BUREAU OF MINERAL RESOURCES, GEOLOGY AND GEOPHYSICS

RECORD

RECORD 1986/29

PRESSURE TESTING

(SINGLE WELL AND INTERFERENCE)

- COMPUTER PROGRAMS

'PULSE' AND 'PUPLT'

by

G.R. Morrison and R.W. De Nardi

RECORD 1986/29

PRESSURE TESTING
(SINGLE WELL AND INTERFERENCE)
- COMPUTER PROGRAMS
'PULSE' AND 'PUPLT'

by

G.R. Morrison and R.W. De Nardi

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

The aim of this record is to describe: the equations used for [oil and gas] well pressure interference testing; the methodology employed by programs PULSE and PUPLT used to analyse these tests; and, some examples of history matching of pressure interference data.

History matching by computer is accomplished in an iterative manner. Firstly, the user must gather all information known about the reservoir being tested. This information is organised into a model. Program PULSE accesses the parameters that describe the model and then generates pressure change values, at increments of time, at the modelled observation well. Secondly, the output from program PULSE is combined with the observed pressure history of the interference test on the graphics screen of the HP-150 (a Hewlett-Packard microcomputer used at BMR) by program PUPLT. This enables deficiencies in the match between the model output and the pressure history data to be observed. The model can then be modified to improve the match with the observed pressure history.

Note that transient conditions are assumed to prevail throughout the interference test. Also note that the defaulting case of an interference test, where the "observation" well is located at the distance of the borehole radius from the "flowing" well, coincides with the geometry of a single well production test - thus increasing the utility of this software package. Variable flow rate cases can also be analysed as program PULSE incorporates the principle of superposition.

Both program PULSE and PUPLT are written in FORTRAN 77 software language. These programs are operated on a HP-1000F mainframe computer within BMR. A HP-150 microcomputer is used as a data entry/retrieval terminal as well as a graphics display unit.

2. EQUATIONS AND THEORY (Dake, 1978 and Matthews et al, 1967)

- refer to section 2A. for the definition of each variable used below.

The partial differential equation governing pressure behaviour, with space and time, of a single-phase fluid is given below (for the rectangular co-ordinate system):

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} + \frac{\partial^2 p}{\partial z^2} = \frac{\phi u C_t}{k} \cdot \frac{\partial p}{\partial t}$$

This equation assumes that Ct is small, the permeability is constant and isotropic, u (viscosity) is independent of pressure, and, if the porosity is constant, that the pressure gradients involved are small. Note that in gas reservoirs (where u is highly pressure dependent, and Ct can be of the order of the inverse of pressure) some error can be expected in this system of matching model output with observed pressure history.

This partial differential equation can be transformed from rectangular co-ordinates into radial or cylindrical co-ordinates. This version is given below:

$$\frac{1}{r} \cdot \frac{\partial}{\partial r}\left\{r \cdot \frac{\partial p}{\partial r}\right\} + \frac{\partial^2 p}{\partial r^2} + \frac{1}{r} \cdot \frac{\partial p}{\partial r} = \frac{\phi u c}{k} \cdot \frac{\partial P}{\partial t}$$

These two forms of the same differential equation, in rectangular and radial co-ordinates, is called the diffusivity equation, and the constant, $k/\phi u C_t$, is known as the hydraulic diffusivity.

The "radial" solution to the diffusivity equation, for the case where the well is assumed to be situated in a porous medium of infinite radial extent, is as follows:

$$p(r,t) = p_i - \frac{qu}{2\pi kh} \cdot \left\{-\tfrac{1}{2}\, Ei\left[\frac{-\phi u C_t r^2}{4kt}\right]\right\}$$

where $-Ei(-x) = \int_x^\infty \frac{e^{-a}}{a}\, da$

This equation - when altered for practical, oilfield units - becomes ...

$$p_{wf} = p_i - \frac{70.6\, quB}{kh}\left\{-Ei\left[\frac{-\phi u C_t r_w^2}{0.00105\, kt}\right] + 2s\right\}$$

... when the "observation" well is placed at the distance of the wellbore radius (rw) from the "pulsing or flowing" well; this is the defaulting case of a well interference test which is equivilant to a single well pressure ~~well~~ test. The skin factor, S, accounts for the pressure change in the reservoir, near to the wellbore, due to formation damage. In the case of a true interference test, where the pulsing or flowing well is placed some significantly large distance from the observation well, then the skin factor is zero, and pwf reverts to p(r,t), where 'r' is the distance between the two wells. The "radial" solution then becomes:

$$p(r,t) = pi - \frac{70.6\ quB}{kh} \cdot \left\{ -Ei \left[\frac{-\emptyset u\ Ctr^2}{0.00105\ kt} \right] \right\}$$

The user of program PULSE must therefore be aware of the situation when an interference test, or alternatively, when a single well (defaulting case of an interference test) pressure test is being modelled, and adjust the skin factor accordingly.

Units for each of the variables appearing in the "radial" solution equation appear below. Note that program PULSE operates in terms of pressure changes due to a specified flow rate, not absolute pressure measurements (i.e. pi - p(r,t) = ..., not p(r,t) = pi - ...). Therefore, the unit of pressure change here is 'psi'.

q = SCF/D for gas wells
  = STB/D for oil wells (or water wells)
B = RB/SCF for gas wells
  = RB/STB for oil wells (or water wells)
u = centipoise
k = millidarcies
h = feet
Ø = fractional value
C_x = psi^{-1}
r = feet
t = hours

Note that total compressibility, Ct, can be calculated from the equation...

$$Ct = CoSo + CgSg + CwSw + Cf$$

...where phases not present in the reservoir during testing can be cancelled.

Up until now, only a single flow rate case has been considered. Program PULSE, however, can account for variable flow rates, including periods of shut-in, using the principle of superposition. This principle allows for the effect of each change in flow rate, on pressure, to be analysed independant of previous flow rates. The final total pressure change, pi - p(r,t), is calculated by summing or superimposing the effect of each of these individual flow rate periods. The flow rate changes are modelled here by step functions.

For example, consider the flow rate schedule as follows (for 'n' different flow rates):

$$\begin{array}{ll} q = q_1 , & 0 \leq t \leq t_1 , \\ q = q_2 , & t_1 < t \leq t_2 , \\ q = q_3 , & t_2 < t \leq t_3 , \\ \vdots & \vdots \\ q = q_n , & t_{n-1} \leq t \end{array}$$

The solution to the "radial" form of the diffusivity equation then becomes:

$$pi - p(r,t) = \frac{70.6\, q_1 uB}{kh} \cdot [-Ei(\theta/t)]$$

$$+ \frac{70.6\, (q_2 - q_1) uB}{kh} \cdot [-Ei(\theta/t-t_1)]$$

$$+ \frac{70.6\, (q_3 - q_2) uB}{kh} \cdot [-Ei(\theta/t-t_2)]$$

$$\ldots + \frac{70.6\, (q_n - q_{n-1}) uB}{kh} \cdot [-Ei(\theta/t-t_{n-1})]$$

where $\theta = (-\phi uCtr^2)/(0.00105k)$

When a single well pressure test is being analysed, and the skin factor is non-zero, the following term must be added to this series:

$$+ \quad \frac{70.6\, q_n uB}{kh} \quad . \qquad (2s)$$

The exponential intergal cannot be evaluated directly by computer; however, it can be approximated by a series expansion or a polynomial approximation (Milton et al, 1970). These two forms of Ei(x) are given below:

for $x > 0$

$$Ei(x) = \gamma + \ln x + \sum_{n=1}^{\infty} x^n/(n.n!)$$

where γ = Euler's constant = 0.57721 56649 ...

for $1 \leq x \leq \infty$

$$Ei(x) = \left[\frac{1}{xe^x}\right] \left\{\frac{x^4+a_1x^3+a_2x^2+a_3x+a_4}{x^4+b_1x^3+b_2x^2+b_3x+b_4}\right\} + \epsilon(x)$$

where $\epsilon(x)$ = the error term

$< 2 \times 10^{-8}$

and

a_1	= 8.57332 87401	b_1	= 9.57332 23454
a_2	= 18.05901 69730	b_2	= 25.63295 61486
a_3	= 8.63476 08925	b_3	= 21.09965 30827
a_4	= 0.26777 37343	b_4	= 3.95849 69228

Re-iterating the constraints of the solution to the diffusivity equation; this solution requires:

- radial flow into the well perforated (or open) over the entire net thickness of the formation
- a homogeneous and isotropic porous medium
- uniform thickness of the medium
- porosity and permeability constant (independent of pressure)
- fluid of small and constant compressibility
- small pressure gradients
- constant fluid viscosity
- negligible gravity forces

2A. DEFINITIONS OF VARIABLES

p	=	pressure
x,y,z	=	variables in the rectangular co-ordinate system
t	=	time variable
Ø	=	effective or connected porosity
u	=	viscosity of the reservoir fluid
Ct	=	the total compressibility of the reservoir. For the case of single-phase flow, Ct = Cp + Cf
Cp	=	compressibility of the reservoir fluid
Cf	=	compressibility of the reservoir rock, formation or pore volume
r	=	variable in the radial co-ordinate system
p(r,t)	=	the pressure in an infinite reservoir at any time, t, and distance, r, from the flowing well
pi	=	the initial reservoir pressure
q	=	the flow rate
k	=	permeability
h	=	the net thickness of the reservoir
Ei	=	the expontial integral function
pwf	=	bottom hole well flowing pressure
rw	=	radius of the wellbore
S	=	the skin factor
B	=	the oil, gas or water formation volume factor
Co	=	the oil compressibility
So	=	the oil saturation
Cg	=	the gas compressibility
Sg	=	the gas saturation
Cw	=	the water compressibility
Sw	=	the water saturation

3. SAMPLE RUNS

Two sample runs have been included in this section (refer to figures 3.1 to 3.6) and these are discussed below.

Figure 3.1 is an example of a model data-file that is input to program PULSE to generate values of pressure drop as a function of time. This particular model is for a drill stem test run over a gas bearing formation. Note that the "DISTANCE BETWEEN WELLS", in the case of a drill stem test, (the defaulting case of an interference test) is the radius of the wellbore (0.3542 feet). The unit of measure of the [GAS] FORMATION VOLUME FACTOR is RB/SCF, and hence, the unit of measure of the [GAS] FLOW RATE is SCF/D. The flow rate schedule is specified for three different time periods; as follows:

for 0 hrs $\leq$ t $\leq$ 10.85 hrs then Q = 16.8 MMSCF/D
10.85 hrs $<$ t $\leq$ 17.97 hrs then Q = 21.45 MMSCF/D
17.97 hrs $<$ t $\leq$ 38.00 hrs then Q = 0

The "TIME INCREMENT FOR OUTPUT..." (Figure 3.1) specifies the time interval at which a calculated pressure drop (due to the flow rate schedule) is placed into an output data-file for the next processing phase (program PUPLT).

Figure 3.2 is an example of a data-file containing graphical scaling information as well as the observed pressure history. This data-file, along with the output data-file from program PULSE, is entered into program PUPLT to generate Figure 3.3. Note that the observed pressure history is in the form of a relative rather than absolute pressure measurement. Should the program user wish to enter the observed absolute pressures, the "INITIAL RESERVOIR PRESSURE" must be entered.

Figure 3.3 is a match of the model output data (solid line) with the observed pressure history (crosses). The flow rate schedule has been drawn as a step function (just as it is treated by program PULSE) on the bottom of the graph (Figure 3.3). Should the match between observed and model data be unacceptable, the model can be altered and the process described above repeated, in an interative manner, until the program user is satisfied.

Figures 3.4 and 3.5 are a repeat of Figures 3.1 and 3.2 respectively, but for a different sample run (No. 2). Figures 3.4 to 3.6 relate to the case of a pressure interference test where the flowing and observing wells are located 10 560.0 ft (2 miles) apart. The reservoir being tested is a naturally fractured gas reservoir, and hence the low porosity of 0.0022.

Copies of programs PULSE and PUPLT are incorporated within Appendix 1 and 2 respectively.

[redacted] DST NO.1-GROSS INTERVAL 3823.0M TO 3868.0MKB

VISCOSITY

0.0237

FORMATION VOLUME FACTOR

0.000688

DISTANCE BETWEEN WELLS (FT)

0.3542

TOTAL COMPRESSIBILITY (PSI-1)

0.000114

POROSITY (FRACTION)

0.15

SKIN FACTOR AT POINT OF PRODUCTION AND OBSERVATION

2.6

NET THICKNESS OF RESERVOIR (FT)

21.3

PERMEABILITY (M-DARCIES)

18.4

TIME INCREMENT FOR OUTPUT IN HOURS

0.1

TIME (HRS.)----TO---->FLOW RATE

10.85	16800000
17.97	21450000
38.00	0

Figure 3.1 - Sample Run 1 Model Data-File

```
TITLE:
TITLE, DST NO.1, OBSERVED DATA (+)
SCALES:
INITIAL RESERVOIR PRESSURE (PSIA)
0.0
DELTA P - TOP (PSI)
-100.0
DELTA P - END (PSI)
1500.0
DELTA P - STEP SIZE
100.0
TIME - START (HRS.)
0.0
TIME - END (HRS.)
50.0
TIME - STEP SIZE
10.0
FLOW RATE - MIN.
0.0
FLOW RATE - MAX.
200000000
FLOW RATE - STEP SIZE
50000000
OBSERVED PRESSURE DATA:
TIME (HRS)---------->PRESSURE DROP (PSI)
  0.0000                 0.00
  0.1333              1435.
  0.1667               979.
  0.2000              1073.
  0.2333              1352.
  0.7333              1427.
  0.8167              1459.
  0.9000              1483.
  0.9833              1407.
  1.0667              1351.
  1.1500              1328.
  1.2333              1322.
  1.3167              1385.
  1.4500              1099.
  1.4833               892.
  1.6500               595.
  1.7333               758.
  1.8167               870.
     .                  .
     .                  .
     .                  .
     .                  .
     .                  .
 33.4833                48.
 34.4833                48.
 35.4833                48.
 36.4833                48.
 37.4833                48.
```

Figure 3.2 - Sample Run 1 Pressure History

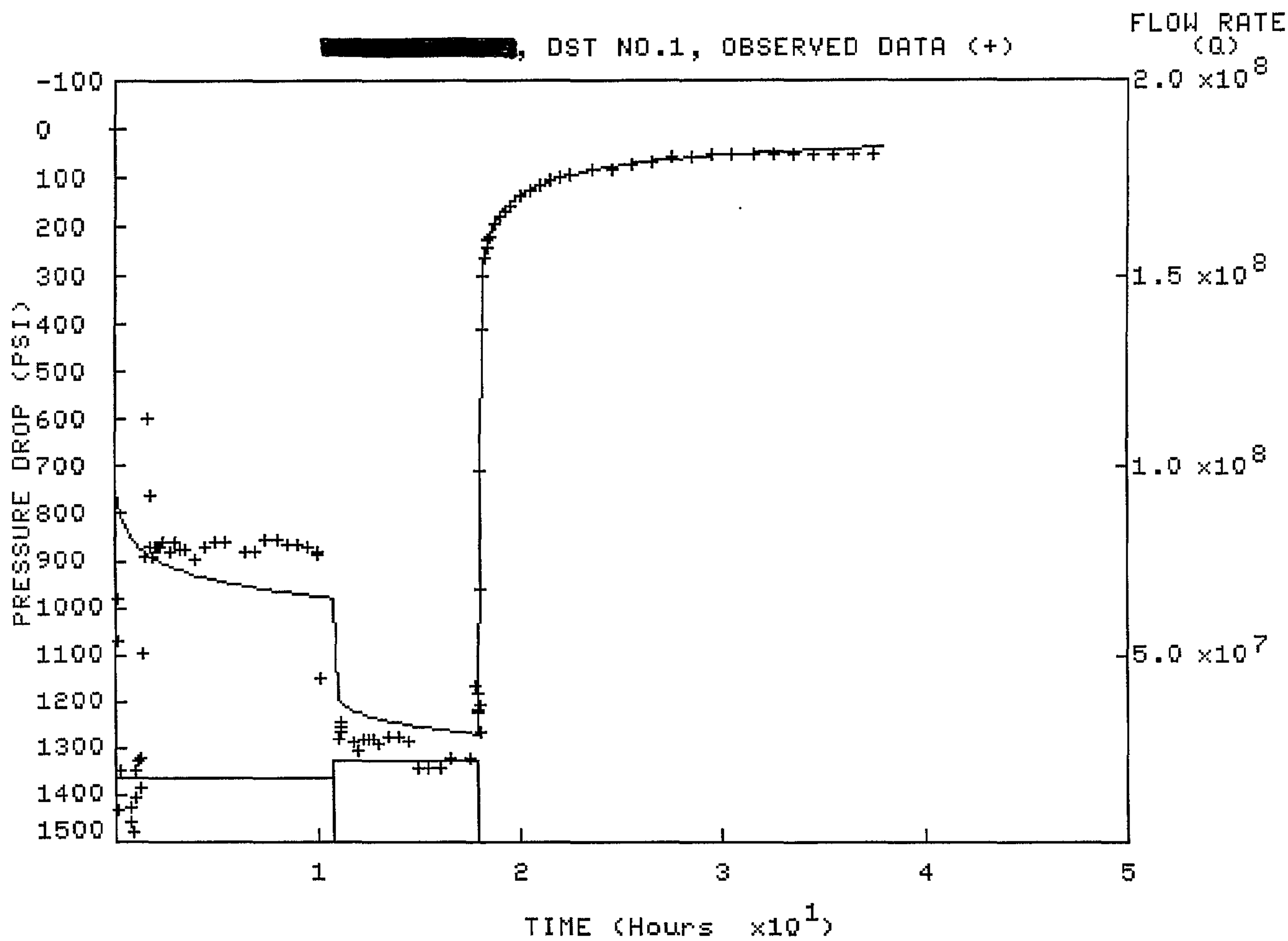

Figure 3.3 - Sample Run 1 Output

```
INTERFERENCE; TITLE
VISCOSITY
0.0189
FORMATION VOLUME FACTOR
0.000921
DISTANCE BETWEEN WELLS (FT)
10560.0
TOTAL COMPRESSIBILITY (PSI-1)
0.000289
POROSITY (FRACTION)
0.0022
SKIN FACTOR AT POINT OF PRODUCTION AND OBSERVATION
0.0
NET THICKNESS OF RESERVOIR (FT)
42
PERMEABILITY (M-DARCIES)
17
TIME INCREMENT FOR OUTPUT IN HOURS
1.0
TIME (HRS.)----TO---->FLOW RATE
  1.67                11042000
  2.67                11858000
 23.42                11867000
 47.42                10627000
 71.42                11060000
 95.42                10479000
119.42                12482000
143.42                11762000
167.42                12002000
215.42                11666000
288.42                11186000
336.42                11282000
360.42                11234000
384.42                12002000
408.42                11522000
444.42                11762000
450.08                11522000
696.08                       0
708.08                11282000
728.08                       0
740.08                11522000
800.00                       0
```

Figure 3.4 - Sample Run 2 Model Data File

```
TITLE:
PLOT TITLE - OBSERVED DATA (+)
SCALES:
INITIAL PRESSURE
0.0
DELTA P - TOP (PSI)
-10.0
DELTA P - END (PSI)
40.0
DELTA P - STEP SIZE
10.0
TIME - START (HRS.)
0.0
TIME - END (HRS.)
1000.0
TIME - STEP SIZE
100.0
FLOW RATE - MIN.
0.0
FLOW RATE - MAX.
100000000
FLOW RATE - STEP SIZE
50000000
OBSERVED PRESSURE DATA:
TIME (HRS)---------->PRESSURE DROP (PSI)
  0                       0.00
 24                       0.85
 48                       2.71
 72                       4.72
 96                       6.66
120                       8.53
144                      10.26
169                      11.93
 .                         .
 .                         .
 .                         .
 .                       18.72
784                      19.12
788                      19.02
792                      18.93
```

Figure 3.5 - Sample Run 2 Pressure History

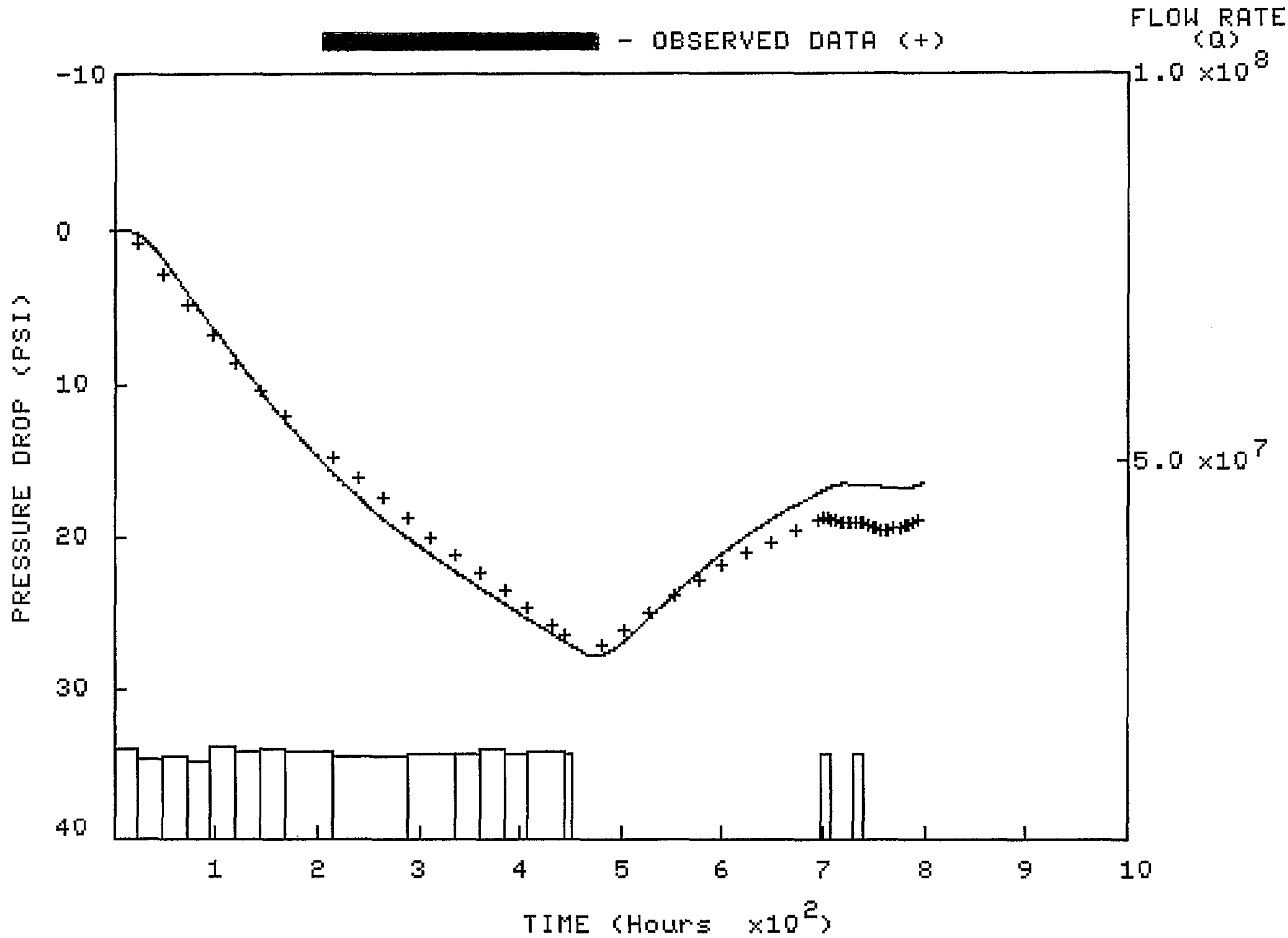

Figure 3.6 - Sample Run 2 Output

4. BIBLIOGRAPHY

(1) Dake, L.P., <u>Fundamentals of Reservoir Engineering</u>, developments in Petroleum Science 8, Elsevier, 1978.

(2) Matthews, C.S., et al, <u>Pressure Buildup and Flow Tests in Wells</u>, Monograph Volume 1, Society of Petroleum Engineers, 1967.

(3) Milton, A., et al, <u>Handbook of Mathematical Functions</u>, Dover, 1970.

Appendix 1 - Program PULSE

```
FTN77,Q,L,C
$FILES(0,2)
      PROGRAM PULSE
C
C        AUTHOR     : R.W.DE NARDI & G.R.MORRISON
C
C        DATE       : NOVEMBER 1985
C
C        FUNCTION   : Program PULSE uses reservoir parameters (perm. porosity
C                     etc.) to calculate pressure drop between a producing
C                     well and a test well. Pressure drop is calculated
C                     using Ei functions.
C
C        INPUT      : Data file containing reservoir parameters and flow rate
C
C        OUTPUT     :- Data file containing time, pressure drop & flow rate.
C                    - Spool file containing input data plus time,day,pressur
C                      drop & flow rate.
C
      DOUBLE PRECISION SIGMA,CALC,EI,DELP
C
      CHARACTER NAME*12,TITLE*40,DUM*1,REPLY*1,RESCRO*13,POSCUR*10,
     +          INVON*4,BLKON*4,WAIT*8,ENHOFF*4
C
      REAL k,Mu,TIME(100),Q(100)
C
      INTEGER Out,Indat
C
C
      RESCRO=CHAR(27)//'H'//CHAR(27)//'J'//CHAR(27)//'&a20c13Y'
      POSCUR=CHAR(27)//'&a-06c+0Y'
      INVON=CHAR(27)//'&dB'
      BLKON=CHAR(27)//'&dC'
      WAIT=CHAR(27)//'@'//CHAR(27)//'@'//CHAR(27)//'@'//CHAR(27)//'@'
      ENHOFF=CHAR(27)//'&d@'
      PAGENO=1
      MAXLIN=20
      LINCNT=0
      TINC=0.1
```

```
      START=0.005D0
      Out=6
C
      POSCUR(5:6)='6'
      RESCRO(8:9)='20'
      WRITE(1,10)RESCRO,INVON,ENHOFF,POSCUR
   10 FORMAT(A,'ENTER DATA FILE NAME ',A,'      ',
     +       2A,'_')
      READ(1,'(A)')NAME
      OPEN(4,FILE=NAME,STATUS='OLD')
C
      READ(4,'(A)')TITLE        ! READ IN TITLE
      READ(4,'(A)')DUM          ! skip a record
      READ(4,*)Mu               ! READ IN VISCOSITY
      READ(4,'(A)')DUM          ! skip a record
      READ(4,*)BETA             ! READ IN FORMATION VOLUME FACTOR
      READ(4,'(A)')DUM          ! skip a record
      READ(4,*)A                ! READ IN DISTANCE BETWEEN WELLS
      READ(4,'(A)')DUM          ! skip a record
      READ(4,*)CT               ! READ IN COMMPRESSIBILITY
      READ(4,'(A)')DUM          ! skip a record
      READ(4,*)PHI              ! READ IN POROSITY
      READ(4,'(A)')DUM          ! skip a record
      READ(4,*)S                ! READ IN SKIN FACTOR
      READ(4,'(A)')DUM          ! skip a record
      READ(4,*)H                ! READ IN THICKNESS
      READ(4,'(A)')DUM          ! skip a record
      READ(4,*)k                ! READ IN PERMEABILITY
      READ(4,'(A)')DUM          ! skip a record
      READ(4,*)OINC             ! READ IN OUTPUT TIME INTERVAL
      READ(4,'(A)')NUM          ! skip a record
      LIM=1
   50 READ(4,*,END=60)TIME(LIM),Q(LIM)   ! READ IN TIME AND FLOW RATES
      LIM=LIM+1                          !
      GOTO 50
   60 CONTINUE
D     WRITE(1,'(''LIM='',I5)')LIM
C
      CLOSE(4)
C
```

```
  100 RESCRO(8:9)='05'
      POSCUR(5:6)='12'
      WRITE(1,120)RESCRO,INVON,ENHOFF,POSCUR
  120 FORMAT(A,'ENTER NAME OF OUTPUT FILE TO BE CREATED (6CHAR:919:9) '
     +       ,A,'              ',2A,'_')
      READ(1,'(A)')NAME
      OPEN(5,FILE=NAME,ERR=140,IOSTAT=IOS,STATUS='NEW')
      GOTO 145
  140 IF(IOS.EQ.502)THEN
          RESCRO(8:9)='10'
          WRITE(1,144)RESCRO,BLKON,ENHOFF,INVON,NAME,ENHOFF,WAIT,WAIT
  144     FORMAT(2A,'WARNING',A,'  YOU ARE ABOUT TO OVERWRITE FILE ',
     +            5A)
          OPEN(5,FILE=NAME,STATUS='OLD')
      END IF
C
  145 WRITE(Out,'(''1'',20X,''Page No. '',I2,5(/))')PAGENO
      WRITE(Out,150)TITLE,Mu,BETA,A,CT,PHI,S,H,k,OINC
  150 FORMAT(20X,'(1) TITLE=',A,/,20X,'(2) VISCOSITY=',F6.4,
     +       /,20X,'(3) FORMATION VOLUME FACTOR=',F8.6,/,20X,
     +       '(4) DISTANCE BETWEEN WELLS (FT)=',F7.1,/,20X,
     +       '(5) COMPRESSIBILITY (PSI-1)=',F8.6,/,20X,
     +       '(6) POROSITY=',F7.4,/,20X,'(7) SKIN FACTOR=',F8.1,/,20X,
     +       '(8) THICKNESS=',F6.1,/,20X,'(9) PERMEABILITY (M-DARCIES)='
     +       ,F6.1,/,20X,'(10) OUTPUT INTERVAL (HRS)=',F6.1,' HRS')
C
      PAGENO=PAGENO+1
      WRITE(Out,180)
  180 FORMAT(3(/),13X,'TIME (HOUR)   DAY    PRESURE DROP(PSI)',
     +       '    FLOW RATE',//,13X,51('-'))
      CALC=70.6*Mu*BETA*Q(1)/(k*H)
D     WRITE(1,'(''CALC='',F8.4)')CALC
C
      PERIOD=TINC
  190 VAL=(AINT(PERIOD*10.))/10.
D     WRITE(1,*)PERIOD,VAL
      IF(AMOD(VAL,OINC).NE.0.0)GOTO 250
      SIGMA=PHI*Mu*CT*A**2/(0.00105*k*PERIOD)
D     WRITE(1,'(''SIGMA='',F8.4)')SIGMA
      IF(SIGMA.GE.88.0)THEN                    ! CHECK SIGMA LT 88.0
```

```
            GOTO 250
      END IF
      DELP=CALC*(EI(SIGMA)+2*S)
      IF(DELP.LE.START)THEN                    ! CHECK PRESSURE DROP > START
            GOTO 250
      END IF
      IF(PAGENO.GT.1)MAXLIN=35
      IF(LINCNT.GE.MAXLIN)THEN
            WRITE(Out,'(''1'',20X,''Page No. '',I2)')PAGENO
             WRITE(Out,180)
            PAGENO=PAGENO+1
            LINCNT=0
      END IF
      LINCNT=LINCNT+1
      DAY=PERIOD/24.
      WRITE(Out,200)PERIOD,DAY,DELP,Q(1)
  200 FORMAT(13X,F7.2,2X,F5.2,10X,F9.4,13X,F9.0)
      WRITE(5,*,ERR=1000,IOSTAT=IOS)PERIOD,DELP,Q(1)   ! OUTPUT RESULTS TO FI
  250 PERIOD=PERIOD+TINC
      IF(PERIOD.LE.TIME(1))GOTO 190
C
C      CALCULATE PRESSURE DROP FOR ALL OTHER FLOW RATES EXCEPT FIRST ONE
C
      DO I=2,LIM-1
            NUM=I
            PERIOD=TIME(I-1)+TINC
            CALL SERIES (TIME,Q,NUM,PERIOD,Mu,BETA,k,H,PHI,CT,A,S,
     +                   TINC,OINC,Out)
      END DO
      CLOSE(5)
      STOP
C
C         ERRORS ON WRITE STATEMENT ? ... COME HERE!
C
 1000 RESCRO(8:9)='05'
      IF(IOS.EQ.507)THEN
            WRITE(1,1100)RESCRO,INVON,ENHOFF,WAIT
 1100       FORMAT(2A,'YOU LEFT OFF THE SECURITY CODE & CARTRIDGE NO.',
     +             ' .... TRY AGAIN !',2A)
            GOTO 100
```

```
      END IF
      END
      SUBROUTINE SERIES (TIME,Q,NUM,PERIOD,Mu,BETA,k,H,PHI,CT,A,S,
     +                    TINC,OINC,Out)
C
C      SUBROUTINE SERIES CALCULATES PRESSURE DROP AS A SERIES FOR ALL
C      FLOW RATES EXCLUDING THE INITIAL FLOW RATE. REFER TO THE FOLLOWING
C      TEXTS FOR MORE INFOFATION.
C                 1. PRESSURE BUILDUP AND FLOW TESTS IN WELLS (C.S. MATTHEW
C                    & D.G. RUSSELL)
C                 2. ADVANCES IN WELL TEST ANALYSIS (R.C. EARLOUGHER. JR.)
C
      DOUBLE PRECISION CALC,SIGMA,DELP,EI,CALMOD
C
      REAL TIME(100),Q(100),Mu,k
C
      INTEGER Out
C
D     WRITE(1,50)Mu,BETA,k,H,PHI,CT,A,Q(1)
D  50 FORMAT(1X,'Mu=',F6.4,2X,'BETA=',F9.6,2X,'k=',F8.3,2X,'H=',F9.2,
D    +       2X,'PHI=',F7.4,2X,'CT=',F9.6,2X,'A=',F7.2,2X,'Q(1)',F9.0)
C
C      CALC. FIRST IN SERIES
C
  100 VAL=AINT(PERIOD*10.)/10.
D     WRITE(1,*)PERIOD,VAL
      IF(AMOD(VAL,OINC).NE.0.0)GOTO 500
      CALC=(70.6*Mu*BETA*Q(1))/(k*H)
      SIGMA=(PHI*Mu*CT*A*A)/(0.00105*k*PERIOD)
      IF(SIGMA.GE.88.)THEN
           DELP=0D0
      ELSE
           DELP=CALC*EI(SIGMA)
      END IF
D     WRITE(1,300)PERIOD,CALC,SIGMA,DELP
C
C        SUM REST OF TERMS IN SERIES
C
      DO 350 J=2,NUM
           SIGMA=(PHI*Mu*CT*A*A)/(0.00105*k*(PERIOD-TIME(J-1)))
```

```
          IF(SIGMA.GE.88.)GOTO 350
          CALC=(70.6*Mu*BETA*(Q(J)-Q(J-1)))/(k*H)
          DELP=DELP+CALC*EI(SIGMA)
          CALMOD=70.6*Mu*BETA*Q(J)/(k*H)
D         WRITE(1,300)PERIOD,CALC,SIGMA,DELP
D 300     FORMAT(1X,'PERIOD=',F6.0,2X,'CALC=',D9.4,2X,'SIGMA=',D9.4,
D    +              2X,'DELP=',D9.4,2X)
  350 CONTINUE
      DAY=PERIOD/24.
      WRITE(Out,400)PERIOD,DAY,DELP+(CALMOD*2.0*S),Q(NUM)
      WRITE(5,*)PERIOD,DELP+(CALMOD*2.0*S),Q(NUM)
  400 FORMAT(13X,F6.2,6X,F5.2,5X,F9.4,9X,F9.0)
  500 PERIOD=PERIOD+TINC
      IF(PERIOD.LE.TIME(NUM))GOTO 100   ! CHECK FOR END OF TIME INTERVAL
      RETURN
      END
      DOUBLE PRECISION FUNCTION EI(SIGMA)
C
C       THIS FUNCTION RETURNS A VALUE DETERMINED USING EXPONENTIAL INTERGRAL
C            FOR MORE INFORMATION REFER TO :- HANDBOOK OF MATHEMATICAL
C            FUNCTION with formulas,graphs, and mathematical tables (Edited
C            by Milton Abramowitz and Irene A. Stegun) pg 229-231.
C
      DOUBLE PRECISION SIGMA,SUM,TERM,A1,A2,A3,A4,B1,B2,B3,B4
D     WRITE(1,'(''SIGMA '',F8.4)')SIGMA
C
C        IF SIGMA IS < 1.0 USE A SERIES
C
      IF(SIGMA.LT.1D0)THEN
           TERM=-SIGMA
           SUM=TERM
           DO NTERMS=2,20
                TERM=-TERM*SIGMA/NTERMS*(NTERMS-1)/NTERMS
                SUM=SUM+TERM
D               WRITE(1,50)NTERMS,TERM,SUM,DABS(TERM/SUM)
D  50           FORMAT(1X,'NTERMS=',I5,3X,'TERM=',D9.6,3X,'SUM=',D9.7,
D    +                 3X,D9.6)
                IF(DABS(TERM/SUM).LT.10E-8)GOTO 100
            END DO
  100           EI=-0.5772156649-DLOG(SIGMA)-SUM
```

```
D                WRITE(1,'(''EI='',D9.7)')EI
C
C         IF SIGMA IS > 1 USE FORMULA
C
      ELSE
           A1=8.5733287401D0
           A2=18.0590169730D0
           A3=8.6347608925D0
           A4=0.2677737343D0
           B1=9.5733223454D0
           B2=25.6329561486D0
           B3=21.0996530827D0
           B4=3.9584969228D0
C
           TOP=(((SIGMA+A1)*SIGMA+A2)*SIGMA+A3)*SIGMA+A4
           BOTTOM=(((SIGMA+B1)*SIGMA+B2)*SIGMA+B3)*SIGMA+B4
           EI=(TOP/BOTTOM)/(SIGMA*DEXP(SIGMA))
D          WRITE(1,'(''EI='',D9.7)')EI
      END IF
      RETURN
      END
```

Appendix 2 - Program PUPLT

```
FTN77,Q,L,C,Y
$FILES(0,2)
      PROGRAM PUPLT
C
C         AUTHOR  : R.W.DE NARDI & G.M.MORRISON (R.E.G)
C
C         DATE    : NOVEMBER 1985
C
C         FUNCTION: Program PUPLT produces a plot on the HP-150 screen (6.30
C                   4.49). Two curves are produced, one contains observed
C                   data the other model information produced by program PULSE
C                   A histogram of flow rate (from model data file) is also
C                   plotted.
C
C                   NOTE: Use LOADER to load the library file %PLTLB
C
C         INPUT   : Two files required - 1. File containing observed data
C                                        2. File containing model and flow
C                                           rate data.
C         OUTPUT  : Graphics plot on HP150 screen. Plot can printed on
C                   HP Thinkjet printer by pressing the 'ENTER' key.
C
C
      CHARACTER NAME*6,DUM*1,TITLE*40,OBSDAT*20,NAME2*6,RESCRT*13,
     +          INVON*4,BLKON*4,WAIT*4,ENHOFF*4,MOVCUR*9
C
      REAL Initp
C
      RESCRT=CHAR(27)//'H'//CHAR(27)//'J'//CHAR(27)//'&a20c13Y'
      INVON=CHAR(27)//'&dB'
      BLKON=CHAR(27)//'&dC'
      WAIT=CHAR(27)//'@'//CHAR(27)//'@'//CHAR(27)//'@'//CHAR(27)//'@'
      ENHOFF=CHAR(27)//'&d@'
      MOVCUR=CHAR(27)//'&a-6c+0Y'
C
      RESCRT(8:9)='08'
   50 WRITE(1,100)RESCRT,INVON,ENHOFF,MOVCUR
  100 FORMAT(A,'ENTER DATA FILE NAME CONTAINING OBSERVED DATA   ',A,
     +       '       ',2A,'_')
```

27

```
      READ(1,'(A)')NAME
      OPEN(4,FILE=NAME,ERR=1000,IOSTAT=IOS,STATUS='OLD') !OBSERVED
  140 RESCRT(8:9)='15'
      WRITE(1,150)RESCRT,INVON,ENHOFF,MOVCUR
  150 FORMAT(A,'ENTER FILE CONTAINING MODEL  ',A,'       ',2A,'_')
      READ(1,'(A)')NAME2
      OPEN(5,FILE=NAME2,ERR=1400,IOSTAT=IOS,STATUS='OLD')! MODEL
C
C
      READ(4,'(A)')DUM              ! skip record
      READ(4,'(A)')TITLE            ! READ IN TITLE
      READ(4,'(A)')DUM              ! skip record
      READ(4,'(A)')DUM              ! skip record
      READ(4,*)Initp                ! READ IN INITIAL PRESSURE (PSIA)
      READ(4,'(A)')DUM              ! skip record
      READ(4,*)YTOP                 ! READ IN MIN. VALUE FOR Y AXIS
      READ(4,'(A)')DUM              ! skip record
      READ(4,*)YBOT                 ! READ IN MIN. VALUE FOR Y AXIS
      READ(4,'(A)')DUM              ! skip a record
      READ(4,*)PDROP                ! READ IN PRESSURE STEP SIZE
      READ(4,'(A)')DUM              ! skip record
      READ(4,*)XLFT                 ! READ IN MIN. VALUE FOR X AXIS
      READ(4,'(A)')DUM              ! skip record
      READ(4,*)XRHT                 ! READ IN MAX VALUE FOR X AXIS
      READ(4,'(A)')DUM              ! skip record
      READ(4,*)TDROP                ! READ IN TIME STEP SIZE
      READ(4,'(A)')DUM              ! skip record
      READ(4,*)FMIN                 ! READ IN MIN FLOW RATE
      READ(4,'(A)')DUM              ! skip record
      READ(4,*)FMAX                 ! READ IN MAX FLOW RATE
      READ(4,'(A)')DUM              ! sip record
      READ(4,*)FDROP                ! READ IN FLOW STEP SIZE
      READ(4,'(A)')DUM              ! skip record
      READ(4,'(A)')DUM              ! skip record
C
      IF(YBOT.GE.1000.)THEN         ! check for y scale values > 1000.
C                                   ! if yes ....
          XSCAL=-0.4                !   start printing Y axis label from this v
      ELSE                          ! else
          XSCAL=-0.3                !   start printing Y axis label from this v
```

```
      END IF                    !
C
C      SET UP PLOT SCREEN, DRAW PLOT BOUNDARIES
C
       CALL PLOTS_(PERROR,-1,1,1)
       CALL PLOT_(0.5,0.5,3)
       CALL PLOT_(5.5,0.5,2)
       CALL PLOT_(0.5,0.5,3)
       CALL PLOT_(0.5,4.0,2)
       CALL PLOT_(5.5,4.0,2)
       CALL PLOT_(5.5,0.5,2)
       CALL PLOT_(0.5,0.5,-3)
C
      CALL SYMB_(1.0,3.6,1,TITLE,0.0,40,1)                         ! WRITE TITLE
      CALL SYMB_(-0.4,1.0,1,'PRESSURE DROP (PSI)',90.0,19,1)       ! LABEL Y AXIS
      CALL SYMB_(2.0,-0.45,1,'TIME (Hours  x10 )',0.0,18,1)        ! LABEL X AXIS
      DEX=ALOG10(XRHT)
      CALL NUMB_(3.39,-0.38,1,DBLE(DEX-1.),0.0,0,1)
      CALL SYMB_(5.0,3.7,1,'FLOW RATE',0.0,9,1)                    ! LABEL FLOW
      CALL SYMB_(5.3,3.6,1,'(Q)',0.0,3,1)                          ! RATES
      YVAL=0.0
      YINC=(PDROP*3.5)/(YBOT-YTOP)
      IF(AMOD((YBOT-YTOP),5.0).EQ.0.0)THEN                   ! CHECK FOR FRAC. INC
C                                                            ! IF NO ...
           CALL NUMB_(XSCAL,0.0,1,DBLE(YBOT),0.0,0,1)        !    WRITE INTEGER VA
      ELSE                                                   ! ELSE ...
           CALL NUMB_(XSCAL,0.0,1,DBLE(YBOT),0.0,1,1)        !    WRITE REAL VAL.
      END IF                                                 !
      YPRESS=YBOT
      FINISH=(YBOT-YTOP)/PDROP
      DO R=1.,FINISH                                               ! LABEL
           YVAL=YVAL+YINC                                          !
           CALL PLOT_(0.0,YVAL,3)                                  ! Y-AXIS
           CALL PLOT_(0.05,YVAL,2)                                 !
           YPRESS=YPRESS-PDROP                                     ! WITH
           IF(AMOD((YBOT-YTOP),5.).NE.0.0)THEN                     !
                CALL NUMB_(XSCAL,YVAL-0.05,1,DBLE(YPRESS),0.0,1,1)!
           ELSE                                                    ! APPROPRIATE
                CALL NUMB_(XSCAL,YVAL-0.05,1,DBLE(YPRESS),0.0,0,1) ! VALUES.
           END IF                                                  !
```

```
      END DO                                                    !
C
      YFLOW=FMIN
      YVAL=0.0
      YINC=(FDROP*3.5)/(FMAX-FMIN)
      FINISH=(FMAX-FMIN)/FDROP
      DO R=1.,FINISH                                            !
           YVAL=YVAL+YINC                                       !
           CALL PLOT_(5.0,YVAL,3)                               !
           CALL PLOT_(4.95,YVAL,2)                              !
           YFLOW=YFLOW+FDROP                                    !
           VAL=ALOG10(YFLOW)                                    !
           POWER=AINT(VAL)                                      !
           REM=DIM(VAL,POWER)                                   !
           WHUM=10**REM                                         !
           CALL NUMB_(5.01,YVAL-0.05,1,DBLE(WHUM),0.0,1,1)      !
           CALL SYMB_(5.33,YVAL-0.05,1,'x10',0.0,3,1)           !
           CALL NUMB_(5.6,YVAL,1,DBLE(POWER),0.0,0,1)           !
      END DO                                                    !
C
      XVAL=0.0
      XINC=(TDROP*5.0)/(XRHT-XLFT)
      FINISH=(XRHT-XLFT)/TDROP
      DO R=1.,FINISH                                            ! LABEL
           XVAL=XVAL+XINC                                       ! X-AXIS
           CALL PLOT_(XVAL,0.0,3)                               ! WITH
           CALL PLOT_(XVAL,0.05,2)                              ! APPROPRIATE
           CALL NUMB_(XVAL-0.05,-0.2,1,DBLE(R),0.0,0,1)         ! VALUES.
      END DO                                                    !
C
C      PLOT TIME V's PRESSURE DROP OF OBSERVED DATA
C
  200 READ(4,*,END=250)TIME,PDROP                       ! READ TILL NO MORE DAT
      XVAL=((TIME-XLFT)*5.0)/(XRHT-XLFT)
      YVAL=((ABS(Initp-PDROP)-YBOT)*3.5)/(YTOP-YBOT)
      CALL SYMB_(XVAL,YVAL,1,'-20',0.0,-15,1)
      GOTO 200                                          ! GO BACK AND READ RECO
C
C      PLOT TIME V's PRESSURE DROP OF MODELED DATA
C
```

```
  250 CALL PLOT_(0.0,0.0,3)
      READ(5,*)HRS,PRESS,FLOW
      XVAL=((HRS-XLFT)*5.0)/(XRHT-XLFT)
      YVAL=((PRESS-YBOT)*3.5)/(YTOP-YBOT)
      CALL PLOT_(XVAL,YVAL,3)
  300 READ(5,*,END=500)HRS,PRESS,FLOW        ! READ DATA TILL EOF REACHED
      XVAL=((HRS-XLFT)*5.0)/(XRHT-XLFT)
      YVAL=((PRESS-YBOT)*3.5)/(YTOP-YBOT)
      CALL PLOT_(XVAL,YVAL,2)
      GOTO 300                               ! GO BACK AND READ A RECORD
C
C     PLOT FLOW RATES AS A HISTOGRAM
C
  500 REWIND(5)                        ! REWIND INPUT FILE TO OBTAIN FLOW RAT
      READ(5,*)TIME,B,FLOW
      YVAL=((FLOW-FMIN)*3.5)/(FMAX-FMIN)
      CALL PLOT_(0.0,YVAL,3)
      XVAL=((TIME-XLFT)*5.0)/(XRHT-XLFT)
  600 READ(5,*,END=999)TIME,B,FLOW     ! READ TIME & FLOW RATE
      XNEW=((TIME-XLFT)*5.0)/(XRHT-XLFT)
      YNEW=((FLOW-FMIN)*3.5)/(FMAX-FMIN)
      CALL PLOT_(XVAL,YVAL,2)
      CALL PLOT_(XVAL,YNEW,2)
      IF(YNEW.NE.YVAL)CALL PLOT_(XVAL,0.0,2)
      CALL PLOT_(XVAL,YNEW,3)
      XVAL=XNEW
      YVAL=YNEW
      GOTO 600                            ! GO BACK AND READ ANOTHER RECORD
  999 CLOSE(4)
      CLOSE(5)
      CALL PLOT_(0.0,0.0,999)             ! END PLOTTING ROUTINE
      STOP
C
C     IF ANY ERRORS OCCUR IN OPENING FILES THEN COME HERE
C
 1000 RESCRT(8:9)='15'
      IF(IOS.EQ.462)THEN
           WRITE(1,1100)RESCRT,BLKON,ENHOFF,WAIT
 1100      FORMAT(2A,'CAN''T FIND FILE ... PLEASE TRY AGAIN !',2A)
           GOTO 50
```

```
      ELSE
            WRITE(1,1200)RESCRT,IOS
1200        FORMAT(A,'RUNTIME ERROR ',I5,' OCCURRED CHECK MANUAL !')
            STOP
      END IF
1400  RESCRT(8:9)='15'
      IF(IOS.EQ.462)THEN
            WRITE(1,1100)RESCRT,BLKON,ENHOFF,WAIT
            GOTO 140
      ELSE
            WRITE(1,1200)RESCRT,IOS
            STOP
      END IF
      END
```