735
11/5/80

Dr. 4003

MASTER

DOE/AL/10563-T4

INTERACTIVE FRACTURE DESIGN MODEL

May 1980

Work Performed Under Contract No. AC32-79AL10563

Republic Geothermal, Inc.
Santa Fe Springs, California

U. S. DEPARTMENT OF ENERGY
Geothermal Energy

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

DISCLAIMER

"This book was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof."

This report has been reproduced directly from the best available copy.

Available from the National Technical Information Service, U. S. Department of Commerce, Springfield, Virginia 22161.

Price: Paper Copy \$8.00
Microfiche \$3.50

INTERACTIVE FRACTURE DESIGN
MODEL

Prepared for:

U.S. Department of Energy
Contract No. DE-AC32-79AL10563

by: Petroleum Training and Technical Services

May 1980

Republic Geothermal, Inc.
11823 E. Slauson Ave., Suite 1
Santa Fe Springs, California 90670

TABLE OF CONTENTS

	<u>Page</u>
Preface	iv
1.0 Introduction	1
1.1 Objective	1
1.2 Fracture Environment	2
2.0 Wellbore Temperature Model	4
3.0 Fracture Generation Model	6
4.0 Fracture Fluid Temperature Determination Model	
4.1 Sinclair Temperature Model	8
4.2 Whitsitt-Dysart Model	11
4.3 Mathematical Model	12
5.0 Combined Approach Model	14
6.0 Reference	18
Appendix A: Regression Analysis of Transient Time Function	19
A-1 General Data	
A-2 Data Input Format	22
A-3 Example Data	24
A-4 Example Output	26
Appendix B: User's Manual	36
B-1 Introduction	36
B-1.1 Interactive	36
B-1.2 Batch	36
B-2 Input Format	37
B-2.1 Data Input Format	38
B-2.1.1 Data Example	40
B-2.1.2 Data Output	42
B-2.2 Special Notes	50
B-2.2.1 Viscosity Data	50
B-2.2.2 Viscosity Relation	51
B-2.2.3 Selection of Time Step Size	51
B-2.2.4 Plot Routines	53
B-2.2.5 Options	53

TABLE OF CONTENTS (continued)

Appendix C: Subprogram Listings.....	54
C-1 Input	54
C-2 Main	57
C-3 Sinclr	59
C-4 Dysart	61
C-5 Ctdx	64
C-6 Echo	65
C-7 Weltem	66
C-8 Gertsm	67
C-9 Plotv	69
C-10 Terp	71
C-11 Dverk	72
C-12 Bxdata	83
C-13 Plott	85
C-14 Block Data	87

Preface

This report is a description of a computer program that can be used to design a fracture stimulation treatment for a geothermal reservoir. The program uses state-of-the-art methods to calculate the temperature of the fracture fluid as a function of time and distance in the fracture. This information is used to determine the temperature dependent properties of the fracture fluid. These fluid properties are utilized to calculate the fracture geometry as a function of time. The fracture geometry and temperature distribution of the fracture fluid are coupled so the subroutines that calculate these distributions have been made interactive.

Because of the high temperatures associated with geothermal reservoirs, standard isothermal calculations of fracture geometry inadequately describe fracturing in these reservoirs. Thus, this program allows for far more realistic evaluation of various design parameters than could be obtained from isothermal estimations. Even though no new mathematical models were developed, the coupling of existing temperature and fracture geometry models required a significant numerical analysis effort. The resulting equations were so complex that computer solution was required.

This model was developed by PTTS under the U.S. Department of Energy contract number DE-AC32-79AL10563.

FRACTURE MODEL REPORT

1.0 INTRODUCTION

The interactive fracture design model involves the development and synthesis of three basic subprograms:

- (a) Fracture Generation Program - This section calculates the fracture parameters based upon design input. The program was developed by Maurer Engineering.
- (b) Heat Transfer Program - This section determines the viscosity of the fluid within the fracture including both conduction and convection heat losses.
- (c) Wellbore Hydraulics Program - This section computes the temperature profiles within the wellbore using program WELTEM based on the published work by Romero-Juarez³.

1.1 OBJECTIVE

The objective for this model is to design a fracture stimulation treatment that includes the effects of temperature changes within the fracture itself as well as incorporating these changes into the design of the stimulation job.

1.2 THE FRACTURE ENVIRONMENT

The classical representation of the fracture geometry is shown in Figure 1. Important assumptions in the fracture environment for the purposes of this model are:

- (1) The fracture is symmetric to the wellbore and extends along the x-axis to a distance L .
- (2) Leakoff occurs in the y-axis in a direction normal to the fracture face.
- (3) This leakoff fluid is flowing counter to the heat flux and provides a substantial cooling effect referred to as heat blockage.
- (4) The temperature distribution in the fracture $T(x)$ varies according to the assumed leakoff rate and fracture parameters.
- (5) The temperature in the fracture $T(x)$ approaches the undisturbed reservoir temperature at a large enough distance.
- (6) The temperature profile in the formation rock $T(y)$ varies with distance from the fracture face and approaches undisturbed reservoir temperature at large 'y'.

FRACTURE REPRESENTATION AT A FIXED TIME

Fig. 1

The fracture length increases with time. The resulting change in viscosity-distance relationship causes variations in the fracture geometry.

2.0 WELLBORE TEMPERATURE MODEL:

The fracture fluid enters the formation at a higher temperature than the surface injection temperature and, therefore, the fluid viscosity entering the formation is lower than that measured at the surface. Proper fracture design calls for a realistic determination of the fracture fluid viscosity. The method of solution proposed by Romero-Juarez³ has been used to calculate the temperature profile within the flowing stream and to allow the determination of sand-face viscosity of the fracture fluid. The equation system for solving the temperature model is incorporated into the overall fracture model as WELTEM.

The governing equations are:

$$T(Z,t) = g_e Z + T_{es} - g_e A + (T_{fs} + g_e A - T_{es})e^{-Z/A} \quad (1)$$

where

$$A = \frac{WC_f[k_{he} + r_{ti}Uf(t)]}{2\pi r_{ti}k_{he}U} \quad (2)$$

and

T_{ma} = temperature in the formation rock matrix, °F.

C_f = specific heat of fluid, Btu/lbm-°F

$f(t)$ = transient heat conduction time function

W = mass flow rate, lbm/D

U = overall heat-transfer coefficient, Btu/D-sq ft-°F

r_{ti} = inner radius of tubing, ft.

T_{es} = rock temperature at surface of earth, °F

Z = depth, measured downward from surface, ft

t = time, D

T_{fs} = surface injection temperature of fluids, °F

g_e = geothermal gradient, °F/ft

k_{he} = thermal conductivity of earth, Btu/D-ft-°F

W_L = fluid leakoff rate, lb/sq ft sec

Generalizing this model to fit any possible system, the following changes were incorporated into the WELTEM program:

- (1) The original equations were used.
- (2) A multiple linear regression analysis on the Ramey time function was made to relate this transient time function to pumping time and casing OD. The data given in Fig. 2 was fitted to a regression curve and this input to the program. A typical form of the data is shown in Fig. 3, and the regression analysis is shown in Appendix A.

The WELTEM program can determine the temperature profile in any wellbore geometry at any time. From this profile the fluid temperature at the formation face is determined for use in the fracture generation program.

Figure 2 - Time function $f(t)$ for calculating well temperature after stimulation

Figure 3 - Temperature profiles after stimulations down 4-1/2" tubing in 7-5/8" casing (6-5/8" ID) with pumping time of 1 hour

3.0 FRACTURE GENERATION MODEL:

A fracture generation model was developed by Maurer Engineering utilizing the Geertsma and de Klerk approach. This model requires the following input:

- (1) Injection rate
- (2) Viscosity
- (3) Fracture height
- (4) Rock shear modulus
- (5) Spurt loss

Output consists of the following:

- (1) Fracture width
- (2) Fracture length
- (3) Fracture volume
- (4) Efficiency

all as functions of time.

4.0 FRACTURE FLUID TEMPERATURE DETERMINATION MODEL:

Techniques have been programmed to permit the engineer to calculate the temperature of the fluid in the fracture. They are the techniques published by Sinclair¹ and by Whitsitt and Dysart.²

The major differences between the techniques are based on the way in which the leakoff rate was handled in each case. Figure 4 illustrates the difference in interpretation of "leak-off rate." Both methods utilize an explicit solution for the temperature in terms of other parameters; i.e.,

$$T(x) = f(x, t, \dots)$$

Figure 4 - Leakoff Representations

In the development of the interactive fracture design model, the differential form of the temperature equation was solved instead of the integrated form. The value for $\frac{dT}{dx}$ was determined from various parameters, then this variable integrated along the fracture. This approach provided greater flexibility in varying the parameters and properties along the fracture.

4.1 SINCLAIR TEMPERATURE MODEL

The heat transfer model is based on the Sinclair¹ equations where heat transfer from the over burden is neglected.

$$T_{Ob} = \frac{1}{2} U (\tau - \delta M_D) \left\{ \exp[2\alpha(\delta + \xi)] \operatorname{erfc} \left(\frac{\delta + \xi}{2\sqrt{\tau - \delta M_D}} + \alpha\sqrt{\tau - \delta M_D} \right) + \operatorname{erfc} \left(\frac{\delta + \xi}{2\sqrt{\tau - \delta M_D}} - \alpha\sqrt{\tau - \delta M_D} \right) \right\} \quad (3)$$

where

$$\tau = \frac{kt}{H^2 M} \quad (4)$$

$$\delta = \frac{1}{-2} \alpha \ln \left| \frac{i_D}{i_D - 2\alpha x_D} \right| \quad (5)$$

$$M_D = \frac{M_{FAW}}{24MH} \quad (6)$$

$$i_D = \frac{i_D C_{FF}}{4Hk}, \text{ dimensionless injection rate} \quad (7)$$

$$\alpha = \frac{M_f uH}{2k} \quad (8)$$

$$\xi = \frac{|y| - \frac{1}{2}W}{H}, \text{ if } |y| > \frac{1}{2}W; \text{ if } |y| \leq \frac{1}{2}W, \xi = 0 \quad (9)$$

$$x_D = x/H \quad (10)$$

and

T_{Db} = dimensionless temperature

k = thermal conductivity of fluid filled-matrix, Btu/hr-ft-°F

t = time, hr

μ = viscosity of fluid, cp

H = fracture height, ft

M_f = volumetric heat capacity of fluid flowing in matrix,
Btu/cu ft-°F

M = volumetric heat capacity, of fluid filled matrix,
Btu/cu ft-°F

M_{fa} = volumetric heat capacity of fluid and sand in fracture,
Btu/cu ft-°F

i_D = dimensionless injection rate

C_{ff} = specific heat capacity of fluid in fracture, Btu/lb - °F

X = fracture width, in.

y = coordinate in the direction perpendicular to the fracture
face, ft

ρ = density, lb/cu ft

i = injection rate, B/D

W = coordinate in the direction of fracture growth, ft

The model was programmed to investigate the effects of the following parameters.

- (1) Injection rate
- (2) Fracture width
- (3) Fluid type
- (4) Fluid efficiency

Figure 5 - Flowchart - Sinclair Model

(5) Leakoff

(6) Formation rock and fluid properties

A flow chart of the model is shown in Figure 5. The model runs interactively to produce the output form shown in the examples found in the Appendices.

4.2 WHITSITT-DYSART MODEL

The Whitsitt-Dysart temperature profile model has been derived using a combination of a mass energy balance in the fracture zone and an energy balance in the formation zone.

Within the formation there is conductive and convective heat loss away from the fracture face. The conductive heat loss was derived from the heat flux in the rock matrix while the convective heat loss was produced by the flow of leakoff fluid into the rock matrix. These two processes can be modelled by the Fourier-Poisson type equation:

$$\frac{\partial^2 T_{ma}}{\partial y^2} - \frac{W_L C_f}{k_{hma} (1 - \phi)} \frac{\partial T_{ma}}{\partial y} = \frac{M_{ma}}{k_{hma}} \frac{\partial T_{ma}}{\partial t}$$

where the physical model is shown in Figure 6.

Solving this with the appropriate boundary conditions provides an analytical answer to the temperature distribution at and away from the fracture face. For the typical fracture operation, only the temperature gradient and the associated heat flux at the fracture face is

Figure 6 - Illustration of thermal model of fluid in fracture
From Whitsitt and Dysart (1970) courtesy of
SPE-AIME

needed. It is, therefore, only necessary to evaluate the heat flux at the fracture face. Differentiation of the temperature equation in the formation matrix and solving for the condition at $y = 0$ gives:

$$u_h(n,t) = \sqrt{\frac{M_{ma} k_{hma}}{\pi t}} (\Delta T_{ma})_0 \Psi(n) \quad (12)$$

The heat flux can be combined with the mass flow in the fracture to obtain the temperature distribution in the fracture zone.

4.3 MATHEMATICAL MODEL

An energy balance on the fluid flowing in the fracture produces the following governing equation:

$$\frac{dT}{dx} = \frac{2h}{w_t [1 - (x/L)^2] C_f} \sqrt{\frac{M_{ma} K_{hma}}{\pi t}} (T_i - T) \Psi(n) \quad (13)$$

$$\Psi(n) = e^{-n^2} - \sqrt{\pi} n \operatorname{erfc}(n) \quad (14)$$

$$n = \frac{W_t x C_f}{h L^2 (1 - \phi)} \sqrt{\frac{t}{M_{ma} k_{hma}}} \quad (15)$$

The following are the units and terms used:

T = temperature of fluid in fracture, °F

x = variable distance from wellbore into fracture, ft

h = vertical fracture height, ft

W_t = total mass flow rate, lb/sec, into one side of a vertical fracture, lb/sec

C_f = specific heat of fracture fluid, Btu/lb-°F

M_{ma} = volumetric heat capacity of matrix ($\rho_{ma} C_{ma}$), Btu/ cu ft-°F

K_{hma} = thermal conductivity of rock, Btu/ft-sec-°F

t = time, sec

ϕ = porosity

L = length of dynamic fracture, ft

This model assumes:

- (1) The leakoff rate varies linearly along the fracture from $x = 0$ to $x = 1$; i.e.

$$W_L = C_1 x + C_2 \quad (16)$$

- (2) Heat transfer from the top and bottom of the fracture is neglected.

(3) Internal energy changes are neglected.

The mathematical model consists of equations (13) through (16) and other fluid property equations as needed.

5.0 COMBINED APPROACH MODEL:

The combination of the fracture generation and the temperature-heat transfer models allows a determination of the fracture parameters considering changing temperatures and fluid viscosities in the fracture.

The flow charts in Figures 5 and 7 demonstrate the information flow. Essentially, the model utilizes an iteration between fracture geometry and temperature computation. Between each cycle, the temperature dependent viscosities are updated and reused; convergence occurs when successive fracture lengths are less than 1%.

The program was set up to run interactively from a terminal. At the present time the user can select either of the available temperature models depending on the type of fracture fluid being used.

Output is provided in tabular and graphical form.

The subprograms used in the model are shown below:

1. FRAC Sets up the fracture problem
2. COTDX Computes the temperature gradient in the future

Figure 7
Combination of Fracture Design
and
Temperature Model

Figure 7 (continued)

3. BLOCK DATA Common block data definition
4. PLOTT Plot routine
5. BXDATA Annotates the graph
6. INPUT Reads input parameters
7. DVERK Solves differential equations
8. PLOTV Plot routine
9. GERTSM Determines fracture parameters
10. MAIN Mainline
11. ECHO Echoes the input data
12. WELTEM Determines input wellbore temperature

6.0 REFERENCES

1. Sinclair, A. R.; "Heat Transfer Effects in Deep Well Fracturing." JPT. December 1971. 1484-1492.
2. Whitsitt, N. F., and Dysart, G. R.; "The Effect of Temperature on Stimulation Design." JPT. April 1970. 493-502.
3. Romero-Juarez, A.; "A Simplified Method for Calculating Temperature Changes in Deep Wells." JPT. June 1979. 763-768.

APPENDIX A

Regression Analysis of Transient Time Function

A-1 General Data

PROBLEM TITLETRANSIENT TIME FUNCTION

NUMBER OF VARIABLES TO READ IN.	3
NUMBER OF VARIABLES ADDED BY TRANSFORMATIONS.	0
TOTAL NUMBER OF VARIABLES	3
NUMBER OF CASES TO READ IN.	1000000
CASE LABELING VARIABLES	
LIMITS AND MISSING VALUE CHECKED BEFORE TRANSFORMATIONS	
BLANKS ARE.	ZEROS
INPUT UNIT NUMBER	0
REWIND INPUT UNIT PRIOR TO READING. . DATA. . .	YES

INPUT FORMAT

(3F10.4)

VARIABLES TO BE USED

	1 TFUNCTION	2 TFUMP	3 OCCASII
REGRESSION INTERCEPT.			NON-ZERO
GROUPING VARIABLE			
WEIGHT VARIABLE			
PRINT COVARIANCE MATRIX			YES
PRINT CORRELATION MATRIX.			YES
PRINT CORRELATION OF REGRESSION COEFFICIENTS.			NO
PRINT RESIDUALS			YES
PRINT NORMAL PROBABILITY PLOT			NO
PRINT DETRENDED NORMAL PROBABILITY PLOT			NO
NUMBER OF CASES READ.			24

VARIABLE	MEAN	STANDARD DEVIATION	COEFFICIENT OF VARIATION	MINIMUM	MAXIMUM
1 TFUNCT	0.94083	0.28797	0.30366	0.42000	1.47000
2 TPUMP	1.75000	1.17028	0.66873	0.0	3.50000
3 ODCASH	6.25000	1.33107	0.21297	4.50000	7.62500

COVARIANCE MATRIX

	TFUNCT 1	TPUMP 2	ODCASH 3
TFUNCT 1	0.0029		
TPUMP 2	0.2902	1.3696	
ODCASH 3	-0.1717	0.0000	1.7717

CORRELATION MATRIX

	TFUNCT 1	TPUMP 2	ODCASH 3
TFUNCT 1	1.0000		
TPUMP 2	0.0612	1.0000	
ODCASH 3	-0.4480	0.0000	1.0000

REGRESSION TITLE. TRANSIENT HEAT LOSS FUNCTION
 DEPENDENT VARIABLE. 1 TFUNCT
 TOLERANCE 0.0100

ALL DATA CONSIDERED AS A SINGLE GROUP

MULTIPLE R	0.9707	STD. ERROR OF EST.	0.0724
MULTIPLE R-SQUARE	0.9423		

ANALYSIS OF VARIANCE

	SUM OF SQUARES	DF	MEAN SQUARE	F-RATIO	F(TAIL)
REGRESSION	1.797	2	0.899	171.591	0.00000
RESIDUAL	0.110	21	0.005		

VARIABLE	COEFFICIENT	STD. ERROR	STD. REG COEFF	T	P(2 TAIL)	TOLERANCE
INTERCEPT	1.10333					
TPUMP 2	0.21190	0.013	0.861	16.434	0.0	1.000000
ODCASH 3	-0.09693	0.011	-0.440	-8.550	0.000	1.000000

ST OF PREDICTED VALUES, RESIDUALS, AND VARIABLES

NOTE - NEGATIVE CASE NUMBER DENOTES A CASE WITH MISSING VALUES.
THE NUMBER OF STANDARD DEVIATIONS FROM THE MEAN IS DENOTED BY UP TO 3 ASTERISKS TO THE
OF EACH RESIDUAL OR VARIABLE.
MISSING VALUES ARE DENOTED BY MORE THAN THREE ASTERISKS.

CASE LABEL	NO.	RESIDUAL	PREDICTED	VARIABLES		
			VALUE	1 TFUNCT	2 TFUMP	3 ODCASH
	1	-0.0242	0.4442	0.4200*	0.0 *	7.6250*
	2	-0.0811 *	0.5411	0.4600*	0.0 *	6.6250
	3	-0.1571 **	0.7471	0.5900*	0.0 *	4.5000*
	4	-0.0198	0.5502	0.5700*	0.5000*	7.6250*
	5	0.0029	0.6471	0.6500*	0.5000*	6.6250
	6	-0.0631	0.8531	0.7900	0.5000*	4.5000*
	7	0.0539	0.6561	0.7100	1.0000	7.6250*
	8	0.0469	0.7531	0.8000	1.0000	6.6250
	9	0.0010	0.9590	0.9600	1.0000	4.5000*
	10	0.0779 *	0.7621	0.8400	1.5000	7.6250*
	11	0.0810 *	0.8590	0.9400	1.5000	6.6250
	12	0.0550	1.0650	1.1200	1.5000	4.5000*
	13	0.0420	0.8600	0.9100	2.0000	7.6250*
	14	0.0750 *	0.9650	1.0400	2.0000	6.6250
	15	0.0691	1.1709	1.2400*	2.0000	4.5000*
	16	-0.0040	0.9740	0.9700	2.5000	7.6250*
	17	0.0491	1.0709	1.1200	2.5000	6.6250
	18	0.0631	1.2769	1.3400*	2.5000	4.5000*
	19	-0.0699	1.0799	1.0100	3.0000*	7.6250*
	20	-0.0169	1.1769	1.1600	3.0000*	6.6250
	21	0.0272	1.3828	1.4100*	3.0000*	4.5000*
	22	-0.1459 **	1.1859	1.0400	3.5000*	7.6250*
	23	-0.0828 *	1.2828	1.2000	3.5000*	6.6250
	24	-0.0188	1.4888	1.4700*	3.5000*	4.5000*

Data Input Format

All information is shown in the required sequence

```

1  YES      (DO YOU WANT TO ECHO THE INPUT?)
2  NO       (DO YOU WANT TO SEE THE TEMPERATURE PLOT?)
3  NO       (DO YOU WANT TO SEE THE VISCOSITY PLOT?)
4  YES      (DO YOU WANT TO USE THE SINCLAIR MODEL?)
5  YES      (DO YOU WANT TO CALL WELTEM?)
6  YES 06   (DO YOU WANT TO READ IN VISCOSITY TABLE?)
7  FLUID TYPE
8  WATER GEL
9  INITIAL TIME (SEC.)
10 XXX.X
11 MAX TIME (SEC.)
12 XXXXX.X
13 TIME INCREMENT (SEC.)
14 XXX.X
15 INJECTION RATE (BPM)
16 XX.X
17 VISCOSITY (CP)
18 XXX.XX
19 RHO
20 XX.X
21 ROCK SHEAR MODULUS (PSI)
22 XXXXXXXX.
23 FRACTURING FLUID COEFF. (FT/MIN**(1/2))
24 X.XXX
25 SPURT LOSS (FT3/FT2=FT)
26 XXX.XX
27 FRACTURE HEIGHT, FT.
28 XXXX.XX
29 SPEC. HEAT OF FRACTURE FLUID, BTU/LB F
30 XX.XX
31 VOL. HEAT CAP. (FLUID), BTU PER CU.FT. F
32 XXX.XX
33 VOL. HEAT CAP. (ROCK), BTU PER CU.FT. F
34 XXX.XX
35 THERMAL CONDUCT. (ROCK), BTU/FT SEC F
36 X.XXXXXX
37 THERMAL CONDUCT. (FLUID), BTU/FT SEC F
38 X.XXX
39 POROSITY, DECIMAL
40 .XXX
41 INITIAL TEMPERATURE
42 XXX.XX
43 RESERVOIR TEMP.
44 XXX.XX
45 DEPTH
46 XXXXX.XX
47 PERMEABILITY
48 XX.XX
49 PRESSURE
50 XXXXX.XX
51 LEAKOFF COEFF. = ZERO

```

52 XX.XXXX
 53 LEAKOFF COEFF. #1
 54 XX.XXXX
 55 DENSITY OF ROCK, LB PER CU.F.T
 56 XXXX.XX
 57 TUBING ID (INCHES)
 58 XXX.XXX
 59 TUBING OD (INCHES)
 60 XXX.XXX
 61 CASING ID (INCHES)
 62 XXX.XXX
 63 CASING OD (INCHES)
 64 XXX.XXX
 65 GEOTHERMAL GRADIENT (DEG F/FT)
 66 XX.XXX
 67 FORMATION SURFACE TEMP. (DEG F)
 68 XXX.XX
 69 FLUID SURFACE TEMP. (DEG F)
 70 XXXX.XX
 71 VISCOSITY TABLE
 72 30. 60.
 73 80. 60.
 74 100. 60.
 75 200. 53.
 76 300. 50.
 77 350. 2.

A-3 EXAMPLE DATA

```

00010 YES          (DO YOU WANT TO ECHO THE INPUT? )
00020 NO           (DO YOU WANT TO SEE THE TEMPERATURE PLOT?)
00030 NO           (DO YOU WANT TO SEE THE VISCOSITY PLOT?)
00040 YES          (DO YOU WANT TO USE SINCLAIR MODEL?)
00050 YES          (DO YOU WANT TO READ IN VISCOSITY TABLE?)
00054 FLUID TYPE
00057 WATER GEL
00060 INITIAL TIME (SEC.)
00070 0.0
00080 MAX TIME      (SEC.)
00090 5400.0
00100 TIME INCREMENT (SEC.)
00110 600.0
00120 INJECTION RATE (BPM)
00130 15.
00140 VISCOSITY (CP)
00150 200.
00160 RHO
00170 62.0
00180 ROCK SHEAR MODULUS (PSI)
00190 2600000.
00200 FRACTURING FLUID COEFF. (FT/MIN**(1/2) )
00210 0.004
00220 SPURT LOSS (FT3/FT2=FT)
00230 0.0
00240 FRACTURE HEIGHT, FT.
00250 200.
00260 SPEC. HEAT OF FRACTURE FLUID, BTU/LB. F
00270 1.
00280 VOL.HEAT CAP.(FLUIDS), BTU PER CU.FT. F
00290 62.4
00300 VOL.HEAT CAP.(ROCK), BTU PER CU.FT. F
00310 40.5
00320 THERMAL CONDUCT.(ROCK), BTU/FT SEC F
00330 0.000417
00331 THERMAL CONDUCT.(FLUID), BTU/FT SEC F
00332 9.432
00340 POROSITY, DECIMAL
00350 0.15
00360 INITIAL TEMPERATURE
00370 80.0
00380 RESERVOIR TEMP.
00390 350.
00400 DEPTH
00410 7200.
00420 PERMEABILITY
00430 10.0
00440 PRESSURE
00450 3000.
00460 LEAKOFF COEF. = ZERO
00470 0.0

```


00480 LEAKOFF COEF. #1
 00490 0.1
 00500 DENSITY OF ROCK, LB PER CU. FT.
 00510 162.0
 00520 TUBING ID (INCHES)
 00530 4.0
 00540 TUBING OD (INCHES)
 00550 4.5
 00560 CASING ID (INCHES)
 00570 7.675
 00571 CASING OD (INCHES)
 00572 8.2
 00580 GEOTHERMAL GRADIENT (DEG F/FT)
 00590 0.01
 00620 FORMATION SURFACE TEMP. (DEG F)
 00630 80.
 00640 FLUID SURFACE TEMP. (DEG F)
 00650 80.
 00760 VISCOSITY TABLE
 00761 50. 60.
 00770 80. 60.
 00780 100. 53.
 00800 200. 50.
 00810 350. 2.
 END OF DATA

HYDRAULIC FRACTURE DESIGN PROGRAM

USING

TEMPERATURE DEPENDENT FRACTURE FLUID PROPERTIES

FRACTURE PARAMETERS

TIME (MINS)	WIDTH (IN)	LENGTH (FT)	VOLUME CU.FT	EFF. %	AVG. FLUID TEMP. DEG.F
0.0	0.0	0.0	0.0	0.0	350.0
10.0	0.050	40.0	89.6	10.6	273.6
20.0	0.065	70.0	153.0	9.1	274.2
30.0	0.072	86.9	209.1	8.3	274.4
40.0	0.078	100.7	260.9	7.8	274.6
50.0	0.082	112.9	309.6	7.4	274.7
60.0	0.086	123.9	356.1	7.1	274.8
70.0	0.090	134.1	400.7	6.8	274.8
80.0	0.093	143.6	443.6	6.6	274.9
90.0	0.096	152.0	485.7	6.4	274.9

FRACTURE DATA

FLUID TYPE: WATER GEL

RATE: 15.00 BPM

MAX TIME: 1.50 HR.

FRAC. WIDTH: 0.00 IN.

FRAC. LENGTH: 152. FT.

INJ. TEMP.: 87.4 °F

RESERVOIR DATA

DEPTH: 7200.0 FT.

HEIGHT: 200.0 FT.

PRESSURE: 3000.0 PSI

TEMP: 350.0 °F

PERM: 10.000 MD

LEGEND

()	TIME=	0.17	HRS
⊙	TIME=	0.33	HRS
Δ	TIME=	0.50	HRS
⊕	TIME=	0.67	HRS
+	TIME=	0.83	HRS
x	TIME=	1.00	HRS
*	TIME=	1.17	HRS
⊖	TIME=	1.33	HRS
○	TIME=	1.50	HRS

FLUID VISCOSITY VS DISTANCE IN FRACTURE

Output from subroutine WELTEM

(This data is representative of two iterations and not the total sample problem.)

DEPTH	TEMPERATURE
0.	0.800000E+02
500.	0.800383E+02
1000.	0.801526E+02
1500.	0.803418E+02
2000.	0.806047E+02
2500.	0.809402E+02
3000.	0.813472E+02
3500.	0.818245E+02
4000.	0.823708E+02
4500.	0.829856E+02
5000.	0.836675E+02
5500.	0.844135E+02
6000.	0.852238E+02
6500.	0.861060E+02
7000.	0.870466E+02
7200.	0.874402E+02

CURVE #1

IN GERTSM: MU = 60.00

INITIAL OUTPUT FROM
FRACTURE GENERATION
MODEL

WIDTH	LENGTH	VOLUME	EFF.	TIME
0.057	49.2	93.0	11.0	10.0

DARCY VELOCITY IS 3.05 FT PER DAY

FRACTURE LENGTH IS 49.24 FT

TEMPERATURE RATIO	DISTANCE RATIO	
0.0000	0.0	
0.0789	0.0500	
0.1654	0.1000	
0.2584	0.1500	
0.3565	0.2000	
0.4576	0.2500	INITIAL TEMPERATURE
0.5588	0.3000	RATIO FROM SINCLAIR
0.6565	0.3500	MODEL
0.7470	0.4000	
0.8263	0.4500	
0.8910	0.5000	
0.9393	0.5500	
0.9712	0.6000	
0.9890	0.6500	
0.9970	0.7000	
0.9995	0.7500	
1.0000	0.8000	
1.0000	0.8500	
1.0000	0.9000	
1.0000	0.9500	
1.0000	1.0000	

DISTANCE	TEMPERATURE	VISCOSITY
0.0	87.44	60.00
2.46	108.16	59.43
4.92	130.85	57.84
7.39	155.28	56.13
9.85	181.05	54.33
12.31	207.59	52.77
14.77	234.15	51.98
17.23	259.82	51.21
19.70	283.58	50.49
22.16	304.38	45.79
24.62	321.38	29.48
27.08	334.05	17.31
29.54	342.43	9.27
32.01	347.12	4.77
34.47	349.20	2.77
36.93	349.86	2.13
39.39	349.99	2.01
41.86	350.00	2.00
44.32	350.00	2.00
46.78	350.00	2.00
49.24	350.00	2.00

TEMPERATURE VISCOSITY
WITHIN THE FRACTURE

IN GERTSM: MU = 50.79

UPDATED VISCOSITY

WIDTH	LENGTH	VOLUME	EFF.	TIME	NEW FRACTURE PARAMETERS
0.054	49.4	89.6	10.6	10.0	

CONVERGENCE REACHED IN SUBROUTINE GERTSM.

MAX TIME REACHED IN MAIN.

CURVE #8

IN GERTSM: MU = 50.76

WIDTH	LENGTH	VOLUME	EFF.	TIME
0.093	143.6	443.8	6.6	80.0

DARCY VELOCITY IS 1.06 FT PER DAY

FRACTURE LENGTH IS 143.56 FT

TEMPERATURE RATIO	DISTANCE RATIO
0.0000	0.0
0.0817	0.0500
0.1709	0.1000
0.2663	0.1500
0.3669	0.2000
0.4693	0.2500
0.5713	0.3000
0.6691	0.3500
0.7583	0.4000
0.8358	0.4500
0.8981	0.5000
0.9439	0.5500
0.9737	0.6000
0.9901	0.6500
0.9973	0.7000
0.9995	0.7500
1.0000	0.8000
1.0000	0.8500
1.0000	0.9000
1.0000	0.9500
1.0000	1.0000

DISTANCE	TEMPERATURE	VISCOSITY
----------	-------------	-----------

0.0	87.44	60.00
7.16	103.89	59.38
14.36	122.31	57.74
21.53	157.42	55.8
28.71	183.76	54.14
35.89	210.72	52.68
43.07	237.49	51.88
50.24	263.12	51.11
57.42	286.58	50.40
64.60	306.89	43.39
71.78	323.24	27.69
78.96	335.26	16.15
86.13	343.08	8.64
93.31	347.39	4.50
100.49	349.28	2.69
107.67	349.88	2.12
114.85	349.99	2.01
122.02	350.00	2.00
129.20	350.00	2.00
136.38	350.00	2.00
143.56	350.00	2.00

IN GERTSM: MU = 50.75

WIDTH	LENGTH	VOLUME	EFF.	TIME
0.093	143.6	443.8	6.6	80.0

CONVERGENCE REACHED IN SUBROUTINE GERTSM.

CURVE #9

IN GERTSM: MU = 50.75

WIDTH	LENGTH	VOLUME	EFF.	TIME
0.096	152.4	485.7	6.4	90.0

DARCY VELOCITY IS 0.99 FT PER DAY

FRACTURE LENGTH IS 152.45 FT

TEMPERATURE RATIO	DISTANCE RATIO
0.0000	0.0
0.0818	0.0500
0.1711	0.1000
0.2669	0.1500
0.3673	0.2000
0.4700	0.2500
0.5720	0.3000
0.6696	0.3500
0.7588	0.4000
0.8361	0.4500
0.8983	0.5000
0.9439	0.5500
0.9737	0.6000
0.9901	0.6500
0.9973	0.7000
0.9995	0.7500
1.0000	0.8000
1.0000	0.8500
1.0000	0.9000
1.0000	0.9500
1.0000	1.0000

UPDATED TEMPERATURE DATA

DISTANCE	TEMPERATURE	VISCOSITY
----------	-------------	-----------

0.0	87.44	60.00
7.62	108.92	59.38
15.24	132.37	57.73
22.37	157.61	55.97
30.49	183.87	54.13
38.11	210.85	52.67
45.73	237.62	51.87
53.36	263.24	51.10
60.98	286.68	50.40
68.60	306.95	43.32
76.22	323.29	27.65
83.85	335.28	16.13
91.47	343.09	8.64
99.09	347.39	4.50
106.71	349.28	2.69
114.33	349.88	2.12
121.96	349.99	2.01
129.58	350.00	2.00
137.20	350.00	2.00
144.82	350.00	2.00
152.45	350.00	2.00

UPDATED TEMPERATURE
VISCOSITY DATA

IN GERTSM: MU = 50.75

UPDATED FRACTURE
FLUID VISCOSITY

WIDTH	LENGTH	VOLUME	EFF.	TIME
0.096	152.4	485.7	6.4	90.0

FINAL AND
UPDATED FRAC.
PARAMETERS

CONVERGENCE REACHED IN SUBROUTINE GERTSM.

MAX TIME REACHED IN MAIN.

B-1 Introduction.

The program is designed to run either:

- (1) interactively
- (2) batch mode

B-1.1 Interactive:

In the interactive mode, a terminal is required to communicate with the host computer for input/output purposes. For interactive graphics, a terminal with plotting capabilities is required. This produces the output on the CRT in graphic form.

B-1.2 Batch:

A regular background or batch job could be set up to produce the same results with the output spooled to a CALCOMP or similar plotting device.

The load module requires 128 K bytes storage to execute.

Input Format

The input data set consists of cards or card image with a literal descriptor on one line and the numeric information on the line immediately following in an F10.0 format.

For example:

The complete input format is shown on the following page.

B-2.1 Data Input Format

All information is shown in the required sequence

```

1  YES      (DO YOU WANT TO ECHO THE INPUT?)
2  NO       (DO YOU WANT TO SEE THE TEMPERATURE PLOT?)
3  NO       (DO YOU WANT TO SEE THE VISCOSITY PLOT?)
4  YES      (DO YOU WANT TO USE THE SINCLAIR MODEL?)
5  YES      (DO YOU WANT TO CALL WELTEM?)
6  YES O6   (DO YOU WANT TO READ IN VISCOSITY TABLE?)
7  FLUID TYPE
8  WATER GEL
9  INITIAL TIME (SEC.)
10 XXX.X
11 MAX TIME (SEC.)
12 XXXXX.X
13 TIME INCREMENT (SEC.)
14 XXX.X
15 INJECTION RATE (BPM)
16 XX.X
17 VISCOSITY (CP)
18 XXX.XX
19 RHO
20 XX.X
21 ROCK SHEAR MODULUS (PSI)
22 XXXXXXXX.
23 FRACTURING FLUID COEFF. (FT/MIN**(1/2))
24 X.XXX
25 SPURT LOSS (FT3/FT2=FT)
26 XXX.XX
27 FRACTURE HEIGHT, FT.
28 XXXX.XX
29 SPEC. HEAT OF FRACTURE FLUID, BTU/LB F
30 XX.XX
31 VOL. HEAT CAP. (FLUID), BTU PER CU.FT. F
32 XXX.XX
33 VOL. HEAT CAP. (ROCK), BTU PER CU.FT. F
34 XXX.XX
35 THERMAL CONDUCT. (ROCK), BTU/FT SEC F
36 X.XXXXXXX
37 THERMAL CONDUCT. (FLUID), BTU/FT SEC F
38 X.XXX
39 POROSITY, DECIMAL
40 .XXX
41 INITIAL TEMPERATURE
42 XXX.XX
43 RESERVOIR TEMP.
44 XXX.XX
45 DEPTH
46 XXXXX.XX
47 PERMEABILITY
48 XX.XX
49 PRESSURE
50 XXXXX.XX
51 LEAKOFF COEFF. # ZERO

```

52 XX.XXXX
 53 LEAKOFF COEFF. #1
 54 XX.XXXX
 55 DENSITY OF ROCK, LB PER CU.F.T
 56 XXXX.XX
 57 TUBING ID (INCHES)
 58 XXX.XXX
 59 TUBING OD (INCHES)
 60 XXX.XXX
 61 CASING ID (INCHES)
 62 XXX.XXX
 63 CASING OD (INCHES)
 64 XXX.XXX
 65 GEOTHERMAL GRADIENT (DEG F/FT)
 66 XX.XXX
 67 FORMATION SURFACE TEMP. (DEG F)
 68 XXX.XX
 69 FLUID SURFACE TEMP. (DEG F)
 70 XXXX.XX
 71 VISCOSITY TABLE
 72 50. 60.
 73 80. 60.
 74 100. 60.
 75 200. 53.
 76 300. 50.
 77 350. 2.

B-2.2 Data Example

B-2.1.1 Data

00010 YES (DO YOU WANT TO ECHO THE INPUT?)
00020 NO (DO YOU WANT TO SEE THE TEMPERATURE PLOT?)
00030 NO (DO YOU WANT TO SEE THE VISCOSITY PLOT?)
00040 YES (DO YOU WANT TO USE SINCLAIR MODEL?)
00050 YES (DO YOU WANT TO READ IN VISCOSITY TABLE?)
00054 FLUID TYPE
00057 WATER GEL
00060 INITIAL TIME (SEC.)
00070 0.0
00080 MAX TIME (SEC.)
00090 5400.0
00100 TIME INCREMENT (SEC.)
00110 600.0
00120 INJECTION RATE (BPM)
00130 15.
00140 VISCOSITY (CP)
00150 200.
00160 RHO
00170 62.0
00180 ROCK SHEAR MODULUS (PSI)
00190 2600000.
00200 FRACTURING FLUID COEFF. (FT/MIN**(1/2))
00210 0.004
00220 SPURT LOSS (FT3/FT2=FT)
00230 0.0
00240 FRACTURE HEIGHT, FT.
00250 200.
00260 SPEC. HEAT OF FRACTURE FLUID, BTU/LB F
00270 1.
00280 VOL.HEAT CAP.(FLUIDS), BTU PER CU.FT. F
00290 62.4
00300 VOL.HEAT CAP.(ROCK), BTU PER CU.FT. F
00310 40.5
00320 THERMAL CONDUCT.(ROCK), BTU/FT SEC F
00330 0.000417
00331 THERMAL CONDUCT.(FLUID), BTU/FT SEC F
00332 9.432
00340 POROSITY, DECIMAL
00350 0.15
00360 INITIAL TEMPERATURE
00370 80.0
00380 RESERVOIR TEMP.
00390 350.
00400 DEPTH
00410 7200.
00420 PERMEABILITY
00430 10.0
00440 PRESSURE
00450 3000.
00460 LEAKOFF COEF. # ZERO
00470 0.0

00480 LEAKOFF COEF. #1
 00490 0.1
 00500 DENSITY OF ROCK, LB PER CU. FT.
 00510 162.0
 00520 TUBING ID (INCHES)
 00530 4.0
 00540 TUBING OD (INCHES)
 00550 4.5
 00560 CASING ID (INCHES)
 00570 7.675
 00571 CASING OD (INCHES)
 00572 8.2
 00580 GEOTHERMAL GRADIENT (DEG F/FT)
 00590 0.01
 00620 FORMATION SURFACE TEMP. (DEG F)
 00630 80.
 00640 FLUID SURFACE TEMP. (DEG F)
 00650 80.
 00760 VISCOSITY TABLE
 00761 50. 60.
 00770 80. 60.
 00780 100. 53.
 00800 200. 50.
 00810 350. 2.
 END OF DATA

B-2.1.2 Data Output

Output from subroutine WELTEM

(This data is representative of two iterations and not the total sample problem.)

DEPTH	TEMPERATURE
0.	0.800000E+02
500.	0.800383E+02
1000.	0.801526E+02
1500.	0.803418E+02
2000.	0.806047E+02
2500.	0.809402E+02
3000.	0.813472E+02
3500.	0.818243E+02
4000.	0.823708E+02
4500.	0.829856E+02
5000.	0.836673E+02
5500.	0.844155E+02
6000.	0.852282E+02
6500.	0.861060E+02
7000.	0.870466E+02
7200.	0.874402E+02

CURVE #1.

IN GERTSM: MU = 60.00

INITIAL OUTPUT FROM
FRACTURE GENERATION
MODEL

WIDTH	LENGTH	VOLUME	EFF.	TIME
0.057	49.2	93.0	11.0	10.0

DARCY VELOCITY IS 3.05^{1/2} FT PER DAY

FRACTURE LENGTH IS 49.24 FT

TEMPERATURE
RATIO

DISTANCE
RATIO

0.0000	0.0
0.0789	0.0500
0.1634	0.1000
0.2584	0.1500
0.3565	0.2000
0.4576	0.2500
0.5568	0.3000
0.6565	0.3500
0.7470	0.4000
0.8263	0.4500
0.8910	0.5000
0.9393	0.5500
0.9712	0.6000
0.9890	0.6500
0.9970	0.7000
0.9995	0.7500
1.0000	0.8000
1.0000	0.8500
1.0000	0.9000
1.0000	0.9500
1.0000	1.0000

INITIAL TEMPERATURE
RATIO FROM SINCLAIR
MODEL

DISTANCE	TEMPERATURE	VISCOSITY
----------	-------------	-----------

0.0	87.44	60.00
2.46	108.16	59.43
4.92	130.85	57.84
7.39	153.28	56.13
9.85	181.05	54.33
12.31	207.59	52.77
14.77	234.15	51.98
17.23	259.82	51.21
19.70	283.58	50.49
22.16	304.38	45.79
24.62	321.38	29.48
27.08	334.05	17.31
29.54	342.43	9.27
32.01	347.12	4.77
34.47	349.20	2.77
36.93	349.86	2.13
39.39	349.99	2.01
41.86	350.00	2.00
44.32	350.00	2.00
46.78	350.00	2.00
49.24	350.00	2.00

TEMPERATURE VISCOSITY
WITHIN THE FRACTURE

IN GERTSM: MU = 50.79

UPDATED VISCOSITY

WIDTH	LENGTH	VOLUME	EFF.	TIME	NEW FRACTURE PARAMETERS
0.054	49.4	89.6	10.6	10.0	

CONVERGENCE REACHED IN SUBROUTINE GERTSM.

MAX TIME REACHED IN MAIN.

CURVE #8

IN GERTSM: MU = 50.76

WIDTH	LENGTH	VOLUME	EFF.	TIME
0.093	143.6	443.8	6.6	30.0

DARCY VELOCITY IS 1.06 FT PER DAY

FRACTURE LENGTH IS 143.56 FT

TEMPERATURE RATIO	DISTANCE RATIO
0.0000	0.0
0.0817	0.0500
0.1709	0.1000
0.2663	0.1500
0.3669	0.2000
0.4695	0.2500
0.5713	0.3000
0.6691	0.3500
0.7585	0.4000
0.8358	0.4500
0.8981	0.5000
0.9439	0.5500
0.9737	0.6000
0.9901	0.6500
0.9973	0.7000
0.9993	0.7500
1.0000	0.8000
1.0000	0.8500
1.0000	0.9000
1.0000	0.9500
1.0000	1.0000

DISTANCE	TEMPERATURE	VISCOSITY
0.0	87.44	60.00
7.18	108.89	59.38
14.36	132.31	57.74
21.53	157.42	55.8
28.71	183.76	54.14
35.89	210.72	52.68
43.07	237.49	51.88
50.24	263.12	51.11
57.42	286.58	50.40
64.60	306.89	43.39
71.78	323.24	27.69
78.96	335.26	16.15
86.13	343.08	8.64
93.31	347.39	4.50
100.49	349.28	2.69
107.67	349.88	2.12
114.85	349.99	2.01
122.02	350.00	2.00
129.20	350.00	2.00
136.38	350.00	2.00
143.56	350.00	2.00

IN GERTSM: MU = 50.75

WIDTH	LENGTH	VOLUME	EFF	TIME
0.093	143.6	443.8	6.6	80.0

CONVERGENCE REACHED IN SUBROUTINE GERTSM.

HYDRAULIC FRACTURE DESIGN PROGRAM
 USING
 TEMPERATURE DEPENDENT FRACTURE FLUID PROPERTIES

FRACTURE PARAMETERS

TIME (MINS)	WIDTH (IN)	LENGTH (FT)	VOLUME CU. FT	EFF. %	AVG. FLUID TEMP. DEG. F
0.0	0.0	0.0	0.0	0.0	350.0
10.0	0.054	40.4	89.6	10.6	273.6
20.0	0.065	70.6	153.0	9.1	274.2
30.0	0.072	86.9	209.1	8.3	274.4
40.0	0.078	100.7	260.9	7.8	274.6
50.0	0.082	112.9	309.6	7.4	274.7
60.0	0.086	123.9	356.1	7.1	274.8
70.0	0.090	134.1	400.7	6.8	274.8
80.0	0.093	143.6	443.8	6.6	274.9
90.0	0.096	152.4	485.7	6.4	274.9

FLUID TEMPERATURE VS DISTANCE IN FRACTURE

FRACTURE DATA

FLUID TYPE: WATER GEL

DATE: 15.00 BPM

MAX TIME: 1.50 HR.

FRAC. WIDTH: 0.00 IN.

FRAC. LENGTH: 152. FT.

INJ. TEMP.: 87.4 °F

RESERVOIR DATA

DEPTH: 7200.0 FT.

HEIGHT: 200.0 FT.

PRESSURE: 3000.0 PSI

TEMP: 350.0 °F

PERM: 10.000 MD

LEGEND

□	TIME=	0.17	HRS
○	TIME=	0.33	HRS
△	TIME=	0.50	HRS
◇	TIME=	0.67	HRS
↑	TIME=	0.83	HRS
×	TIME=	1.00	HRS
*	TIME=	1.17	HRS
◊	TIME=	1.33	HRS
◊	TIME=	1.50	HRS

B-2.2.1

Viscosity Data

Viscosity data is input by specifying two bits of data from the viscosity-temperature plot. The data is fitted to the following semi-logarithmic relation:

$$\mu = b10^{at}$$

where.

μ = viscosity of fluid

a, b = constants

T = temperature of fluid

Computation:

$$a = \frac{\log \mu_1 - \log \mu_2}{T_1 - T_2}$$

Example:

$$T_1 = 100^\circ\text{F}, \mu_1 = 150 \text{ cp}$$

$$T_2 = 300^\circ\text{F}, \mu_2 = 10 \text{ cp}$$

$$a = \frac{\log 150 - \log 10}{(T_1 - T_2)}$$

$$= 0.01354$$

$$b = \frac{150}{10^{-.00528} \times 100}$$

$$= \frac{150}{.258} = 580.95$$

8-2.2.2 Viscosity Relation:

$$\mu = 581 \times 10^{-.00588 T}$$

8-2.2.3 Selection of Time Step Size

The program produces the fracture parameters at selected intervals depending on the increment specified on input (card #110).

A sample output is shown below.

FRACTURE PARAMETERS

TIME (MINS)	WIDTH (IN)	LENGTH (FT)	VOLUME CU.FT	EFF. %	AUG. FLUID TEMP. DEG.F
0.0	0.0	0.0	0.0	0.0	300.0
10.0	0.146	299.6	727.3	43.2	257.8
20.0	0.176	443.4	1301.2	38.7	259.6
30.0	0.197	556.4	1824.1	36.1	260.5
40.0	0.213	633.1	2315.1	34.4	261.0
50.0	0.226	739.1	2783.1	33.1	261.4
60.0	0.237	817.4	3233.3	32.0	261.7

Since the program is set up to handle nine time increments, it is recommended that the input increment be selected such that the total number of steps be less than nine.

For example:

Initial time = 0.0 secs.

Increment = 300. secs.

Total time = 1,300. secs.

This will produce: $\frac{1,300}{300} = 6$ time steps.

If the total time were 3,600 secs., then the maximum increment step would be: $\frac{3,600}{9} = 400$ secs.

Thus, increment is calculated from the following:

$$\text{MIN INCREMENT} = \frac{\text{MAX.TIME (Secs)}}{9}$$

3-2.2.4 Plot Routines

The program is designed to produce plotted output on any graph device that uses CALCOMP type routines. As implemented the present program will produce graphic output on a CRT e.g., the Tektronix system, if the interactive plot package is available, or on a hard copy plotter.

For a CRT device the user is prompted to select the frame size and overall graph dimensions by the plot routines. On a hard copy device e.g., CALCOMP plotter, the plots are automatically scaled to fit the plot width. The temperature plot is 8.5" X 14" and 8.5" X 11.5" for the viscosity plot.

3-2.2.5 Options

The first five data cards of input are self-explanatory and allow the user to select the options for use in running the program. The fracture fluid temperature calculation routine option is implemented as follows:

On card #4:

DO YOU WANT TO USE SINCLAIR MODEL

YES - use constant leakoff model by Sinclair

NO - use variable leakoff model by Whitsitt-Dysart

The user can dispense with the wellbore temperature calculation by using the appropriate response.

Appendix C: Subprogram Listings

C-1 Input

```

03460      SUBROUTINE INPUT
03470      DIMENSION FLTYPE(5)
03480      COMMON/C05/ JC(10,36),ECKVAL(36),JF(10)
03490      COMMON/C10/ WORK(2,9),H,RINJ,C,FH,FK,TIME,PHI,LENGTH,TEMP,W,TRES
03500      COMMON/C20/IN,OUT,X,IX
03510      COMMON/CLAIR/Q,MU,FH,G,CB,SP
03520      COMMON/C40/FLTYPE,DEPTH,PERM,PRESS,WIDTH
03530      COMMON/UJ10/CZERO,C1,C2,RHO,HKF
03540      COMMON/CAUG1/TINIT,TMAX,TINCR
03550      COMMON/COCT10/ DENROK,RHOFCE
03560      COMMON/ROCT11/DTE,DTE,DCE,GE,SFT,TES,TFS
03570      COMMON/TERCON/A,B,TLOW,THIGH,VLOW,VHIGH
03580      COMMON/INTTAB/NR,TEHINT(10),VISINT(10)
03590      DATA NW/10/
03600      INTEGER OUT,FLTYPE
03610      REAL LENGTH,MU
03620      READ(IN,20) (JF(K),K=1,NW)
03630      READ(IN,32) FLTYPE
03640      READ(IN,20) (JC(K,1),K=1,NW)
03650      READ(IN,4) TINIT
03660      ECKVAL(1) = TINIT
03670      READ(IN,20) (JC(K,2),K=1,NW)
03680      READ(IN,4) TMAX
03690      ECKVAL(2) = TMAX
03700      READ(IN,20) (JC(K,3),K=1,NW)
03710      READ(IN,4) TINCR
03720      ECKVAL(3) = TINCR
03730      READ(IN,20) (JC(K,4),K=1,NW)
03740      READ(IN,4) Q
03750      ECKVAL(4) = Q
03760      RINJ = Q
03770      READ(IN,20) (JC(K,5),K=1,NW)
03780      READ(IN,4) MU
03790      ECKVAL(5) = MU
03800      READ(IN,20) (JC(K,6),K=1,NW)
03810      READ(IN,4) RHO
03820      ECKVAL(6) = RHO
03830      READ(IN,20) (JC(K,7),K=1,NW)
03840      READ(IN,4) G
03850      ECKVAL(7) = G
03860      READ(IN,20) (JC(K,8),K=1,NW)
03870      READ(IN,4) CB
03880      ECKVAL(8) = CB
03890      READ(IN,20) (JC(K,9),K=1,NW)
03900      READ(IN,4) SP
03910      ECKVAL(9) = SP
03920      READ(IN,20) (JC(K,10),K=1,NW)
03930      20 FORMAT( 18A4 )
03940      25 FORMAT( 2X, 18A4 )
03950      READ(IN,4) H
03960      ECKVAL(10) = H

```

```

03970      FH = H
03980      4 FORMAT( F10.2)
03990      30 FORMAT( 2X, F10.2 )
04000      31 FORMAT( 2X, F10.4 )
04010      READ(IN,20) (JC(K,11),K=1,NW)
04020      READ(IN,4) C
04030      ECKVAL(11) = C
04040      READ(IN,20) (JC(K,12),K=1,NW)
04050      READ(IN,4) RHOFCE
04060      ECKVAL(12) = RHOFCE
04070      READ(IN,20) (JC(K,13),K=1,NW)
04080      READ(IN,4) FH
04090      ECKVAL(13) = FH
04100      READ(IN,20) (JC(K,14),K=1,NW)
04110      READ(IN,4) FK
04120      ECKVAL(14) = FK
04130      READ(IN,20) (JC(K,15),K=1,NW)
04140      READ(IN,4) HKF
04150      ECKVAL(15) = HKF
04160      READ(IN,20) (JC(K,16),K=1,NW)
04170      READ(IN,4) PHI
04180      ECKVAL(16) = PHI
04190      READ(IN,20) (JC(K,17),K=1,NW)
04200      READ(IN,4) TEMP
04210      ECKVAL(17) = TEMP
04220      READ(IN,20) (JC(K,18),K=1,NW)
04230      READ(IN,4) TRES
04240      ECKVAL(18) = TRES
04250      READ(IN,20) (JC(K,19),K=1,NW)
04260      READ(IN,4) DEPTH
04270      ECKVAL(19) = DEPTH
04280      READ(IN,20) (JC(K,20),K=1,NW)
04290      READ(IN,4) PERM
04300      ECKVAL(20) = PERM
04310      34 FORMAT(2X,F10.5 )
04320      READ(IN,20) (JC(K,21),K=1,NW)
04330      READ(IN,4) PRESS
04340      ECKVAL(21) = PRESS
04350      READ(IN,20) (JC(K,22),K=1,NW)
04360      READ(IN,4) CZERO
04370      ECKVAL(22) = CZERO
04380      READ(IN,20) (JC(K,23),K=1,NW)
04390      READ(IN,4) C1
04400      ECKVAL(23) = C1
04410      READ(IN,20) (JC(K,24),K=1,NW)
04420      READ(IN,4) DENROK
04430      ECKVAL(24) = DENROK
04440      READ(IN,20) (JC(K,25),K=1,NW)
04450      READ(IN,4) DTI
04460      ECKVAL(25) = DTI
04470      READ(IN,20) (JC(K,26),K=1,NW)

```

```

04480 READ(IN,4) DTE
04490 ECKVAL(26) = DTE
04500 READ(IN,20) (JC(K,27),K=1,NW)
04510 READ(IN,4) DCI
04520 ECKVAL(27) = DCI
04530 READ(IN,20) (JC(K,28),K=1,NW)
04540 READ(IN,4) DCE
04550 ECKVAL(28) = DCE
04560 READ(IN,20) (JC(K,29),K=1,NW)
04570 READ(IN,4) GE
04580 ECKVAL(29) = GE
04590 READ(IN,20) (JC(K,30),K=1,NW)
04600 READ(IN,4) TES
04610 ECKVAL(30) = TES
04620 READ(IN,20) (JC(K,31),K=1,NW)
04630 READ(IN,4) TFS
04640 ECKVAL(31) = TFS
04650 IF (NR.GT.0) GO TO 51
04660 READ(IN,20) (JC(K,32),K=1,NW)
04670 READ(IN,4) TLOW
04680 ECKVAL(32) = TLOW
04690 READ(IN,20) (JC(K,33),K=1,NW)
04700 READ(IN,4) ULOW
04710 ECKVAL(33) = ULOW
04720 READ(IN,20) (JC(K,34),K=1,NW)
04730 READ(IN,4) THIGH
04740 ECKVAL(34) = THIGH
04750 READ(IN,20) (JC(K,35),K=1,NW)
04760 READ(IN,4) VHIGH
04770 ECKVAL(35) = VHIGH
04780 32 FORMAT( 3A4 )
04790 GO TO 41
04800 51 READ(IN,20) (JC(K,32),K=1,NW)
04810 DO 40 JR=1,NR
04820 40 READ(IN,39) TEMINT(JR),VISINT(JR)
04830 39 FORMAT(2F10.3)
04840 41 CONTINUE
04850 RETURN
04860 END

```



```

12630 C      THIS IS THE MAIN ROUTINE
12640      COMMON/C10/ WORK(2,9),H,RINJ,C,FH,FK,TIME,PHI,LENGTH,TINJ,W,TRF
12650      COMMON/C20/IN,OUT,X,DX
12660      COMMON/C40/FLTYPE,DEPTH,PERM,PRESS,WIDTH
12670      COMMON/C50/ SAVAUT(10), NUMPTS(9),NPASS
12680      COMMON/C60/KURVE
12690      COMMON/CLAIR/G,MU,FH,G,CR,SP
12700      COMMON/CAUG1/TINIT,THAX,TINCR
12710      COMMON/CMG/MODELS
12720      COMMON/FRACV/WS(10),XLS(10),VOLS(10),EFFS(10),TIM(10)
12730      COMMON/TERCON/A,B,TLOW,THIGH,VLOW,VHIGH
12740      COMMON/EXOCT/ BXLEN
12750      COMMON/NEWINT/KALINT
12760      COMMON/INTTAB/NR,TEMINT(10),VISINT(10)
12770      DATA MAXROW/10/,IDOUT/3/
12780      INTEGER OUT,YES/'Y'/
12790 C      TINIT = INITIAL TIME,THAX=MAX TIME,TINCR=TIME INCRMT.
12800 C      TIME UNITS ARE READ IN SUBROUTINE INPUT, UNITS: SECONDS.
12810      REAL LENGTH,MU
12820 C      DO YOU WANT TO ECHO THE INPUT?
12830      READ(IN,4) JIN
12840      4 FORMAT( A1 )
12850 C      DO YOU WANT TO SEE THE TEMPERATURE PLOT?
12860      READ(IN,4) JPLOTT
12870 C      DO YOU WANT TO SEE THE VISCOSITY PLOT?
12880      READ(IN,4) JPLOTV
12890 C      DO YOU WANT TO USE SINCLAIR MODEL?
12900      READ(IN,4) MODELS
12910 C      DO YOU WANT TO CALL WELTEM ?
12920      READ(IN,4) KALLW
12930 C      DO YOU WANT TO READ IN VISCOSITY TABLE?
12940      READ(IN,6) KALINT,NR
12950      6 FORMAT(A1,4X,I2)
12960      IF (NR.LE. MAXROW) GO TO 17
12970      WRITE (6,16) NR, MAXROW
12980      16 FORMAT(/,3X,'*** INPUT ERROR. VALUE GIVEN FOR NUMBER OF ROWS',
12990      7 ' IN VISCOSITY TABLE',/,8X,'IS TOO LARGE. IT WAS READ AS ',
13000      7 ' I2,' MAXIMUM ALLOWABLE IS ',I2,'.',/,
13010      7 ' 8X,'PLEASE CORRECT AND RESUBMIT. JOB TERMINATED.')
```

STOP

```

13030      17 CONTINUE
13040      CALL INPUT
13050      IF ( (THAX/TINCR).LE.9.) GO TO 19
13060      WRITE(6,18)
13070      18 FORMAT(/,4X,'***** INPUT ERROR. MAX TIME STEPS IS NINE. ',
13080      7 '/,11X,'YOU MUST ADJUST INPUT VALUES 'MAX TIME' AND ',
13090      7 ' 'TIME INCREMENT' SUCH THAT:',/,11X,'MAX TIME DIVIDED BY',
13100      7 ' TIME INCREMENT IS LESS THAN OR EQUAL TO 9. ',
13110      7 '/,11X,'JOB TERMINATED. ')
```

19 CONTINUE

```

13130      IF (NR.GT.0) GO TO 30
13140      A =(ALOG10(VLOW)-ALOG10(VHIGH))/(TLOW-THIGH)
13150      B = VLOW / (10.*(A*TLOW))
13160 C      A AND B ARE USED IN SUBROUTINE TERP.
```

```

13170      30 IF (JIN.EQ.YES) CALL ECHO
13180      T = TINIT
13190      THAXFL = THAX + 1.
13200      IF (KALLW.EQ.YES) CALL WELTEX(TINJ,MU)
13210      5 T = T + TINCR
13220      IF (T.GT.THAXFL) GO TO 40
13230      T IS TIME IN SECONDS.
13240      CALL GERTSM(T,TINJ)
13250      GO TO 5
13260      40 WRITE(6,55)
13270      55 FORMAT(//,'          MAX TIME REACHED IN MAIN.      ',//)
13280      WRITE(IOUT,502)
13290      502 FORMAT(5(//))
13300      WRITE(IOUT,503)
13310      503 FORMAT(1X,78('X'))
13320      WRITE(IOUT,504)
13330      504 FORMAT(1X,'*',76X,'*')
13340      WRITE(IOUT,504)
13350      WRITE(IOUT,505)
13360      505 FORMAT(1X,'*',20X,'HYDRAULIC FRACTURE DESIGN PROGRAM',T79,'*')
13370      WRITE(IOUT,504)
13380      WRITE(IOUT,506)
13390      506 FORMAT(1X,'*',36X,'USING',T79,'*')
13400      WRITE(IOUT,504)
13410      WRITE(IOUT,507)
13420      507 FORMAT(1X,'*',12X,'TEMPERATURE DEPENDENT FRACTURE FLUID',
13430      ? ' PROPERTIES',T79,'*')
13440      WRITE(IOUT,504)
13450      WRITE(IOUT,503)
13460      WRITE(IOUT,504)
13470      WRITE(IOUT,508)
13480      508 FORMAT(1X,'*',25X,'FRACTURE PARAMETERS',T79,'*')
13490      WRITE(IOUT,504)
13500      KM1 = KURVE - 1
13510      DO 20 L=1,KM1
13520      20 SAVAUT(KURVE-L+1) = SAVAUT(KURVE-L)
13530      SAVAUT(1) = TRES
13540      WRITE(IOUT,21)
13550      21 FORMAT(1X,'*',5X,'TIME',6X,'WIDTH',6X,'LENGTH',6X,'VOLUME',
13560      ? 5X,'EFF.',4X,'AVG.FLUID TEMP.',T79,'*',
13570      ? /,1X,'*',4X,'(HINS)',6X,'(IN)',7X,'(FT)',8X,'CU.FT',6X,'X',
13580      ? 11X,'DEG.F',T79,'*')
13590      WRITE(IOUT,504)
13600      WRITE(IOUT,22) (TIH(MH),WS(MH),XLS(MH),VOLS(MH),EFFS(MH),SAVAUT(M
13610      ? ),MH=1,KURVE)
13620      22 FORMAT(1X,'*',F10.1,F10.3,2X,F10.1,2X,F10.1,F9.1,4X,F10.1,T79,'
13630      WRITE(IOUT,504)
13640      WRITE(IOUT,503)
13650      WRITE(IOUT,509)
13660      509 FORMAT( 10(//) )
13670      BXLEN = XLS(KURVE)
13680      IF (JPLOTT.EQ.YES) CALL PLOTT(NUMPTS)
13690      IF (JPLOTV.EQ.YES) CALL PLOTV(NUMPTS)
13700      IF (JPLOTT.EQ.YES.OR.JPLOTV.EQ.YES) CALL PLOT(0.,0.,999)
13710      STOP
13720      END

```

```

14510 SUBROUTINE SINCLR(Q,TRES,TINJ,W,H,T,KK,L,VISC,RH1CV,EFFPER)
14520 DIMENSION TEMP(21),RATIO(21)
14530 COMMON/PLAY/DIST(202,9),TREAL(202,9),VISCOS(202,9)
14540 COMMON/C50/SAVAVT(10),NUMPTS(9),NFPASS
14550 COMMON/COCT10/RHO1C1,RHOFCF
14560 COMMON/C60/KURVE
14570 COMMON/NEWINT/KALINT
14580 REAL LMAX,LMAX1,L,K,KK,M
14590 DATA EFF/0.5/
14600 INTEGER YES/'Y'/
14610 RHO1C1 = RH1CV
14620 EFF = EFFPER / 100.
14630 IF (KURVE.LE.9) GO TO 8
14640 WRITE(6,7)
14650 7 FORMAT(/,5X,'*** ERROR. ARRAY SUBSCRIPT TOO LARGE. SEE KURVE
14660 ? 'IN SUBROUTINE SINCLR. ',/,9X,'JOB TERMINATED. ')
14670 STOP
14680 8 CONTINUE
14690 T = T / 60.
14700 K = KK * 3600.
14710 Q = Q * 60.*24.
14720 XI = 1.
14730 DXN = 0.01
14740 IF (EFF.EQ.1.) GO TO 20
14750 10 CONTINUE
14760 XNUM = XI * DXN
14770 F = EXP(-XNUM)+2.*EFF*XNUM-1.
14780 IF (F.GT.0.0) GO TO 30
14790 XI = XI + 1.
14800 GO TO 10
14810 20 XNUM = 0.0
14820 30 CONTINUE
14830 WRITE(6,11)
14840 11 FORMAT(/)
14850 VF = 2.*W*XNUM/T
14860 WRITE(6,14) VF
14870 14 FORMAT(2X,'DARCY VELOCITY IS',F6.2,2X,'FT PER DAY',/)
14880 WRITE(6,16) L
14890 16 FORMAT(2X,'FRACTURE LENGTH IS',F10.2,2X,'FT',/)
14900 RHOFCF = 43.
14910 ALPHA = RHOFCF*VF*H/(2.*K*24.)
14920 TAU = K*T/(H*H*RHO1C1)
14930 M = RHOFCF*W/(24.*RHO1C1*H)
14940 QG = Q
14950 TEMP(1) = 0.0
14960 DO 100 I=1,21
14970 XI = I - 1
14980 RATIO(I) = .05*XI
14990 X = L*RATIO(I)
15000 DIST(I,KURVE) = X
15010 GAM = X/H
15020 Q = QG*RHOFCF*5.61/(4.*H*K*24.)
15030 DEL = (1./(2.*ALPHA))*ALOG(ABS(Q/(Q-2.*ALPHA*GAM)))
15040 RAD = SQRT(TAU-DEL*M)
15050 CRLS = DEL/2./RAD
15060 CRRS = ALPHA*RAD

```

```

15070      ARG1 = CRLS+CRRS
15080      ARG2 = CRLS-CRRS
15090      IF (ARG1.GT.1.0E01) GO TO 93
15100      XX1 = ERFC(ARG1)
15110      GO TO 94
15120      93 XX1 = 0.
15130      94 CONTINUE
15140      IF (ARG2.GT.1.0E01) GO TO 95
15150      XX2 = ERFC(ARG2)
15160      GO TO 96
15170      95 XX2 = 0.
15180      96 CONTINUE
15190      IF (XX1.LT.0.1E-10) XX1 = 0.1E-10
15200      IF (XX2.LT.0.1E-10) XX2 = 0.1E-10
15210      ZCR1 = (Q-2.*ALPHA*GAM)
15220      ZCR2 = Q / ZCR1
15230      ZCR3 = ZCR2 * XX1
15240      CRTPI = (ZCR3 + XX2) / 2.
15250      TEMP(1) = 1. - CRTPI
15260      100 CONTINUE
15270      TOT = 0.0
15280      DO 210 J=1,21
15290      TREAL(J,KURVE) = TINJ + TEMP(J) * (TRES-TINJ)
15300      IF (KALINT.NE.YES) GO TO 105
15310      CALL INTERP(TREAL(J,KURVE),VIS)
15320      GO TO 106
15330      105 CALL TERP(TREAL(J,KURVE),VIS)
15340      106 CONTINUE
15350      VISCOS(J,KURVE) = VIS
15360      210 TOT = TOT + TREAL(J,KURVE)
15370      AVGT = TOT / 21.
15380      SAVAVT(KURVE) = AVGT
15390      WRITE(6,21)
15400      21 FORMAT(6X,'TEMPERATURE',5X,'DISTANCE',/,8X,'RATIO',9X,'RATIO',/)
15410      DO 200 J=1,21
15420      WRITE(6,22) TEMP(J),RATIO(J)
15430      22 FORMAT(2X,F13.4,F14.4)
15440      200 CONTINUE
15450      IF (KALINT.NE.YES) GO TO 205
15460      CALL INTERP(AVGT,VISC)
15470      GO TO 206
15480      205 CALL TERP(AVGT,VISC)
15490      206 CONTINUE
15500      WRITE(6,23)
15510      23 FORMAT(///, '          DISTANCE      TEMPERATURE      VISCOSITY',/
15520      T = T * 60.
15530      NUMPTS(KURVE) = 21
15540      NPASS = KURVE
15550      KN = KURVE
15560      WRITE(6,5566) (DIST(HB,KN),TREAL(HB,KN),VISCOS(HB,KN),
15570      ? HB=1,21)
15580      5566 FORMAT( 3X,3F15.2)
15590      Q = QQ/(60.*24.)
15600      RETURN
15610      END

```

```

00010      SUBROUTINE DYSART(TIME,LENGTH,WIDTH,EFF,VAUG)
00020 C      INCOMING:  TIME,LENGTH,WIDTH,EFF  PASSED OUT:  VAUG
00030 C      VAUG IS VISCOSITY CORRESPONDING TO AVG. TEMP.
00040      COMMON/C10/WORK(2,9),H,RINJ,C,FH,FK,OLDTIM,PHI,OLDLEN,TEMPV,U,TI
00050      COMMON/C20/IN,OUT,X,IX
00060      COMMON/C30/YPRI(1)
00070      COMMON/C40/FLTYPE,DEPTH,PERH,PRESS,OLDWID
00080      COMMON/C50/SAVAUT(10),NUMPTS(9),NFASS
00090      COMMON/C60/KURVE
00100      COMMON/FLAY/ XF(202,9),YF(202,9),VISCOS(202,9)
00110      COMMON/NEWINT/KALINT
00120      DIMENSION TEMP(1),CAR(24),FLTYPE(5)
00130      DATA POS/0.1E-20/,MAX/200/,KPL/1/
00140      INTEGER OUT,ONE/1/,TWO/2/,FLTYPE,YES/'Y'/
00150      REAL LENGTH
00160      EXTERNAL CINTX
00170      KEG = KURVE
00180      OLDTIM = TIME
00190      OLDLEN = LENGTH
00200      OLDWID = WIDTH
00210      NEQT = 1
00220      MSFLAG = 0
00230      WRITE(6,39) KPL
00240 39 FORMAT(/,5X,'CALL TO DYSART #',I2 )
00250      KPL = KPL + 1
00260      IF (KEG.LE.9) GO TO 42
00270      WRITE(OUT,46)
00280 46 FORMAT(/,3X,'*** VALUE OF KEG IN SUB.DYSART EXCEEDS ',
00290      ? 'ARRAY SIZE 9. JOB TERMINATED. ')
00300      STOP
00310 42 CONTINUE
00320      X = 0.0
00330      KOUNT = 1
00340      XF(1,KEG) = 0.0
00350      YF(1,KEG) = TEMPV
00360      IF (KALINT.NE.YES) GO TO 37
00370      CALL INTERP(TEMPV,U)
00380      GO TO 38
00390 37 CALL TERR(TEMPV,U)
00400 38 CONTINUE
00410      VISCOS(1,KEG) = U
00420 C      WRITE(6,41) EFF,TIME
00430 C 41 FORMAT(/,2X,' EFF= ',E14.6,7X,' TIME = ',E14.6,' SECONDS. ')
00440 C      WRITE(6,43) LENGTH,WIDTH
00450 C 43 FORMAT(2X,' LENGTH=',E14.6,' FT.',3X,' WIDTH =',E14.6,' IN.',/)
00460 C      WRITE(3,44) TIME
00470 C 44 FORMAT('TIME=',E15.7)
00480 C      WRITE(6,42)
00490 C      W = RINJ * 5.833333 * EFF / 100.
00500      W = RINJ * 5.833333 * 0.5

```

```

00510      TEMP(1) = TEMPV
00520      KNIT = 0
00530      40 CONTINUE
00540      KNIT = KNIT + 1
00550      IF (KNIT.GT.200) GO TO 79
00560      XEND = X + DX
00570      IF (ABS(LENGTH-XEND).LE.POS) GO TO 50
00580      ONE = 1
00590      CALL DVERK(NEGT,CDTDX,X,TEMP,XEND,0.01,ONE,CAR,TWO,WORK,NEH)
00600      ZTEMP = TEMP(1)
00610      XOL = X / LENGTH
00620 C      WRITE(OUT,45) X,XOL,TEMP(1),YPRIME(1)
00630 C 45 FORMAT(6X,F6.1,4X,3E15.6)
00640 C      WRITE(3,48) X,TEMP(1)
00650 C 48 FORMAT(F3.1,E15.7)
00660 C      WRITE(OUT,49) X,TEMP(1)
00670 C 49 FORMAT(2X,F3.1,2X,E14.6 )
00680 C.....PLOT OPTIONS AND SELECT NEW TIME HERE.
00690      IF (KOUNT.LE.MAX) GO TO 58
00700      IF( MSFLAG.EQ.1) GO TO 55
00710      WRITE(OUT,57) MAX
00720 57 FORMAT(/,' MAX ARRAY SIZE ',I3,' REACHED IN SUB.DYSART. ',
00730 ? ' NO MORE POINTS SAVED.',/, ' PROGRAM CONTINUES.',/)
00740      MSFLAG = 1
00750      GO TO 56
00760 55 CONTINUE
00770 58 KOUNT = KOUNT + 1
00780 56 CONTINUE
00790      XF(KOUNT,KEG) = X
00800      YF(KOUNT,KEG) = ZTEMP
00810      IF (KALINT.NE.YES) GO TO 59
00820      CALL INTERF(ZTEMP,V)
00830      GO TO 60
00840 59 CALL TERP(ZTEMP,V)
00850 60 CONTINUE
00860      VISCOS(KOUNT,KEG) = V
00870      IF (X.LT.LENGTH) GO TO 40
00880 50 WRITE(OUT,52)
00890 52 FORMAT(/,' *** CONVERGENCE IN DVERK. '/')
00900      KOUNT = KOUNT - 1
00910      XKOUNT = KOUNT
00920      TOT = 0.0
00930      DO 153 K=1,KOUNT
00940 153 TOT = TOT + YF(K,KEG)
00950      AVGTMP = TOT / XKOUNT
00960      IF (KALINT.NE.YES) GO TO 61
00970      CALL INTERF(AVGTMP,VAUG)
00980      GO TO 62
00990 61 CALL TERP(AVGTMP,VAUG)
01000 62 CONTINUE

```

```

01010      SAVAUT(KURVE+1) = AVGTMP
01020 C      WRITE(6,76) AVGTMP,VAUG
01030 C 76 FORMAT(/,3X,'TRACE AVG.TEMP=',E14.6,4X,'% VISCOS=',E14.6,/)
01040      WRITE(6,75)
01050 75 FORMAT(5X,'LENGTH',7X,'TEMP',8X,'VISCOSITY',/)
01060      WRITE(OUT,72) (XF(KU,KEG),YP(KU,KEG),VISCOS(KU,KEG),
01070      ? KU=1,KOUNT,20)
01080 72 FORMAT(2X,F6.1,5X,F8.3,5X,F8.3)
01090      KONTEV = KOUNT/20*20
01100      IF(KONTEV.NE.KOUNT) WRITE(OUT,72) XF(KOUNT,KEG),YP(KOUNT,KEG),
01110      ? VISCOS(KOUNT,KEG)
01120      WRITE(OUT,73)
01130 73 FORMAT(/)
01140 C      SAVE KOUNT (# OF POINTS) PER EACH CURVE, FOR USE BY PLOT ROUTINE
01150      NUMPTS(KEG) = KOUNT
01160      NPASS = KEG
01170 C 42 FORMAT(/,T10,'X',T23,'X/L',T36,'TEMP',T51,'DT/DX')
01180 C 42 FORMAT(/,T6,'X',T16,'TEMP' )
01190 C      ENDFILE 3
01200 C      REWIND 3
01210      RETURN
01220 79 WRITE(6,80)
01230 80 FORMAT(///,5X,'*** UNABLE TO CONVERGE IN 200 ITERATIONS.',
01240      ? /,9X,'JOB TERMINATED IN SUBROUTINE IYSART. ')
01250      STOP
01260      END

```

C-5 Ccdtx

```

01270      SUBROUTINE CDTDX(NEQT,X,TEMP,YPRIME)
01280      COMMON/C10/ WORK(2,9),H,RINJ,C,FM,FK,TIME,PHI,LENGTH,TEMP,W
01290      COMMON/C20/IN,OUT,XX,DX
01300      COMMON/C30/YDUP(1)
01310      COMMON/VJ10/CZERO,C1,C2,RHO,HKF
01320      DIMENSION TEMP(NEQT),YPRIME(1)
01330      INTEGER OUT
01340      REAL LENGTH
01350      DATA PI/3.14159/,POS/0.1E-20/,SQRTPI/1.772453/
01360 C.....CALCULATE NU (ZU)
01370      DPR = FM * FK
01380      IF (ABS(DPR).GE.POS) GO TO 12
01390      WRITE(OUT,33)
01400 33 FORMAT(/' *** ERROR. DIVIDE CHECK CALC. ZN. ')
01410      STOP
01420 12 FR = SQRT( TIME/DPR )
01430 C      WL = CZERO * X * X + C1 * X + C2
01440 C      COMPUTE LEAKOFF RATE IN LB/FT**2-SEC
01450 C      CZERO IS THE LEAKOFF VELOCITY AT THE BASE OF FRACTURE.
01460 C      C1 IS LEAKOFF VELOCITY IN FT/DAY
01470      WL = CZERO + X/LENGTH * (C1-CZERO)
01480      WL = WL * RHO / 14400.
01490      DPL = H * (LENGTH*LENGTH) * (1.0-PHI)
01500 C      TO CONVERT TO LB/FT*FT-SEC
01510      WL = WL * RHO / 14000.
01520      IF (ABS(DPL).GE.POS) GO TO 18
01530      WRITE(OUT,44)
01540 44 FORMAT(/' *** ERROR. DIVIDE CHECK CALC. DPL. ')
01550      STOP
01560 18 FLN = W * X * C
01570      ZN = FLN / DPL * FR
01580 C.....CALCULATE PSI (PSI)
01590      EN = ZN * ZN
01600      PSI = EXP(-EN) - SQRTPI * ZN * ERFC(ZN)
01610 C.....CALCULATE DT/DX (DTDX)
01620      DPR3 = PI * TIME
01630      IF (ABS(DPR3).GE.POS) GO TO 28
01640      WRITE(OUT,55)
01650 55 FORMAT(/' *** ERROR. DIVIDE CHECK CALC. DPR3. ')
01660      STOP
01670 28 FR = SQRT(DPR / DPR3)
01680      TST = X / LENGTH
01690      ST = TST * TST
01700      ST = ST -1.0E-06
01710      D3 = W * (1.-ST) * C
01720      IF (ABS(D3).GE.POS) GO TO 24
01730      WRITE(OUT,66)
01740 66 FORMAT(/' *** ERROR. DIVIDE CHECK CALC. D3. ')
01750      STOP
01760 24 DTDX = (2.*H) / D3 * FR * ( TRES - TEMP(1) ) * PSI
01770      YPRIME(1) = DTDX
01780      YDUP(1) = DTDX
01790 C      WRITE(OUT,1111) D3,ST,W,TEMP(1),PSI,DTDX
01800 C1111 FORMAT(' INTERMED ..D3,ST,W,TEMP,PSI,DTDX...',6E15.7)
01810      RETURN
01820      END

```


C-6 Echo.

```
13730      SUBROUTINE ECHO
13740      DIMENSION FLTYPE(5)
13750      COMMON/C05/ JC(10,36),ECKVAL(36),JF(10)
13760      COMMON/C10/ WORK(2,9),H,RINJ,C,FH,FK,TIME,PHI,LENGTH,TEMP,W,T
13770      COMMON/C20/IN,OUT,X,DX
13780      COMMON/CLAIR/Q,MU,FH,G,CB,SP
13790      COMMON/C40/FLTYPE,DEPTH,PERM,PRESS,WIDTH
13800      COMMON/VJ10/CZERO,C1,C2,RHO,HKF
13810      COMMON/CAUG1/TINIT,TMAX,TINCR
13820      COMMON/TERCON/A,B,TLOW,THIGH,VLOW,VHIGH
13830      COMMON/INTTAB/NR,TEHINT(10),VISINT(10)
13840      DATA NW/10/
13850      INTEGER OUT,FLTYPE
13860      REAL LENGTH,MU
13870      WRITE(OUT,6)
13880      6 FORMAT('/',5X,'INPUT VALUES ARE: ',/)
13890      WRITE(OUT,7) (JF(L),L=1,NW)
13900      WRITE(OUT,7) FLTYPE
13910      NT = 35
13920      IF (NR.GT.0) NT = 31
13930      DO 10 K=1,NT
13940      WRITE(OUT,7) (JC(L,K),L=1,NW)
13950      7 FORMAT( 2X, 18A4 )
13960      WRITE(OUT,8) ECKVAL(K)
13970      8 FORMAT(2X,F15.5)
13980      10 CONTINUE
13990      IF (NR.LE.0) GO TO 16
14000      WRITE(OUT,7) (JC(L,32),L=1,NW)
14010      DO 15 K=1,NR
14020      15 WRITE(OUT,14) TEHINT(K),VISINT(K)
14030      14 FORMAT(2X,2F10.3)
14040      16 CONTINUE
14050      WRITE(OUT,12)
14060      12 FORMAT(//)
14070      RETURN
14080      END
```

C-7 Weltem

```

15760 SUBROUTINE WELTEM(T,XMU)
15770 DIMENSION LFTYPE(5)
15780 COMMON/C10/ WORK(2,9),H,RINJ,C,FM,FK,TIME,PHI,LENGTH,TEMP,WX,TR
15790 COMMON/UJ10/CZERO,C1,C2,RHO,HKF
15800 COMMON/CAUG1/TINIT,THAX,TINCR
15810 COMMON/ROCT11/DTI,DTE,DCI,DCE,GE,SFT,TES,TFS
15820 COMMON/INTTAB/NR,TEMINT(10),VISINT(10)
15830 COMMON/C40/LFTYPE,DEPTH,PERM,PRESS,WIDTH
15840 DELTZ = 500.
15850 GAMMA = RHO / 62.4
15860 WRITE(6,5)
15870 5 FORMAT(/,8X,'WELTEM CALCULATIONS',//,5X,'DEPTH',8X,'TEMPERATURE
15880 C NR > 0 IF VISCOSITY TABLE IS TO BE READ IN.
15890 IF (NR.LE.0) GO TO 10
15900 FMU = VISINT(1)
15910 GO TO 15
15920 10 CALL TERP(TFS,FMU)
15930 15 FVN = C * FMU / HKF
15940 HH=(10.29*HKF*(RINJ*GAMMA*1440.))**0.8/((DTI**1.8)*(FMU**0.8))
15950 ? FVN**0.4
15960 U =1./((1./HH+(DCE+DTE-DCI-DTI)/14400.+(DCI-DTE)/226.4)
15970 THOURS = THAX / 3600.
15980 F = 1.18333 + 0.2119*THOURS - 0.09693 * DCE
15990 W = RINJ * 350. * 1440.
16000 RTI = DTI/2.
16010 FKDAY = FK * 86400.
16020 A = (W*C*(FKDAY+RTI*U*F))/(2.*3.14159*RTI*FKDAY*U)
16060 Z = 0.0
16070 WRITE(6,25) Z,TES
16080 20 Z = Z + DELTZ
16090 IF (Z.GE.DEPTH) GO TO 13
16100 T = GE * Z + TES - GE * A + (TFS+GE*A-TES) * EXP(-Z/A)
16110 WRITE(6,25) Z,T
16120 25 FORMAT(5X,F6.0,6X,E14.6)
16130 GO TO 20
16140 13 IF ( (Z-DEPTH).LT.1.0) GO TO 31
16141 T = GE * DEPTH + TES - GE * A + (TFS+GE*A-TES) * EXP(-DEPTH/A)
16142 WRITE(6,25) DEPTH,T
16150 31 IF (NR.GT.0) GO TO 37
16160 CALL TERP(T,XMU)
16170 GO TO 39
16180 37 CALL INTERP(T,XMU)
16190 39 CONTINUE
16200 WRITE(6,26)
16210 26 FORMAT(//)
16220 RETURN
16230 END

```

C-8 Gertsm

```

11650      SUBROUTINE GERTSM(T,TINJ)
11660      COMMON/C10/WORK(2,9),H,RINJ,COLD,FH,FK,OLDTIM,PHI,LENGTH,TEMP
11670      ? OLIM,TRES
11680      COMMON/C20/IN,OUT,X,DX
11690      COMMON/C40/FLTYPE,DEPTH,PERM,PRESS,WIDTH
11700      COMMON/CLAIR/Q,MU,FH,G,C,SP
11710      COMMON/C60/KURVE
11720      COMMON/CHG/MODELS
11730      COMMON/FRACV/WS(10),XLS(10),VOLS(10),EPPS(10),TIM(10)
11740      INTEGER YES/'Y'/
11750      REAL LENGTH,MU
11760      REAL MT,MU
11770      C      NO EXIT FROM HERE UNTIL SUCCESSIVE XL'S CONVERGE.
11780      TIME = T/60.
11790      PI = 3.14159
11800      SAUXL = -1.
11810      TROOT = SQRT(TIME)
11820      WRITE(6,1) KURVE
11830      1 FORMAT(/,5X,'***** CURVE # ',I1,' *****',/)
11840      2 ZAP1 = FH * (C*C)
11850      ZAP2 = .00561 * Q / ZAP1
11860      ZAP3 = ZAP2 * ZAP2 * ZAP2
11870      WRITE(6,3) MU
11880      3 FORMAT(/,5X,'IN GERTSM: MU =',F8.2,/)
11890      DKHL = 8.724 * ZAP3 * (145. * MU / (G*60. * TIME))
11900      DKU = 1
11910      DKUP = DKU
11920      KIT = 0
11930      50 CONTINUE
11940      KIT = KIT + 1
11950      IF (KIT.LT.500) GO TO 52
11960      WRITE(3,53) KURVE
11970      53 FORMAT(/,5X,'Z-',I1),
11980      GO TO 100
11990      52 CONTINUE
12000      IF (SP.NE.0.0) GO TO 34
12010      A = PI
12020      GO TO 5
12030      34 DKS = C * TROOT / SP
12040      A = PI + 8. * DKU / DKS
12050      5 CONTINUE
12060      ARG = 9. * SQRT(PI) * DKU / A
12070      IF (ARG.LT.2.) GO TO 10
12080      NUM = 1
12090      EAE = 1
12100      DO 40 N=1,8
12110      NUM = (2 * N - 1) * NUM
12120      EAE = EAE + (-1) ** N * NUM/(2*(ARG*ARG))**N
12130      40 CONTINUE
12140      EAE = EAE/ARG/SQRT(PI)
12150      DKL = (1.-(A/16./DKU)*(1.-EAE))/2./PI

```

```

12160      GO TO 11
12170 10 CONTINUE
12180      ARG2 = ARG * ARG
12190      CX = ERFC(ARG)
12200      DKL = (1.-(A/16./DKU) * (1.-EXP(ARG2)*CX))/2./PI
12210 11 CONTINUE
12220      DKU = (1./DKML/DKL/DKL) **.25
12230      DDKU = DKU - DKUP
12240      DDKU = ABS(DDKU)
12250      IF(DDKU.LT..0001) GO TO 100
12260      DKUP = DKU
12270      GO TO 50
12280 100 CONTINUE
12290      W = 12.*CX*TR00T/DKU
12300      XL = 3.61 * Q * TR00T * DKL/C/FH
12310      VOL = 2. * W * XL * FH / 12.
12320      EFF = VOL * 100. / (Q*5.61*TIME)
12330      WRITE(6,21)
12340 21 FORMAT(//,5X,'WIDTH',4X,'LENGTH',4X,'VOLUME',5X,'EFF.',
12350      ? 7X,'TIME' ,/)
12360      WRITE(6,22) W,XL,VOL,EFF,TIME
12370 22 FORMAT( F10.3,2F10.1,F9.1,F11.1)
12380      VERG = ABS(XL-SAUXL) / XL
12390      SAUXL = XL
12400      IF (VERG.LT.0.01) GO TO 300
12410      IF (MODELS.NE.YES) GO TO 24
12420      ALTRES = TRES
12430 C      TINJ, RETURNED FROM WELTEX, IS INPUT TO SINCLA.
12440      CALL SINCLA(RINJ,ALTRES,TINJ,W,H,TIME,FK,XL,MU,FH,EFF)
12450      GO TO 2
12460 24 CONTINUE
12470      CALL DYSART(T,XL,W,EFF,MU)
12480      GO TO 2
12490 300 CONTINUE
12500      WRITE(6,38)
12510 38 FORMAT(//,5X,'CONVERGENCE REACHED IN SUBROUTINE GERTSM. ',//)
12520 C      NOW INCREMENT POINTER TO NEXT CURVE. (SECOND DIMENSION OF
12530 C      ARRAYS XP,YF,VISCOS IN SUBROUTINE FRAC.
12540      KURVE = KURVE + 1
12550      KU = KURVE
12560      WS(KU) = W
12570      XLS(KU) = XL
12580      VOLS(KU) = VOL
12590      EFFS(KU) = EFF
12600      TIM(KU) = TIME
12610      RETURN
12620      END

```

C-9 Plotv

```

10760 SUBROUTINE PLOTV( NUMPTS)
10770 DIMENSION XA(202),UA(202),LUMA(9),NUMPTS(9),DIF(9),DIFV(9)
10780 COMMON/PLAY/ XD(202,9),YD(202,9),VISCOS(202,9)
10790 COMMON/C20/IN,OUT,XXDUM,IXDUM
10800 INTEGER OUT
10810 COMMON/C50/ SAVAUT(10), NIUMMY(9),NPASS
10820 DATA MAX/200/,LUMA/0,1,2,5,6,10,14,36,37/,PLOTHT/6.5/,PLOTWD/9.5
10830 SMALL = .1E+10
10840 BIG = .1E-10
10850 SMLV = .1E+10
10860 BIGV = .1E-10
10870 DO 5 K=1,NPASS
10880 NCR = NUMPTS(K)
10890 NCRLIM = NCR
10900 IF (NCRLIM.GT.200) NCRLIM = 200
10910 DO 8 L=1,NCRLIM
10920 XPL = XD(L,K)
10930 YPL = VISCOS(L,K)
10940 IF (XPL.LT.SMALL) SMALL = XPL
10950 IF (XPL.GT.BIG) BIG = XPL
10960 IF (YPL.LT.SMLV) SMLV = YPL
10970 IF (YPL.GT.BIGV) BIGV = YPL
10980 8 CONTINUE
10990 DIF(K) = BIG - SMALL
11000 5 DIFV(K) = BIGV - SMLV
11010 BIGDIF = .1E-10
11020 IUSE = 0
11030 DO 9 K=1,NPASS
11040 IF (DIF(K).LE.BIGDIF) GO TO 9
11050 BIGDIF = DIF(K)
11060 IUSE = K
11070 9 CONTINUE
11080 BIGDV = .1E-10
11090 IUSEV = 0
11100 DO 14 K=1,NPASS
11110 IF (DIFV(K).LE.BIGDV) GO TO 14
11120 BIGDV = DIFV(K)
11130 IUSEV = K
11140 14 CONTINUE
11150 C IUSE & IUSEV INDICATE WHICH CURVES HAVE LARGEST MAX & MIN VALUES
11160 IF (IUSE.GT.0.AND.IUSEV.GT.0) GO TO 11
11170 WRITE(OUT,12)
11180 12 FORMAT(//,2X,'*** ERROR IN SUB.PLOTV. VALUE OF IUSE IS ',
11190 7 'INVALID. JOB TERMINATED. ')
11200 STOP
11210 11 CONTINUE
11220 LPTS = NUMPTS(IUSE)
11230 IF (LPTS.GT.200) LPTS = 200
11240 DO 10 K=1,LPTS
11250 UA(K) = VISCOS(K,IUSE)
11260 10 XA(K) = XD(K,IUSE)

```

```

11270      CALL PLOT(0.,0.,-3)
11280      CALL PLOT(11.5,0.,2)
11290      CALL PLOT(11.5,8.5,2)
11300      CALL PLOT(0.,8.5,2)
11310      CALL PLOT(0.,0.,2)
11320      CALL PLOT(1.,1.,-3)
11330      CALL SCALE(XA,PLOTWD,LPTS,1)
11340      CALL SCALE(VA,PLOTHT,LPTS,1)
11350      BEGINX = XA(LPTS+1)
11360      DELTX  = XA(LPTS+2)
11370      BEGINV = VA(LPTS+1)
11380      BEGINV = VA(LPTS+1) - ( DIFV(IUSEV)*.1)
11390      DELTV  = VA(LPTS+2)
11400      CALL AXIS(0.,0.,' ',-1,PLOTWD,0.,BEGINX,DELT X )
11410      CALL SYMBOL(3.,-0.75,0.175,'FRACTURE DISTANCE - FEET',
11420      ? 0.,24)
11430      CALL AXIS(0.,0.,' ',1,PLOTHT,90.,BEGINV,DELT V )
11440      CALL SYMBOL(-.5,2.5,0.175,'VISCOSITY - CP.',90.,17)
11450      CALL SYMBOL(1.3,7.0,0.175,
11460      ? 'FLUID VISCOSITY VS DISTANCE IN FRACTURE',0.0,39)
11470      20 CONTINUE
11480      DO 30 KEG=1,NPASS
11490      KPTS = NUMPTS(KEG)
11500      IF (KPTS.GT.200) KPTS = 200
11510      DO 25 K=1,KPTS
11520      XA(K) = XD(K,KEG)
11530      25 VA(K) = VISCOS(K,KEG)
11540      XA(KPTS+1) = BEGINX
11550      XA(KPTS+2) = DELTX
11560      VA(KPTS+1) = BEGINV
11570      VA(KPTS+2) = DELTV
11580      30 CALL LINE(XA,VA,KPTS,1,10,LUMA(KEG))
11590      RETURN
11600      77 WRITE(6,84) KNT,MAX
11610      84 FORMAT(/,' **** ERROR. KNT=',I3,' IS GREATER THAN MAX. (',I3,')
11620      ? ' PROGRAM TERMINATED.')
11630      STOP
11640      END

```

```
10710 SUBROUTINE TERP(TEMP,VIS)
10720 COMMON/TERCON/A,B,TLOW,THIGH,VLOW,VHIGH
10730 VIS = B * 10.** (A*TEMP)
10740 RETURN
10750 END
```

C-11 Dverk

```

SUBROUTINE DVERK (N,FCN,X,Y,XEND,TOL,IND,C,NW,W,IER)

C
INTEGER          N,IND,NW,K
INTEGER          IER
C1  DOUBLE PRECISION X,Y(N),XEND,TOL,C(1),W(NW,9),TEMP
REAL             X,Y(N),XEND,TOL,C(1),W(NW,9),TEMP
C1  DOUBLE PRECISION ZERO,ONE,TWO,THREE,FOUR,FIVE,SEVEN,TEN,HALF,P
REAL             ZERO,ONE,TWO,THREE,FOUR,FIVE,SEVEN,TEN,HALF,P
1  C4D15,C2D3,C5D6,C1D6,C1D15,C2D96
C1  DOUBLE PRECISION RK(39),REFS,RTOL
REAL             RK(39),REFS,RTOL
C1  DATA          ZERO/0.0D0/,ONE/1.0D0/,TWO/2.0D0/,THREE/3.0D0/
DATA          ZERO/0.0/,ONE/1.0/,TWO/2.0/,THREE/3.0/
C1  DATA          FOUR/4.0D0/,FIVE/5.0D0/,SEVEN/7.0D0/
DATA          FOUR/4.0/,FIVE/5.0/,SEVEN/7.0/
C1  DATA          TEN/10.0D0/,HALF/0.5D0/,P9/0.9D0/
DATA          TEN/10.0/,HALF/0.5/,P9/0.9/
C1  DATA          C4D15/.2666666666666667D0/
DATA          C4D15/.26666667/
C1  DATA          C2D3/.6666666666666667D0/
DATA          C2D3/.66666667/
C1  DATA          C5D6/.8333333333333333D0/
DATA          C5D6/.83333333/
C1  DATA          C1D6/.1666666666666667D0/
DATA          C1D6/.16666667/
C1  DATA          C1D15/.6666666666666667D-1/
DATA          C1D15/.66666667E-1/
C1  DATA          C2D96/120.4272910821709D0/
DATA          C2D96/120.4273/
C1  DATA          REFS/Z34100000000000000/
DATA          REFS/Z3C100000/
C1  DATA          RTOL/Z0D1000000000000000/
DATA          RTOL/Z0S100000/
C1  DATA          RK( 1)/.1666666666666667D+00/
DATA          RK( 1)/.16666667E+00/
C1  DATA          RK( 2)/.5333333333333333D-01/
DATA          RK( 2)/.53333333E-01/
C1  DATA          RK( 3)/.2133333333333333D+00/
DATA          RK( 3)/.21333333E+00/
C1  DATA          RK( 4)/.8333333333333333D+00/
DATA          RK( 4)/.83333333E+00/
C1  DATA          RK( 5)/.2666666666666667D+01/
DATA          RK( 5)/.26666667E+01/
C1  DATA          RK( 6)/.2500000000000000D+01/
DATA          RK( 6)/.25000000E+01/
C1  DATA          RK( 7)/.2578125000000000D+01/
DATA          RK( 7)/.2578125E+01/
C1  DATA          RK( 8)/.9166666666666667D+01/
DATA          RK( 8)/.91666667E+01/
C1  DATA          RK( 9)/.6640625000000000D+01/
DATA          RK( 9)/.6640625E+01/
C1  DATA          RK(10)/.9854166666666667D+00/
DATA          RK(10)/.9854167E+00/
C1  DATA          RK(11)/.2400000000000000D+01/
DATA          RK(11)/.24000000E+01/
C1  DATA          RK(12)/.8000000000000000D+01/

```


	DATA	RK(12)/.8000000E+01/
C1	DATA	RK(13)/.6560457516337869D+01/
	DATA	RK(13)/.6560458E+01/
C1	DATA	RK(14)/.3055555555555556D+00/
	DATA	RK(14)/.3055556E+00/
C1	DATA	RK(15)/.3450980392156863D+00/
	DATA	RK(15)/.3450980E+00/
C1	DATA	RK(16)/.5508666666666667D+00/
	DATA	RK(16)/.5508667E+00/
C1	DATA	RK(17)/.1653333333333333D+01/
	DATA	RK(17)/.1653333E+01/
C1	DATA	RK(18)/.9455882352941176D+00/
	DATA	RK(18)/.9455882E+00/
C1	DATA	RK(19)/.3240000000000000D+00/
	DATA	RK(19)/.3240000E+00/
C1	DATA	RK(20)/.2337882352941176D+00/
	DATA	RK(20)/.2337882E+00/
C1	DATA	RK(21)/.2035465116279070D+01/
	DATA	RK(21)/.2035465E+01/
C1	DATA	RK(22)/.6976744186046512D+01/
	DATA	RK(22)/.6976744E+01/
C1	DATA	RK(23)/.5648179814561484D+01/
	DATA	RK(23)/.5648180E+01/
C1	DATA	RK(24)/.1373215676141258D+00/
	DATA	RK(24)/.1373216E+00/
C1	DATA	RK(25)/.2863022661036103D+00/
	DATA	RK(25)/.2863023E+00/
C1	DATA	RK(26)/.1441785567164738D+00/
	DATA	RK(26)/.1441786E+00/
C1	DATA	RK(27)/.7500000000000000D-01/
	DATA	RK(27)/.7500000E-01/
C1	DATA	RK(28)/.3899286987522282D+00/
	DATA	RK(28)/.3899287E+00/
C1	DATA	RK(29)/.3194444444444444D+00/
	DATA	RK(29)/.3194444E+00/
C1	DATA	RK(30)/.1350383631713555D+00/
	DATA	RK(30)/.1350384E+00/
C1	DATA	RK(31)/.1078329882677709D-01/
	DATA	RK(31)/.1078330E-01/
C1	DATA	RK(32)/.6980519480519481D-01/
	DATA	RK(32)/.6980519E-01/
C1	DATA	RK(33)/.6250000000000000D-02/
	DATA	RK(33)/.6250000E-02/
C1	DATA	RK(34)/.6963012477712360D-02/
	DATA	RK(34)/.6963012E-02/
C1	DATA	RK(35)/.6944444444444444D-02/
	DATA	RK(35)/.6944444E-02/
C1	DATA	RK(36)/.6138107416879795D-02/
	DATA	RK(36)/.6138107E-02/
C1	DATA	RK(37)/.6818181818181818D-01/
	DATA	RK(37)/.6818182E-01/
C1	DATA	RK(38)/.1078329882677709D-01/
	DATA	RK(38)/.1078330E-01/
C1	DATA	RK(39)/.6980519480519481D-01/
	DATA	RK(39)/.6980519E-01/

C

```

      BEGIN INITIALIZATION, PARAMETER
      CHECKING, INTERRUPT RE-ENTRIES
      IER = 0
      IF (IND.LT.1.OR.IND.GT.6) GO TO 290
      CASES - INITIAL ENTRY, NORMAL
      RE-ENTRY, INTERRUPT RE-ENTRIES
      GO TO (5,5,40,145,265,265), IND
      CASE 1 - INITIAL ENTRY (IND .EQ. 1
      OR 2) ABORT IF N.GT.NW OR TOL.LE.
5 IF (N.GT.NW.OR.TOL.LE.ZERO) GO TO 295
  IF (IND.EQ.2) GO TO 15
      INITIAL ENTRY WITHOUT OPTIONS (IND
      .EQ. 1) SET C(1) TO C(9) EQUAL TO
      0
      DO 10 K=1,9
      C(K) = ZERO
10 CONTINUE
      GO TO 30

```

BEGIN INITIALIZATION, PARAMETER
CHECKING, INTERRUPT RE-ENTRIES

ABORT IF IND OUT OF RANGE 1 TO 6
TO 290

CASES - INITIAL ENTRY, NORMAL
RE-ENTRY, INTERRUPT RE-ENTRIES

CASE 1 - INITIAL ENTRY (IND .EQ. 1
OR 2) ABORT IF N.GT.NW OR TOL.LE.

```
INITIAL ENTRY WITHOUT OPTIONS (IND
.ER. 1) SET C(1) TO C(9) EQUAL TO
0
```

SUMMARY OF THE COMPONENTS OF THE
COMMUNICATIONS VECTOR
PRESCRIBED AT THE OPTION
OF THE USER

C(1) ERROR CONTROL INDICATOR
C(2) FLOOR VALUE
C(3) HMIN SPECIFICATION
C(4) HSTART SPECIFICATION
C(5) SCALE SPECIFICATION
C(6) HMAX SPECIFICATION
C(7) MAX NO OF FCN EVALS
C(8) INTERRUPT NO 1
C(9) INTERRUPT NO 2

DETERMINED BY THE PROGRAM

INITIAL ENTRY WITH OPTIONS (IND .EQ.

C		2) MAKE C(1) TO C(9) NON-NEGATIVE
	DO 20 K=1,9	
C1	C(K) = DABS(C(K))	
	C(K) = ABS(C(K))	
	20 CONTINUE	
C		MAKE FLOOR VALUES NON-NEGATIVE IF
C		THEY ARE TO BE USED
C		
	IF (C(1).NE.FOUR.AND.C(1).NE.FIVE) GO TO 30	
	DO 25 K=1,N	
C1	C(K+30) = DABS(C(K+30))	
	C(K+30) = ABS(C(K+30))	
	25 CONTINUE	
	30 CONTINUE	
C		INITIALIZE RRES, DWARF, PREV XEND,
C		FLAG, COUNTS
	C(10) = REFS	
	C(11) = RTOL	
C		SET PREVIOUS XEND INITIALLY TO
C		INITIAL VALUE OF X
	C(20) = X	
	DO 35 K=21,24	
	C(K) = ZERO	
	35 CONTINUE	
	GO TO 45	
C		CASE 2 - NORMAL RE-ENTRY (IND .EQ.
C		3) ABORT IF XEND REACHED, AND
C		EITHER X CHANGED OR XEND NOT
	40 IF (C(21).NE.ZERO.AND.(X.NE.C(20).OR.XEND.EQ.C(20))) GO TO 285	
C		RE-INITIALIZE FLAG
C	C(21) = ZERO	
C		CASE 3 - RE-ENTRY FOLLOWING AN
C		INTERRUPT (IND .EQ. 4 TO 6)
C		TRANSFER CONTROL TO THE
C		APPROPRIATE RE-ENTRY POINT. THIS
C		HAS ALREADY BEEN HANDLED BY THE
C		COMPUTED GO TO.
	45 CONTINUE	
C		LOOP THROUGH THE FOLLOWING 4 STAGES
C		ONCE FOR EACH TRIAL STEP UNTIL TH
C		OCCURRENCE OF ONE OF THE FOLLOWIN
C		(A) THE NORMAL RETURN (WITH IND
C		.EQ. 3) ON REACHING XEND IN STAGE
C		4 (B) AN ERROR RETURN (WITH IND
C		.LT. 0) IN STAGE 1 OR STAGE 4 (C)
C		AN INTERRUPT RETURN (WITH IND .EQ
C		4, 5 OR 6), IF REQUESTED, IN STAG
C		1 OR STAGE 4
	50 CONTINUE	
C		STAGE 1 - PREPARE - DO CALCULATIONS
C		OF HMIN, HMAX, ETC., AND SOME
C		PARAMETER CHECKING, AND END UP
C		WITH SUITABLE VALUES OF HMAX,
C		XTRIAL AND HTRIAL IN PREPARATION
C		FOR TAKING AN INTEGRATION STEP.

```

C                                     ERROR RETURN (WITH IND=-1) IF NO
C                                     OF FCN EVALS TOO GREAT
C
C   IF (C(7).EQ.ZERO.OR.C(24).LT.C(7)) GO TO 55
C   IND = -1
C   GO TO 9005
55 CONTINUE

C                                     CALCULATE SLOPE (ADDING 1 TO NO OF
C                                     FCN EVALS) IF IND .NE. 6
C
C   IF (IND.EQ.6) GO TO 60
C   CALL FCN (N,X,Y,W(1,1))
C   C(24) = C(24)+ONE
60 CONTINUE

C                                     CALCULATE HMIN - USE DEFAULT UNLESS
C                                     VALUE PRESCRIBED
C
C   C(13) = C(3)
C   IF (C(3).NE.ZERO) GO TO 120
C
C                                     CALCULATE DEFAULT VALUE OF HMIN
C                                     FIRST CALCULATE WEIGHTED NORM Y -
C                                     C(12) - AS SPECIFIED BY THE ERROR
C                                     CONTROL INDICATOR C(1)
C
C   TEMP = ZERO
C   IF (C(1).NE.ONE) GO TO 70
C
C                                     ABSOLUTE ERROR CONTROL - WEIGHTS ARE
C                                     1
C
C1  DO 65 K=1,N
C    TEMP = DMAX1(TEMP,DABS(Y(K)))
C    TEMP = AMAX1(TEMP,ABS(Y(K)))
65 CONTINUE
C   C(12) = TEMP
C   GO TO 115
70 IF (C(1).NE.TWO) GO TO 75
C
C                                     RELATIVE ERROR CONTROL - WEIGHTS ARE
C                                     1/DABS(Y(K)) SO WEIGHTED NORM Y IS
C                                     1
C
C   C(12) = ONE
C   GO TO 115
75 IF (C(1).NE.THREE) GO TO 85
C
C                                     WEIGHTS ARE 1/MAX(C(2),ABS(Y(K)))
C
C1  DO 80 K=1,N
C    TEMP = DMAX1(TEMP,DABS(Y(K))/C(2))
C    TEMP = AMAX1(TEMP,ABS(Y(K))/C(2))
80 CONTINUE
C1  C(12) = DMIN1(TEMP,ONE)
C   C(12) = AMIN1(TEMP,ONE)
C   GO TO 115
85 IF (C(1).NE.FOUR) GO TO 95
C
C                                     WEIGHTS ARE 1/MAX(C(K+30),ABS(Y(K)))
C
C1  DO 90 K=1,N
C    TEMP = DMAX1(TEMP,DABS(Y(K))/C(K+30))
C    TEMP = AMAX1(TEMP,ABS(Y(K))/C(K+30))
90 CONTINUE
C1  C(12) = DMIN1(TEMP,ONE)
C   C(12) = AMIN1(TEMP,ONE)
C   GO TO 115
95 IF (C(1).NE.FIVE) GO TO 105

```

```

C                                WEIGHTS ARE 1/C(K+30)
      DO 100 K=1,N
C1      TEMP = DMAX1(TEMP,DABS(Y(K))/C(K+30))
      TEMP = AMAX1(TEMP,ABS(Y(K))/C(K+30))
100    CONTINUE
      C(12) = TEMP
      GO TO 115
105    CONTINUE
C                                DEFAULT CASE - WEIGHTS ARE
C                                1/MAX(1,ABS(Y(K)))
      DO 110 K=1,N
C1      TEMP = DMAX1(TEMP,DABS(Y(K)))
      TEMP = AMAX1(TEMP,ABS(Y(K)))
110    CONTINUE
C1      C(12) = DMIN1(TEMP,ONE)
      C(12) = AMIN1(TEMP,ONE)
115    CONTINUE
C1      C(13) = TEN*DMAX1(C(11),C(10)*DMAX1(C(12)/TOL,DABS(X)))
      C(13) = TEN*AMAX1(C(11),C(10)*AMAX1(C(12)/TOL,ABS(X)))
120    CONTINUE
C                                CALCULATE SCALE - USE DEFAULT UNLESS
C                                VALUE PRESCRIBED
      C(15) = C(5)
      IF (C(5).EQ.ZERO) C(15) = ONE
C                                CALCULATE HMAX - CONSIDER 4 CASES
C                                CASE 1 - BOTH HMAX AND SCALE
C                                PRESCRIBED
C1      IF (C(6).NE.ZERO.AND.C(5).NE.ZERO) C(16) = DMIN1(C(6),TWO/C(5))
      IF (C(6).NE.ZERO.AND.C(5).NE.ZERO) C(16) = AMIN1(C(6),TWO/C(5))
C                                CASE 2 - HMAX PRESCRIBED, BUT SCALE
C                                NOT
C1      IF (C(6).NE.ZERO.AND.C(5).EQ.ZERO) C(16) = C(6)
C                                CASE 3 - HMAX NOT PRESCRIBED, BUT
C                                SCALE IS
C1      IF (C(6).EQ.ZERO.AND.C(5).NE.ZERO) C(16) = TWO/C(5)
C                                CASE 4 - NEITHER HMAX NOR SCALE IS
C                                PROVIDED
C1      IF (C(6).EQ.ZERO.AND.C(5).EQ.ZERO) C(16) = TWO
C                                ERROR RETURN (WITH IND=-2) IF HMIN
C                                .GT. HMAX
      IF (C(13).LE.C(16)) GO TO 125
      IND = -2
      GO TO 9005
125    CONTINUE
C                                CALCULATE PRELIMINARY HMAX -
C                                CONSIDER 3 CASES
C1      IF (IND.GT.2) GO TO 130
C                                CASE 1 - INITIAL ENTRY - USE

```

```

C                                     PRESCRIBED VALUE OF HSTART, IF
C                                     ANY, ELSE DEFAULT
C
C(14) = C(4)
IF (C(4).EQ.ZERO) C(14) = C(16)*TOL**C1D6
GO TO 140
130 IF (C(23).GT.ONE) GO TO 135
C                                     CASE 2 - AFTER A SUCCESSFUL STEP,
C                                     AT MOST ONE FAILURE, USE MIN(2,
C                                     .9*(TOL/EST)**(1/6))*HMAG, BUT
C                                     AVOID POSSIBLE OVERFLOW. THEN
C                                     AVOID REDUCTION BY MORE THAN HALF
C
C      TEMP = TWO*C(14)
C      IF (TOL.LT.C2D96*C(19)) TEMP = F9*(TOL/C(19))*C1D6*C(14)
C1     C(14) = DMAX1(TEMP,HALF*C(14))
C      C(14) = AMAX1(TEMP,HALF*C(14))
C      GO TO 140
135 CONTINUE
C                                     CASE 3 - AFTER TWO OR MORE
C                                     SUCCESSIVE FAILURES
C
C      C(14) = HALF*C(14)
140 CONTINUE
C
C      C(14) = DMIN1(C(14),C(16))
C      C(14) = AMIN1(C(14),C(16))
C                                     CHECK AGAINST HMAX
C
C      C(14) = DMAX1(C(14),C(13))
C      C(14) = AMAX1(C(14),C(13))
C                                     CHECK AGAINST HMIN
C
C      IF (C(8).EQ.ZERO) GO TO 145
C      IND = 4
C      GO TO 9005
C                                     INTERRUPT NO 1 (WITH IND=4) IF
C                                     REQUESTED
C
C      RESUME HERE ON RE-ENTRY WITH IND
C      .EQ. 4
145 CONTINUE
C                                     CALCULATE HMAG, XTRIAL - DEPENDING
C                                     ON PRELIMINARY HMAG, XEND
C
C1     IF (C(14).GE.DABS(XEND-X)) GO TO 150
C      IF (C(14).GE.ABS(XEND-X)) GO TO 150
C
C      DO NOT STEP MORE THAN HALF WAY TO
C      XEND
C
C1     C(14) = DMIN1(C(14),HALF*DABS(XEND-X))
C      C(14) = AMIN1(C(14),HALF*ABS(XEND-X))
C1     C(17) = X+DSIGN(C(14),XEND-X)
C      C(17) = X+SIGN(C(14),XEND-X)
C      GO TO 155
150 CONTINUE
C                                     HIT XEND EXACTLY
C1     C(14) = DABS(XEND-X)
C      C(14) = ABS(XEND-X)
C      C(17) = XEND
155 CONTINUE
C                                     CALCULATE HTRIAL

```



```

C(18) = C(17)-X

C                                     STAGE 2 - CALCULATE YTRIAL (ADDING
C                                     TO NO OF FCN EVALS). W(X,2), ...
C                                     W(X,8) HOLD INTERMEDIATE RESULTS
C                                     NEEDED IN STAGE 3. W(X,9) IS
C                                     TEMPORARY STORAGE UNTIL FINALLY
C                                     HOLDS YTRIAL.

DO 160 K=1,N
  W(K,9) = Y(K)+C(18)*W(K,1)*RK(1)
160 CONTINUE
  CALL FCN (N,X+C(18)*C1D6,W(1,9),W(1,2))

C
DO 165 K=1,N
  W(K,9) = Y(K)+C(18)*(W(K,1)*RK(2)+W(K,2)*RK(3))
165 CONTINUE
  CALL FCN (N,X+C(18)*C4D15,W(1,9),W(1,3))

C
DO 170 K=1,N
  W(K,9) = Y(K)+C(18)*(W(K,1)*RK(4)-W(K,2)*RK(5)+W(K,3)*RK(6))
170 CONTINUE
  CALL FCN (N,X+C(18)*C2D3,W(1,9),W(1,4))

C
DO 175 K=1,N
  W(K,9) = Y(K)+C(18)*(-W(K,1)*RK(7)+W(K,2)*RK(8)-W(K,3)*RK(9)
1  +W(K,4)*RK(10))
175 CONTINUE
  CALL FCN (N,X+C(18)*C5D6,W(1,9),W(1,5))

C
DO 180 K=1,N
  W(K,9) = Y(K)+C(18)*(W(K,1)*RK(11)-W(K,2)*RK(12)+W(K,3)*RK(13)
1  -W(K,4)*RK(14)+W(K,5)*RK(15))
180 CONTINUE
  CALL FCN (N,X+C(18),W(1,9),W(1,6))

C
DO 185 K=1,N
  W(K,9) = Y(K)+C(18)*(-W(K,1)*RK(16)+W(K,2)*RK(17)-W(K,3)
1  *RK(18)-W(K,4)*RK(19)+W(K,5)*RK(20))
185 CONTINUE
  CALL FCN (N,X+C(18)*C1D15,W(1,9),W(1,7))

C
DO 190 K=1,N
  W(K,9) = Y(K)+C(18)*(W(K,1)*RK(21)-W(K,2)*RK(22)+W(K,3)*RK(23)
1  -W(K,4)*RK(24)+W(K,5)*RK(25)+W(K,7)*RK(26))
190 CONTINUE
  CALL FCN (N,X+C(18),W(1,9),W(1,8))

C                                     CALCULATE YTRIAL, THE EXTRAPOLATED
C                                     APPROXIMATION AND STORE IN W(X,9)

DO 195 K=1,N
  W(K,9) = Y(K)+C(18)*(W(K,1)*RK(27)+W(K,3)*RK(28)+W(K,4)*RK(29)
1  +W(K,5)*RK(30)+W(K,7)*RK(31)+W(K,8)*RK(32))
195 CONTINUE

C                                     ADD 7 TO THE NO OF FCN EVALS

C(24) = C(24)+SEVEN

C                                     STAGE 3 - CALCULATE THE ERROR
C                                     ESTIMATE EST. FIRST CALCULATE THE
C                                     UNWEIGHTED ABSOLUTE ERROR ESTIMATE

```


<pre> GO TO 260 250 CONTINUE C C DO 255 K=1,N C1 TEMP = DMAX1(TEMP,DABS(W(K,2))/DMAX1(ONE,DABS(Y(K)))) TEMP = AMAX1(TEMP,ABS(W(K,2))/AMAX1(ONE,ABS(Y(K)))) 255 CONTINUE 260 CONTINUE C C C(19) = TEMP*C(14)*C(15) C C IND = 5 IF (C(19).GT.TOL) IND = 6 C IF (C(9).NE.ZERO) GO TO 9005 C 265 CONTINUE C IF (IND.EQ.6) GO TO 275 C C X = C(17) DO 270 K=1,N Y(K) = W(K,9) 270 CONTINUE C(22) = C(22)+ONE C(23) = ZERO C C IF (X.NE.XEND) GO TO 50 IND = 3 C(20) = XEND C(21) = ONE GO TO 9005 275 CONTINUE C C C(23) = C(23)+ONE C C IF (C(14).GT.C(13)) GO TO 50 IND = -3 GO TO 9005 C C </pre>	<pre> DEFAULT CASE - WEIGHTS ARE 1/MAX(1,ABS(Y(K))) CALCULATE EST - (THE WEIGHTED MAX NORM OF W(=,2))*HMAX*SCALE - EST IS INTENDED TO BE A MEASURE OF THE ERROR PER UNIT STEP IN YTRIAL STAGE 4 - MAKE DECISIONS. SET IND=5 IF STEP ACCEPTABLE, ELSE SET IND= INTERRUPT NO 2 IF REQUESTED RESUME HERE ON RE-ENTRY WITH IND .EQ. 5 OR 6 STEP ACCEPTED (IND .EQ. 5), SO UPDATE X, Y FROM XTRIAL, YTRIAL, ADD 1 TO THE NO OF SUCCESSFUL STEPS, AND SET THE NO OF SUCCESSIVE FAILURES TO ZERO RETURN(WITH IND=3, XEND SAVED, FLAG SET) IF X .EQ. XEND STEP NOT ACCEPTED (IND .EQ. 6), SO ADD 1 TO THE NO OF SUCCESSIVE FAILURES ERROR RETURN (WITH IND=-3) IF HMAX .LE. PMIN END STAGE 4 END LOOP </pre>
---	---

C	280	CONTINUE	
C			RELATIVE ERROR OPTION SELECTED AND
			Y(K) IS ZERO
		IER = 132	
		GO TO 9000	
	285	CONTINUE	
C			X OR XEND WAS NOT CHANGED FROM
C			PREVIOUS CALL
		IER = 131	
		GO TO 9000	
C			
	290	CONTINUE	
C			IND OUT OF RANGE
		IER = 130	
		GO TO 9000	
	295	CONTINUE	
C			N.GT.NW OR TOL.LE.O
		IER = 129	
C			BEGIN ABORT ACTION
	9000	CONTINUE	
		CALL UERTST (IER,6HDVERK)	
	9005	CONTINUE	
		RETURN	
		END	

```

02810      SUBROUTINE BXDATA
02820      DIMENSION FLTYPE(5),LUMA(9)
02830      INTEGER FLTYPE
02840      COMMON/C10/WORK(2,9),H,RINJ,C,FH,FK,TIME,PHI,XLEN,TEMP,W,TRES
02850      COMMON/C40/FLTYPE,DEPTH,PERM,PRESS,WIDTH
02860      COMMON/C50/SAVAUT(10),NDUMMY(9),NFASS
02870      COMMON/CAUG1/TINIT,TMAX,TINCR
02880      COMMON/BXOCT/ BXLEN
02890      DATA LUMA/0,1,2,5,6,10,14,36,37/
02900      TIMEH = TIME / 60.
02910      THOURN = TINCR / 3600.
02920      TIMLEG = TINIT / 3600.
02930      THOURM = TMAX / 3600.
02940      CALL PLOT(10.5,-1.,-3)
02950      CALL PLOT(2.5,0.0,2)
02960      CALL PLOT(2.5,8.5,2)
02970      CALL PLOT(0.0,8.5,2)
02980      CALL SYMBOL(0.2,8.0,0.15,'FRACTURE DATA',0.,13)
02990      CALL SYMBOL(0.2,7.6,0.1,'FLUID TYPE:',0.,11)
03000      CALL SYMBOL(1.35,7.6,.1,FLTYPE,0.,20)
03010      CALL SYMBOL(.2,7.2,.1,'RATE:',0.,5)
03020      CALL NUMBER(1.25,7.2,.1,RINJ,0.,2)
03030      CALL SYMBOL(2.,7.2,.1,'BPM',0.,3)
03040      CALL SYMBOL(.2,6.8,.1,'MAX TIME:',0.,9)
03050      CALL NUMBER(1.25,6.8,.1,THOURM,0.,2)
03060      CALL SYMBOL(2.,6.8,.1,'HR.',0.,4)
03070      CALL SYMBOL(.2,6.4,.1,'FRAC. WIDTH:',0.,12)
03080      CALL NUMBER(1.5,6.4,.1,WIDTH,0.,2)
03090      CALL SYMBOL(2.,6.4,.1,'IN.',0.,3)
03100      CALL SYMBOL(.2,6.0,.1,'FRAC. LENGTH:',0.,12)
03110      CALL NUMBER(1.5,6.0,.1,BXLEN,0.,0)
03120      CALL SYMBOL(2.1,6.0,.1,'FT.',0.,3)
03130      CALL SYMBOL(.2,5.6,.1,'INJ. TEMP.:',0.,11)
03140      CALL NUMBER(1.5,5.6,.1,TEMP,0.,1)
03150      CALL SYMBOL(2.0,5.68,.1,75,0.,-1)
03160      CALL SYMBOL(2.1,5.6,.1,'F',0.,1)
03170      CALL SYMBOL(.2,5.1,.15,'RESERVOIR DATA',0.,14)
03180      CALL SYMBOL(.2,4.7,.1,'DEPTH:',0.,6)
03190      CALL NUMBER(1.25,4.7,.1,DEPTH,0.,1)
03200      CALL SYMBOL(2.,4.7,.1,'FT.',0.,3)
03210      CALL SYMBOL(.2,4.35,.1,'HEIGHT:',0.,7)
03220      CALL NUMBER(1.25,4.35,.1,H,0.,1)
03230      CALL SYMBOL(2.,4.35,.1,'FT.',0.,3)
03240      CALL SYMBOL(.2,4.0,.1,'PRESSURE:',0.,9)
03250      CALL NUMBER(1.25,4.0,.1,PRESS,0.,1)
03260      CALL SYMBOL(2.,4.0,.1,'PSI',0.,3)
03270      CALL SYMBOL(.2,3.65,.1,'TEMP:',0.,5)
03280      CALL NUMBER(1.25,3.65,.1,TRES,0.,1)
03290      CALL SYMBOL(2.,3.73,.1,75,0.,-1)
03300      CALL SYMBOL(2.1,3.65,.1,'F',0.,1)
03310      CALL SYMBOL(.2,3.3,.1,'PERM:',0.,5)

```

```

03320 CALL NUMBER(1.25,3.3,.1,PERM,0.,3)
03330 CALL SYMBOL(2.,3.3,.1,'MD',0.,2)
03340 CALL SYMBOL(.8,2.25,.15,'LEGEND',0.,6)
03350 YPS = 2.5
03360 DO 18 K=1,NPASS
03370 CALL SYMBOL(.3,YPS+.05,.1,LUMA(K),0.,-1)
03380 CALL SYMBOL(.67,YPS,.1,'TIME=',0.,6)
03390 TIMLEG = TIMLEG + THOURN
03400 CALL NUMBER(1.5,YPS,.1,TIMLEG,0.,2)
03410 CALL SYMBOL(2.1,YPS,.1,'HRS',0.,3)
03420 18 YPS = YPS - .24
03430 CALL PLOT(-10.5,1.,-3)
03440 RETURN
03450 END

```

C-13 Plott

```

01930      SUBROUTINE PLOTT( NUMPTS)
01940      INTEGER OUT
01950      DIMENSION XA(202),YA(202),LUMA(9),NUMPTS(9),DIF(9)
01960      COMMON/PLAY/ XD(202,9),YD(202,9),VISCOS(202,9)
01970      COMMON/C10/WORK(2,9),H,RINJ,C,FM,FK,TIME,PHI,LENGTH,TEMP,W,TRES
01980      COMMON/C50/ SAVAUT(10),NDUMMY(9),NPASS
01990      COMMON/C20/IN,OUT,XDUM,DXDUM -
02000      DATA MAX/200/,LUMA/0,1,2,5,6,10,14,36,37/,PLOTHT/6.5/,PLOTWD/9.5/
02010 C      PLOTHT & PLOTWD ARE PLOT FRAME HEIGHT & WIDTH.
02020 C      FIRST,FIND OUT WHICH OF THE NPASS CURVES TO USE TO CALL AXIS.
02030 C      (I.E.,ONE FOR WHICH MAX X VALUE MINUS MIN X VALUE IS GREATEST)
02040      SMALL = .1E+10
02050      BIG = .1E-10
02060      DO 5 K=1,NPASS
02070      NCR = NUMPTS(K)
02080      NCRLIM = NCR
02090      IF (NCRLIM.GT.200) NCRLIM = 200
02100      DO 8 L=1,NCRLIM
02110      XPL = XD(L,K)
02120      IF (XPL.LT.SMALL) SMALL = XPL
02130      IF (XPL.GT.BIG ) BIG = XPL
02140      8 CONTINUE
02150      BIF(K) = BIG - SMALL
02160      5 CONTINUE
02170      BIGDIF = .1E-10
02180      IUSE = 0
02190      DO 9 K=1,NPASS
02200      IF (DIF(K).LE.BIGDIF) GO TO 9
02210      IUSE = K
02220      9 CONTINUE
02230 C      IUSE INDICATES WHICH CURVE TO USE IN CALLING AXIS.
02240      IF (IUSE.GT.0) GO TO 11
02250      WRITE(OUT,12)
02260      12 FORMAT(//,2X,'*** ERROR IN SUB.PLOTT. VALUE OF IUSE IS
02270      ? 'INVALID. JOB TERMINATED. ')
02280      STOP
02290      11 CONTINUE
02300      LPTS = NUMPTS(IUSE)
02310      IF (LPTS.GT.200) LPTS=200
02320      DO 10 K=1,LPTS
02330 C      YA(K) = YD(K,IUSE)
02340      10 XA(K) = XD(K,IUSE)
02350      CALL PLOT(-1.,-1.,3)
02360      CALL PLOT(10.5,-1.,2)
02370      CALL PLOT(10.5,7.5,2)
02380      CALL PLOT(-1.,7.5,2)
02390      CALL PLOT(-1.,-1.,2)
02400      CALL PLOT(1.,1.,3)
02410      CALL SCALE(XA,PLOTWD,LPTS,1)
02420 C      CALL SCALE(YA,PLOTHT,LPTS,1)
02430      BEGINX = XA(LPTS+1)

```

```

02440      DELTX = XA(LPTS+2)
02450 C      BEGINY = YA(LPTS+1)
02460 C      DELTY = YA(LPTS+2)
02470      INDELT = (TRES+20.-TEMP) / PLOTHT
02480      DELTY = INDELT
02490      BEGINY = TEMP
02500      CALL AXIS(0.,0.,',',-1,PLOTWD,0.,BEGINX,DELTX)
02510      CALL SYMBOL(3.,-0.75,0.175,'FRACTURE DISTANCE - FEET'
02520      ? 0.,24)
02530      CALL AXIS(0.,0.,',',1,PLOTHT,90.,BEGINX,DELTY)
02540      CALL SYMBOL(-.5,2.5,0.175,'TEMPERATURE - F',90.,15)
02550      CALL SYMBOL(-.69,4.8,.2,75,0.0,-1)
02560      CALL SYMBOL(1.3,7.0,0.175,'FLUID TEMPERATURE VS DISTANCE IN FRAC
02570      ?RE',0.0,41)
02580      CALL BXDATA
02590 20 CONTINUE
02600      DO 30 KEG=1,NPASS
02610      KPTS = NUMPTS(KEG)
02620      IF (KPTS.GT.200) KPTS = 200
02630      DO 25 K=1,KPTS
02640      XA(K) = XD(K,KEG)
02650 25 YA(K) = YD(K,KEG)
02660      XA(KPTS+1) = BEGINX
02670      XA(KPTS+2) = DELTX
02680      YA(KPTS+1) = BEGINY
02690      YA(KPTS+2) = DELTY
02700      NUMSYH = KPTS / 20
02710      CALL LINE(XA,YA,KPTS,1,NUMSYH,LUHA(KEG))
02720 30 CONTINUE
02730      CALL PLOT(0.,0.,-3)
02740      CALL PLOT(15.,-1.,-3)
02750      RETURN
02760 77 WRITE(6,84) KNT,MAX
02770 84 FORMAT(/,' **** ERROR. KNT=',I3,' IS GREATER THAN MAX. (',I3,')'
02780      ? ' PROGRAM TERMINATED.')
02790      STOP
02800      END

```

C-14 Block Data

```
01830      BLOCK DATA
01840      INTEGER OUT
01850      REAL LENGTH
01860      COMMON/C10/ WORK(2,9),H,RINJ,C,FM,FK,TIME,PHI,LENGTH,TEMP,W,TRE
01870      COMMON/C20/IN,OUT,X,DX
01880      COMMON/PLAY/ XP(202,9),YP(202,9),VISCOS(202,9)
01890      COMMON/C60/KURVE
01900      DATA XP/1818*0./,YP/1818*0./,VISCOS/1818*0./
01910      DATA IN/2/,DX/5.0/,X/0.0/,OUT/6/,KURVE/1/
01920      END
```