DO 23 IDOT=1,7
XDOT=XDOT+0.0625
YDOT=0.0
DO 24 JDOT=1,9
YDOT=YDOT+0.11
CALL PLOT(XDOT,YDOT-0.03,3)
CALL PLOT(XDOT,YDOT,2)
24 CONTINUE
23 CONTINUE
C
C PLOT OUT THE MATRIX I.D.
CALL PLOT(1.0,0.0,3)
CALL ORIGI(1.0,0.0,1)
CALL PLOT(0.5,0.0,2)
CALL PLOT(0.5,1.0,2)
CALL PLOT(0.0,1.0,2)
CALL PLOT(0.0,0.0,2)
CALL SYMB(0.25,1.1,0.1,21HMATRIX = 1 - POROSITY,90.0,21)
XDD=0.0
DO 25 IDD=1,7
XDD=XDD+0.0625
YDD=0.0
DO 26 JDD=1,3
CALL PLOT(XDD,YDD,3)
CALL PLOT(XDD,YDD+0.08,2)
CALL PLOT(XDD,YDD+0.16,3)
CALL PLOT(XDD,YDD+0.19,2)
CALL PLOT(XDD,YDD+0.27,3)
YDD=YDD+0.27
26 CONTINUE
CALL PLOT(XDD,YDD+0.08,2)
CALL PLOT(XDD,YDD+0.16,3)
CALL PLOT(XDD,YDD+0.19,2)
25 CONTINUE
C
C PLOT OUT THE WATER I.D.
CALL PLOT(-2.0,4.5,3)
CALL ORIGI(-2.0,4.5,1)

```

```

CALL PLOT(0.5,0.0,2)
CALL PLOT(0.5,1.0,2)
CALL PLOT(0.0,1.0,2)
CALL PLOT(0.0,0.0,2)
CALL SYMB(0.25,1.1,0.1,5HWATER,90.0,5)

```

C
C

```

PLOT OUT THE HYDROCARBON I.D.
CALL PLOT(1.0,0.0,3)
CALL ORIGI(1.0,0.0,1)
CALL PLOT(0.5,0.0,2)
CALL PLOT(0.5,1.0,2)
CALL PLOT(0.0,1.0,2)
CALL PLOT(0.0,0.0,2)
CALL SYMB(0.25,1.1,0.1,11HHYDROCARBON,90.0,11)
CALL SYMB(0.4,1.1,0.1,9H(OR COAL),90.0,9)
XSHADE=0.0
DO 27 ISHADE=1,50
XSHADE=XSHADE+0.01
CALL PLOT(XSHADE,0.0,3)
CALL PLOT(XSHADE,1.0,2)
CONTINUE

```

27
C
C

```

PLOT OUT TRACK ONE
CALL ORIGI(6.0,1.0,0)
CALL PLOT(0.0,0.0,3)
CALL PLOT(2.0,0.0,2)
CALL PLOT(2.0,2.5,3)
CALL PLOT(0.0,2.5,2)
CALL PLOT(0.0,0.0,2)

```

```

C PLOT SCALES ON TOP OF TRACK ONE
CALL SYMB(0.2,0.34,0.1,18HFORMATION ANALYSIS,90.0,18)
CALL SYMB(0.65,0.85,0.1,8HPOROSITY,90.0,8)
CALL SYMB(0.75,0.05,0.1,24H100<----->0,90.0,24)
CALL SYMB(0.85,0.9,0.1,7HPERCENT,90.0,7)
CALL SYMB(1.65,0.5,0.1,15HVOLUME OF SHALE,90.0,15)
CALL SYMB(1.75,0.05,0.1,24H0<----->100,90.0,24)
CALL SYMB(1.85,0.9,0.1,7HPERCENT,90.0,7)
IF(MDPH.EQ.1) THEN
CALL SYMB(0.0,2.875,0.1,15HDEPTH IN METRES,0.0,15)
ELSE
CALL SYMB(0.0,2.875,0.1,13HDEPTH IN FEET,0.0,13)
END IF
CALL PLOT(2.0,0.0,3)
CALL ORIGI(2.0,0.0,1)
CALL PLOT(XMAX,0.0,2)
CALL PLOT(XMAX,2.5,2)
CALL PLOT(0.0,2.5,2)
CALL PLOT(0.0,0.0,2)
CALL PLOT(0.0,1.25,3)
CALL PLOT(XMAX,1.25,2)

```

C

```

C      PLOT 5 METRE OR 20 FOOT LINES AND DEPTH NUMBERS
      DL=DPHT-5.0
      DL=DPHT-5.0
30     DL=DL+5.0
      IF(DL.GT.DPHB) THEN
      GOTO 35
      ELSE
      XDL=(DL-DPHT)*DPHCON
      CALL PLOT(XDL,0.0,3)
      CALL PLOT(XDL,2.5,2)
      CALL PLOT(XDL,3.25,3)
      CALL PLOT(XDL,8.25,2)
      CALL NUMB(XDL,2.7,0.1,DL,90.0,0)
      GOTO 30
      END IF

C      -----
C      PLOT OUT THE CURVES IN TRACK ONE
35     DELDPH=DEPTH(1)-DPHT
      XT=DELDPH*DPHCON

C      PLOT OUT THE VOLUME OF SHALE CURVE
C      YVT=VSH(1)*2.5
      CALL PLOT(XT,YVT,3)
      CFLAG=0
      DO 40 IV=2,NSP
C      SET FLAG IF COAL IS PRESENT
      IF(VSH(IV).EQ.2.0.AND.PHIE(IV).EQ.0.0.AND.SWE(IV).EQ.1.0) THEN
      CFLAG=CFLAG+1
      GOTO 40
      END IF
      X=(DEPTH(IV)-DEPTH(1)+DELDPH)*DPHCON
      Y=VSH(IV)*2.5
      CALL PLOT(X,Y,2)
C      IF COAL IS PRESENT THEN BLOCK OUT TRACKS 1 AND 2
      IF(CFLAG.GT.0) THEN
      XTCOAL=((DEPTH(IV-CFLAG)+DEPTH(IV-CFLAG-1))/2.0-DEPTH(1)+DELDPH)
2*DPHCON
      XBCOAL=((DEPTH(IV)+DEPTH(IV-1))/2.0-DEPTH(1)+DELDPH)*DPHCON
37     CALL PLOT(XTCOAL,0.0,3)
      CALL PLOT(XTCOAL,2.5,2)
      CALL PLOT(XTCOAL,3.25,3)
      CALL PLOT(XTCOAL,5.75,2)
      XTCOAL=XTCOAL+0.01
      IF(XTCOAL.LE.XBCOAL) GOTO 37
      CFLAG=0
      CALL PLOT(X,Y,3)
      END IF

C      PLOT OUT DASH CODING INSIDE VOLUME OF SHALE CURVE
      YDASH=Y
38     YDASH=YDASH-0.08
      IF(YDASH.LT.0.0) THEN

```

IV PLOG

(a) Introduction

Program 'PLOG' improves upon the plotting routine used in program 'XYPLT' (De Nardi, 1983/15). Apart from the standard 1:200 scale plot, program 'PLOG' has the added facility to compact the plot in the vertical and horizontal directions.

Figure 1 indicates that input to program 'PLOG' may be of two data file types. These two types are:

- (i) a data file generated by program XYPLT;
- (ii) a data file containing the log interpretation parameters determined by program RW.

The plots generated by program PLOG are produced on BMR's Hewlett-Packard plotter; HP2631G while the program itself uses BMR's Hewlett-Packard computer (HP1000F series).

(b) Operating Instructions

The program has been designed to operate in an interactive manner. Program 'PLOG' will request specific information from the operator. If any responses are invalid or unrecognisable by the program, then suitable 'help' comments will appear. Also, there are a number of internal checks within the program that guard against processing of corrupt data.

To avoid any confusion, the format and nomenclature appearing in the plots are similar to that used on the standard wireline logs. The combinations of wireline logs which may be plotted are limited to the following three suites:

	<u>Box 1</u>	<u>Box 2</u>
Suite 1	gamma	Density/Porosity
Suite 2	gamma/SP	Resistivity
Suite 3	gamma/SP	Sonic

A sample run starts on the next page:

Sample Run:

Appearing below is an example run using program 'PLOG':

NOTE: All users responses are underlined.

assuming the user has 'logged on', type in:

!RP,PLOG

! Restores the program to
main memory.

!RU,PLOG

! Begin program execution.

PLEASE ENTER DATA FILE NAME

DATA1

IS THIS DATA FILE LOG4 COMPATIBLE (Y/N)

Y

DO YOU WISH TO CONVERT THE LOG DEPTHS
TO ANOTHER DATUM (Y/N)

Y

Enter the distance from the current log zero depth
to the new datum depth

30.0

The data is now being read in, it won't take very long.

ENTER SCALE FOR X DIRECTION. (available range 200 1000) 1:

200

ENTER SCALING FACTOR FOR Y DIRECTION
(available range 0.25 1.0)

1.0

THERE ARE 3 SUITES OF LOGS AVAILABLE :-

	TRACK 1	TRACK 2
Suite1	GAMMA RAY	DENSITY/NEUTRON
Suite2	GAMMA/SP	RESISTIVITY
Suite3	GAMMA/SP	SONIC

SELECT THE SUITE OF LOGS YOU REQUIRE BY ENTERING :-
1 , 2 OR 3

Entering a 4 gives you the option
of selecting the X-Y scales again

1

The following scale values are pre-set for suite 1

LOG	SCALE
GAMMA RAY	0 - 200
DENSITY	1.95 - 2.95
POROSITY	0.45 - -0.15

IS THERE ANY SCALE THAT NEEDS TO BE CHANGED (Y/N)

N

Please wait
I'll be back shortly

Nearly finished

DO YOU WANT TO PLOT ANOTHER SUITE OF LOGS (Y/N)

Y

ENTER SCALE FOR X DIRECTION. (available range 200 1000) 1:

500

ENTER SCALING FACTOR FOR Y DIRECTION
(available range 0.25 1.0)

0.5

THERE ARE 3 SUITES OF LOGS AVAILABLE :-

	TRACK 1	TRACK 2
Suite1	GAMMA RAY	DENSITY/NEUTRON
Suite2	GAMMA/SP	RESISTIVITY
Suite3	GAMMA/SP	SONIC

SELECT THE SUITE OF LOGS YOU REQUIRE BY ENTERING :-
1 , 2 OR 3

Entering a 4 gives you the option
of selecting the X-Y scales again

2

The following scale values are pre-set for suite 2

LOG	SCALE
GAMMA RAY	0 - 200
SP scale is calculated from the data.	

RESISTIVITY

0.2 - 2000

IS THERE ANY SCALE THAT NEEDS TO BE CHANGED (Y/N)

N

Please wait
I'll be back shortly

SP Log NOT plotted (ie. no values)

Nearly finished

DO YOU WANT TO PLOT ANOTHER SUITE OF LOGS (Y/N)

N

DO YOU WANT TO ENTER ANOTHER DATA FILE (Y/N)

Y

PLEASE ENTER DATA FILE NAME

DATA2

IS THIS DATA FILE LOG4 COMPATIBLE (Y/N)

N

IS THIS WELL IN FT OR M
ENTER FT or M

FT

Enter todays date as follows XX-XX-XX

08-08-84

Do you want to convert the depths to METRES

Y

DO YOU WISH TO CONVERT THE LOG DEPTHS
TO ANOTHER DATUM (Y/N)

N

The data is now being read in, it won't take very long.

ENTER SCALE FOR X DIRECTION. (available range 200 1000) 1:

200

ENTER SCALING FACTOR FOR Y DIRECTION
(available range 0.25 1.0)

1.0

THERE ARE 3 SUITES OF LOGS AVAILABLE :-

	TRACK 1	TRACK 2
Suite1	GAMMA RAY	DENSITY/NEUTRON
Suite2	GAMMA/SP	RESISTIVITY
Suite3	GAMMA/SP	SONIC

SELECT THE SUITE OF LOGS YOU REQUIRE BY ENTERING :-
1 , 2 OR 3

Entering a 4 gives you the option
of selecting the X-Y scales again

3

The following scale values are pre-set for suite 3

LOG	SCALE
GAMMA RAY	0 - 200
SP scale is calculated from the data.	
SONIC	140 - 40

IS THERE ANY SCALE THAT NEEDS TO BE CHANGED (Y/N)

N

Please wait
I'll be back shortly

SP Log NOT plotted (ie. no values)

Nearly finished

SONIC Log NOT plotted (ie. no values)

DO YOU WANT TO PLOT ANOTHER SUITE OF LOGS (Y/N)

N

DO YOU WANT TO ENTER ANOTHER DATA FILE (Y/N)

N

NORMAL TERMINATION

PLOG LUN 06 SPOOL FILE -0214

(c) Program 'PLOG' Source Listing

FTN77,Q,L,C

\$FILES(0,1)

PROGRAM PLOG

C *****

C FUNCTION : Produce a plot of digitised wire-line log data.

C Allow for compaction in both 'X' and 'Y' directions.

C INPUT : Output data file created by LOG4 or XYPLT programs.

C OUTPUT : Plot of wire-line logs in standard or compacted form.

C There are three suite of logs available :

C logs available.

C TRACK 1

C 1. GAMMA RAY

C 2. GAMMA/SP

C 3. GAMMA/SP

C TRACK 2

C DENSITY/NEUTRON

C RESISTIVITY

C SONIC

C AUTHOR : Rob DeNardi (X9755).

C DATE : May 1984.

C *****

C PARAMETER (LARG=300,LUN=6)

C REAL DPH(LARG),GRY(LARG),DENS(LARG),POR(LARG),SP(LARG),SONIC(LARG)

C * ,Msfl(LARG),RLS(LARG),RLD(LARG),Len1,Len2

C INTEGER Title(25),Date(4),Suite,Ptype

C CHARACTER*1 NAME*12,DUMMY,YENO1,YENO,LINTYP*5,METFT,LOGTYP,

C * LOG*8,ZETA

C

C 5 WRITE(1,10)

C 10 FORMAT(//,20X,'PLEASE ENTER DATA FILE NAME')

C READ(1,'(A)')NAME

C

C OPEN(4,FILE=NAME,STATUS='OLD')

C

C FIND OUT WHICH PROGRAM PRODUCED THE DATA FILE

C ie. LOG4 OR XYPLT

C

C WRITE(1,40)

C 40 FORMAT(//,15X,'IS THIS DATA FILE LOG4 COMPATIBLE (Y/N)')

C READ(1,'(A1)')YENO1

C IF(YENO1.NE.'Y')THEN

C WRITE(1,50)

C 50 FORMAT(//,23X,'IS THIS WELL IN FT OR M',/,28X,


```

*      'ENTER FT or M')
      READ(1, '(A1)') METFT
      WRITE(LUN, 90) METFT
      WRITE(1, 60)
60     FORMAT(//, 17X, 'Enter todays date as follows XX-XX-XX')
      READ(1, '(4A2)') (Date(J), J=1, 4)
      WRITE(LUN, 70) (Date(J), J=1, 4)
70     FORMAT(10X, 'Date=', 4A2)
      READ(4, '(25A2)') (Title(J), J=1, 25)
      WRITE(LUN, 80) (Title(J), J=1, 25)
80     FORMAT(10X, 'Title=', 25A2)
      ELSE
        READ(4, '(25A2)') (Title(J), J=1, 25)
        WRITE(LUN, 80) (Title(J), J=1, 25)
        READ(4, '(4X, 4A2)') (Date(J), J=1, 4)
        WRITE(LUN, 70) (Date(J), J=1, 4)
        READ(4, '(9X, A1)') METFT
        WRITE(LUN, 90) METFT
90     FORMAT(10X, 'METFT=', A1)
        READ(4, *) NSP
        WRITE(LUN, 100) NSP
100    FORMAT(10X, 'NSP=', I4)
        DO 200 I=1, 3
          READ(4, '(A1)') DUMMY
200    CONTINUE
      END IF
C
C
      FTMET=1.0
      AFFACT=1.0
      IF (METFT.EQ. 'F'.OR.METFT.EQ. 'f'.OR.METFT.EQ. '0') THEN
        WRITE(1, 220)
220    FORMAT(//, 13X, 'Do you want to convert the depths to ',
*          'METRES')
        READ(1, '(A1)') YENO
        IF (YENO.EQ. 'Y') THEN
          FTMET=0.3048
          AFFACT=0.254/0.3048
          METFT='1'
          WRITE(LUN, 230) FTMET, METFT
230    FORMAT(10X, 'FTMET=', F7.4, 2X, 'METFT= ', A)
        ELSE
          FTMET=1.0
          AFFACT=1.0
        END IF
      END IF
C
C
      CHANGE LOG DEPTHS TO ANOTHER DATUM IF NECESSARY
      WRITE(1, 250)
250  FORMAT(//, 15X, 'DO YOU WISH TO CONVERT THE LOG DEPTHS ',

```

```

*          /,22X,'TO ANOTHER DATUM (Y/N)')
READ(1,'(A1)')YENO
IF(YENO.EQ.'Y')THEN
  WRITE(1,260)
260  FORMAT(/,12X,'Enter the distance from the current log',
*      ' zero depth',/,23X,'to the new datum depth')
  READ(1,*)DTND
ELSE
  DTND=0.0
END IF

```

C
C
C

COPY DATA INTO ARRAYS

```

WRITE(1,280)
280  FORMAT(/,10X,'The data is now being read in, it won't ',
*      'take very long.',/)
NUM=0
300  READ(4,*,END=400)A,B,C,D,E,F,G,H,P
  NUM=NUM+1
  DPH(NUM)=(A-DTND)*FTMET
  WRITE(LUN,310)DPH(NUM)
310  FORMAT(10X,'DPH= ',F10.4)
  GRY(NUM)=B
  DENS(NUM)=C
  POR(NUM)=D
  SONIC(NUM)=E
  SP(NUM)=F
  Msfl(NUM)=G
  RLS(NUM)=H
  RLD(NUM)=P
  GOTO 300
400  CLOSE(4)
  WRITE(LUN,410)NUM
410  FORMAT(10X,'NUM=',I4)

```

C
C
C
C

NOTE: If NSP does not equal NUM for a LOG4 type data file,
then NSP is the value used rather than NUM.

```
IF(YENO1.NE.'Y')NSP=NUM
```

C
C
C
C

SELECT SCALE FOR SP DATA

```

TEMP1=SP(1)
TEMP2=SP(1)
DO 420 I=2,NSP
  BIG=MAX(TEMP1,SP(I))
  TEMP1=BIG
  SMALL=MIN(TEMP2,SP(I))
  TEMP2=SMALL
420  CONTINUE
WRITE(LUN,430)BIG,SMALL

```

```

430 FORMAT(10X,'BIG=',F8.3,2X,'SMALL=',F8.3)
    SPLFT=(INT(SMALL/10.))*10
    SPRHT=(INT(BIG/10.+1.))*10
    WRITE(LUN,440)SPLFT,SPRHT
440 FORMAT(10X,'SPLFT=',F8.3,2X,'SPRHT=',F8.3)
C
C      READ IN PLOT SCALING FACTORS & THE SUITE OF WIRE-LINE
C      LOGS TO BE PLOTTED
C
450 WRITE(1,500)
500 FORMAT(/,10X,'ENTER SCALE FOR X DIRECTION.',
*      ' (available range 200 <> 1000) 1:')
    READ(1,*)IXSC
    WRITE(LUN,510)IXSC
510 FORMAT(10X,'IXSC=',I4)
C
C      CHECK 'X' SCALE IS WITHIN RANGE
C
    IF (IXSC.LT.100.OR.IXSC.GT.1000) THEN
        WRITE(1,520)
520    FORMAT(/,15X,'THE SCALE VALUE YOU ENTERED IS OUT OF RANGE',
*      //,30X,'TRY AGAIN !')
        GOTO 450
    END IF
C
530 WRITE(1,550)
550 FORMAT(/,20X,'ENTER SCALING FACTOR FOR ','Y',' DIRECTION',/,
*23X,'(available range 0.25 <> 1.0)')
    READ(1,*)YFACT
    WRITE(LUN,560)YFACT
560 FORMAT(10X,'YFACT=',F8.3)
C
C      CHECK 'Y' SCALE IS WITHIN RANGE
C
    IF (YFACT.LT.0.25.OR.YFACT.GT.1.0) THEN
        WRITE(1,520)
        GOTO 530
    END IF
570 WRITE(1,600)
600 FORMAT(/,19X,'THERE ARE 3 SUITES OF LOGS AVAILABLE :- ',
*//,30X,'TRACK 1          TRACK 2',
*//,15X,'Suite1          GAMMA RAY          DENSITY/NEUTRON',/,
*15X,'Suite2          GAMMA/SP          RESISTIVITY',/,
*15X,'Suite3          GAMMA/SP          SONIC',//,
*10X,'SELECT THE SUITE OF LOGS YOU REQUIRE BY ENTERING :-',/,
*27X,'1 , 2 OR 3',//,20X,'Entering a 4 gives you the option',
*  //,20X,' of selecting the X-Y scales again')
    READ(1,*)Suite
    WRITE(LUN,603)Suite
603 FORMAT(10X,'Suite=',I2)
    IF(Suite.EQ.4)GOTO 450

```

```

C
C
C      CHECK SUITE OF LOGS SELECTED IS VALID
C
C      IF(Suite.LT.1.OR.Suite.GT.4)THEN
C          WRITE(1,604)
604      FORMAT(//,25X,'YOU MUST ENTER EITHER 1,2,3, OR 4',/,35X,
C          *      'TRY AGAIN')
C          GOTO 570
C      END IF
C
C          SET UP PARAMETERS FOR PLOT
C
C      XUP=INT(DPH(1)/10.)*10.
C      XLOW=INT(DPH(NSP)/10.+1.)*10.
C
C      IF(METFT.EQ.'1'.OR.METFT.EQ.'M'.OR.METFT.EQ.'m')THEN
C          XMAX=(XLOW-XUP)*0.19765625*AFFACT*2/(IXSC/100)
C          XDIST=5.0
C      ELSE
C          XMAX=(XLOW-XUP)*0.19765625*0.254*2/(IXSC/100)
C          XDIST=20.0
C      END IF
C      WRITE(LUN,605)XUP,XLOW,XMAX
605  FORMAT(10X,'XUP=',F8.3,2X,'XLOW=',F8.3,2X,'XMAX=',F8.3)
C      YMAX=10.0
C      XVAL=XMAX
C      TOTX=XMAX+4.0
C      Len1=2.5*YFACT
C      Len2=5.0*YFACT
C      WRITE(LUN,610)XVAL,TOTX
610  FORMAT(10X,'XVAL=',F8.3,5X,'TOTX=',F8.3)
C
C          MAXIMUM LENGTH OF PLOT SET TO 6 FT.
C
C      IF(TOTX.GT.72.0)STOP 'LENGTH OF PLOT EXCEEDS 6 FT.'
C
C          PRE-SET SCALES FOR WIRE-LINE LOGS
C
C      DLFT=1.950
C      DRHT=2.950
C      PLFT=0.45
C      PRHT=-0.15
C      GLFT=0.0
C      GRHT=200.0
C      RESL=0.2
C      RESR=2000.0
C      SLFT=140.
C      SRHT=40.
C
C          CHANGE WIRE-LINE LOG SCALE IF NECESSARY

```

```

C
  WRITE(1,620)Suite
620  FORMAT(///,19X,'The following scale values are pre-set for suite',
  *          I2,///,30X,
  *'LOG          SCALE',/,27X,
  *'GAMMA RAY    0 - 200')
  IF(Suite.EQ.1)THEN
    WRITE(1,622)
622    FORMAT(27X,'DENSITY          1.95 - 2.95',/,27X,
  *          'POROSITY          0.45 - -0.15')
    ELSE IF(Suite.EQ.2)THEN
      WRITE(1,623)
623    FORMAT(27X,'SP scale is calculated from the data.',/,27X,
  *          'RESISTIVITY        0.2 - 2000')
    ELSE IF(Suite.EQ.3)THEN
      WRITE(1,624)
624    FORMAT(27X,'SP scale is calculated from the data.',/,27X,
  *          'SONIC              140 - 40')
    END IF
    WRITE(1,645)
645  FORMAT(//,20X,'IS THERE ANY SCALE THAT NEEDS TO BE CHANGED ',
  *          '(Y/N)')

C
  READ(1,'(A1)')YENO
  IF(YENO.NE.'Y')GOTO 655
625  WRITE(1,630)
630  FORMAT(/,25X,'ENTER LOG TO BE CHANGED')
  READ(1,'(A1)')LOGTYP

C
  IF(LOGTYP.EQ.'G')THEN
    WRITE(1,640)
640    FORMAT(/,10X,'ENTER NEW SCALE VALUE',/,28X,'eg. 1.85,2.85')
  READ(1,*)GLFT,GRHT
  ELSE IF(LOGTYP.EQ.'D')THEN
    WRITE(1,640)
    READ(1,*)DLFT,DRHT
  ELSE IF(LOGTYP.EQ.'P')THEN
    WRITE(1,640)
    READ(1,*)PLFT,PRHT
  ELSE IF(LOGTYP.EQ.'S')THEN
    WRITE(1,640)
    READ(1,*)SLFT,SRHT
  ELSE IF(LOGTYP.EQ.'R')THEN
    WRITE(1,640)
    READ(1,*)RESL,RESR
  END IF

C
  WRITE(1,650)
650  FORMAT(//,10X,'DO YOU WANT TO CHANGE ANOTHER SCALE (Y/N)')
  READ(1,'(A1)')YENO
  IF(YENO.EQ.'Y')GOTO 625

```

```

C
655 WRITE(1,652)
652 FORMAT(//,22X,'Do you want the output on the ZETA plotter. ',
*          '(Y/N)')
      READ(1,'(A1)')ZETA
      IF(ZETA.EQ.'Y')THEN
        Ptype=11
        WRITE(LUN,653)Ptype
653      FORMAT(10X,'Ptype= ',I3)
        CALL NEWPE(1)
        CALL SPEED(10)
        WRITE(1,654)
654      FORMAT(10X,'NEWPE=1')
      ELSE
        Ptype=10
      END IF

C
      WRITE(1,660)
660 FORMAT(//,33X,'Please wait',/,28X,
*          'I'll be back shortly',/)
C *****
C          PLOTTING ROUTINE BEGINS
C *****
      CALL PLOTS(TOTX,YMAX,Ptype)
      IF(Suite.EQ.1)THEN
        CALL HEAD1(YFACT,GLFT,GRHT,PLFT,PRHT,DLFT,DRHT,
*                Title,Date,METFT,IXSC)
        CALL BOX(XVAL,YFACT,XUP,XLOW,XMAX,LUN,XDIST,0)
        LINTYP='SOLID'
        LOG=' GAMMA'
        WRITE(LUN,680)LINTYP,LOG
680      FORMAT(20X,'LINTYP=',A5,2X,'LOG=',A8)
        CALL PLT(DPH,GRY,NSP,XUP,XLOW,XMAX,GLFT,GRHT,Len1,LUN,0,
*              LINTYP,LOG,ZETA)
        CALL ORIGI(0.0,Len1+0.75,1)
        LINTYP='SOLID'
        LOG='DENSITY '
        WRITE(LUN,680)LINTYP,LOG
        CALL PLT(DPH,DENS,NSP,XUP,XLOW,XMAX,DLFT,DRHT,Len2,LUN,0,
*              LINTYP,LOG,ZETA)
        WRITE(1,690)
690      FORMAT(//,31X,'Nearly finished',/)
        LINTYP='LDASH'
        LOG='POROSITY'
        WRITE(LUN,680)LINTYP,LOG
        CALL PLT(DPH,POR,NSP,XUP,XLOW,XMAX,PLFT,PRHT,Len2,LUN,0,
*              LINTYP,LOG,ZETA)
      ELSE IF(Suite.EQ.2)THEN
        CALL HEAD2(YFACT,SPLFT,SPRHT,GLFT,GRHT,Title,Date,
*                METFT,IXSC,LUN)
        CALL BOX(XVAL,YFACT,XUP,XLOW,XMAX,LUN,XDIST,1)

```

```

LINTYP='SOLID'
LOG=' GAMMA '
WRITE(LUN,680)LINTYP,LOG
CALL PLT(DPH,GRY,NSP,XUP,XLOW,XMAX,GLFT,GRHT,Len1,LUN,0,
*      LINTYP,LOG,ZETA)
LINTYP='LDASH'
LOG=' SP '
WRITE(LUN,680)LINTYP,LOG
CALL PLT(DPH,SP,NSP,XUP,XLOW,XMAX,SPLFT,SPRHT,Len1,LUN,0,
*      LINTYP,LOG,ZETA)
CALL ORIGI(0.0,Len1+0.75,1)
LINTYP='SOLID'
LOG=' Msfl '
WRITE(LUN,680)LINTYP,LOG
CALL PLT(DPH,Msf1,NSP,XUP,XLOW,XMAX,RESL,RESR,Len2,LUN,1,
*      LINTYP,LOG,ZETA)
LINTYP='LDASH'
LOG=' RLS '
WRITE(LUN,680)LINTYP,LOG
CALL PLT(DPH,RLS,NSP,XUP,XLOW,XMAX,RESL,RESR,Len2,LUN,1,
*      LINTYP,LOG,ZETA)
WRITE(1,690)
LINTYP='FDASH'
LOG=' RLD '
WRITE(LUN,680)LINTYP,LOG
CALL PLT(DPH,RLD,NSP,XUP,XLOW,XMAX,RESL,RESR,Len2,LUN,1,
*      LINTYP,LOG,ZETA)
ELSE
CALL HEAD3(YFACT,GLFT,GRHT,SPLFT,SPRHT,SLFT,SRHT,
*      Title,Date,METFT,IXSC)
CALL BOX(XVAL,YFACT,XUP,XLOW,XMAX,LUN,XDIST,0)
LINTYP='SOLID'
LOG=' GAMMA '
WRITE(LUN,680)LINTYP,LOG
CALL PLT(DPH,GRY,NSP,XUP,XLOW,XMAX,GLFT,GRHT,Len1,LUN,0,
*      LINTYP,LOG,ZETA)
LINTYP='FDASH'
LOG=' SP '
WRITE(LUN,680)LINTYP,LOG
CALL PLT(DPH,SP,NSP,XUP,XLOW,XMAX,SPLFT,SPRHT,Len1,LUN,0,
*      LINTYP,LOG,ZETA)
CALL ORIGI(0.0,Len1+0.75,1)
LINTYP='SOLID'
WRITE(1,690)
LOG=' SONIC '
WRITE(LUN,680)LINTYP,LOG
CALL PLT(DPH,SONIC,NSP,XUP,XLOW,XMAX,SLFT,SRHT,Len2,LUN,0,
*      LINTYP,LOG,ZETA)
END IF
C
CALL PLOT(XMAX,-3.5,999)

```



```
C *****
C                                     CHECK FOR MORE FILES
C
C      WRITE(1,700)
700  FORMAT(/,17X,'DO YOU WANT TO PLOT ANOTHER SUITE OF LOGS',
      *          '(Y/N)')
      READ(1,'(A1)') YENO
      IF(YENO.EQ.'Y') THEN
        GOTO 450
      ELSE
800    WRITE(1,800)
        FORMAT(/,18X,'DO YOU WANT TO ENTER ANOTHER DATA FILE (Y/N)')
        READ(1,'(A1)') YENO
        IF(YENO.EQ.'Y') THEN
          GOTO 5
        ELSE
          STOP 'NORMAL TERMINATION'
        END IF
      END IF
END IF
END
REAL FUNCTION SCALE(ARRY,Lft,RHT,SIZE,IRES)
      FUNCTION SCALE RETURNS A SCALED VALUE (REAL) TO BE PLOTTED
REAL Lft
IF(IRES.EQ.0) THEN
  SCALE=((ARRY-Lft)*SIZE)/(RHT-Lft)
ELSE
  SCALE=((LOG10(ARRY)-LOG10(Lft))*SIZE)/(LOG10(RHT)-LOG10(Lft))
END IF
RETURN
END
SUBROUTINE BOX(XVAL,YFACT,XUP,XLOW,XMAX,LUN,XDIST,IRES)
      SUBROUTINE BOX DRAWS THE FRAME FOR THE PLOTS ie. TRACK1
      AND TRACK2, AND ALSO THE DEPTH VALUES.
      DEFINE PARAMETERS FOR DRAWING THE TWO TRACKS
      SIZE=(0.04*YFACT)+0.06
      YPOS4=2.5*YFACT
      YFIX=YPOS4+0.75
      YPOS=YFIX+(5.0*YFACT)
      OUTLINE TRACK ONE
```

```

C      CALL ORIGI(3.50,0.5,0)
      CALL PLOT(0.0,0.0,3)
      CALL PLOT(XVAL,0.0,2)
      CALL PLOT(XVAL,YPOS4,2)
      CALL PLOT(0.0,YPOS4,2)
      CALL PLOT(0.0,YPOS4/2.,3)
      CALL PLOT(XVAL,YPOS4/2.,2)

C      C
C      OUTLINE TRACK TWO

      CALL PLOT(0.0,YFIX,3)
      CALL PLOT(XVAL,YFIX,2)
      CALL PLOT(XVAL,YPOS,2)
      CALL PLOT(0.0,YPOS,2)

C      C
C      IF IRES = 1 THEN DRAW TRACK 2 AS A LOG SCALE

      IF(IRES.EQ.0) THEN
        YPOS1=YFIX+(1.25*YFACT)
        YPOS2=YFIX+(2.50*YFACT)
        YPOS3=YFIX+(3.75*YFACT)
        CALL PLOT(0.0,YPOS1,3)
        CALL PLOT(XVAL,YPOS1,2)
        CALL PLOT(XVAL,YPOS2,3)
        CALL PLOT(0.0,YPOS2,2)
        CALL PLOT(0.0,YPOS3,3)
        CALL PLOT(XVAL,YPOS3,2)
      ELSE
        YPOS1=YFIX+(0.875*YFACT)
        YPOS2=YFIX+(2.125*YFACT)
        YPOS3=YFIX+(3.375*YFACT)
        YPOS5=YFIX+(4.625*YFACT)
        CALL PLOT(0.0,YPOS1,3)
        CALL PLOT(XVAL,YPOS1,2)
        CALL PLOT(XVAL,YPOS2,3)
        CALL PLOT(0.0,YPOS2,2)
        CALL PLOT(0.0,YPOS3,3)
        CALL PLOT(XVAL,YPOS3,2)
        CALL PLOT(XVAL,YPOS5,3)
        CALL PLOT(0.0,YPOS5,2)
      END IF

C      C
      XINC=XMAX/((XLOW-XUP)/XDIST)
      ISTEP=((XLOW-XUP)/XDIST)+1.
      STEPX=0.
      DEPTH=XUP
      WRITE(LUN,400) XINC,ISTEP
400  FORMAT(10X,'XINC=',F8.5,2X,'ISTEP=',I4)
      ILINE=0

```

WRITE OUT DEPTHS (AT EITHER 5M OR 20FT INTERVALS)

DO 600 I=1, ISTEP

CALL NUMB(STEPX, YPOS4+0.2, SIZE, DEPTH, 90.0, -1)

IF (MOD (ILINE, 2) .EQ. 0) THEN

CALL PLOT (STEPX, YPOS4+0.10, 3)

CALL PLOT (STEPX, 0.0, 2)

CALL PLOT (STEPX, YPOS, 3)

CALL PLOT (STEPX, YFIX-0.10, 2)

ELSE

CALL PLOT (STEPX, YPOS4+0.05, 3)

CALL PLOT (STEPX, YPOS4, 2)

CALL PLOT (STEPX, YFIX, 3)

CALL PLOT (STEPX, YFIX-0.05, 2)

ENDIF

DEPTH=DEPTH+XDIST

STEPX=STEPX+XINC

ILINE=ILINE+1

WRITE (LUN, 500) DEPTH, STEPX, ILINE

500 FORMAT (10X, 'DEPTH=', F8.0, 2X, 'STEPX=', F9.5, 2X, 'ILINE=', I4)

600 CONTINUE

RETURN

END

SUBROUTINE HEAD1(YFACT, Lft1, RHT1, Lft2, RHT2, Lft3, RHT3,

* Title, Date, METFT, IXSC)

PREPARE TITLES & SCALE INDEX FOR FIRST SUITE OF LOGS

REAL Lft1, Lft2, Lft3

INTEGER Title(20), Date(4)

CHARACTER*1 METFT

DEFINE PEN LOCATIONS

REDUCT=(1-YFACT)*100.

YVAL=0.5+(2.5*YFACT)

YPOS2=YVAL+0.75

YPOS=YPOS2+((5.0*YFACT)-0.5)

YPOS1=YVAL-0.5

SIZE=(0.02*YFACT)+0.08

ISTEP=(5.0*YFACT)/SIZE

IDASH=0

WRITE OUT HEADING INFORMATION

XSCAL=IXSC

CALL SYMB(0.25, 1.3, 0.15, 6HTITLE:, 90.0, 6)

CALL SYMB(0.25, 2.2, 0.15, Title, 90.0, 50)

CALL SYMB(0.75, 1.3, 0.10, 10HX SCALE 1:, 90.0, 10)

```

CALL NUMB(0.75,2.3,0.10,XSCAL,90.0,-1)
CALL SYMB(0.5,1.3,0.1,22HY SCALE REDUCED BY    %,90.0,22)
CALL NUMB(0.5,3.15,0.1,REDUCT,90.0,-1)
CALL SYMB(1.0,1.3,0.10,5HDATE:,90.0,5)
CALL SYMB(1.0,1.8,0.10,Date,90.0,8)

```

C
C
C

PRODUCE HEADING FOR TRACK ONE

```

CALL PLOT(1.25,0.5,3)
CALL PLOT(3.5,0.5,2)
CALL PLOT(3.5,YVAL,3)
CALL PLOT(1.25,YVAL,2)
CALL PLOT(1.25,0.5,2)
CALL PLOT(1.25,YPOS2,3)
CALL PLOT(3.5,YPOS2,2)
CALL PLOT(2.938,0.5,3)
CALL PLOT(2.938,YVAL,2)
CALL NUMB(3.138,0.6,SIZE,Lft1,90.0,1)
CALL NUMB(3.138,YPOS1,SIZE,RHT1,90.0,1)
YVAL=0.5+(2.5*YFACT-((0.1*YFACT)+0.71))/2.
CALL SYMB(2.888,YVAL,SIZE,9HGAMMA RAY,90.0,9)

```

C
C
C

PRODUCE HEADING FOR TRACK TWO

```

YVAL=YPOS2+(5.0*YFACT)
CALL PLOT(3.5,YVAL,3)
CALL PLOT(1.25,YVAL,2)
CALL PLOT(1.25,YPOS2,2)
CALL PLOT(2.938,YPOS2,3)
CALL PLOT(2.938,YVAL,2)
CALL PLOT(2.375,YPOS2,3)
YVAL=YPOS2+SIZE
DO 200 I=1,ISTEP
    IF(MOD(IDASH,2).EQ.0) THEN
        CALL PLOT(2.375,YVAL,2)
    ELSE
        CALL PLOT(2.375,YVAL,3)
    ENDIF
    IDASH=IDASH+1
    YVAL=YVAL+SIZE
200 CONTINUE
CALL NUMB(2.575,YPOS2+0.05,SIZE,Lft2,90.0,2)
CALL NUMB(2.575,YPOS,SIZE,RHT2,90.0,2)
YVAL=YPOS2+(5.0*YFACT-((0.1*YFACT)+0.65))/2.
CALL SYMB(2.325,YVAL,SIZE,8HPOROSITY,90.0,8)
CALL SYMB(2.888,YVAL,SIZE,7HDENSITY,90.0,7)
CALL NUMB(3.138,YPOS2+0.05,SIZE,Lft3,90.0,2)
CALL NUMB(3.138,YPOS,SIZE,RHT3,90.0,2)
YVAL=(2.5*YFACT)+0.60
CALL SYMB(3.0,YVAL,SIZE,5HDEPTH,90.0,5)
IF (METFT.EQ. '1'.OR.METFT.EQ. 'M'.OR.METFT.EQ. 'm') THEN

```

```

        YVAL=(2.5*YFACT)+0.70
        CALL SYMB(3.2,YVAL,SIZE,3H(M),90.0,3)
ELSE
        CALL SYMB(3.2,YVAL,SIZE,4H(FT),90.0,4)
END IF
C
RETURN
END
SUBROUTINE HEAD2(YFACT,Lft1,RHT1,Lft2,RHT2,Title,Date,METFT,IXSC,
*
        LUN)

```

PREPARE TITLES & SCALE INDEX FOR SECOND SUITE OF LOGS

```

REAL Lft1,Lft2
INTEGER Title(20),Date(4)
CHARACTER*1 METFT

```

DEFINE PEN LOCATIONS

```

REDUCT=(1-YFACT)*100.
YVAL=0.5+(2.5*YFACT)
YPOS2=YVAL+0.75
YPOS=YPOS2+((5.0*YFACT)-0.5)
YPOS1=YVAL-0.5
SIZE=(0.03*YFACT)+0.07
ISTEP=(2.5*YFACT)/SIZE
IDASH=0

```

WRITE OUT HEADING INFORMATION

```

XSCAL=IXSC
CALL SYMB(0.25,1.3,0.15,6HTITLE:,90.0,6)
CALL SYMB(0.25,2.2,0.15,Title,90.0,50)
CALL SYMB(0.75,1.3,0.10,10HX SCALE 1:,90.0,10)
CALL NUMB(0.75,2.3,0.10,XSCAL,90.0,-1)
CALL SYMB(0.5,1.3,0.1,22HY SCALE REDUCED BY %,90.0,22)
CALL NUMB(0.5,3.15,0.1,REDUCT,90.0,-1)
CALL SYMB(1.0,1.3,0.10,5HDATE:,90.0,5)
CALL SYMB(1.0,1.8,0.10,Date,90.0,8)

```

PREPARE HEADING FOR TRACK ONE

```

CALL PLOT(1.25,0.5,3)
CALL PLOT(3.5,0.5,2)
CALL PLOT(3.5,YVAL,3)
CALL PLOT(1.25,YVAL,2)
CALL PLOT(1.25,0.5,2)
CALL PLOT(1.25,YPOS2,3)
CALL PLOT(3.5,YPOS2,2)
CALL PLOT(2.375,0.5,3)
YVAL=0.5+SIZE

```

C
C
C

PRODUCE DASHED LINE FOR SP HEADING

```

DO 100 I=1,ISTEP
  IF(MOD(IDASH,2).EQ.0)THEN
    CALL PLOT(2.375,YVAL,2)
  ELSE
    CALL PLOT(2.375,YVAL,3)
  ENDIF
  IDASH=IDASH+1
  YVAL=YVAL+SIZE

```

```

100 CONTINUE
YVAL=0.5+(2.5*YFACT)
CALL PLOT(2.938,0.5,3)
CALL PLOT(2.938,YVAL,2)
CALL NUMB(2.575,0.6,SIZE,Lft1,90.0,1)
CALL NUMB(2.575,YPOS1,SIZE,RHT1,90.0,1)
CALL NUMB(3.138,0.6,SIZE,LHT2,90.0,1)
CALL NUMB(3.138,YPOS1,SIZE,RHT2,90.0,1)
YVAL=0.5+(2.5*YFACT-0.2)/2.
CALL SYMB(2.325,YVAL,SIZE,2HSP,90.0,2)
YVAL=0.5+(2.5*YFACT-((0.1*YYFACT)+0.71))/2.
CALL SYMB(2.888,YVAL,SIZE,9HGAMMA RAY,90.0,9)

```

C
C
C

PRODUCE HEADING FOR TRACK TWO

```

YVAL=YPOS2+(5.0*YFACT)
CALL PLOT(3.5,YVAL,3)
CALL PLOT(1.25,YVAL,2)
CALL PLOT(1.25,YPOS2,2)
CALL PLOT(1.813,YPOS2,3)
CALL PLOT(1.813,YVAL,2)
CALL PLOT(2.375,YPOS2,3)
ISTEP=(5.0*YFACT)/SIZE
IDASH=0
YVAL=YPOS2+SIZE

```

C
C
C

PRODUCE FIRST DASH LINE (COURSE)

```

DO 200 I=1,ISTEP
  IF(MOD(IDASH,2).EQ.0)THEN
    CALL PLOT(2.375,YVAL,2)
  ELSE
    CALL PLOT(2.375,YVAL,3)
  ENDIF
  IDASH=IDASH+1
  YVAL=YVAL+SIZE
200 CONTINUE
CALL PLOT(2.938,YPOS2,3)
ISTEP=(5.0*YFACT)/((0.01*YFACT)+0.06)
IDASH=0

```

```

        YSTEP2=(0.01*YFACT)+0.06
        YVAL=YPOS2+YSTEP2
        WRITE(LUN,210) ISTEP, IDASH, YSTEP2, YVAL
210  FORMAT(10X, 'ISTEP=', I4, 2X, 'IDASH=', I4, 2X, 'YSTEP2=', F7.4,
        *2X, 'YVAL=', F7.4)

```

```

C
C      PRODUCE SECOND DASH LINE (FINE)
C
      DO 300 I=1, ISTEP
        IF (MOD(IDASH, 2) .EQ. 0) THEN
          CALL PLOT(2.938, YVAL, 2)
        ELSE
          CALL PLOT(2.938, YVAL, 3)
        ENDIF
        IDASH=IDASH+1
        YVAL=YVAL+YSTEP2
        WRITE(LUN, 220) IDASH, YVAL
220    FORMAT(10X, 'IDASH=', I4, 5X, 'YVAL=', F7.4)
300  CONTINUE
      CALL NUMB(2.013, YPOS2+0.05, SIZE, 0.2, 90.0, 1)
      CALL NUMB(2.013, YPOS-0.1, SIZE, 2000.0, 90.0, 1)
      CALL NUMB(2.575, YPOS2+0.05, SIZE, 0.2, 90.0, 1)
      CALL NUMB(2.575, YPOS-0.1, SIZE, 2000.0, 90.0, 1)
      CALL NUMB(3.138, YPOS2+0.05, SIZE, 0.2, 90.0, 1)
      CALL NUMB(3.138, YPOS-0.1, SIZE, 2000.0, 90.0, 1)
      YVAL=YPOS2+(5.0*YFACT-1.0)/2.
      CALL SYMB(1.763, YVAL, SIZE, 11HRLLS (OHMM), 90.0, 11)
      CALL SYMB(2.325, YVAL, SIZE, 11HRLLM (OHMM), 90.0, 11)
      CALL SYMB(2.888, YVAL, SIZE, 11HRLLD (OHMM), 90.0, 11)
      YVAL=(2.5*YFACT)+0.60
      CALL SYMB(3.0, YVAL, 0.1, 5HDEPTH, 90.0, 5)
      IF (METFT.EQ. '1'.OR.METFT.EQ. 'M'.OR.METFT.EQ. 'm') THEN
        YVAL=(2.5*YFACT)+0.70
        CALL SYMB(3.2, YVAL, 0.1, 3H(M), 90.0, 3)
      ELSE
        CALL SYMB(3.2, YVAL, 0.1, 4H(FT), 90.0, 4)
      END IF

```

```

C
      RETURN
      END

```

```

      SUBROUTINE HEAD3(YFACT, Lft1, RHT1, Lft2, RHT2, Lft3, RHT3,
*      Title, Date, METFT, IXSC)

```

```

C
C      PREPARE TITLES & SCALE INDEX FOR THE THIRD SUITE OF LOGS
C

```

```

      REAL Lft1, Lft2, Lft3
      INTEGER Title(20), Date(4)
      CHARACTER*1 METFT

```

```

C
C      DEFINE PEN LOCATIONS
C

```



```

REDUCT=(1-YFACT)*100.
YVAL=0.5+(2.5*YFACT)
YPOS2=YVAL+0.75
YPOS=YPOS2+((5.0*YFACT)-0.5)
YPOS1=YVAL-0.5
SIZE=(0.04*YFACT)+0.06
ISTEP=(2.5*YFACT)/SIZE
IDASH=0

```

```

C
C      WRITE OUT HEADING INFORMATION
C

```

```

XSCAL=IXSC
CALL SYMB(0.25,1.3,0.15,6HTITLE:,90.0,6)
CALL SYMB(0.25,2.2,0.15,Title,90.0,50)
CALL SYMB(0.75,1.3,0.10,10HX SCALE 1:,90.0,10)
CALL NUMB(0.75,2.3,0.10,XSCAL,90.0,-1)
CALL SYMB(0.5,1.3,0.1,22HY SCALE REDUCED BY %,90.0,22)
CALL NUMB(0.5,3.15,0.1,REDUCT,90.0,-1)
CALL SYMB(1.0,1.3,0.10,5HDATE:,90.0,5)
CALL SYMB(1.0,1.8,0.10,Date,90.0,8)

```

```

C
C      PREPARE HEADING FOR TRACK 1
C

```

```

CALL PLOT(1.25,0.5,3)
CALL PLOT(3.5,0.5,2)
CALL PLOT(3.5,YVAL,3)
CALL PLOT(1.25,YVAL,2)
CALL PLOT(1.25,0.5,2)
CALL PLOT(1.25,YPOS2,3)
CALL PLOT(3.5,YPOS2,2)
CALL PLOT(2.375,0.5,3)
YVAL=0.5+YSTEP

```

```

C
C      PRODUCE DASH LINE FOR SP HEADING
C

```

```

DO 100 I=1,ISTEP
  IF(MOD(IDASH,2).EQ.0)THEN
    CALL PLOT(2.375,YVAL,2)
  ELSE
    CALL PLOT(2.375,YVAL,3)
  ENDIF
  IDASH=IDASH+1
  YVAL=YVAL+SIZE

```

```

100 CONTINUE

```

```

YVAL=0.5+(2.5*YFACT)
CALL PLOT(2.938,0.5,3)
CALL PLOT(2.938,YVAL,2)
CALL NUMB(3.138,0.6,SIZE,Lft1,90.0,1)
CALL NUMB(3.138,YPOS1,SIZE,RHT1,90.0,1)
CALL NUMB(2.575,0.6,SIZE,Lft2,90.0,1)
CALL NUMB(2.575,YPOS1,SIZE,RHT2,90.0,1)

```

```

YVAL=0.5+(2.5*YFACT-0.2)/2.
CALL SYMB(2.325,YVAL,SIZE,2HSP,90.0,2)
YVAL=0.5+(2.5*YFACT-((0.1*YFACT)+0.71))/2.
CALL SYMB(2.888,YVAL,SIZE,9HGAMMA RAY,90.0,9)

```

```

C
C      PREPARE HEADING FOR SECOND TRACK
C

```

```

YVAL=YPOS2+(5.0*YFACT)
CALL PLOT(3.5,YVAL,3)
CALL PLOT(1.25,YVAL,2)
CALL PLOT(1.25,YPOS2,2)
CALL PLOT(2.938,YPOS2,3)
CALL PLOT(2.938,YVAL,2)
CALL NUMB(3.138,YPOS2+0.05,SIZE,Lft3,90.0,1)
CALL NUMB(3.138,YPOS,SIZE,RHT3,90.0,1)
YVAL=YPOS2+(5.0*YFACT-0.5)/2.
CALL SYMB(2.888,YVAL,SIZE,5HSONIC,90.0,5)
YVAL=(2.5*YFACT)+0.60
CALL SYMB(3.0,YVAL,0.1,5HDEPTH,90.0,5)
IF(METFT.EQ.'1'.OR.METFT.EQ.'M'.OR.METFT.EQ.'m') THEN
    YVAL=(2.5*YFACT)+0.70
    CALL SYMB(3.2,YVAL,0.1,3H(M),90.0,3)
ELSE
    CALL SYMB(3.2,YVAL,0.1,4H(FT),90.0,4)
END IF

```

```

C
RETURN
END

```

```

SUBROUTINE PLT(DPH,ARRY,NSP,XUP,XLOW,XMAX,SCLFT,SCRHT,Len,
*          LUN,IRES,LINTYP,LOG,ZETA)

```

```

C
C      PLOT THE LOG VALUES EITHER AS

```

1. SOLID LINES
2. LARGE DASH LINES
3. SMALL DASH LINES

```

C *****

```

```

C      (XONE,YONE) & (XTWO,YTWO) are the two points joined by either
C      a solid or dashed line.

```

```

C      DASLEN = Length of dash.

```

```

C      LINTYP = Describes the type of line to be plotted.

```

```

C *****

```

```

C      LOGICAL empty
C      REAL DPH(NSP),ARRY(NSP),Len
C      CHARACTER LINTYP*5,LOG*8,ZETA*1

```

```

C
C      CHECK FOR BLANK ARRAY

```

```

C
CALL BLANK(ARRY,LUN,NSP,empty)
IF(empty) THEN
  WRITE(1,100) LOG
100   FORMAT(//,20X,A8,' Log NOT plotted (ie. no values)',/)
  RETURN
END IF

C
C
C      CHOSE LENGTH OF DASH

IF(LINTYP.EQ.'LDASH'.OR.LINTYP.EQ.'FDASH') THEN
  CALL DLEN(NSP,DPH,ARRY,XUP,XLOW,XMAX,IRES,LUN,Len,
*      SCLFT,SCRHT,DASLEN)
END IF
IF(LINTYP.EQ.'FDASH') DASLEN=DASLEN/2.

C
C
C      SELECT PEN COLOUR IF ZETA PLOT

IF(ZETA.EQ.'Y') THEN
  IF(LINTYP.EQ.'LDASH') THEN
    CALL NEWPE(3)
    WRITE(LUN,150)
150   FORMAT(10X,'RED PEN SELECTED')
  ELSE IF(LINTYP.EQ.'FDASH') THEN
    CALL NEWPE(4)
    WRITE(LUN,160)
160   FORMAT(10X,'GREEN PEN SELECTED')
  END IF
END IF

C
C
C      PREPARE PEN LOCATION FOR START OF PLOT

IDASH=0
XONE=SCALE(DPH(1),XUP,XLOW,XMAX,0)
YONE=SCALE(ARRY(1),SCLFT,SCRHT,Len,IRES)
WRITE(LUN,200) XONE,YONE
200  FORMAT(20X,'XONE='F8.4,5X,'YONE='F8.4)

C
C
C      POSITION PEN AT START OF PLOT

CALL PLOT(XONE,YONE,3)

C
C
C      PLOT WIRE-LINE LOG

DO 300 I=2,NSP
  XTWO=SCALE(DPH(I),XUP,XLOW,XMAX,0)
  Y TWO=SCALE(ARRY(I),SCLFT,SCRHT,Len,IRES)
  WRITE(LUN,250) XTWO,Y TWO
250  FORMAT(20X,'XTWO='F8.4,5X,'YTWO='F8.4)
  IF(LINTYP.EQ.'LDASH') THEN
    CALL DASLN(XONE,XTWO,YONE,Y TWO,DASLEN,LUN,IDASH)

```

```

      ELSE IF (LINTYP.EQ.'FDASH') THEN
        CALL DASLN(XONE,XTWO,YONE,YTWO,DASLEN,LUN,IDASH)
      ELSE
        CALL PLOT(XTWO,YTWO,2)
      END IF
      XONE=XTWO
      YONE=YTWO
300 CONTINUE
C
      RETURN
      END
      SUBROUTINE DASLN(XONE,XTWO,YONE,YTWO,DASLEN,LUN,IDASH)
C
C       DASHES BETWEEN POINTS - LENGTH OF DASH DEPENDS UPON 'DASLEN'
C
C       *****
C
C       SLOPE = Slope between two points (ie. (XTWO,YTWO) & (XONE,YONE))
C       DIST  = Distance between two points ( as above )
C       XINC  = Incremental step in X direction.
C       Xystep= No. of incremental steps in XY direction ( between the
C               two points - INTEGERIZED )
C       XVAL  = X value to be plotted.
C       YVAL  = Y value to be plotted.
C
C       *****
C       INTEGER Xystep
C
C       SLOPE=(YTWO-YONE)/(XTWO-XONE)
C       DIST=SQRT((XTWO-XONE)**2 + (YTWO-YONE)**2)
C       XINC=DASLEN*COS(ATAN(SLOPE))
C       Xystep=DIST/DASLEN
C       XVAL=XONE
C       XVAL=XVAL + XINC
C       WRITE(LUN,40) XVAL,IDASH,SLOPE,DIST,Xystep,XINC
40  FORMAT(5X,'XVAL=',F8.3,2X,'IDASH=',I3,2X,'SLOPE=',F13.10,2X,
* 'DIST=',F13.10,2X,'Xystep=',I4,2X,'XINC=',F8.4)
      IF(Xystep.GT.100) STOP 'CHECK DATA FILE FOR ERRORS'
C
C       DASHING PROCESS BEGINS
C
      DO 100 I=1,Xystep
        YVAL=SLOPE*(XVAL-XONE)+YONE
        WRITE(LUN,80) XVAL
80    FORMAT(20X,'XVAL=',F8.4)
        IF (MOD(IDASH,2).EQ.0) THEN
          CALL PLOT(XVAL,YVAL,2)
        ELSE
          CALL PLOT(XVAL,YVAL,3)
        END IF
        IDASH=IDASH+1

```

```

      XVAL=XVAL+XINC
100 CONTINUE
C
C      THIS ENSURES THE PLOTTER PEN REACHES THE SECOND OF THE TWO POINTS
C
      CALL PLOT(XTWO,YTWO,3)
C
      RETURN
      END
      SUBROUTINE BLANK(ARRY,LUN,NSP,empty)
C
C          CHECK FOR BLANK ARRAY(S)
C
      LOGICAL empty
      REAL ARRY(NSP)
C
      DO 100 I=1,NSP
          IF(ARRY(I).NE.0.00000)GOTO 200
100      CONTINUE
C
      empty=.TRUE.
      RETURN
200 empty=.FALSE.
C
      RETURN
      END
      SUBROUTINE DLEN(NSP,DPH,ARRY,XUP,XLOW,XMAX,IRES,LUN,Len,
*          SCLFT,SCRHT,DASLEN)
C
C          FIND THE OPTIMUM LENGTH OF THE DASH
C
      REAL DPH(NSP),ARRY(NSP),Len
C
C
      TEMP=10.0
      XONE=SCALE(DPH(1),XUP,XLOW,XMAX,0)
      YONE=SCALE(ARRY(1),SCLFT,SCRHT,Len,IRES)
C
      DO 100 I=2,NSP
          XTWO=SCALE(DPH(I),XUP,XLOW,XMAX,0)
          YTWO=SCALE(ARRY(I),SCLFT,SCRHT,Len,IRES)
          DIST=SQRT((XTWO-XONE)**2 + (YTWO-YONE)**2)
C
          DASLEN=MIN(TEMP,DIST)
          TEMP=DASLEN
C
          XONE=XTWO
          YONE=YTWO
100 CONTINUE
      WRITE(LUN,200) DASLEN
200 FORMAT(10X,'DASLEN=',F10.7)

```

C

RETURN
END

C

END\$

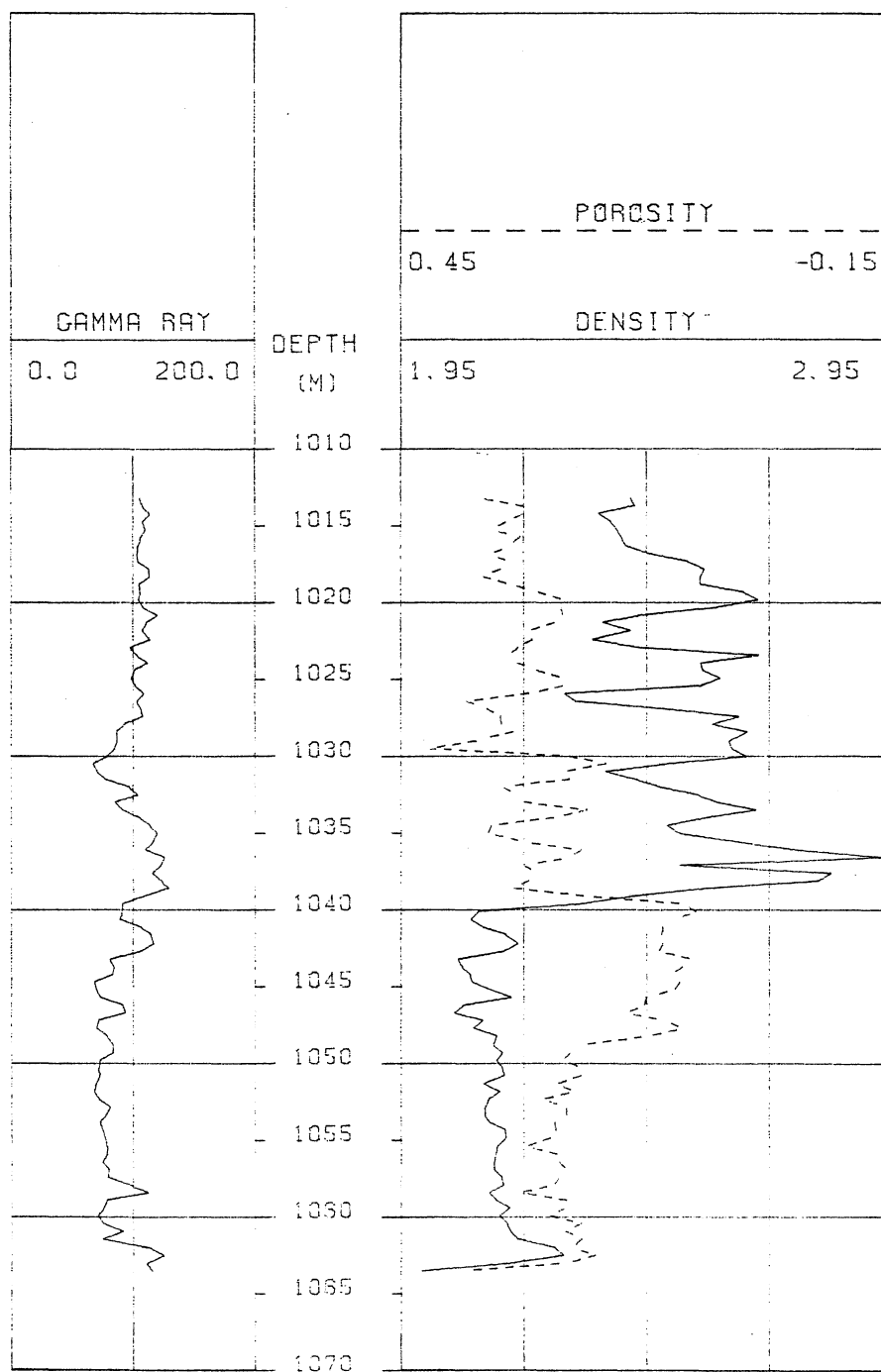
(d) Sample plots

TITLE: WELLXY NO. 1 (1000-1065)

Y SCALE REDUCED BY 50 %

X SCALE 1.500

DATE: 08-08-84

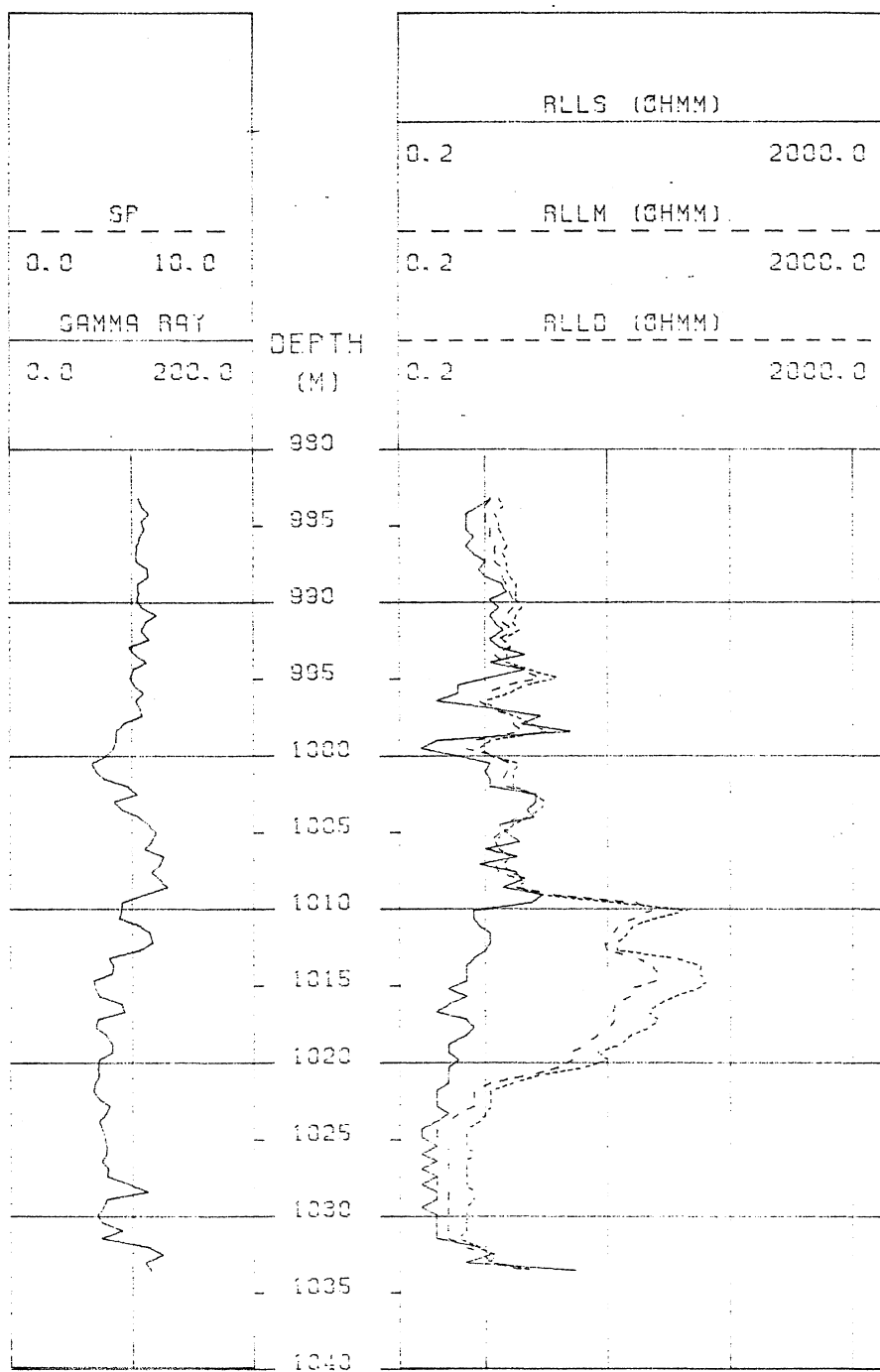


TITLE: WELLY NO. 1 (1000-1065)

Y SCALE REDUCED BY 50 %

X SCALE 1.500

DATE: 08-08-84

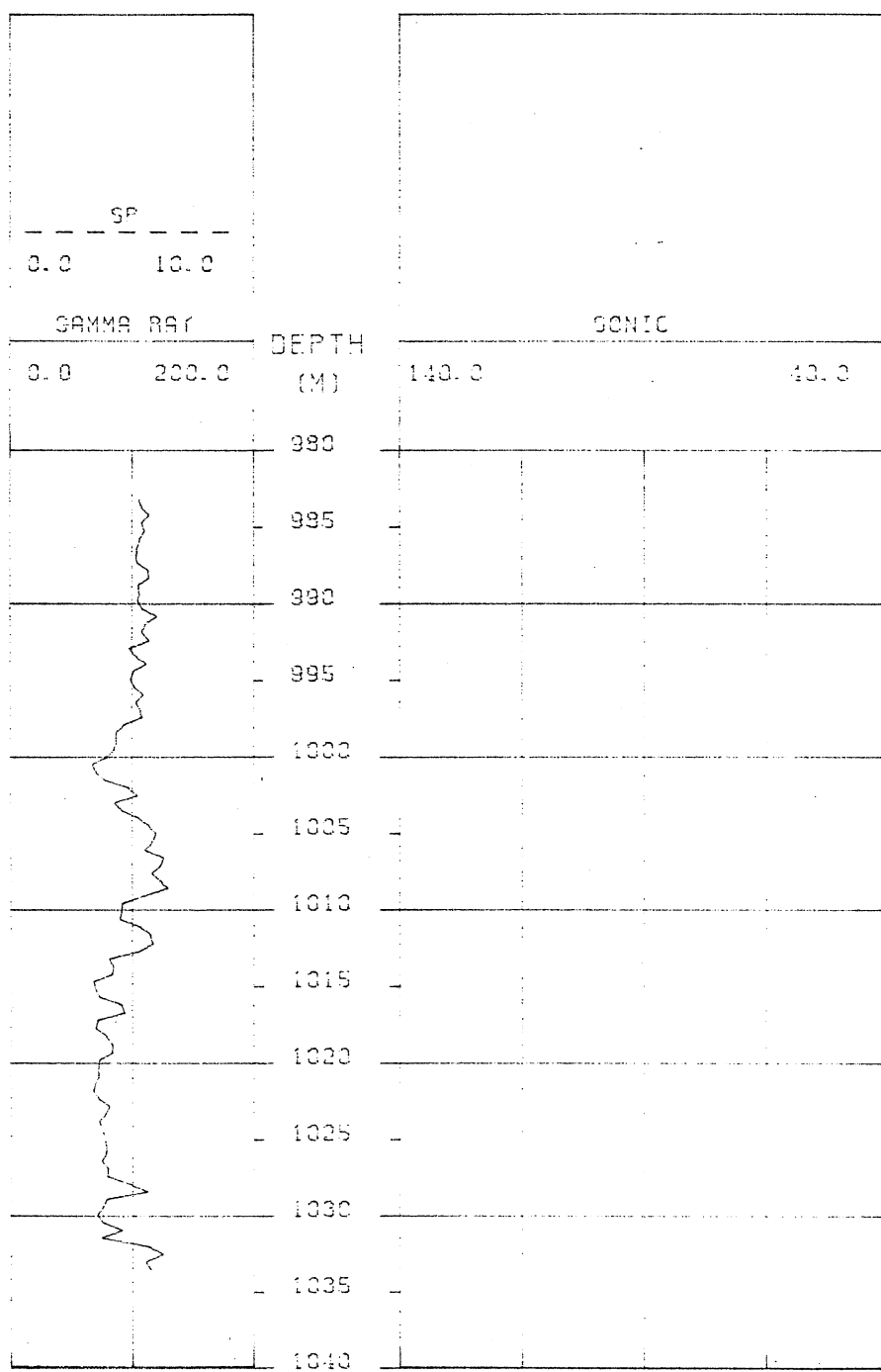


TITLE WELLY NO. 1 (1000-1035)

Y SCALE REDUCED BY 50 %

X SCALE 1:500

DATE: 08-08-84



V Bibliography

(i) R.W. De Nardi, 1983

"XYPLT - A Program to Convert Incrementally Digitised Data to 'LOG4' Compatible Input Format",

a BMR publication, 1983/15

(ii) L.E. Kurylowicz, 1978

"A Review of the Concepts and Practices of Wireline Log Interpretation",

a BMR internal publication, 1978/3

(iii) G.R. Morrison, 1984

"A Dual Water Wireline Log Interpretation Model, Computer Program DW.",

a BMR internal publication, 1984/8

(iv) G.R. Morrison, 1984

"RW (A Two Pass Method of Wireline Log Interpretation for Hydrocarbon Reservoir Evaluation; Computer Program RW)",

a BMR publication, 1984/26.

VI Acknowledgment

The authors wish to thank Ms Anne Felton, Mrs Evelyn Nicholas and Mr Ken Stillwell for the support they have given for this project. We also wish to thank Ms Chris Stokman for the excellent typing services she has provided.