

ADVANCED WELLBORE THERMAL SIMULATOR

GEOTEMP2,

APPENDIX. COMPUTER PROGRAM LISTING

By

Robert F. Mitchell

ENERTECH ENGINEERING AND RESEARCH CO.

2600 S.W. Freeway, Suite 300

Houston, Texas 77098

(713) 521-9033

Sandia Laboratories Contract 46-5670

Department of Energy

Geothermal Technology Development

Division 4741

November 1981

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.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

leg

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.

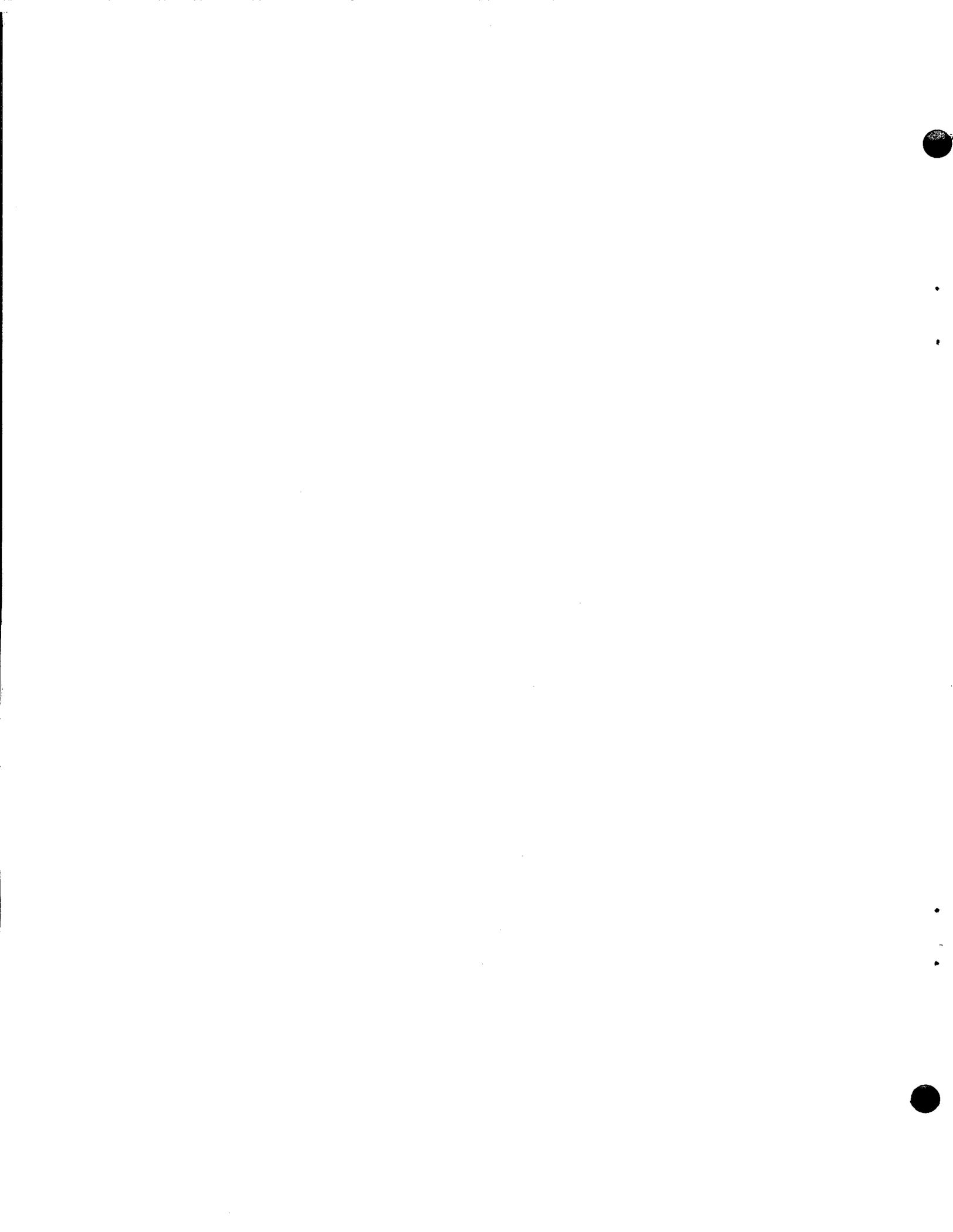


TABLE OF CONTENTS

	page
1. Introduction	1
2. MAIN Program	2
3. Initialization Programs	13
READ	14
PROP	18
GRID	19
4. Wellbore Flow Programs	21
A. Liquid Flow Subroutines	24
UPFLOW	25
DWNFLO	28
B. Gas Flow Subroutines	30
FLOW	31
GVISC	38
FRIC	39
CD	40
VEXIT	41
ACHNG	42
NOZL	43
VSLIP	44
VMIXT	45
C. Two-phase Flow subroutines	47
TUFAS	48
PSAT	57
TSAT	58
DVSAT	59
DLSAT	60
ESAT	61
SVVIS	62
SLVIS	63
KSAT	64
SURTN	65
STEAM	66
DENCR	68
FR2P	70
RLORK	71
GMORK	72
HMIK	73
H2PR	73a

5. Wellbore Heat Transfer Programs	74
COND	75
PROP	79
GPROP	80
KTUBE	82
CONV	83
CONVN	84
HTUBE	85
CONAN	86
CA	87
VISC	88
COEF	90
WELL	94
6. References	98

1. INTRODUCTION

This appendix gives the program listing of GEOTEMP2 with comments and discussion to make the program organization more understandable. As shown in the table of contents, this appendix is divided into an introduction and four main blocks of code:

1. Main program
2. Program Initialization
3. Wellbore flow
4. Wellbore heat transfer

In each section, the purpose and use of each subprogram is discussed and the program listing is given. Flowcharts will be included to clarify code organization when needed.

GEOTEMP2 has been written in FORTRAN IV. Efforts have been made to keep the programing as conventional as possible so that GEOTEMP2 will run without modification on most computers. GEOTEMP2 was developed on a Prime 550 mini-computer, and clearly should run on most mainframe computers. Smaller 16 bit mini and micro computers probably will not be suitable because of the program's size. The original GEOTEMP has been run on a DEC PDP 11/44 computer and the code executed very slowly. GEOTEMP is a much smaller code than GEOTEMP2, and the PDP 11/44 is a pretty capable 16 bit mini-computer. The inference is that this type of machine is too small.

2. MAIN Program

The MAIN program reads the input, controls the execution of the subprograms, and outputs the results. The overall organization is given in a flow chart as Figure 1. Unfortunately, MAIN does not break into blocks as cleanly as sketched in the flow chart, but for the most part this chart is correct, and comments in the code itself should take care of the exceptions.

The code first reads the required input data (see the GEOTEMP2 user manual-reference 1.) and initializes the program. This process is discussed in more detail in section 3. of this appendix. The actual subroutine calls occur on the first page of the MAIN listing along with the temperature initialization.

The next step in the execution of the program is to check for control input data and to read that data. Otherwise, the code will stop. Data are read beginning at line number 6 and continue through line number 54, depending on the options invoked.

The next general operation undertaken by the code is to determine the state of the wellbore fluids. This operation is discussed in section 4. of this report, and takes place for the most part between line number 54 and line number 11 and between line number 14 and line number 40.

Having determined the state of the wellbore fluids, the coefficients in the heat transfer equations can now be determined. These calculations are discussed in section 5. of this report. The subroutines that perform these calculations are called from a number of places in the code, for instance, between line numbers 11 and 4, 12 and 14, and between 42 and 43. Changing flow conditions, such as shut-in during drilling, require the re-evaluation of these coefficients.

The solution of the energy equation takes place at line number 43. This calculation is also discussed in section 5. Immediately following this computation, the current time is compared to the time to change data. If the time step is not complete, the program returns to line number 4. Otherwise, the code continues through the print section following line number 51. The code then returns to line number 6 to look for more control data. If no more data are found, the program exits.

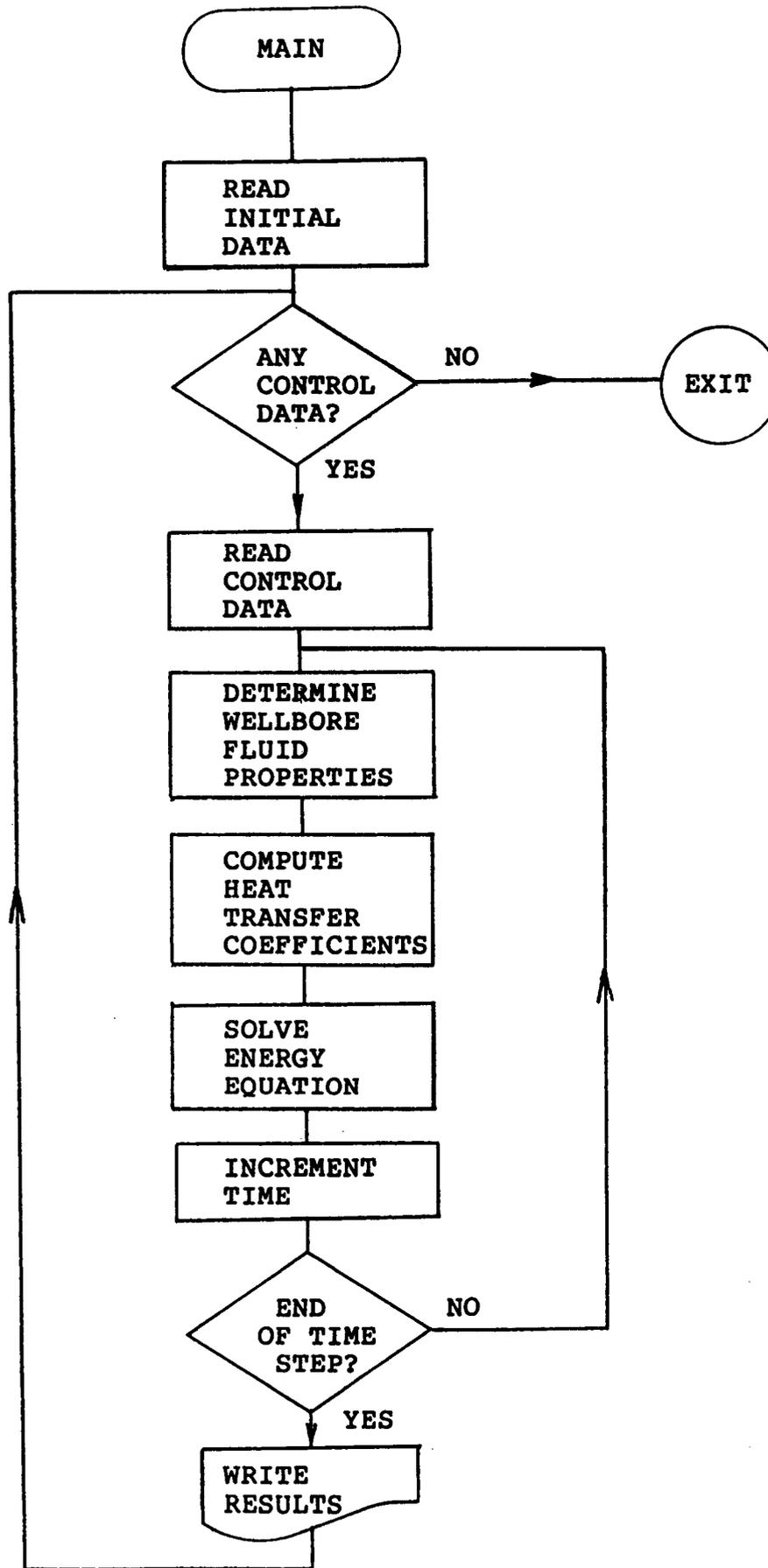


FIGURE 1: MAIN PROGRAM FLOW CHART

GEOTEMP2

THIS PROGRAM COMPUTES THE TEMPERATURES IN AND SURROUNDING A WELL
 CONSTRUCTED UNDER A CONTRACT WITH DOE THROUGH SANDIA FOR
 APPLICATION TO GEOTHERMAL WELL COMPLETIONS.

THIS REPRESENTS AN EXTENDED VERSION OF A PREVIOUS CODE
 CALLED GEOTEMP. IMPROVEMENTS INCLUDE GAS AND MIST DRILLING
 AND TWO PHASE STEAM INJECTION AND PRODUCTION.

SUBMITTED TO SANDIA IN SEPTEMBER 1981

```

COMMON /BLK1/ TW(151,20),TWJ(151,21)
COMMON /BLK3/ KDR,NZ,NZP1,TD,Z(151),ZC(15),DZ(15),TVD,DDVN
COMMON /BLK8/ NR,RMAX,R(21),DR(21),AR(20),RI(15),RO(15)
COMMON /BLK4/ U(151,20),Q(151,2)
COMMON /BLK6/ IFLOW,PI,DT,TIN1,TIN2,FR1,FR2,FR3,NIT
COMMON /BLK12/ VFR2
COMMON /BLK11/ VFR,DDCHG,COSDVN
COMMON /BLK13/ DENFP(5),PVFP(5),YPFP(5),SHFP(5),CONFP(5),
@IPF,IAF,ISF
COMMON /BLK14/ TSUR,BHT,TDEPTH,DEPTHHT
COMMON /BLK17/ ZI(10,2),NA(10,2),NI(2),NOF
COMMON /BLK18/ DENG(151,2),PRESS(151,2),VGAS(151,2)
COMMON /BLK19/ DEPTH,BITR,BITA,DCLRL,DCLRI,DCLRO,PSTAND
COMMON /BLK20/ DCUT(151),VCUT(151),DLIQ(151,2),DVAP(151,2),DRT,
& VFR3
COMMON /BLK21/ IREGM(151,3),QUALS,PSTEM,I2PH,ISKIP(151,2)
DIMENSION BITN(3)
DATA PI/3.1415927/
DATA TIME/0./,DTMAX/24./
  
```

READ IN VARIABLES

CALL READ

MATERIAL PROPERTIES

CALL PROP

DEFINE GRID

CALL GRID

INITIALIZE TEMPERATURES

```

COSDVN=(TVD-DDVN)/(TD-DDVN)
DTDZ1=(TDEPTH-TSUR)/DEPTHHT
DTDZ2=(BHT-TDEPTH)/(TVD-DEPTHHT)
NR1 = NR + 1
DO 28 K=1,NZP1
VD=Z(K)
IF(VD.GT.DDVN)VD=DDVN+(Z(K)-DDVN)*COSDVN
TT=DTDZ1*VD+TSUR
IF(VD.GT.DEPTHHT)TT=TDEPTH+DTDZ2*(VD-DEPTHHT)
  
```

```
DO 28 I=1,NR1
TWJ(K,I) = TT
28 TW(K,I) = TT
```

C

```
DCRIM=DCLRI*24.
DCROM=DCLRO*24.
BITRM=BITR*24.
```

C

C

C

READ AND WRITE NEW FLOWING DATA

C

C

IFLOW = FLOWING STREAM OPTION

C

1 = INJECTION

C

2 = PRODUCTION

C

3 = FORWARD CIRCULATION

C

4 = REVERSE CIRCULATION

C

5 = FORWARD CIRCULATION COMPRESSIBLE

C

6 = DRILLING, LIQUID SYSTEMS

C

(RESET TO IFLOW = 3)

C

7 = DRILLING, GAS SYSTEMS

C

(RESET TO IFLOW = 5)

C

8 = DRILLING, MIST

C

(RESET TO IFLOW = 5)

C

9 = STEAM PRODUCTION

C

10 = STEAM INJECTION

C

TIN1 = INLET TEMPERATURE OF FLOWING FLUID, F

C

VFRP = VOLUME FLOW RATE OF FLUID, GPM

C

VFRP = GAS FLOW RATE, SCF/MIN GAS FLOW OPTIONS

C

VFR2P=0.0

VFR3=0.

VFR3P=0.0

NFLOW=0

DAYS=0.0

6 READ(5,*,END=50,ERR=52)DUMMY,IFLOW,ISEC,IPF,TIN1,VFRP,DAYCH

I2PH=0

TCHG = DAYCH*24.

DDCHG=0.0

HRC=0.0

C

C

CHANGE CARD OPTIONS

C

C

GAS FLOW OPTIONS (IFLOW 5,7,8) READ PSTAND

C

IF((IFLOW-5)*(IFLOW-7)*(IFLOW-8).EQ.0)READ(5,*,END=50,ERR=52)
& DUMMMY,PSTAND

C

C

MIST FLOW OPTION (IFLOW 8) READ VFR3P

C

IF(IFLOW.EQ.8)READ(5,*,END=50,ERR=52)DUMMY,VFR3P

VFR3=VFR3P*60./7.4805

C

C

READ DRILLING PARAMETERS (IFLOW 6,7,8)

C

IBH = 0 NO BOTTOM HOLE ASSY CHANGE

C

IBH = 1 READ BOTTOM HOLE ASSY CARD

C

HRC = HOURS OF CIRCULATION DAILY

C

DDCHG = DEPTH REACHED AT TIME DAYCH

C

```
IF((IFLOW-6)*(IFLOW-7)*(IFLOW-8).EQ.0)READ(5,*,END=50,ERR=52)
& DUMMY,IBH,HRC,DDCHG
```

```
C
C READ BOTTOM HOLE ASSY DATA
C DCLRL = LENGTH OF DRILL COLLAR ASSEMBLY, FT.
C DCRIM = DRILL COLLAR I.D., IN.
C DCROM = DRILL COLLAR O.D., IN.
C BITRM = DRILL BIT DIAMETER, IN.
C BITN(I) = BIT NOZZLE DIAMETER, IN.
```

```
C
C IF(IBH.EQ.1)READ(5,*,END=50,ERR=52)DUMMY,DCLRL,DCROM,DCRIM,BITRM,
& (BITN(K),K=1,3)
```

```
C
C SECONDARY FLOW
```

```
C
C ISEC = 0 NO SECONDARY FLOW CHANGE
C ISEC = 1 SECONDARY FLOW CHANGE
```

```
C
C IF(ISEC.EQ.1)READ(5,*,END=50)DUMMY,ISF,TIN2,VFR2P
```

```
C
C VOLUME FLOW RATE FT3/HR
```

```
C
C VFR2=VFR2P*60./7.4805
```

```
C
C TWO-PHASE STEAM PRODUCTION/INJECTION
```

```
C
C IF(IFLOW.LT.9)GO TO 54
```

```
C
C IFLOW = 9 TWP-PHASE STEAM PRODUCTION
C IFLOW = 10 TWO-PHASE STEAM INJECTION
```

```
C
C READ (5,*)DUMMY,ISTM,XXXX
C IF(ISTM.EQ.1)QUALS=XXXX
C IF(ISTM.EQ.2)PSTEM=XXXX
```

```
C
C TINK=(TIN1+459.67)/1.8
C CALL PSAT(PSTST,XX,TINK)
C PSTST=PSTST/6894.757
```

```
C
C IF(ISTM.EQ.1)PSTEM=PSTST
C IF(ISTM.EQ.1)GO TO 55
C IF(PSTEM.GT.PSTST)QUALS=0.
C IF(PSTEM.LE.PSTST)QUALS=1.
```

```
55 CONTINUE
C IF(IFLOW.EQ.9)IFLOW=2
C IF(IFLOW.EQ.10)IFLOW=1
C NFLOW=1
C VFR=VFRP*60./7.4805
C I2PH=1
```

```
C
C WRITE(6,104)DAYS
C IF(IFLOW.EQ.2)WRITE(6,153)
C IF(IFLOW.EQ.1)WRITE(6,154)
C IF(ISTM.EQ.1)WRITE(6,155)QUALS
C IF(ISTM.EQ.2)WRITE(6,156)PSTEM
C WRITE(6,116)TIN1,VFRP,DAYCH
```

CALL TUFAS
GO TO 11

C
C

54 CONTINUE

IF (IFLOW.EQ.6) IFLOW=3
IF (IFLOW.EQ.7) IFLOW=5
IF (IFLOW.EQ.8) IFLOW=5
WRITE (6,104) DAYS
IF (IFLOW .EQ. 1) WRITE (6,108)
IF (IFLOW .EQ. 2) WRITE (6,110)
IF (IFLOW .EQ. 3) WRITE (6,112)
IF (IFLOW .EQ. 5) WRITE (6,112)
IF (IFLOW .EQ. 4) WRITE (6,114)
IF (IFLOW.EQ.5.AND.IPF.EQ.1)WRITE(6,139)
IF (IFLOW.EQ.5.AND.IPF.EQ.2)WRITE(6,140)
IF (IFLOW.EQ.5.AND.IPF.EQ.3)WRITE(6,141)
IF (IFLOW.EQ.5.AND.VFR3.GT.0.01)WRITE(6,149)VFR3P
IF (NFLOW.EQ.0.AND.IFLOW.LT.5)GO TO 34
NI(1)=1
NI(2)=1
NA(1,1)=IPF
NA(1,2)=IPF
TD1=TD
IF (DDCHG.GT.0.10)TD1=DEPTH
ZI(1,1)=TD1
ZI(1,2)=TD1
NFLOW=1
IF (IFLOW.GT.4)GO TO 47
NFLOW=0

34 CONTINUE

IF (IFLOW .LT. 1 .OR. IFLOW .GT. 5) WRITE (6,115)
IF (IFLOW.EQ.1.OR.IFLOW.EQ.3)WRITE(6,130)IPF
IF (IFLOW.EQ.2)WRITE(6,132)IPF
IF (IFLOW.EQ.4)WRITE(6,134)IPF
IF (NI(1).EQ.1.AND.NI(2).EQ.1)GO TO 45
DO 46 JNT=1,2
IF (JNT.EQ.1)WRITE(6,124)
IF (JNT.EQ.2)WRITE(6,126)
NIJ=NI(JNT)
DO 46 INT=1,NIJ
NN1=NIJ-INT+1
ZTOP=0.
IF (INT.GT.1) ZTOP=ZI (NN1+1, JNT)
46 WRITE (6,128) NA (NN1, JNT), ZTOP, ZI (NN1, JNT)
GO TO 47
45 IF (NA(1,1).EQ.NA(1,2))WRITE(6,136)NA(1,1)
IF (NA(1,1).NE.NA(1,2))WRITE(6,138)NA(1,1),NA(1,2)
47 CONTINUE
IF (ISEC.EQ.1)WRITE(6,152)ISF,TIN2,VFR2P
IF (IFLOW.GT.4)GO TO 35
WRITE (6,116) TIN1,VFRP,DAYCH
GO TO 36
35 VFR=VFRP
WRITE(6,117)TIN1,VFR,DAYCH
GO TO 37

```

C
C CONVERT UNITS ON FLOW RATE, GAL/MIN TO FT3/HR
C   60 MIN/HR AND 7.4805 GAL/FT3
C
36 VFR = VFRP*60./7.4805
37 IF (HRC .GT. 0.) DRT = 24.*(DDCHG - DEPTH)/((TCHG - TIME)*HRC)
   IF (HRC.LE.0.0)DRT=0.0
   VFRC = VFR
   IF (DDCHG.LT.0.1)GO TO 11
   WRITE (6,145)DDCHG,HRC
   WRITE (6,146)DCLRL,DCRIM,DCROM
   WRITE (6,147)BITRM, (BITN(LL),LL=1,3)
   DCLRI=DCRIM/24.
   DCLRO=DCROM/24.
   BITR=BITRM/24.
   BITA=0.0
   DO 9 LL=1,3
   9 BITA=BITA+3.1416*(BITN(LL)/24.)**2
   IF (BITA.LE.0.0)BITA=3.14159*DCLRI**2
   KDR=DEPTH/(Z(2)-Z(1))+1.5
   IF (KDR.LE.1)KDR=2
11 CONTINUE

C
C FORM WELLBORE CONDUCTANCES
C
   DELT = 0.01
   DT = DELT*DTMAX
   IF (IFLOW.GT.4)CALL FLOW
   CALL COND
   CALL COEF

C
C INCREMENT TIME
C
   IDRLL=0
   4 CONTINUE
   IF (DDCHG .GT. 0.1) GO TO 8

C
C TIME INCREMENTS - NOT DRILLING
C
   DT=DELT*DTMAX
   GO TO 12

C
C TIME INCREMENTS - DRILLING
C
   8 CONTINUE
   IF (HRC.GE.0.1)GO TO 10
   HRC=0.0
   VFR=0.
   DRT=0.0
   IDRLL=3
10 IF (IDRLL.GE.3)GO TO 16

C
C DRILLING
C
   DT=HRC/3.0
   IF (HRC.LE.2.0)DT=HRC/2.0

```

```
IF(HRC.LE.1.0)DT=HRC
IF(HRC.LE.2.0)IDRLL=1
IF(HRC.LE.1.0)IDRLL=2
VFR=VFCR
IDRLL=IDRLL+1
GO TO 12
```

C
C
C

SHUT-IN

```
16 VFR=0.
IF(IDRLL.EQ.3.AND.IFLOW.GT.4)CALL FLOW
IF(IDRLL.EQ.3)CALL COND
DT=8.0-HRC/3.0
IF(HRC.GT.22.0)DT=12.-HRC/2.0
IF(HRC.GT.23.0)DT=24.-HRC
IF(HRC.GT.22.0)IDRLL=4
IF(HRC.GT.23.0)IDRLL=5
IDRLL=IDRLL+1
IF(IDRLL.EQ.6)IDRLL=0
IF(HRC.GT.23.9)HRC=24.0
IF(HRC.GT.23.9)GO TO 8
```

C
C

```
12 IF(TIME.GE.TCHG)GO TO 39
IF(TIME+DT.GT.TCHG)DT=TCHG-TIME
TIME = TIME + DT
IF(DT.LT.0.01)DT=0.01
DAYS = TIME/24.
DELT=1.25*DELT
IF (DDCHG .LT. 0.1) GO TO 14
DRLRT = DRT
IF (VFR .LT. 0.1) DRLRT = 0.
DEPTH = DEPTH + DT*DRLRT
KDRO = KDR
ZCELL=Z(2)-Z(1)
KDR=DEPTH/ZCELL+1.5
IF(KDR.LE.1)KDR=2
NZ=KDR+1
NZP1=KDR+2
IF (KDR .EQ. KDRO) GO TO 14
IF(IFLOW.GT.4)CALL FLOW
CALL COND
CALL COEF
```

C
C
C

COMPUTE WELLBORE TEMPERATURES

```
14 IF(IFLOW.GT.4)GO TO 40
IF(VFR.LT.1.0)GO TO 40
IF(NI(1).EQ.1.AND.NI(2).EQ.1.AND.IPF.EQ.NA(1,1))GO TO 40
```

C
C
C

UPDATE WELLBORE FLUID INTERFACES

```
VF=VFR*DT
VF2=0.
IF(ISEC.EQ.1)VF2=VFR2*DT
IF(IFLOW.EQ.2)GO TO 41
```

```

JFLOW=1
IF (IFLOW.EQ.4) JFLOW=2
CALL DWNFLO(VF,IPF,IFLOW,JFLOW)
IF (IFLOW.EQ.1) GO TO 42
41 CALL UPFLOW(VF,VF2,IPF,IFLOW)
42 CALL COND
WRITE(6,120) DAYS
DO 44 JNT=1,2
IF (JNT.EQ.1) WRITE(6,124)
IF (JNT.EQ.2) WRITE(6,126)
NIJ=NI(JNT)
DO 44 INT=1,NIJ
NN1=NIJ-INT+1
ZTOP=0.
IF (INT.GT.1) ZTOP=ZI(NN1+1,JNT)
44 WRITE(6,128) NA(NN1,JNT),ZTOP,ZI(NN1,JNT)
GO TO 43
40 IF (IFLOW.GT.4) CALL FLOW
CALL COEF
43 CALL WELL(DAYS)
IF ( TIME .LT. (TCHG-1.E-3) .AND. DDCHG .LT. 0.1) GO TO 4
TLEFT=TCHG-TIME
IF (IDRLLEQ.0.AND.TLEFT.LE.23.99) GO TO 51
IF (IDRLLEQ.3.AND.TLEFT.LE.23.99) GO TO 51
GO TO 4
51 CONTINUE

```

```

C
C PRINT OPTION
C

```

```

GPM = VFR*7.4805/60.
WRITE (6,101) DAYS,NIT
IF (DDCHG .GT. 0.1.AND.IFLOW.LE.4) WRITE (6,122) GPM,DEPTH
IF (DDCHG.GT.0.1.AND.IFLOW.GE.5) WRITE(6,123) VFR,DEPTH
WRITE (6,118) (R(I),I=1,5),R(NR1)
WRITE (6,106) (Z(K),(TWJ(K,I),I=1,5),TWJ(K,NR1),K=1,NZP1)
IF (I2PH.EQ.0) GO TO 58
IF (IFLOW.EQ.2) WRITE(6,157)
IF (IFLOW.EQ.1) WRITE(6,158)
WRITE(6,159)
DO 57 I=1,KDR
57 WRITE(6,160) Z(I),PRESS(I,1),TWJ(I,1),
& DVAP(I,1),DLIQ(I,1),VGAS(I,1),(IREGM(I,J),J=1,3)
58 CONTINUE
IF (IFLOW.LT.5) GO TO 39
WRITE(6,142)
WRITE(6,143)
DO 38 I=1,KDR
WRITE(6,144) Z(I),PRESS(I,1),TWJ(I,1),DENG(I,1),DLIQ(I,1),
& DVAP(I,1),VGAS(I,1)
38 CONTINUE
WRITE(6,150)
DO 338 I=1,KDR
WRITE(6,151) Z(I),PRESS(I,2),TWJ(I,2),DENG(I,2),DLIQ(I,2),
& DVAP(I,2),DCUT(I),VGAS(I,2),VCUT(I)
338 CONTINUE
39 CONTINUE

```

```

        IF (ABS((TIME - TCHG)/TIME) .LT. 0.001) GO TO 6
        IF (TIME.GE.TCHG) GO TO 6
        GO TO 4
50 STOP
C
C INPUT ERROR
C
52 WRITE(6,148)
    STOP
C
C FORMAT STATEMENTS
C
101 FORMAT (///1X,'TIME =',F10.3,' DAYS',30X,'ITERATIONS =',I3)
104 FORMAT ('1'//2X,'S E T   V A R I A B L E S   AT TIME =',
    & F10.3,' DAYS')
106 FORMAT (3X,F8.0,3X,5F9.1,4X,F9.1)
108 FORMAT (10X,'FLOWING OPTION = INJECTION')
110 FORMAT (10X,'FLOWING OPTION = PRODUCTION')
112 FORMAT (10X,'FLOWING OPTION = FORWARD CIRCULATION')
114 FORMAT (10X,'FLOWING OPTION = REVERSE CIRCULATION')
115 FORMAT (//10X,'FLOWING OPTION INCORRECTLY DEFINED'//)
116 FORMAT (10X,'INLET TEMPERATURE =',F6.0,' F'/
    & 10X,'FLOW RATE =',F6.0,' GAL/MIN'/
    & 10X,'TIME TO CHANGE DATA =',F10.3,' DAYS')
117 FORMAT(10X,'INLET TEMPERATURE=',F6.0,' F'/
    & 10X,'FLOW RATE =',F6.0,' SCF/MIN'/
    & 10X,'TIME TO CHANGE DATA =',F10.3,' DAYS')
118 FORMAT (/17X,'T E M P E R A T U R E   ',
    & 'D I S T R I B U T I O N'//30X,'RADIAL POSITIONS, FEET'/
    & 4X,'DEPTH, FT',1X,5F9.1,4X,F9.1/)
120 FORMAT(/' FLUIDS IN WELL UPDATED AT TIME=',F10.3,' DAYS')
122 FORMAT (1X,'CONDITIONS SINCE LAST TIME STEP: '/
    & 5X,'FLOW RATE =',F5.0,' GAL/MIN',
    & 5X,'CIRCULATION DEPTH =',F6.0,' FT')
123 FORMAT(' CONDITIONS SINCE LAST TIME STEP: '/
    & 5X,'FLOW RATE =',F5.0,' SCF/MIN',
    & 5X,'CIRCULATION DEPTH =',F6.0,' FT')
124 FORMAT(18X,' FLUIDS IN TUBING')
126 FORMAT(18X,' FLUIDS IN ANNULUS')
128 FORMAT(20X,' FLUID #',I2,' FROM ',F7.0,' FT. TO ',F7.0,' FT.')
130 FORMAT(20X,' FLUID #',I2,' INJECTED INTO TUBING')
132 FORMAT(20X,' FLUID #',I2,' PRODUCED')
134 FORMAT(20X,' FLUID #',I2,' INJECTED INTO ANNULUS')
136 FORMAT(20X,' FLUID #',I2,' IN WELL')
138 FORMAT(20X,' FLUID #',I2,' IN TUBING'/
    @20X,' FLUID #',I2,' IN ANNULUS')
139 FORMAT(20X,' AIR INJECTED INTO TUBING')
140 FORMAT(20X,' NITROGEN INJECTED INTO TUBING')
141 FORMAT(20X,' FOAM INJECTED INTO TUBING')
142 FORMAT(//17X,'G A S   &   M I S T   D R I L L I N G'/
    & 10X,'F L O W I N G   S T R E A M   P R O P E R T I E S'//)
143 FORMAT(27X,'T U B I N G'//2X,'DEPTH   PRESSURE   TEMP',8X,
    & 'DENSITY-LBM/CF',7X,'VELOCITY'/3X,'FT',9X,'PSIA',5X,'F',8X,
    & 'GAS   WATER   VAPOR',6X,'FT/SEC')
144 FORMAT(1X,F7.0,3X,F7.1,2X,F6.1,1X,3F8.3,5X,F6.1)
145 FORMAT (10X,'DEPTH TO CHANGE DATA =',F6.0,' FT'/

```

```

& 10X,'CIRCULATION TIME PER DAY =',F6.1,' HRS'/)
146 FORMAT(10X,'BOTTOM HOLE ASSEMBLY: '/12X,'DRILL COLLARS: '/
& 14X,'LENGTH=',F5.0,' FT, I.D.=',F6.3,' IN, O.D.=',F6.3,' IN')
147 FORMAT( 12X,'DRILL BIT: '/14X,'DIAMETER=',F7.3,' IN, NOZZLE',
& ' SIZES=',3F7.3,' IN')
148 FORMAT('//'***** ERROR *****'// ' CHECK CHANGE OPTIONS')
149 FORMAT(10X,'MIST DRILLING: WATER ADDED AT',F6.1,' GAL/MIN')
150 FORMAT(/27X,'A N N U L U S'//2X,'DEPTH PRESSURE TEMP',12X,
& 'DENSITY-LBM/CF',10X,'VELOCITY-FT/SEC'/3X,'FT',9X,'PSIA',5X,'F',
& 8X,'GAS WATER VAPOR ROCKS',5X,'GAS ROCKS')
151 FORMAT(1X,F7.0,3X,F7.1,2X,F6.1,1X,4F8.3,2F8.1)
152 FORMAT(10X,'SECONDARY FLOW'/20X,'FLUID #',I2/
& 20X,'INLET TEMPERATURE=',F6.0,' F'/20X,'FLOW RATE=',F6.0,
& ' GAL/MIN')
153 FORMAT(10X,'FLOWING OPTION = TWO-PHASE STEAM PRODUCTION')
154 FORMAT(10X,'FLOWING OPTION = TWO-PHASE STEAM INJECTION')
155 FORMAT(10X,'STEAM QUALITY =',F6.3)
156 FORMAT(10X,'INLET PRESSURE =',F8.1,' PSIA')
157 FORMAT('//17X,'S T E A M P R O D U C T I O N'//)
158 FORMAT('//17X,'S T E A M I N J E C T I O N'//)
159 FORMAT(8X,'F L O W I N G S T R E A M P R O P E R T I E S'//
& 4X,' DEPTH PRESSURE TEMP DENSITY-LBM/FT3 VELOCITY'/
& 4X,' FT PSIA F VAPOR LIQUID FT/SEC ')
160 FORMAT(4X,F7.0,3X,F7.1,3X,F6.1,3X,F8.3,1X,F8.3,F9.1,7X,3A2)
200 FORMAT (20X,3I5,3F10.0)
201 FORMAT(20X,5F10.0)
202 FORMAT(20X,4F10.0)
203 FORMAT(A2)

```

C

END

3. INITIALIZE PROGRAM

Three subroutines are called from MAIN to initialize GEOTEMP2 for the particular problem specified in the input data. Subroutine READ reads the input that specifies the geometry of the wellbore, the undisturbed temperature of the ground, the fluid properties to be used, and the initial state of the wellbore. Subroutine PROP defines the soil, cement, and steel thermal properties and computes fluid thermal properties for liquid systems. Subroutine GRID generates the finite difference grid's coordinates.

SUBROUTINE READ

C
C
C

THIS SUBROUTINE READS THE PROBLEM PARAMETERS

```
COMMON /BLK3/ KDR,NZ,NZP1,TD,Z(151),ZC(15),DZ(15),TVD,DDVN
COMMON /BLK8/ NR,RMAX,R(21),DR(21),AR(20),RI(15),RO(15)
COMMON /BLK6/ IFLOW,PI,DT,TIN1,TIN2,FR1,FR2,FR3,NIT
COMMON /BLK11/ VFR,DDCHG,COSDVN
COMMON /BLK12/ VFR2
COMMON /BLK13/ DENFP(5),PVFP(5),YFPF(5),SHFP(5),CONFP(5),
@IPF,IAF,ISF
COMMON /BLK14/ TSUR,BHT,TDEPTH,DEPTH
COMMON /BLK16/ NPF,NTS,KZP
COMMON /BLK17/ ZI(10,2),NA(10,2),NI(2),NOF
COMMON /BLK19/ DEPTH,BITR,BITA,DCLRL,DCLRI,DCLRO,PSTAND
DIMENSION TITLE(20)
```

C
C
C

READ AND WRITE PROBLEM TITLE

```
READ (5,212) TITLE
212 FORMAT (20A4)
WRITE (6,124) TITLE
124 FORMAT ('1'//1X,20A4//)
```

C

31 READ(5,*)DUMMY,NTS

C

C

READ TUBING PARAMETERS

C

C

C

C

C

C

```
RI(J) = I.D. OF TUBING,IN.
RO(J) = O.D. OF TUBING,IN.
ZC(J) = BOTTOM OF TUBING INTERVAL,FT.
DZ(J) = CEMENT INTERVAL LENGTH,FT.
```

```
READ(5,*)(J,RI(J),RO(J),ZC(J),DZ(J),I=1,NTS)
DO 1 I=1,NTS
IF(RO(1).GE.RO(I))GO TO 2
RO(1)=RO(I)
2 IF(RI(1).LT.RI(I))GO TO 3
RI(1)=RI(I)
3 J=I+5
RO(J)=RO(I)
RI(J)=RI(I)
ZC(J)=ZC(I)
1 DZ(J)=DZ(I)
WRITE(6,101)
101 FORMAT(///1X,' TUBING CONFIGURATION'/
&8X,'TUBING ID,IN. OD,IN. TOP,FT. BASE,FT. CEMENT,FT.')
```

DT1=0.

```
DO 5 J=1,NTS
JJ=J+5
IF(J.GT.1)DT1=ZC(JJ-1)
5 WRITE(6,103)J,RI(JJ),RO(JJ),DT1,ZC(JJ),DZ(JJ)
103 FORMAT(9X,I2,2X,2F8.3,F9.0,F10.0,F10.1)
```

```

C
C READ CASING PARAMETERS
C   RI(J) = ID OF CASING, IN
C   RO(J) = OD OF CASING, IN
C   ZC(J) = SETTING DEPTH OF CASING, FT
C   DZ(J) = CEMENT INTERVAL LENGTH ON CASING J, FT
C
32 READ(5,*)DUMMY,NCS
   IF(NCS.EQ.0)GO TO 10
   DO 6 I=1,NCS
   READ(5,*)J,RI1,RO1,ZC1,DZ1
   J=J+1
   RI(J)=RI1
   RO(J)=RO1
   ZC(J)=ZC1
   6 DZ(J)=DZ1
202 FORMAT (20X,I5,4F10.0)
   WRITE (6,102)
102 FORMAT (///1X,'C A S I N G   P R O G R A M'/
& 8X,'CASING',6X,'ID, IN',4X,'OD, IN',
& 6X,'DEPTH, FT',2X,'CEMENT INTERVAL, FT')
   DO 66 I=1,NCS
   J=I+1
   66 WRITE(6,104)I,RI(J),RO(J),ZC(J),DZ(J)
   10 CONTINUE
104 FORMAT (1X,I10,5X,2F10.3,2X,F10.0,7X,F10.0)
C
C READ BOREHOLE GEOMETRY
C
C DEPTH = INITIAL DEPTH, FT.
C TD = TOTAL MEASURED DEPTH, FT.
C TVD = TRUE VERTICAL DEPTH, FT.
C DDVN = DEPTH OF DEVIATION, FT.
C R(3) = BOREHOLE DIAMETER, IN.
C
33 READ(5,*)DUMMY,DEPTH,TD,TVD,DDVN,R(3)
210 FORMAT(20X,5F10.0)
   WRITE(6,222)
222 FORMAT(//' W E L L   G E O M E T R Y'/)
   WRITE(6,224)TD,R(3)
224 FORMAT(10X,' TOTAL DEPTH=',F10.0,' FT.'/
@10X,' BORE DIAMETER=',F10.3,' IN.')
   IF(TD.GT.TVD)WRITE(6,226)DDVN,TVD
226 FORMAT(10X,' NOTE: DEVIATED WELL'/
@10X,' DEPTH OF DEVIATION=',F10.0,' FT.'/
@10X,' TRUE VERTICAL DEPTH=',F10.0,' FT.')

```

```

C
C COMPLETE CASING DESCRIPTION
C
38 IF(R(3).LE.RO(NCS+1))R(3)=RO(NCS+1)+FLOAT(5-NCS)
  IF(NCS.EQ.4)GO TO 22
  DO 8 II=NCS,3
    I=II+1
    J=II+2
    RO(J)=RO(I)+1.1
    RI(J)=RO(I)+.9
    ZC(J)=1.0
  8 DZ(J)=0.0
22 CONTINUE

```

```

C
C READ INITIAL TEMPERATURES
C
34 READ (5,*)DUMMY,TSUR,BHT,TDEPTH,DEPTHT
214 FORMAT (20X,4F10.0)
  IF(ABS(TD-DDVN).LT.1.)TD=TD+1
  IF(ABS(DEPTHT*(TVD-DEPTHT)).LT.1.)DEPTHT=DEPTHT+1.
  IF(TVD.GT.TD)TVD=TD-1.
  WRITE(6,208)
208 FORMAT(//' NOTE: TRUE DEPTH=MEASURED DEPTH')

```

```

C
C READ FLUID PROPERTIES
C
C DENFP(I) = DENSITY OF FLUID I,LBM/GAL
C PVFP(I) = PLASTIC VISCOSITY OF FLUID I, CENTIPOISE
C YFPF(I) = YIELD POINT OF FLUID I, LBF/100FT2
C
35 READ(5,*)DUMMY,NPF
  WRITE(6,107)
107 FORMAT(///1X,'W E L L B O R E   F L U I D   P R O P E R T ',
  @'I E S')
  DO 9 I=1,NPF
    READ(5,*)J,DENFP(J),PVFP(J),YFPF(J)
216 FORMAT(20X,I5,3F10.0,6A4)
  WRITE(6,116)J
116 FORMAT(/10X,'FLUID TYPE NO.',I5/)
  WRITE(6,117)DENFP(J)
117 FORMAT(10X,'DENSITY=',F6.1,' LBM/GAL')
  WRITE(6,118)PVFP(J)
118 FORMAT(10X,'PLASTIC VISCOSITY=',F5.0,' CENTIPOISE')
  WRITE(6,119)YFPF(J)
119 FORMAT(10X,'YIELD POINT=',F4.0,' LBF/100FT2')
  9 CONTINUE

```

```

C
C INITIALIZE FLUID PROPERTIES
C   IPF=PRIMARY FLUID #
C   ISF=SECONDARY FLUID #
C   IAF=ANNULAR FLUID #
C   TIN2=INLET TEMP FOR SECONDARY FLUID, F.
C   VFR2=VOLUME FLOW RATE OF SECONDARY FLUID, GPM
C
  36 READ(5,*)DUMMY,IPF,IAF
      ISF=1
      VFR2=0.
      TIN2=70.
  204 FORMAT(20X,3I5,2F10.0)
      WRITE(6,218)
  218 FORMAT(//1X,'W E L L B O R E   I N I T I A L   S T A T E'/)
      WRITE(6,219)IPF,IAF
  219 FORMAT(10X,' FLUID #',I2,' IN TUBING & TUBING ANNULUS'/
    @10X,' FLUID #',I2,' IN CASING - CASING ANNULI')
      DO 12 I=1,NPF
        IF((DENFP(I)-21.)*(8.32-DENFP(I)).LT.0.)WRITE(6,106)I,DENFP(I)
  106 FORMAT (/1X,10('*'),2X,'CAUTION FLUID',I3,' DENSITY =',
    & F6.1,' LBM/GAL',2X,10('*')/)
  12 CONTINUE
C
C   INITIALIZE INTERFACE PARAMETERS
C
      TD1=TD
      IF(DDCHG.GT..10)TD1=DEPTH
      ZI(1,1)=TD1
      ZI(1,2)=TD1
      NA(1,1)=IPF
      NA(1,2)=IPF
      NI(1)=1
      NI(2)=1
C
C CONVERT UNITS ON FLOW RATE, GAL/MIN TO FT3/HR
C   60 MIN/HR AND 7.4805 GAL/FT3
C
      VFR = VFR*60./7.4805
      VFR2 = VFR2*60./7.4805
  37 RETURN
      END

```

SUBROUTINE PROP

THIS SUBROUTINE DEFINES PROPERTIES OF THE FOLLOWING MATERIALS:
STEEL, CEMENT, WELLBORE FLUIDS, AND SOIL

COMMON /BLK9/ DENS(150), SHS(150), CONS(150)
COMMON /BLK10/ DENST, SHST, CONST, DENC, SHC, CONC
COMMON /BLK11/ VFR, DDCHG, COSDVN
COMMON /BLK12/ VFR2
COMMON /BLK13/ DENFP(5), PVFP(5), YFP(5), SHFP(5), CONFP(5),
@IPF, IAF, ISF
COMMON /BLK16/ NPF, NTS, KZP

SOIL PROPERTIES

DENS = DENSITY OF SOIL, LBM/FT3
SHS = SPECIFIC HEAT CAPACITY OF SOIL, BTU/LBM-F
CONS = THERMAL CONDUCTIVITY OF SOIL, BTU/HR-FT-F

DATA DENS/150*140./, SHS/150*0.30/, CONS/150*2.0/

STEEL AND CEMENT PROPERTIES

DENST = DENSITY OF STEEL, LBM/FT3
SHST = SPECIFIC HEAT CAPACITY OF STEEL, BTU/LBM-F
CONST = THERMAL CONDUCTIVITY OF STEEL, BTU/HR-FT-F
DENC = DENSITY OF CEMENT, LBM/FT3
SHC = SPECIFIC HEAT CAPACITY OF CEMENT, BTU/LBM-F
CONC = THERMAL CONDUCTIVITY OF CEMENT, BTU/HR-FT-F

DATA DENST/490./, SHST/0.11/, CONST/26.2/
DENC=104.
SHC=0.20
CONC=0.50

FLUID PROPERTIES

DENFP(I) = DENSITY OF FLUID I, LBM/GAL TO LBM/FT3
SHFP(I) = SPECIFIC HEAT CAPACITY OF FLUID I, BTU/LBM-F
CONFP(I) = THERMAL CONDUCTIVITY OF FLUID I, BTU/HR-FT-F

DO 1 I=1, NPF
IF (DENFP(I) .GT. 8.32) GO TO 2
WRITE (6, 102) I, DENFP(I)
102 FORMAT (/2X, 10(' ')/5X, 'FLUID TYPE=', I2, ' DENSITY =',
& F8.3, ' LBM/GAL'/9X, 'COMPARED TO 8.33 FOR WATER'/2X, 10(' ')/)
2 IF (DENFP(I) .GT. 10.3) GO TO 4
SF = 0.0798*(DENFP(I) - 8.33)
IF (SF.LT.0.0) SF=0.0
GO TO 6
4 SF = 0.0318*(DENFP(I) - 10.3) + 0.162
6 DENFP(I) = DENFP(I)*7.4805
SHFP(I) = 1. - 0.777*SF
CONFP(I) = 0.399 + 9.6*SF
1 CONTINUE
RETURN
END

SUBROUTINE GRID

C THIS SUBROUTINE FORMS THE R-Z GRID

C
 COMMON /BLK3/ KDR,NZ,NZP1,TD,Z(151),ZC(15),DZ(15),TVD,DDVN
 COMMON /BLK6/ IFLOW,PI,DT,TIN1,TIN2,FR1,FR2,FR3,NIT
 COMMON /BLK8/ NR,RMAX,R(21),DR(21),AR(20),RI(15),RO(15)
 COMMON /BLK15/ ATA(10,2)
 COMMON /BLK16/ NPF,NTS,KZP
 COMMON /BLK19/ DEPTH,BITR,BITA,DCLRL,DCLRI,DCLRO,PSTAND
 DATA NR/10/,RMAX/50.0/,DDEPTH/200./

C CHANGE DIAMETERS TO RADII, CHANGE INCHES TO FEET

C
 K=NTS+5
 DO 2 I=1,K
 RI(I) = RI(I)/24.
 2 RO(I) = RO(I)/24.

C INITIALIZE DRILLING DATA

C
 DCLRL=-200.
 DCLRI=RI(K)
 DCLRO=RO(K)
 BITA=PI*DCLRI**2
 BITR=RI(2)

C SOIL RADII

C
 R(1) = RI(1)/2.
 R(2) = (RI(2) + RO(1))/2.
 R(3) = R(3)/24.
 DR(3) = R(3) - R(2)
 CR = 0.1
 DCR = 1.0
 NR1 = NR + 1
 NIT = 0
 6 CR = CR + DCR
 NIT = NIT + 1
 DO 8 I=4,NR1
 8 DR(I) = DR(I-1)*CR
 DO 10 I=4,NR1
 10 R(I) = R(I-1) + (DR(I) + DR(I-1))/2.
 IF (RMAX .GT. R(NR1)) GO TO 6
 CR = CR - DCR
 DCR = DCR/10.
 IF (ABS(R(NR1) - RMAX)/RMAX .LT. 1.E-3) GO TO 12
 IF (NIT .LT. 50) GO TO 6
 WRITE (6,102)
 102 FORMAT (/2X,10('*'),2X,'RADIAL COORDINATES NOT CONVERGED',2X,
 & 10('*'))/
 12 CONTINUE

C
C
C

COMPUTE AREAS

```
DO 14 I=1, NR
  R1 = (R(I) + R(I+1))/2.
  RR = 0.
  IF (I .GT. 1) RR = R(I-1)
  R2 = (RR + R(I))/2.
14 AR(I) = PI*(R1*R1 - R2*R2)
DO 15 I=1, NTS
  ATA(I,1)=PI*RI(I+5)**2
15 ATA(I,2)=PI*(RI(2)**2-RO(I+5)**2)
```

C
C
C

DEFINE VERTICAL SECTIONS

```
KDR = 1.5 + (TD + 1.)/DDEPTH
NZ = KDR + 1
NZP1 = NZ + 1
DO 4 K=1, NZP1
4 Z(K) = (K-1)*DDEPTH
RETURN
END
```

4. Wellbore Flow Programs

The subroutines associated with wellbore flow determine the flowing stream properties necessary to perform the heat transfer calculations discussed in section 5. Three basic types of fluid flow are considered in GEOTEMP2: single phase liquid flow, single phase gas flow, and two phase steam flow.

A. Liquid flow

In liquid flow, most flowing stream properties of concern are constant or can be calculated independantly of the flow momentum equations. The principal concern in liquid flow in GEOTEMP2 is the location of interfaces when multiple fluids are present in the wellbore. Two subprograms are used for these calculations: UPFLOW and DWNFLO. As suggested by their names, UPFLOW is concerned with interfaces moving upward in the wellbore and DWNFLO is concerned with interfaces moving downward. These subroutines provide the following functions:

- a. move interfaces the appropriate distance for a given fluid influx
- b. renumber interfaces when a fluid is circulated out of the wellbore
- c. transfer fluid from tubing to annulus or vice-versa for circulation cases
- d. account for additional fluid influx for secondary flow.

The fluid interfaces are set to zero for gas flow and two phase steam flow. The problem of displacing wellbore fluids with a gas or two-phase steam is not attempted.

B. Gas flow

The principal subprogram used for gas flow calculations is called FLOW. This subroutine calculates the flowing steam properties from the tubing inlet to the annulus outlet using the temperatures determined by the wellbore heat transfer subprograms. Subroutine FLOW calls the following subprograms in the course of its computations:

- a. FUNCTION GVISC-This function subroutine computes the gas viscosity based on the correlations in reference 2.
- b. FUNCTION FRIC-This function subroutine computes the D'Arcy friction factor based on the correlation in reference 3.
- c. FUNCTION CD-This function subroutine computes the aerodynamic drag coefficients for cuttings from the correlation in reference 4.

- d. SUBROUTINE VEXIT-This subroutine computes the exit flow properties given the inlet properties, for gas flow. The equations used in VEXIT are given in reference 5.
- e. SUBROUTINE ACHNG-This subroutine determines the change in flow properties for a discontinuous change in flow area.
- f. SUBROUTINE NOZL-This subroutine tests for choked flow in the bit nozzles.
- g. SUBROUTINE VSLIP-This subroutine computes the initial slip velocity of the cuttings with respect to the flowing gas, based on the correlation in reference 4.
- h. SUBROUTINE VMIXT-This subroutine computes the exit flow properties for a gas, mist, and cuttings mixture given the inlet conditions. The calculations in VMIXT are based on the equations in reference 5.

C. Two-phase steam flow

The principal subprogram used for two-phase steam flow is called TUFAS. This subroutine calculates the flowing stream properties for injection or production in the tubing. Subroutine TUFAS calls the following subprograms in computing the flowing conditions:

- a. The following subroutines calculate saturated vapor/liquid properties for water, based on the correlations in reference 6:
 - PSAT - saturated pressure
 - TSAT - saturated temperature
 - DVSAT - saturated vapor density
 - DLSAT - saturated liquid density
 - ESAT - saturated enthalpy
 - SVVIS - vapor viscosity
 - SLVIS - liquid viscosity
 - KSAT - saturated thermal conductivity
 - SURTN - surface tension
- b. SUBROUTINE STEAM-This subroutine calculates pressure and internal energy, and their first partial derivatives, as a function of density and temperature. Both vapor and liquid phases are included in the correlation.
- c. SUBROUTINE DENCRC-This subroutine determines the density consistent with the input pressure, temperature, and initial guess for the density.
- d. FUNCTION FR2P-This function subroutine computes the D'Arcy friction factor for two phase flow. Correlations in references 3. and 7. are used.

- e. FUNCTION RLORK-This function subroutine computes the liquid fraction for two phase bubble flow based on reference 7.
- f. SUBROUTINE GMORK-This subroutine computes the bubble rise velocity and Orkiszewski's gamma function for two phase slug flow based on reference 7.
- g. FUNCTION HMIX-This function subroutine computes the two phase heat transfer film coefficient based on references 8. and 9.
- h. FUNCTION H2PR-This function subroutine computes the overall conductivity between the two-phase steam and the center of the tubing-casing annulus.

A. LIQUID FLOW SUBROUTINES

**UPFLOW
DWNFLO**

SUBROUTINE UPFLOW(VF,VF2,IPF,IFLOW)

C
C
C
C

VF = VOLUME PRODUCED
VOF = VOLUME CIRCULATED BY SUB DWNFLO

COMMON /BLK3/ KDR,NZ,NZP1,TD,Z(151),ZC(15),DZ(15),TVD,DDVN
COMMON /BLK11/ VFR,DDCHG,COSDVN
COMMON /BLK15/ ATA(10,2)
COMMON /BLK16/ NPF,NTS,KZP
COMMON /BLK17/ ZI(10,2),NA(10,2),NI(2),NOF
COMMON /BLK19/ DEPTH,BITR,BITA,DCLRL,DCLRI,DCLRO,PSTAND

C

J=1
VF1=VF+VF2
RCP=VF/VF1
TD1=TD
IF(DDCHG.GT..01)TD1=DEPTH
K=KTUBE(TD1)-5
NCO=0

C
C
C

RENUMBER INTERFACES IF DEPTH CHANGES

IF(IFLOW.EQ.3)J=2
J1=J+1
IF(J1.GT.2)J1=1
IF((ZI(1,J)-TD1)**2.LT.1.0)GO TO 5
NIJ=NI(J)
NUNF=0
NTD1=0
DO 6 I=1,NIJ
DTST=ZI(I,J)-TD1
IF(DTST.GT.1.0)NUNF=NUNF+1
6 IF(DTST**2.LE.1.0)NTD1=I
IF(NUNF.EQ.0.AND.NTD1.EQ.0)GO TO 8
IF(NTD1.EQ.0)NTD1=NUNF
ZI(NTD1,J)=TD1
DO 7 I=NTD1,NIJ
NN=I+1-NTD1
ZI(NN,J)=ZI(I,J)
7 NA(NN,J)=NA(I,J)
NI(J)=NI(J)-NTD1+1
GO TO 5
8 IF(NA(1,J).EQ.NA(1,J1))GO TO 16
NI(J)=NI(J)+1
NIJ=NI(J)
DO 9 I=2,NIJ
NN=NIJ-I+2
ZI(NN,J)=ZI(NN-1,J)
9 NA(NN,J)=NA(NN-1,J)
NA(1,J)=NA(1,J1)
16 ZI(1,J)=TD1
5 CONTINUE
IF(IFLOW.LT.3)GO TO 10

```

C
C ADD INTERFACES CIRCULATED BY DWNFLO
  RCP=RCP*ATA(K,J)/ATA(K,J1)
  NOF1=NOF
  IF(NA(1,1).EQ.NA(1,2))NOF1=NOF1-1
  IF(NOF1.LE.0)GO TO 13
  NI(J)=NI(J)+NOF1
  NIJ=NI(J)
  DO 1 I=1,NOF1
  NN=NIJ-I+1
  NA(NN,J)=NA(NN-NOF1,J)
  1 ZI(NN,J)=ZI(NN-NOF1,J)
  13 DI=TD1
  DO 12 I=1,NOF
  NN=NOF+1-I
  NA(NN,J)=NA(I,J1)
  DI=DI+(ZI(I,J1)-ZI(I+1,J1))/RCP
  12 ZI(NN,J)=DI
C
C RENUMBER INTERFACES FOR INJECTION SIDE
C
  NIJ=NI(J1)-NOF
  DO 14 I=1,NIJ
  ZI(I,J1)=ZI(I+NOF,J1)
  14 NA(I,J1)=NA(I+NOF,J1)
  NI(J1)=NI(J1)-NOF
  GO TO 20
C
C PRODUCTION
C
  10 IF(IPF.EQ.NA(1,J))GO TO 15
  NI(J)=NI(J)+1
  N=NI(J)-1
  DO 11 I=1,N
  II=N+2-I
  NA(II,J)=NA(II-1,J)
  11 ZI(II,J)=ZI(II-1,J)
  ZI(1,J)=ZI(2,J)+VF1/ATA(K,J)
  NA(1,J)=IPF
  GO TO 20
  15 ZI(1,J)=ZI(1,J)+VF1/ATA(K,J)
C
C CIRCULATE INTERFACES
C
  20 CONTINUE
  NIJ=NI(J)
  DO 30 I=1,NIJ
  VF3=VF1
  K=KTUBE(ZI(I,J))-5
  4 A=ATA(K,J)
  ZTRY=VF3/A
  IF(K.LE.1)GO TO 2
  IF(ZTRY.GT.ZI(I,J)-ZC(K+4))GO TO 3
  2 ZI(I,J)=ZI(I,J)-ZTRY
  IF(ZI(I,J).LT.0.)NCO=NCO+1
  GO TO 30

```

```
3 VF3=VF3-A*(ZI(I,J)-ZC(K+4))
  ZI(I,J)=ZC(K+4)
  K=K-1
  GO TO 4
30 CONTINUE
  NI(J)=NI(J)-NCO
  RETURN
  END
```

SUBROUTINE DWNFLO(VF,IPF,IFLOW,J)

C
 C VF = VOLUME OF FLUID INJECTED
 C Z(I,J) = DEPTH OF INTERFACE I
 C NA(I,J) = FLUID TYPE ABOVE INTERFACE I
 C NI(J) = NUMBER OF INTERFACES
 C IPF = FLUID TYPE INJECTED
 C

COMMON /BLK3/ KDR,NZ,NZP1,TD,Z(151),ZC(15),DZ(15),TVD,DDVN
 COMMON /BLK11/ VFR,DDCHG,COSDVN
 COMMON /BLK15/ ATA(10,2)
 COMMON /BLK16/ NPF,NTS,KZP
 COMMON /BLK17/ ZI(10,2),NA(10,2),NI(2),NOF
 COMMON /BLK19/ DEPTH,BITR,BITA,DCLRL,DCLRI,DCLRO,PSTAND

C
 C
 C NOF=0
 C TD1=TD
 C IF(DDCHG.GE..01)TD1=DEPTH
 C IF(DDCHG.LT.0.1)GO TO 15
 C NIJ=NI(J)
 C NUNF=0
 C NTD1=0
 C DO 16 I=1,NIJ
 C DTST=ZI(I,J)-TD1
 C IF(DTST.GT.1.)NUNF=NUNF+1
 16 IF(DTST**2.LE.1.)NTD1=I
 C IF(NUNF.EQ.0.AND.NTD1.EQ.0)GO TO 18
 C IF(NTD1.EQ.0)NTD1=NUNF
 C ZI(NTD1,J)=TD1
 C DO 17 I=NTD1,NIJ
 C NN=I+1-NTD1
 C ZI(NN,J)=ZI(I,J)
 17 NA(NN,J)=NA(I,J)
 C NI(J)=NI(J)-NTD1+1
 C GO TO 15
 18 ZI(1,J)=TD1
 15 CONTINUE
 C NN=NI(J)
 C IF(IPF.EQ.NA(NN,J))GO TO 10

C
 C NEW INTERFACE GENERATED

C
 C NN=NN+1
 C NI(J)=NI(J)+1
 C NA(NN,J)=IPF
 C ZI(NN,J)=0.

C
 C CIRCULATE INTERFACES DOWN

C
 C
 C 10 CONTINUE
 C NIJ=NI(J)
 C DO 20 I=1,NIJ
 C VF1=VF
 2 K=KTUBE(ZI(I,J))
 C A=ATA(K-5,J)

```

ZTRY=VF1/A
IF(ZTRY.GT.(ZC(K)-ZI(I,J)))GO TO 4
3 ZI(I,J)=ZI(I,J)+ZTRY
GO TO 5
4 IF(K.GE.NTS+5)GO TO 3
VF1=VF1-A*(ZC(K)-ZI(I,J))
ZI(I,J)=ZC(K)+.1
GO TO 2
5 IF(ZI(I,J).GT.TD1)NOF=NOF+1
20 CONTINUE

```

```

C
C GENERATE INTERFACE AT BOTTOMHOLE
C

```

```

IF(NOF.EQ.NI(J))GO TO 31
IF(TD1-ZI(NOF+1,J).GT.1.0)GO TO 33
ZI(NOF+1,J)=TD1
GO TO 34
33 NIJ=NI(J)
JJ=NI(J)-NOF
IF(JJ.LE.0)GO TO 31
DO 30 I=1,JJ
II=NIJ-I+2
ZI(II,J)=ZI(II-1,J)
30 NA(II,J)=NA(II-1,J)
31 JJ=NOF+1
NA(JJ,J)=NA(JJ-1,J)
ZI(JJ,J)=TD1
NI(J)=NI(J)+1
34 IF(IFLOW.GT.1)GO TO 40

```

```

C
C RENUMBER INTERFACES FOR INJECTION CASE IFLOW = 1
C

```

```

NIJ=NI(J)-NOF
DO 32 I=1,NIJ
ZI(I,J)=ZI(I+NOF,J)
32 NA(I,J)=NA(I+NOF,J)
NI(J)=NI(J)-NOF
40 RETURN
END

```

B. GAS FLOW SUBROUTINES

**FLOW
GVISC
FRIC
CD
VEXIT
ACHNG
NOZL
VSLIP
VMIXT**

SUBROUTINE FLOW

C
C COMPUTES FLOWING STREAM PROPERTIES
C

```

COMMON /BLK1/ TW(151,20),TWJ(151,21)
COMMON /BLK3/ KDR,NZ,NZP1,TD,Z(151),ZC(15),DZ(15),TVD,DDVN
COMMON /BLK8/ NR,RMAX,R(21),DR(21),AR(20),RI(15),RO(15)
COMMON /BLK11/ VFR,DDCHG,COSDVN
COMMON /BLK15/ ATA(10,2)
COMMON /BLK17/ ZI(10,2),NA(10,2),NI(2),NOF
COMMON /BLK18/ DENG(151,2),PRESS(151,2),VGAS(151,2)
COMMON /BLK19/ DEPTH,BITR,BITA,DCLRL,DCLRI,DCLRO,PSTAND
COMMON /BLK20/ DCUT(151),VCUT(151),DLIQ(151,2),DVAP(151,2),DRT,
& VFR3
COMMON /S1/ D2,V2,DPD,PHI
COMMON /S2/ DIAS,DENS,DELTA
COMMON /S3/ G3,G4,D3,D4
DATA DFTR/16.0184625/
DATA PFTR/6894.757/
DATA VFTR/.3048/
DATA DELTA/1.5/
DATA DIAS/0.01/
DATA DENS/2000./
DATA DLH2O/1000./

```

C
C INITIALIZE PARAMETERS
C

```

IGAS=NA(1,1)
IF((IGAS-1)*(IGAS-2).NE.0)GO TO 9
GO TO (1,2),IGAS
1 RG=287.06
DSCF=1.2229
GO TO 10
2 RG=296.08
DSCF=1.1828
GO TO 10
9 WRITE(6,1000)IGAS
1000 FORMAT(//' ***** ERROR *****'//' GAS #',I2,
& ' UNDEFINED'/)
STOP

```

C
10 VFRG=VFR
12 A1=ATA(1,1)*VFTR**2
IF(DCLRL.GE.Z(KDR))A1=3.14159*DCLRI**2*VFTR**2
VFRM=VFRG*VFTR**3/60.
G=VFRM*DSCF/A1
G3=VFR3*DLH2O*VFTR**3/3600./A1
T1=(TWJ(1,1)+459.67)/1.8
P1=PSTAND*PFTR
D1=P1/RG/T1
V1=G/D1

C
GG=9.80665
DL=(Z(2)-Z(1))*VFTR
DHYD=2.*SQRT(A1/3.14159)
ZDCL=(Z(KDR)-DCLRL)*VFTR

```

C      PRESS(1,1)=P1/PFTR
      DENG(1,1)=D1/DFTR
      VGAS(1,1)=V1/VFTR
C
      I=1
C      WRITE(6,2200) I, PRESS(I,1), DENG(I,1), VGAS(I,1), TWJ(I,1)
C2200  FORMAT(1X,I4,' P=',E11.4,' D=',E11.4,' V=',E11.4,' T=',E11.4)
C
C      SELECT STATIC OR FLOWING GAS
C
      IF(VFRG.LT.1.0)GO TO 40
      D3=G3/V1
      CALL DVSAT(D4,DX4,T1)
      IF(D4.GT.D3)D4=D3
      G4=D4*V1
      DLIQ(1,1)=(D3-D4)/DFTR
      DVAP(1,1)=D4/DFTR
C
C      DETERMINE FLOWING PROPERTIES IN TUBING
C
      I=1
14  I=I+1
      T2=(TWJ(I,1)+459.67)/1.8
      GCOS=GG
      IF(Z(I).GT.DDVN)GCOS=GG*COSDVN
C
C      CHECK FOR AREA CHANGES
C
      ZT=Z(I-1)*VFTR
      ZB=Z(I)*VFTR
C
C      DRILL COLLAR TO TUBING AREA CHANGE
C
      IF(ZT.GT.ZDCL)GO TO 15
      IF(ZB.GT.ZDCL)GO TO 16
C
C      TUBING AREA CHANGE
C
      K1=KTUBE(Z(I-1))
      K2=KTUBE(Z(I))
      IF(K1.EQ.K2)GO TO 15
      A2=ATA(K2-5,1)*VFTR**2
      DHYD2=2.*SQRT(A2/3.14159)
      DL1=ZC(K1)*VFTR-ZT
      GO TO 17
16  A2=3.14159*(DCLRI*VFTR)**2
      DHYD2=2.*DCLRI*VFTR
      DL1=ZDCL-ZT
17  TM=T1+(T2-T1)*DL1/DL
      RE=DHYD*D1*V1/GVISC(T1,IGAS)
      FPD=FRIC(RE,DHYD,1)/DHYD*D1*V1*DL1/4.
      GPD=GCOS*DL1*(D1+D3)
      CALL VEXIT(P1,D1,V1,T1,TM,FPD,GPD,IGAS)
      IF(IGAS.GE.100)GO TO 29

```

```

CALL ACHNG(P1,D1,T1,V1,A1,A2,IGAS)
IF(IGAS.GE.100)GO TO 29
DHYD=DHYD2
15 RE=DHYD*D1*V1/GVISC(T1,IGAS)
FPD=FRIC(RE,DHYD,1)/DHYD*D1*V1*DL1/4.
GPD=GCOS*DL1*(D1+D3)
CALL VEXIT(P1,D1,V1,T1,T2,FPD,GPD,IGAS)
IF(IGAS.GE.100)GO TO 29
C
PRESS(I,1)=P1/PFTR
DENG(I,1)=D1/DFTR
VGAS(I,1)=V1/VFTR
C
CALL DVSAT(D4,DX4,T1)
IF(D4.GT.D3)D4=D3
DLIQ(I,1)=(D3-D4)/DFTR
DVAP(I,1)=D4/DFTR
C
WRITE(6,2200)I,PRESS(I,1),DENG(I,1),VGAS(I,1),TWJ(I,1)
IF(I.EQ.KDR)GO TO 20
GO TO 14
C
CHECK BIT NOZZLES FOR CHOKED FLOW
C
20 A2=BITA*VFTR**2
CALL NOZL(P1,T1,D1,V1,A1,A2,IGAS)
IF(IGAS.GE.100)GO TO 29
C
AREA CHANGE: TUBING TO ANNULUS
C
IK=2
RI1=DCLRO*VFTR
IF(Z(I).LE.ZDCL)RI1=RO(KTUBE(Z(I)))*VFTR
RO1=BITR*VFTR
IF(Z(I).GT.ZC(2))GO TO 19
IK=1
RO1=RI(2)*VFTR
19 A2=3.14159*(RO1**2-RI1**2)
DHYD=2.*(RO1-RI1)
CALL ACHNG(P1,D1,T1,V1,A1,A2,IGAS)
IF(IGAS.GE.100)GO TO 29
T1=(TWJ(I,2)+459.67)/1.8
RI2=RI1
RO2=RO1
C
PRESS(I,2)=RG*T1*D1/PFTR
DENG(I,2)=D1/DFTR
VGAS(I,2)=V1/VFTR
C
CALL DVSAT(D4,DX4,T1)
IF(D4.GT.D3)D4=D3
DLIQ(I,2)=(D3-D4)/DFTR
DVAP(I,2)=D4/DFTR
C
DCUT(I)=0.
VCUT(I)=0.

```

C WRITE(6,2200)I,PRESS(I,2),DENG(I,2),VGAS(I,2),TWJ(I,2)

C
C
C

DETERMINE INNER CASING

ICASE=1
21 ICASE=ICASE+1
ZCS=ZC(ICASE)*VFTR
RCS=RI(ICASE)*VFTR
IF(Z(KDR).GE.ZC(ICASE))GO TO 22
IF(ICASE.EQ.5)GO TO 22
GO TO 21

C
C
C

DETERMINE SOLIDS MASS FLOW RATE IF DRILLING

22 ISOL=0
G2=DRT*DENS*3.14159*BITR**2*VFTR**3/AL/3600.
IF(DRT.LT.0.01)GO TO 28
ISOL=1
CALL VSLIP(D1,T1,V1,G2,GCOS,IGAS)
PRESS(I,2)=RG*D1*T1/PFTR
DENG(I,2)=D1/DFTR
VGAS(I,2)=V1/VFTR

C

CALL DVSAT(D4,DX4,T1)
IF(D4.GT.D3)D4=D3
DLIQ(I,2)=(D3-D4)/DFTR
DVAP(I,2)=D4/DFTR
DCUT(I)=D2/DFTR
VCUT(I)=V2/VFTR

C
C
C

FLOWING PROPERTIES IN ANNULUS

28 I=I-1
T2=(TWJ(I,2)+459.67)/1.8
GCOS=-GG
IF(Z(I).GT.DDVN)GCOS=-GG*COSDVN

C

ZT=Z(I)*VFTR
ZB=Z(I+1)*VFTR
K1=KTUBE(Z(I))
K2=KTUBE(Z(I+1))
ZTA=ZC(K1)*VFTR

C

23 ZM=ZT
IAR=0
IF(ZM.GT.ZDCL.OR.ZB.LE.ZDCL)GO TO 24

C
C
C

AREA CHANGE: DRILL COLLAR TO PIPE

RI2=RO(K2)*VFTR
ZM=ZDCL
IAR=1
GO TO 25

C

24 IF(ZM.GT.ZTA.OR.ZB.LE.ZTA)GO TO 25

```

C
C AREA CHANGE:TUBING SIZE
C
    RI2=RO(K1)*VFTR
    ZM=ZTA
    IAR=1
C
25 IF(ZM.GT.ZCS.OR.ZB.LE.ZCS)GO TO 26
C
C AREA CHANGE: OPEN HOLE TO CASING
C
    IAR=2
    RO2=RCS
    ZM=ZCS
C
26 DL1=ZB-ZM
    TM=T1+(T2-T1)*DL1/DL
    VIS1=GVIS(T1,IGAS)
    RE=DHYD*D1*V1/VIS1
    FPD=FRIC(RE,DHYD,IK)/DHYD*D1*V1*DL1/4.
    GPD=GCOS*DL1*(D1+D3)
    IF(ISOL.EQ.1)DPD=.5*CD(D1,DENS,VIS1,DIAS,1)*DELTA*D1*PHI
    & *DL1/DIAS
    IF(ISOL.EQ.0)CALL VEXIT(P1,D1,V1,T1,TM,FPD,GPD,IGAS)
    IF(ISOL.EQ.1)CALL VMIXT(P1,D1,V1,T1,TM,FPD,GPD,IGAS)
    IF(IGAS.GE.200)GO TO 30
    IF(IGAS.GE.100)GO TO 29
    IF(IAR.EQ.0)GO TO 27
    A2=3.14159*(RO2**2-RI2**2)
    DHYD=2.*(RO2-RI2)
    G2=G2*A1/A2
    D2=D2*A1/A2
    CALL ACHNG(P1,D1,T1,V1,A1,A2,IGAS)
    IF(ISOL.EQ.1)CALL VSLIP(D1,T1,V1,G2,GCOS,IGAS)
    IF(IGAS.GE.100)GO TO 29
    ZB=ZM
    IF(IAR.EQ.2)IK=1
    GO TO 23
C
27 CONTINUE
    PRESS(I,2)=P1/PFTR
    DENG(I,2)=D1/DFTR
    VGAS(I,2)=V1/VFTR
C
    CALL DVSAT(D4,DX4,T1)
    IF(D4.GT.D3)D4=D3
    DLIQ(I,2)=(D3-D4)/DFTR
    DVAP(I,2)=D4/DFTR
    DCUT(I)=0.
    VCUT(I)=0.
    IF(ISOL.EQ.1)DCUT(I)=D2/DFTR
    IF(ISOL.EQ.1)VCUT(I)=V2/VFTR

```

```

C
C   WRITE(6,2200) I,PRESS(I,2),DENG(I,2),VGAS(I,2),TWJ(I,2)
C
C   IF(I.EQ.1) RETURN
C   GO TO 28
C
C   CHOKED FLOW
C
C   29 PINC=P1/PFTR
C     IF(PINC.LE.5.0)PINC=5.0
C     IF(PINC.GT.100.0)PINC=100.0
C     PSTAND=PSTAND+PINC
C     IGAS=NA(1,1)
C     WRITE(6,1001)PSTAND
1001 FORMAT(/' CHOKED FLOW: STANDPIPE PRESSURE INCREASED TO',F8.1,
      & ' PSI'/)
C     GO TO 12
C
C   INSUFFICIENT LIFTING CAPACITY
C
C   30 VFR=VFR*1.05
C     IGAS=NA(1,1)
C     WRITE(6,1002)VFR
1002 FORMAT(/' GAS FLOW RATE TOO LOW- INCREASED TO:',F8.0,' SCF')
C     GO TO 10
C
C   STATIC COLUMN
C
C   40 P1=14.7*PFTR
C     T1=(TWJ(1,2)+459.67)/1.8
C     D1=P1/RG/T1
C
C     DENG(1,2)=D1/DFTR
C     PRESS(1,2)=P1/PFTR
C     VGAS(1,2)=0.0
C     DLIQ(1,2)=0.0
C     DVAP(1,2)=0.0
C
C     DO 41 I=2,KDR
C       GCOS=GG
C       IF(Z(I).GT.DDVN) GCOS=GG*COSDVN
C       P1=P1+D1*DL*GCOS
C       T1=(TWJ(I,2)+459.67)/1.8
C       D1=P1/RG/T1
C
C     DENG(I,2)=D1/DFTR
C     PRESS(I,2)=P1/PFTR
C     VGAS(I,2)=0.0
C     DLIQ(I,2)=0.0
C     DVAP(I,2)=0.0
41 CONTINUE
C
C     DENG(KDR,1)=DENG(KDR,2)
C     PRESS(KDR,1)=PRESS(KDR,2)
C     VGAS(KDR,1)=0.0
C     DLIQ(KDR,1)=0.0

```

```
DVAP(KDR,1)=0.0  
DCUT(KDR)=0.0  
VCUT(KDR)=0.0
```

C

```
DO 42 II=2,KDR  
I=KDR-II+1  
GCOS=GG  
IF(Z(I+1).GT.DDVN)GCOS=GG*COSDVN  
P1=P1-D1*DL*GCOS  
T1=(TWJ(I,1)+459.67)/1.8  
D1=P1/RG/T1
```

C

```
DENG(I,1)=D1/DFTR  
PRESS(I,1)=P1/PFTR  
VGAS(I,1)=0.0  
DLIQ(I,1)=0.0  
DVAP(I,1)=0.0  
DCUT(I)=0.  
VCUT(I)=0.
```

C

```
42 CONTINUE  
RETURN  
END
```

FUNCTION GVISC(T,IGAS)

C
C
C

COMPUTES VISCOSITIES OF GASES

```
IF(IGAS.EQ.3)GO TO 3
IF(IGAS.EQ.2)GO TO 1
D=.552795+2.810892E2/T-135083.40/T/T+39353086./T**3
& -41.419387E8/T**4
GO TO 2
1 D=.579561+2.847486E2/T-13.232490E4/T/T+37.106107E6/T**3
& -37.549675E8/T**4
2 GVISC=SQRT(T)/D*1.E-6
RETURN
3 GVISC=10.
RETURN
END
```

FUNCTION FRIC(RE,DHYD,IK)

C
C
C

COMPUTES DARCY FRICTION FACTOR

```
RR=.0000457/DHYD
IF(IK.EQ.2)RR=.00305/DHYD
A=.026*RR**.225+0.133*RR
B=22.*RR**0.44
C=-1.62*RR**0.134
FRIC=A+B*RE**C
FRIC=4.*FRIC
FRICL=64./RE
IF(FRICL.GT.FRIC)FRIC=FRICL
RETURN
END
```

FUNCTION CD(DEN1,DEN2,VIS1,D2,IND)

C
C THIS FUNCTION COMPUTES THE DRAG COEFFICIENT FOR SOLID
C PARTICLES MOVING IN A FLOWING STREAM BASED ON
C SWANSON'S CORRELATION.
C

C DEN1=DENSITY OF THE FLUID
C DEN2=DENSITY OF THE SOLID PARTICLE
C VIS1=VISCOSITY OF THE FLUID
C D2=THE AVERAGE PARTICLE DIAMETER
C IND=INDICATOR OF PARTICLE TYPE
C 1 IRREGULAR QUARTZ
C 2 CUBICAL GALENA
C 3 SPHERICAL
C 4 IRREGULAR KCL
C 5 IRREGULAR SPHALERITE
C

DIMENSION A(5),B(5)
DATA A/1.277,1.082,.942,1.870,1.022/
DATA B/2.80,3.11,3.27,2.56,2.18/
DATA G/9.80665/
VN=4./3.*G*D2*(DEN2-DEN1)/DEN1
VN=SQRT(VN)
CD1=A(IND)*(1.+6.93*B(IND)*VIS1/D2/DEN1/VN)
CD=CD1**2
RETURN
END

```

SUBROUTINE VEXIT(P1,D1,V1,T1,T2,FPD,GPD,IGAS)
C
C DETERMINES EXIT FLOW PROPERTIES FOR PIPE ELEMENT
C
COMMON /S3/ G3,G4,D3,D4
C
G1=D1*V1
G=G1+G3
R=P1/D1/T1
A=G+FPD+GPD/V1/2.
B=-P1-G*V1+FPD*V1-1.5*GPD
C=R*T2*G1
C
D=B*B-4.*A*C
IF(D.LE.0.)GO TO 1
V1=(-B-SQRT(D))/A/2.
D1=G1/V1
D3=G3/V1
P1=R*D1*T2
T1=T2
RETURN
1 IGAS=IGAS+100
X=C/P1
Y=B+P1
B=-2.*Y-4.*A*X
C=Y*Y
P2=(-B+SQRT(B*B-4.*C))/2.
P1=P2-P1
IF(P1.LE.35000.)P1=35000.
RETURN
END

```

SUBROUTINE ACHNG(P1,D1,T1,V1,A1,A2,IGAS)

C
C COMPUTES CHANGE IN FLOWING PROPERTIES
C DUE TO ABRUPT AREA CHANGE
C

COMMON /S3/ G3,G4,D3,D4

CP=1004.0

IF(IGAS.EQ.2)CP=1038.3

CW=0.0

IF(D3.GT.1.E-6)CALL DLSAT(D3A,XX1,T1)

IF(D3.GT.1.E-6)CALL STEAM(XX1,XX2,XX3,XX4,XX5,CW,D3A,T1)

R=P1/D1/T1

C
DM1=D1*V1*A1

DM3=G3*A1

DMT=DM1+DM3

C
CPA=(DM1*CP+DM3*CW)/DMT

C
A=DMT-.5*DM1*R/CPA

B=-P1*A2-DMT*V1

C=R*DM1*(T1+V1**2*.5/CPA)

DD=B*B-4.*A*C

IF(DD.LE.0.0)GO TO 1

V1A=(-B-SQRT(DD))/A/2.

T1=T1+.5/CPA*(V1**2-V1A**2)

D1=DM1/A2/V1A

D3=DM3/A2/V1A

P1=R*D1*T1

V1=V1A

A1=A2

RETURN

1 IGAS=IGAS+100

G=DM1/A2

B=2.*G*V1-4.*G*G/D1

C=(G*V1)**2

P2=(-B+SQRT(B*B-4.*C))/2.

P1=P2-P1

IF(P1.LE.35000.)P1=35000.

RETURN

END

SUBROUTINE NOZL(P,T,D,V,A1,A2,IGAS)

C
C THIS SUBROUTINE CHECKS FOR CHOKED FLOW IN THE BIT NOZZLES
C IF FLOW IS NOT CHOKED-RETURN TO FLOW
C IF FLOW IS CHOKED-IGAS INCREASED BY 100 AND P SET
C TO ESTIMATED STANDPIPE PRESSURE INCREASE REQUIRED
C

CP=1004.0
IF(IGAS.EQ.2)CP=1038.3
GO=D*V
G=GO*A1/A2
Z=P/D/T
A=.5/CP-1./Z
B=D*T/G+V/Z
C=T+.5*V*V/CP
DD=B*B+4.*A*C
IF(DD.GE.0.0)RETURN
P1=SQRT(-4.*A*C)*Z*G-V*G
P=P1-P
IF(P.LT.35000.)P=35000.
IGAS=IGAS+100
RETURN
END

SUBROUTINE VSLIP(DEN1,T,V1,G2,GG,IGAS)

C
C
C

INITIAL SLIP VELOCITY

COMMON /S1/ D2,V2,DPD,PHI
COMMON /S2/ D,DENS,DELTA

C

GCOS=ABS(GG)
G1=DEN1*V1
VN=SQRT(4.*GCOS*D*(DENS-DEN1)/DEN1/3.)
VIS1=GVIS(T,IGAS)
CD1=CD(DEN1,DENS,VIS1,D,1)
VS=VN*SQRT(1./CD1)
A=VS
B=G2/DENS+G1/DEN1-VS
C=G2/DENS
PHI=(SQRT(B*B+4.*A*C)-B)/2./A
V1=G1/(1.-PHI)/DEN1
V2=V1-VS
D1=G1/V1
D2=G2/V2
RETURN
END

SUBROUTINE VMIXT(P1,D1,V1,T1,T2,FPD,GPD,IGAS)

C

COMMON /S1/ D2,V2,DPD,PHI
COMMON /S2/ DIAS,DENS,DELTA
COMMON /S3/ G3,G4,D3,D4

C

R=P1/T1/D1
G1=D1*V1
G=G1+G3
G2=D2*V2
D3=G3/V1
DENM=D1+D2+D3
GPD=GPD/D1
FPD=FPD/D1

C

W10=GPD*DENM
W11=-.5*W10/V1
W12=-.5*W10/V2
F10=-FPD/G
F11=F10*G
F12=F10*G2
F10=2.*F10*(G*V1+G2*V2)

C

W20=GPD*PHI*(DENS-D1)
W22=-.5*W20/V2
PH0=DPD
PH1=.5*PH0*(V1*V1-V2*V2)
PH2=-PH1/V2
PH1=PH1/V1
PH0=PH0*(V1-V2)*(V1-V2)

C

DV2D=G2-W22-PH2
DV20=(W20+PH0)/DV2D
DV2V1=PH1/DV2D

C

D1V1=G-W11-F11+(G2-W12-F12)*DV2V1
D10=P1+W10+F10-(G2-W12-F12)*DV20

C

V1P=V1*(1.-PHI)+V1/V2*PHI*DV20
DV1P=1.0-PHI+V1/V2*PHI*DV2V1

C

A=D1V1*DV1P
B=D1V1*V1P-D10*DV1P
C=G1*R*T2-D10*V1P
DTEST=B*B-4.*A*C
IF(DTEST.LE.0.0)GO TO 100
DV1=(-B-SQRT(DTEST))/2./A
V1=V1+DV1
IF(V1.LE.0.0)GO TO 200
DV2=DV2V1*DV1+DV20
V2=V2+DV2
IF(V2.LE.0.0)GO TO 200
D1=G1/V1
D2=G2/V2
D3=G3/V1
PHI=D2/DENS

```
T1=T2  
P1=R*D1*T1
```

C

```
RETURN  
100 IGAS=IGAS+100  
P1=DTEST/(B*DV1P-2.*A*V1P)  
IF(P1.LE.35000.)P1=35000.  
RETURN  
200 IGAS=IGAS+200  
RETURN  
END
```

C. TWO-PHASE FLOW SUBROUTINES

TUFAS
PSAT
TSAT
DVSAT
DLSAT
ESAT
SVVIS
SLVIS
KSAT
SURTN
STEAM
DENCR
FR2P
RLORK
GMORK
HMIX
H2PR

SUBROUTINE TUFAS

C
C COMPUTES FLOWING STREAM PROPERTIES
C FOR TWO-PHASE FLOW
C

```
COMMON /BLK1/ TW(151,20),TWJ(151,21)
COMMON /BLK3/ KDR,NZ,NZP1,TD,Z(151),ZC(15),DZC(15),TVD,DDVN
COMMON /BLK6/ IFLOW,PI,DT,TIN1,TIN2,FR1,FR2,FR3,NIT
COMMON /BLK8/ NR,RMAX,R(21),DR(21),AR(20),RI(15),RO(15)
COMMON /BLK11/ VFR,DDCHG,COSDVN
COMMON /BLK15/ ATA(10,2)
COMMON /BLK17/ ZI(10,2),NA(10,2),NI(2),NOF
COMMON /BLK18/ DENG(151,2),PRESS(151,2),VGAS(151,2)
COMMON /BLK19/ DEPTH,BITR,BITA,DCLRL,DCLRI,DCLRO,PSTAND
COMMON /BLK20/ DCUT(151),VCUT(151),DLIQ(151,2),DVAP(151,2),DRT,
& VFR3
COMMON /BLK21/ IREGM(151,3),QUALS,PSTEM,I2PH,ISKIP(151,2)
COMMON /S3/ G3,G4,D3,D4
DIMENSION IFRNM(3,6)
DATA IFRNM/'LI','QU','ID','BU','BB','LE','SL','UG',' ',' ',
& 'TR','AN','S','MI','ST',' ','VA','PO','R '/
DATA DFTR/16.0184625/
DATA PFTR/6894.757/
DATA VFTR/.3048/
```

C
C INITIALIZE
C
C

```
G3=0.0
G4=0.0
D3=0.0
D4=0.0
TSCF=288.15
CALL DLSAT(DLSCF,DD,TSCF)
FLM=VFR/60.*4.719474E-4*DLSCF
```

C
C
C
C
C
C
C
C
C
C
C

```
KAR=1
IF(IFLOW.EQ.2)KAR=KDR
KT=KTUBE(Z(KAR))-5
AREA=ATA(KT,1)*VFTR*VFTR
```

```
G1=FLM*QUALS/AREA
G2=FLM*(1.-QUALS)/AREA
```

```
P1=PSTEM*PFTR
T1=(TIN1+459.67)/1.8
IF(T1.GT.647.2)GO TO 2
CALL PSAT(PS1,DP,T1)
IF(P1.GT.PS1*1.0001)GO TO 1
IF(P1.LT.PS1*.9999)GO TO 2
```

C
C TWO-PHASE FLOW
C

```
CALL DLSAT(D2,DD,T1)
CALL DVSAT(D1,DD,T1)
VNS=G1/D1+G2/D2
```

```

C      GO TO 3
C
C      LIQUID FLOW
C
1  CONTINUE
   CALL DLSAT(D2,DD,T1)
   D2=DENCR(P1,T1,D2)
   G2=G1+G2
   G1=0.0
   D1=0.0
   VNS=G2/D2
   GO TO 3
C
C      VAPOR FLOW
C
2  CONTINUE
   D1=1.0
   IF(T1.LE.647.2)CALL DVSAT(D1,DD,T1)
   D1=DENCR(P1,T1,D1)
   G1=G1+G2
   G2=0.0
   D2=0.0
   VNS=G1/D1
C
3  CONTINUE
   IF(VFR.LT.1.0)GO TO 80
   IND=0
   PRESS(KAR,1)=P1/PFTR
   DLIQ(KAR,1)=G2/VNS/DFTR
   DVAP(KAR,1)=G1/VNS/DFTR
   VGAS(KAR,1)=VNS/VFTR
   DO 4 I=1,3
4  IREGM(KAR,I)=0
   ISKIP(KAR,1)=0
C
   KN=KAR
C
C      TEST FOR TWO-PHASE FLOW
C
10 CONTINUE
   K=KN
   IND=0
   KN=K-1
   IF(IFLOW.EQ.1)KN=K+1
   IF(KN.EQ.0)RETURN
   IF(KN.GT.KDR)RETURN
   GG=9.80665
   DZ=(Z(K)-Z(KN))*VFTR
   IF(DZ.LT.0.0)GG=-GG
   DZ=ABS(DZ)

```

C
C
C

CHECK FOR AREA CHANGES

KT1=KTUBE(Z(K))-5
KT2=KTUBE(Z(KN))-5
IF(KT1.EQ.KT2)GO TO 12
RA=ATA(KT1,1)/ATA(KT2,1)
G1=G1*RA
G2=G2*RA

12 CONTINUE
AREA=ATA(KT2,1)*VFTR*VFTR
DHYD=2.*SQRT(AREA/PI)

C

T1=(TWJ(K,1)+459.67)/1.8
P1=PRESS(K,1)*PFTR
NCYC=4
ICYC=1
DZ=DZ/FLOAT(NCYC)

90 CONTINUE
IF(T1.GT.647.2)GO TO 52
CALL PSAT(P1,DPDT,T1)
GT=G1+G2
IF(G2.LT.GT*1.E-12.AND.P1.LT.1.001*PS1)GO TO 50
IF(IFLOW.EQ.1)GO TO 72
IF(P1.GT.PS1*1.001.AND.G1/GT.LE.1.E-12)GO TO 40
GO TO 74

72 IF(G1.LE.GT*1.E-12.AND.P1.GT.0.999*PS1)GO TO 40

74 CONTINUE
IF(P1.LT.PS1*.999.AND.G2.GT.0.0)GO TO 13
IF(P1.GT.PS1*1.001.AND.G1.GT.0.0)GO TO 14
GO TO 16

13 T1=TSAT(P1)
PS1=P1
GO TO 16

14 IF(P1.LT.2.212E7)T1=TSAT(P1)
IF(P1.GE.2.212E7)GO TO 42
PS1=P1

16 CONTINUE
IF(P1.GT.PS1*1.0001)GO TO 40
IF(P1.LT.PS1*.9999)GO TO 50

C TWO PHASE FLOW

C
C
C
C

INITIAL VELOCITIES

17 CALL DVSAT(D1,DD1,T1)
CALL DLSAT(D2,DD2,T1)
V1=G1/D1
V2=G2/D2
VNS=V1+V2

C

CALL ESAT(E1,E2,DE1,DE2,D1,D2,DD1,DD2,T1)

C
C
C

TEST FOR FLOW REGIME

ITRAN=0
GT=G1+G2
IF (G1/GT.LE.1.E-12)GO TO 20
IF (G2/GT.LE.1.E-6)GO TO 26
IF (V1.LE.1.E-12)GO TO 20
SRTN=SURTN(T1)
IF (SRTN.LE.1.E-6) SRTN=1.E-6
VTG=V1*SQRT(SQRT(D2/ABS(GG)/SRTN))
ELB=1.071-0.2218*VNS**2/DHYD/.3048
ELS=50.+36.*VTG*V2/V1
ELM=75.+84.*(VTG*V2/V1)**.75
IF (ELB.LT.0.13)ELB=0.13
IF (V1.LE.VNS*ELB)GO TO 20
IF (V1.GT.VNS*ELB.AND.ELS.GT.VTG)GO TO 22
IF (ELM.GT.VTG.AND.VTG.GT.ELS)ITRAN=1
IF (ITRAN.EQ.1)GO TO 22
GO TO 26

C
C
C
C
C

BALANCE OF MOMENTUM

BUBBLE FLOW

20 RE=D2*DHYD*V2/SLVIS(T1)
FRAC=FR2P(RE,WN,SN,DHYD,2)
RL=RLORK(V1,V2)
DPFR=.5*FRAC/DHYD*(V2/RL)**2*DZ
DPGR=GG*(RL*(D2-D1)+D1)*DZ
IFRG=2
GO TO 30

C
C
C

SLUG FLOW

22 GT=G1+G2
RE=D2*DHYD*VNS/SLVIS(T1)
FRAC=FR2P(RE,WN,SN,DHYD,2)
CALL GMORK(T1,DHYD,D2,GT,VNS,GAM,VR)
DPFR=.5*FRAC/DHYD*D2*VNS**2*((V2+VR)/(VNS+VR)+GAM)*DZ
DPGR=GG*((GT+D2*VR)/(VNS+VR)+D2*GAM)*DZ
IF (ITRAN.EQ.1)GO TO 24
IFRG=3
GO TO 30

C
C
C

TRANSITION FLOW

24 PSF=(ELM-VTG)/(ELM-ELS)
PMF=(VTG-ELS)/(ELM-ELS)
VOLFG=1./(1.+V2/V1)
RE=D1*DHYD*V1/SVVIS(T1)
SRTN=SURTN(T1)
IF(SRTN.LE.1.E-6)SRTN=1.E-6
WN=(V1*SLVIS(T1)/SRTN)**2*D2/D1
SN=SRTN/(D1*V1**2*DHYD)
FRAC=FR2P(RE,WN,SN,DHYD,3)
DPFR=PSF*DPFR+PMF*.5*FRAC/DHYD*D1*V1**2*DZ
DPGR=PSF*DPGR+PMF*(VOLFG*D1+(1.-VOLFG)*D2)*GG*DZ
IFRG=4
GO TO 30

C
C
C

MIST FLOW

26 VOLFG=1./(1.+V2/V1)
RE=D1*DHYD*V1/SVVIS(T1)
SRTN=SURTN(T1)
IF(SRTN.LE.1.E-6)SRTN=1.E-6
WN=(V1*SLVIS(T1)/SRTN)**2*D2/D1
SN=SRTN/(D1*V1*V1*DHYD)
FRAC=FR2P(RE,WN,SN,DHYD,3)
DPFR=.5*FRAC/DHYD*D1*V1**2*DZ
DPGR=(VOLFG*D1+(1.-VOLFG)*D2)*DZ*GG
IFRG=5

C

30 P1A=P1-DPFR-DPGR
IF(P1A.LE.0.0)GO TO 34
P1=P1A
IF(P1.GT.2.212E7)T2=647.2
IF(P1.LE.2.212E7)T2=TSAT(P1)
DT=T2-T1

C
C
C

BALANCE OF ENERGY

PRMX=V2/(V1+V2)
H2PH=HMIX(G1,G2,D1,D2,T1,PRMX,DHYD)
H2PH=H2PR(H2PH,DHYD,KT2,K,KN)
TANN=(TWJ(K,2)+TWJ(KN,2))/2.
IF(K.EQ.1.OR.KN.EQ.1)TANN=2.*TWJ(2,2)-TWJ(3,2)
TANN=(TANN+459.67)/1.8
QFLUX=4.*H2PH*DZ*(T1+DT/2.-TANN)/DHYD
DG2=-G2
IF(ABS(E2-E1).LT.0.0001)GO TO 32
DG2=(DT*(G1*DE1+G2*DE2)+GT*GG*DZ+QFLUX)/(E1-E2)

C

32 PF=1.0
IF(G1-DG2.LE.0.0)PF=G1/DG2
IF(G2+DG2.LE.0.0)PF=-G2/DG2

```

C
T1=T1+DT
G1=G1-DG2*PF
G2=G2+DG2*PF

C
CALL DVSAT(D1,DD1,T1)
CALL DLSAT(D2,DD2,T1)
V1=G1/D1
V2=G2/D2
VNS=V1+V2

C
IF(G1.LT.0.0)G1=0.0
IF(G2.LT.0.0)G2=0.0

C
TOLD=TWJ(KN,1)
TNEW=T1*1.8-459.67
TOLD=TNEW
TWJ(KN,1)=.8*TNEW+.2*TOLD
IND=1
GO TO 60

C
C FLASHING FLOW
C
34 IF(P1-DPFR.LE.0.0)GO TO 70
DPGR=D1*GG*DZ
P1=P1-DPGR-DPFR
IF(P1.LE.0.0)GO TO 70
DT=(-G2*(E1-E2)+QFLUX)/(G1*DE1+G2*DE2)
T1=T1+DT
TS=TSAT(P1)
IF(T1.LT.TS)T1=TS
IFRG=5
G1=G1+G2
G2=0.0
D1=DENCR(P1,T1,D1)
D2=0.0
VNS=G1/D1
GO TO 60

C
C LIQUID FLOW
C
40 D2=DLIQ(K,1)*DFTR
42 D1=0.0
IFRG=1
G2=G1+G2
G1=0.0
VNS=G2/D2
RE=G2*DHYD/SLVIS(T1)
FRAC=FR2P(RE,0.0,0.0,DHYD,2)
DPFR=.5*FRAC*G2*VNS*DZ/DHYD
DPGR=GG*D2*DZ
P1A=P1-DPFR-DPGR
IF(P1A.GT.0.0)GO TO 43
CALL DVSAT(D1,DD1,T1)
CALL DLSAT(D2,DD2,T1)
CALL ESAT(E1,E2,DE1,DE2,D1,D2,DD1,DD2,T1)

```

```

GO TO 34
43 T2=(TWJ(KN,1)+459.67)/1.8
IF(T2.GT.647.2)GO TO 44
CALL PSAT(PS2,DPS2,T2)
IF(P1A.LT.PS2)GO TO 44
T1=T2
P1=P1A
CALL DLSAT(D2S,DD,T1)
IF(D2.LT.D2S)D2=D2S
D2=DENCR(P1,T1,D2)
VNS=G2/D2
GO TO 60
44 P1B=PS1*(P1A-P1)-P1*(PS2-PS1)
P1B=P1B/(P1A-P1+PS1-PS2)
DZ=DZ*(P1A-P1B)/(P1A-P1)
P1=P1B
T1=TSAT(P1)
PS1=P1
GO TO 17

```

C
C
C

VAPOR FLOW

```

50 D1=DVAP(K,1)*DFTR
GO TO 54
52 D2=0.0
G1=G1+G2
G2=0.0
IF(D1.LE.0.0)D1=P1/461.50/T1
D1=DENCR(P1,T1,D1)
54 VNS=G1/D1
IFRG=6
RE=G1*DHYD/SVVIS(T1)
FRAC=FR2P(RE,0.0,0.0,DHYD,2)/DHYD
DPFR=.5*FRAC*G1*VNS
DPGR=GG*D1*DZ
P1A=P1
P1=P1-DPFR-DPGR
IF(P1.LE.0.0)GO TO 70
T1=(TWJ(KN,1)+459.67)/1.8
D1=DENCR(P1,T1,D1)
VNS=G1/D1
IF(T1.GT.647.2)GO TO 60
CALL PSAT(PS2,DPS2,T1)
IF(P1.LE.PS2)GO TO 60
T1=TSAT(P1A)
PS1=P1A
GO TO 17

```

```

C
C STORE CALCULATED VARIABLES
C
  60 ICYC=ICYC+1
    IF(ICYC.LE.NCYC)GO TO 90
    DLIQ(KN,1)=G2/VNS/DFTR
    DVAP(KN,1)=G1/VNS/DFTR
    DO 61 LL=1,3
  61 IREGM(KN,LL)=IFRNM(LL,IFRG)
    VGAS(KN,1)=VNS/VFTR
    PRESS(KN,1)=P1/PFTR
    ISKIP(KN,1)=1
    IF(IFRG.EQ.1)ISKIP(KN,1)=0
    IF(IFRG.EQ.6)ISKIP(KN,1)=0

C
    GO TO 10

C
C ERROR HANDLING
C
  70 P1=P1/PFTR
    DPFR=DPFR/PFTR
    DPGR=DPGR/PFTR
    TF=T1*1.8-459.67
    WRITE(6,100)K,KN,(IFRNM(KK,IFRG),KK=1,3),P1,TF,DPFR,DPGR
100 FORMAT(/' TWO-PHASE PRESSURE LESS THAN ZERO'/
& ' FROM K=',I3,' TO K=',I3,' FLOW TYPE= ',3A2/
& ' INITIAL PRESSURE=',F6.2,' PSIA   TEMPERATURE=',F6.1,' F'/
& ' FRICTIONAL PRESSURE DROP=',F8.2,' PSIA'/
& ' GRAVITATIONAL PRESSURE DROP=',F8.2,' PSIA')
    STOP

C
C SHUT-IN TWO PHASE
C
  80 PRESS(KAR,1)=P1/PFTR
    DLIQ(KAR,1)=D2*(1.-QUALS)/DFTR
    DVAP(KAR,1)=D1*QUALS/DFTR
    VGAS(KAR,1)=0.0
    IF(KAR.EQ.KDR)GO TO 85

C
C INJECTION - SURFACE PRESS DEFINED
C
    DD=D1+D2
    DO 82 I=2,KDR
    DZ=(Z(I)-Z(I-1))*VFTR
    P1=P1+9.80665*DD*DZ
    T1=(TWJ(I,1)+459.67)/1.8
    DD=DENCR(P1,T1,DD)
    IF(T1.GT.647.2)GO TO 83
    CALL PSAT(PS1,DP,T1)
    IF(P1.LE.PS1)GO TO 83
    DLIQ(I,1)=DD/DFTR
    DVAP(I,1)=0.0
    IFRG=1
    GO TO 84
  83 DVAP(I,1)=DD/DFTR
    DLIQ(I,1)=0.0

```

```

      IFRG=6
84  PRESS(I,1)=P1/PFTR
      VGAS(I,1)=0.0
      DO 82 K=1,3
82  IREGM(I,K)=IFRNM(K,IFRG)
      RETURN

```

```

C
C  PRODUCTION - BOTTOM HOLE PRESSURE DEFINED
C

```

```

85  DD=D1+D2
      ISPR=0
      DO 86 II=2,KDR
          I=KDR-II+1
          ISKIP(I,1)=0
          DZ=(Z(I+1)-Z(I))*VFTR
          DPGR=9.80665*DD*DZ
          T1=(TWJ(I,1)+459.67)/1.8
          PS1=1.E12
          IF(T1.GT.647.2)GO TO 89
          CALL PSAT(PS1,DP,T1)
          IF(P1.GT.PS1.AND.P1-DPGR.GT.PS1)GO TO 89
          DPGR=P1-PS1
          ISPR=1
          CALL DVSAT(DD,DPD,T1)
89  P1=P1-DPGR
          DD=DENCR(P1,T1,DD)
          IF(ISPR.EQ.1)GO TO 87
          IF(P1.LE.PS1)GO TO 87
          DLIQ(I,1)=DD/DFTR
          DVAP(I,1)=0.0
          IFRG=1
          GO TO 88
87  DVAP(I,1)=DD/DFTR
          DLIQ(I,1)=0.0
          TWJ(I,1)=TSAT(P1)*1.8-459.67
          ISKIP(I,1)=1
          IFRG=6
88  PRESS(I,1)=P1/PFTR
          VGAS(I,1)=0.0
          DO 86 K=1,3
86  IREGM(I,K)=IFRNM(K,IFRG)
          RETURN
      END

```

C
SUBROUTINE PSAT(P,DP,T)

DIMENSION F(8)

DATA F/-741.9242,-29.72100,-11.55286,-0.8685635,.1094098,
@ .439993,.2520658,.05218684/
DATA TC,PC/647.3,22120.0E3/

C
T1=3.382-0.01*T

TR=TC/T

F1=F(1)+F(2)*T1

F2=F(2)

DO 1 I=3,8

F1=F1+F(I)*T1**(I-1)

1 F2=F2+FLOAT(I-1)*F(I)*T1**(I-2)

P=PC*EXP(.01*(TR-1.0)*F1)

DP=-.01*(TR/T*F1+.01*(TR-1.0)*F2)*P

RETURN

END

FUNCTION TSAT(P)

C
C
C
C

COMPUTES SATURATION TEMPERATURE IN KELVINS AS A FUNCTION
OF PRESSURE IN PASCALS

DIMENSION B(5)

DATA B/1.0158658,.53542626,.070704624,-.26191199,0.10003160/

DATA A/-.45800227/

C

IF(P.LE.0.0)GO TO 2

AP=ALOG(P)/ALOG(22.120E6)

F=A

DO 1 I=1,5

F=F+B(I)*AP**I

1 CONTINUE

TSAT=472.5944/(1.73010-F)

N=0

3 CALL PSAT(PS,DPS,TSAT)

DP=P-PS

IF(ABS(DP/P).LE.1.E-5)GO TO 4

DT=DP/DPS

TSAT=TSAT+DT

N=N+1

IF(N.GT.10)GO TO 4

GO TO 3

4 RETURN

2 TSAT=100.

RETURN

END

SUBROUTINE DVSAT(D,DD,TDUM)

C

```
T=TDUM
IF(T.GE.647.0)GO TO 1
IF(T.LT.273.15)T=273.16
TR=(T-273.15)/374.12
IF(TR.GT.1.0)TR=1.0
R1=460.330
R2=107.779
X=2.6-.6*TR
TRX=TR**X
IF(TRX.GT.1.0)TRX=.99999
R=SQRT(1.0-TRX)*(R1-R2)+R2
CALL PSAT(P,DP,T)
D=P/R/T
DR=.5*(R1-R2)/SQRT(1.-TRX)*TRX*(X/TR-.6*ALOG(TR))/374.12
DD=D*(DP/P+DR/R-1./T)
RETURN
1 DD=0.0
D=317.
RETURN
END
```

SUBROUTINE DLSAT(D,DD,T)

C

DATA T1,T2/273.15,647.29/
DATA D1,D2/1000.0,317.00903/
DATA X1,X2/1.6160,.40873/

C

IF(T.GE.647.0)GO TO 1
TR=(T-T1)/(T2-T1)
IF(TR.LE.0.0)TR=.00001
X=X1*EXP(X2*TR**5)
TRX=TR**X
SQX=SQRT(1.-TRX)
D=D2+(D1-D2)*SQX
DD=.5*(D2-D1)/(T2-T1)/SQX*X*TRX*(ALOG(TR)*X2*5.*TR**4+1./TR)
RETURN

1 D=317.0
DD=0.0
RETURN
END

```
SUBROUTINE ESAT(E1,E2,DE1,DE2,D1,D2,DD1,DD2,T)
CALL STEAM(P,DPD,DPT,E1,DED,DET,D1,T)
E1=E1+P/D1
DP=0.0
IF(T.LE.647.2)CALL PSAT(P,DP,T)
DE1=DED*DD1+DET+DP/D1-P*DD1/D1**2
CALL STEAM(P,DPD,DPT,E2,DED,DET,D2,T)
DP=0.0
IF(T.LE.647.2)CALL PSAT(P,DP,T)
E2=E2+P/D2
DE2=DED*DD2+DET+DP/D2-P*DD2/D2**2
RETURN
END
```

FUNCTION SVVIS(T)

```
C
C SVVIS = SATURATED VAPOR VISCOSITY PA-S
C T = TEMP KELVIN
C TC = TEMP CELSIUS
C D = DENSITY GM/CC
C
  TC=T-273.15
  V1=.407*TC+80.4
  IF(TC.GT.300.)GO TO 1
  CALL DVSAT(D,DD,T)
  D=D/1000.
  V2=-D*(1858.0-5.9*TC)
  SVVIS=(V1+V2)*1.E-7
  RETURN
1 CALL DVSAT(D1,DD,573.15)
  CALL DVSAT(D2,DD,648.15)
  D2=D2/1000.
  VV=-D1*88./1000.
  VC=353.*D2+676.5*D2*D2+107.1*D2**3
  SVVIS=(V1+VV+(VC-VV)*(TC-300.)/75.)*1.E-7
  RETURN
END
```

FUNCTION SLVIS(T)

C
C
C
C
C

SLVIS = SAT LIQ VISCOSITY PA-S
T = TEMP KELVIN
D = DENS KG/M3

TK=T

IF(T.GT.573.15)TK=573.15

SLVIS=241.4*10**(247.8/(T-140.))*1.E-7

IF(T.LE.573.15)RETURN

CALL DLSAT(D3,DD,573.15)

CALL DLSAT(DL,DD,T)

CALL DVSAT(DV,DD,T)

SLVIS=SVVIS(T)*(D3-DL)/(D3-DV)+SLVIS*(DL-DV)/(D3-DV)

RETURN

END

SUBROUTINE KSAT(CON,T,IND)

```

C
C IND = 1 VAPOR
C IND = 2 LIQUID
C CON WATT/M-K
C T KELVIN
C
      IF(IND.EQ.1)GO TO 10
      IF(IND.EQ.2)GO TO 20
      WRITE(6,100)
100  FORMAT('KSAT ERROR IND OUT OF RANGE')
      STOP
10  CONTINUE
      TC=T-273.15
      CON1=17.6+5.87E-2*TC+1.04E-4*TC*TC-4.51E-8*TC**3
      IF(T.LE.647.2)CALL DVSAT(D,DD,T)
      IF(T.GT.647.3)D=317.0
      D=D/1000.
      IF(T.GT.647.2)GO TO 30
      CALL PSAT(PS,DP,T)
      IF(PS.GT.5.E7)GO TO 30
      IF(TC.LE.0.0)TC=0.01
      CON=(CON1+(103.51+0.4198*TC-2.771E-5*TC*TC)*D+
& 2.1482E14*D*D/TC**4.2)/1000.
      RETURN
20  CONTINUE
      TA=T/273.15
      CON=-922.47+2839.5*TA-1800.7*TA*TA+525.77*TA**3-73.440*TA**4
      CON=CON/1000.
      RETURN
30  CON=39.44895+750.64429*D-564.34229*D*D+786.34375*D**3
      CON=CON/1000.
      RETURN
      END

```

FUNCTION SURTN(T)

C
C
C
C
C
C
C

SURFACE TENSION FOR WATER

SURTN = SURFACE TENSION N/M2

T = TEMP KELVIN

VIS = VISCOSITY PA-S

IF(T.GT.647.28)GO TO 1

DVISC=(SLVIS(T)-SVVIS(T))*1.E6

IF(DVISC.LE.1.E-3)DVISC=1.E-3

SURTN=78.609*EXP(-77.225/DVISC)*.10

RETURN

1 SURTN=0.

RETURN

END

SUBROUTINE STEAM(P,DPD,DPT,U,DUD,DUT,DSI,T)

DIMENSION A(7,10),C(8),AJ(7),DAJ(7),D2AJ(7)

DATA A/29.492937,-5.1985860,6.8335354,-0.1564104,-6.3972405,
@ -3.9661401,-0.69048554,
@ -132.13917,7.7779182,-26.149751,-0.72546108,26.409282,
@ 15.453061,2.7407416,
@ 274.64632,-33.301902,65.326396,-9.2734289,-47.740374,
@ -29.142470,-5.1028070,
@ -360.93828,-16.254622,-26.181978,4.3125840,56.323130,
@ 29.568796,3.9636085,
@ 342.18431,-177.31074,0.0,0.0,0.0,0.0,0.0,
@ -244.50042,127.48742,0.0,0.0,0.0,0.0,0.0,
@ 155.18535,137.46153,0.0,0.0,0.0,0.0,0.0,
@ 5.9728487,155.97836,0.0,0.0,0.0,0.0,0.0,
@ -410.30848,337.31180,-137.46618,6.7874983,136.87317,
@ 79.847970,13.041256,
@ -416.05860,-209.88866,-733.96848,10.401717,645.81880,
@ 399.17570,71.531353/
DATA C/1857.065,3229.12,-419.465,36.6649,-20.5516,4.85233,
@ 46.000,-1011.249/

DATA TAC,E/1.544912,4.800/
DATA R/.46151/

TA=1000./T
D=DSI/1000.
X=EXP(-E*D)

DM=D-0.634
AJ(1)=A(1,1)+A(1,2)*DM
DAJ(1)=A(1,2)
D2AJ(1)=0.0
DO 10 I=3,8
AJ(1)=AJ(1)+A(1,I)*DM**(I-1)
D2AJ(1)=D2AJ(1)+FLOAT((I-1)*(I-2))*DM**(I-3)
10 DAJ(1)=DAJ(1)+A(1,I)*FLOAT(I-1)*DM**(I-2)
FX=X*(A(1,9)+A(1,10)*D)
AJ(1)=AJ(1)+FX
DAJ(1)=DAJ(1)+X*A(1,10)-E*FX
D2AJ(1)=D2AJ(1)+E*E*FX-2.*E*X*A(1,10)

DM=D-1.0
DO 11 J=2,7
AJ(J)=A(J,1)+A(J,2)*DM
DAJ(J)=A(J,2)
D2AJ(J)=0.0
DO 12 I=3,8
AJ(J)=AJ(J)+A(J,I)*DM**(I-1)
D2AJ(J)=D2AJ(J)+FLOAT((I-1)*(I-2))*DM**(I-3)
12 DAJ(J)=DAJ(J)+A(J,I)*FLOAT(I-1)*DM**(I-2)
FX=X*(A(J,9)+A(J,10)*D)
AJ(J)=AJ(J)+FX
D2AJ(J)=D2AJ(J)+E*E*FX-2.*E*X*A(J,10)
11 DAJ(J)=DAJ(J)+X*A(J,10)-E*FX

C

```
TMC=TA-TAC
TM=TA-2.5
Q=AJ(1)
D2QDD=D2AJ(1)
DQD=DAJ(1)
DO 14 J=2,7
Q=Q+TMC*AJ(J)*TM**(J-2)
D2QDD=D2QDD+TMC*D2AJ(J)*TMC**(J-2)
14 DQD=DQD+TMC*DAJ(J)*TM**(J-2)
```

C

```
DQT=0.0
D2QDT=0.0
DO 15 J=2,7
D2QDT=D2QDT+DAJ(J)*TM**(J-3)*(TM+TMC*FLOAT(J-2))
15 DQT=DQT+AJ(J)*TM**(J-3)*(TM+TMC*FLOAT(J-2))
```

C

```
D2QTT=2.*AJ(3)+(2.*TM+2.*TMC)*AJ(4)
DO 16 J=5,7
16 D2QTT=D2QTT+(2.*TM**(J-3)+TMC*TM**(J-4)*FLOAT((J-2)*(J-3)))*AJ(J)
```

C

```
DPSI=C(1)+C(7)*(ALOG(T)-1.0)-C(8)/TA
D2PSI=-C(7)*TA+C(8)
DO 4 I=3,6
D2PSI=D2PSI+C(I)*FLOAT((2-I)*(1-I))*TA**(2-I)
4 DPSI=DPSI+C(I)*FLOAT(2-I)*TA**(1-I)
D2PSI=-D2PSI/1000.
```

C

```
P=D*R*T*(1.0+D*Q+D*D*DQD)*1.E6
DPD=(R*T+D*R*T*(2.*Q+4.*D*DQD+D*D*D2QDD))*1000.0
DPT=D*R+D*D*R*(Q-TA*DQT+D*DQD-D*TA*D2QDT)*1.E6
U=(R*T*D*TA*DQT+DPSI)*1000.
DUT=(-R*D*TA*TA*D2QTT+D2PSI)*1000.0
DUD=(R*DQT+R*D*D2QDT)*1000.0
RETURN
END
```

FUNCTION DENCRC(P,T,D)

C
C COMPUTES DENSITY CONSISTENT WITH P,T
C DENSITY NEAR D TO INITIALIZE
C
M=0
N=0
CALL STEAM(PTEST,X1,X2,X3,X4,X5,D,T)
IF(ABS(P-PTEST)/P.LT. 1.E-3)GO TO 1
IF(PTEST.GT.P)GO TO 2

C
C PTEST<P
C
D1=D
P1=PTEST
D2=D1
4 D2=D2*1.10
IF(M.GT.50)D2=D2*1.50
9 CALL STEAM(P2,X1,X2,X3,X4,X5,D2,T)
IF(P2.GT.P)GO TO 3
M=M+1
IF(M.GT.200)GO TO 8
D1=D2
P1=P2
GO TO 4

C
C PTEST>P
C
2 D2=D
P2=PTEST
D1=D2
5 D1=D1*.90
IF(M.GT.50)D1=D1*.50
M=M+1
IF(M.GT.200)GO TO 8
CALL STEAM(P1,X1,X2,X3,X4,X5,D1,T)
IF(P1.LT.P)GO TO 3
D2=D1
P2=P1
GO TO 5

C
C SPLIT INTERVAL
C
3 D=D2-(D2-D1)*(P2-P)/(P2-P1)
CALL STEAM(P3,X1,X2,X3,X4,X5,D,T)
N=N+1
IF(N.GT.100)GO TO 7
IF(ABS(P-P3)/P.LT.1.E-3)GO TO 1
IF(ABS(D-D1)/D.LT.1.E-5)GO TO 1
IF(P3.GT.P)GO TO 6
P1=P3
D1=D
GO TO 3
6 P2=P3
D2=D
GO TO 3

C

```
8 D=.5*(D1+D2)
7 WRITE(6,100)T,D,D1,D2,P,P1,P2
100 FORMAT(' WARNING-DENSITIES UNRELIABLE: T=',F6.0/
& ' D=',E11.4,' D1=',E11.4,' D2=',E11.4/
& ' P=',E11.4,' P1=',E11.4,' P2=',E11.4)
1 DENC=D
RETURN
END
```

FUNCTION FR2P(RE,WN,SN,DHYD,IK)

C
C Darcy Friction Factor for Two Phase Flow
C
C IK = 1 Commercial Steel
RR=.0000457/DHYD
C
C IK = 2 Galvanized Iron with Scale
IF(IK.EQ.2)RR=.000456/DHYD
C
C IK = 3 Dunn and Ros Mist Flow E/D
IF(IK.NE.3)GO TO 1
IF(WN.GT.0.005)RR=174.8*SN*WN**0.302
IF(WN.LE.0.005)RR=34.*SN
IF(RR.GT.0.5)RR=0.5
IF(RR.LT.1.E-3)RR=1.E-3
1 CONTINUE
A=.026*RR**.225+0.133*RR
B=22.*RR**0.44
C=-1.62*RR**0.134
FRIC=A+B*RE**C
FR2P=4.*FRIC
FRICL=64./RE
IF(FRICL.GT.FR2P)FR2P=FRICL
RETURN
END

FUNCTION RLORK(V1,V2)

C
C
C

ORKIZEWSKI'S RL FUNCTION FOR BUBBLE FLOW

```
VNS=V1+V2
VBUB=.2438
VNR=VNS/VBUB+1.
V1R=V1/VBUB
RAD=VNR*VNR-4.*V1R
IF(RAD.LE.1.E-12)RAD=1.E-12
EG=.5*(VNR-SQRT(RAD))
RLORK=1.-EG
IF(RLORK.LE.1.E-12)RLORK=1.E-12
RETURN
END
```

SUBROUTINE GMORK (T, DHYM, D2M, GTM, VNSM, GAM, VR)

```

C
C THIS SUBROUTINE CALCULATES ORKIZEWSKI'S GAMMA FUNCTION
C AND BUBBLE RISE VELOCITY VR
C
C T = TEMP KELVIN
C DHYM = HYDRAULIC DIAMETER M
C D2M = LIQUID DENSITY KG/M3
C GTM = TOTAL MASS FLUX DENSITY KG/M2-S
C VNSM = NO SLIP VELOCITY M/S
C
C CALCULATE ENGLISH UNIT EQUIVALENTS
C
    DHYD=DHYM/.3048
    D2=D2M/16.01845
    GT=GTM/4.882428
    VNS=VNSM/.3048
C
    VIS=SLVIS(T)*1000.0
    RE=1488.*D2*DHYD*VNS/VIS
    ENB=1488.*D2*DHYD/VIS
    D12=SQRT(DHYD)
C
C COMPUTE BUBBLE RISE VELOCITY
C
    VB=5.6745*D12
    VB1=VB*(0.546+8.74E-6*RE)
    VB2=VB*(0.350+8.74E-6*RE)
    ALF=VB*(0.251+8.74E-6*RE)
    VB3=.5*ALF+SQRT(ALF*ALF+13.59*VIS/D2/D12)
    VR=VB1
    IF (VB2*ENB.GE.8000.0) VR=VB2
    IF (VB3*ENB.GT.3000.0.AND.VB3*ENB.LT.8000.0) VR=VB3
    VISLG=ALOG10(VIS)
    DHLG=ALOG10(DHYD)
    IF (VNS.GT.10.0) GO TO 1
    GAM=0.0130*VISLG/DHYD**(1.38)-0.681
    & +0.232*ALOG10(VNS)-0.428*DHLG
    IF (GAM.LE.-0.065*VNS) GAM=-0.065*VNS
    GO TO 2
1  GAM=0.045*VISLG/DHYD**(0.799)-0.709-0.888*DHLG
    & -0.162*ALOG10(VNS)
    GTEST=VR*(GT/D2-VNS)/(VR+VNS)/VNS
    IF (GAM.LT.GTEST) GAM=GTEST
2  VR=VR*.3048
    RETURN
    END

```

```

      FUNCTION HMIX(G1,G2,D1,D2,T1,P2,DHYD)
C
C   FILM COEFFICIENT DETERMINED BY SIMILARITY ANALYSIS
C
C   CONMX = THERMAL CONDUCTIVITY OF MIXTURE
C   CPMIX = HEAT CAPACITY OF MIXTURE
C   GMIX  = MASS FLUX DENSITY OF THE MIXTURE
C   VSMIX = VISCOSITY OF THE MIXTURE
C
      CALL KSAT(CON1,T1,1)
      CALL KSAT(CON2,T1,2)
      CALL STEAM(X1,X2,X3,X4,X5,DE1,D1,T1)
      CALL STEAM(X1,X2,X3,X4,X5,DE2,D2,T1)
      P1=1.-P2
C
      CONMX=P1*CON1+P2*CON2
      CPMIX=P1*DE1+P2*DE2
      GMIX=G1+G2
      VSMIX=P1*SVVIS(T1)+P2*SLVIS(T1)
C
      RE=GMIX*DHYD/VSMIX
      PR=VSMIX*CPMIX/CONMX
C
      HMIX=0.023*CONMX*RE**0.80*PR**0.35/DHYD
      RETURN
      END

```

FUNCTION H2PR(H2PH,DHYD,KT2P,K,KN)

THIS FUNCTION COMPUTES THE OVERALL CONDUCTIVITY
BETWEEN THE TWO-PHASE STEAM AND THE CENTER OF THE
TUBING-CASING ANNULUS

H2PR = WATT/M-K

COMMON /BLK1/ TW(151,20),TWJ(151,21)
COMMON /BLK3/ KDR,NZ,NZP1,TD,Z(151),ZC(15),DZ(15),TVD,DDVN
COMMON /BLK8/ NR,RMAX,R(21),DR(21),AR(20),RI(15),RO(15)
COMMON /BLK10/ DENST,SHST,CONST,DENC,SHC,CONC
COMMON /BLK13/ DENFP(5),PVFP(5),YPFP(5),SHFP(5),CONFP(5),
@IPF,IAF,ISF

KTP=KT2P+5
R1=RI(KTP)
R2=RO(KTP)
R3=R(2)
A21=ALOG(R2/R1)
A32=ALOG(R3/R2)
CON21=CONST*1.730735

IF(.5*(Z(K)+Z(KN)).GT.ZC(KTP)-DZ(KTP))GO TO 10

CONVECTING FLUID IN TUBING-CASING ANNULUS

T1=.5*(TW(K,1)+TW(KN,1))
T2=.5*(TW(K,2)+TW(KN,1))
VISA=VISC(T1,T2,R1,R2,DENFP(IAF),PVFP(IAF),YPFP(IAF))
HANN=CONVN(VISA,DENFP(IAF),SHFP(IAF),CONFP(IAF),R2,R3,T1,T2)
HANN=HANN*1.730735
R1=DHYD/2.

H2PR=H2PH/(1.+H2PH*R1*(A21/CON21+A32/HANN))
RETURN

CEMENT IN ANNULUS

10 CON32=CONC*1.730735
R1=DHYD*.5
H2PR=H2PH/(1.+H2PH*R1*(A21/CON21+A32/CON32))
RETURN
END



5. WELLBORE HEAT TRANSFER

The wellbore heat transfer calculations are based on the computational scheme developed for the original GEOTEMP. This scheme consists of three parts. The first part is the computation of the wellbore conductivity in subroutine COND. COND calls the following subprograms:

- a. SUBROUTINE FPROP-This subroutine determines the fluid properties at a given location in the wellbore.
- b. SUBROUTINE GPROP-This subroutine determines the gas properties at a given location in the wellbore.
- c. FUNCTION KTUBE-This function subroutine determines the tubing size at a given location in the wellbore.
- d. SUBROUTINE CONV-This subroutine determines the film coefficient for forced convection in a tube or annulus.
- e. SUBROUTINE CONVN-This subroutine determines the film coefficient for natural convection in an annulus.
- f. FUNCTION HTUBE-This function subroutine determines the film coefficient for natural convection in a tube.
- g. SUBROUTINE CONAN-This subroutine determines the overall conductivity of the wellbore tubulars and annuli, excluding the tubing and tubing annulus.
- h. FUNCTION CA-This function subroutine determines average properties for an annulus with a cement-packer fluid interface.
- i. FUNCTION VISC-This function subroutine determines the fluid viscosity for use in CONV, CONVN, and HTUBE.

The second part of the calculation is the determination of the coefficients in the finite difference heat transfer equations. These calculations are executed in subroutine COEF. The final part of the heat transfer calculation is the determination of the new temperature field at the end of a time step. This calculation is done in subroutine WELL.

The principal change in these subroutines is the integration of the other flowing conditions in gas flow and two-phase steam flow. The operation of these subroutines, with the exception of these minor changes, can be found in reference 10.

SUBROUTINE COND

C
C
C

THIS SUBROUTINE DEFINES THE CONDUCTANCES BETWEEN CELLS.

```
COMMON /BLK1/ TW(151,20),TWJ(151,21)
COMMON /BLK3/ KDR,NZ,NZP1,TD,Z(151),ZC(15),DZ(15),TVD,DDVN
COMMON /BLK8/ NR,RMAX,R(21),DR(21),AR(20),RI(15),RO(15)
COMMON /BLK4/ U(151,20),Q(151,2)
COMMON /BLK6/ IFLOW,PI,DT,TIN1,TIN2,FR1,FR2,FR3,NIT
COMMON /BLK9/ DENS(150),SHS(150),CONS(150)
COMMON /BLK10/ DENST,SHST,CONST,DENC,SHC,CONC
COMMON /BLK11/ VFR,DDCHG,COSDVN
COMMON /BLK13/ DENFP(5),PVFP(5),YPFP(5),SHFP(5),CONFP(5),
@IPF,IAF,ISF
COMMON /BLK5/ CAN(5)
COMMON /BLK16/ NPF,NTS,KZP
COMMON /BLK19/ DEPTH,BITR,BITA,DCLRL,DCLRI,DCLRO,PSTAND
COMMON /BLK21/ IREGM(151,3),QUALS,PSTEM,I2PH,ISKIP(151,2)
DIMENSION RTI(10,2)
```

C
C
C

RESISTANCE TERMS

```
DO 1 I=1,NTS
RTI(I,1) = ALOG(RO(I+5)/RI(I+5))/CONST
1 RTI(I,2) = ALOG(R(2)/RO(I+5))
RT3=ALOG(RI(2)/R(2))
RT4 = ALOG(RO(2)/RI(2))/CONST
RT5 = ALOG(RI(3)/RO(2))
RT6 = ALOG(RO(3)/RI(3))/CONST
RT7 = ALOG(RI(4)/RO(3))
RT8 = ALOG(RO(4)/RI(4))/CONST
RT9 = ALOG(RI(5)/RO(4))
RT10 = ALOG(RO(5)/RI(5))/CONST
RT11 = ALOG(R(3)/RO(5))
RT12 = ALOG(R(3)/RI(5))
RT13 = ALOG(R(3)/RI(4))
RT14 = ALOG(R(3)/RI(3))
```

C
C
C

COMPUTE WELLBORE CONDUCTANCES

```
DO 22 K=1,KDR
ZKP = Z(KDR)
IF (K .LT. KDR) ZKP = (Z(K) + Z(K+1))/2.
ZKM = Z(1)
IF (K .GT. 1) ZKM = (Z(K) + Z(K-1))/2.
```

C
C
C

DETERMINE CASING STRINGS THROUGH CELL AT Z(K)

```
J=5
26 IF (ZKP .LT. ZC(J)+0.1) GO TO 28
J=J-1
IF(J.EQ.1)GO TO 28
GO TO 26
28 II = 0
```

C
C
C

AVERAGE TUBING GEOMETRY

```
KZP=KTUBE(ZKP)-5
RT1=RTI(KZP,1)
RT2=RTI(KZP,2)
RIK=RI(KZP+5)
ROK=RO(KZP+5)
KZM=KTUBE(ZKM)-5
IF(KZP.EQ.KZM)GO TO 2
ZMM=ZC(KZM+5)
PF1=(ZKP-ZMM)/(ZKP-ZKM)
PF2=(ZMM-ZKM)/(ZKP-ZKM)
RT1=RT1*PF1+PF2*RTI(KZM,1)
RT2=RT2*PF1+PF2*RTI(KZM,2)
RIK=PF1*RIK+PF2*RI(KZM+5)
ROK=PF1*ROK+PF2*RO(KZM+5)
2 CONTINUE
TMT=TWJ(K,1)
TMA=TWJ(K,2)
TMM=TMA*(R(1)-RIK)/(R(1)-R(2))+TMT*(RIK-R(2))/(R(1)-R(2))
```

C
C
C

SUM RESISTANCES

```
36 R12 = RT1
R23=1.E-20
GO TO (10,12,14,16,18),J
10 CALL CONAN(2,K,ZKP,ZKM)
IF (DEPTH .GT. ZC(1)) R23 = RT14/CONS(K)
GO TO 20
12 CALL CONAN(2,K,ZKP,ZKM)
IF (DEPTH .GT. ZC(3)) R23 = RT14/CONS(K)
GO TO 20
14 CALL CONAN(3,K,ZKP,ZKM)
IF (DEPTH .GT. ZC(4)) R23 = RT13/CONS(K)
IF (DEPTH .GT. ZC(3)) R23 = R23 + RT7/CAN(3) + RT6
GO TO 20
16 CALL CONAN(4,K,ZKP,ZKM)
IF (DEPTH .GT. ZC(5)) R23 = RT12/CONS(K)
IF (DEPTH .GT. ZC(4)) R23 = R23 + RT9/CAN(4) + RT8
IF (DEPTH .GT. ZC(3)) R23 = R23 + RT7/CAN(3) + RT6
GO TO 20
18 CALL CONAN(5,K,ZKP,ZKM)
R23 = 1.E-20
IF (DEPTH .GT. ZC(5)) R23 = RT11/CAN(5) + RT10
IF (DEPTH .GT. ZC(4)) R23 = R23 + RT9/CAN(4) + RT8
IF (DEPTH .GT. ZC(3)) R23 = R23 + RT7/CAN(3) + RT6
20 IF (DEPTH .GT. ZC(2)-1.) R23 = R23 + RT5/CAN(2) + RT4
```

C

```
IF(IFLOW.GT.4)GO TO 200
IF(VFR.LT.1.0)GO TO 100
IF(I2PH.GT.0)GO TO 310
```

```

C
C TUBING CIRCULATION-LIQUIDS
C
  CALL FPROP(DENF,PVF,YPF,SHF,CONF,ZKP,ZKM,1)
  VIS=VISC(TMT,0.,0.,RIK,VFR,PVF,YPF)
  H1=CONV(VFR,VIS,DENF,SHF,CONF,0.,RIK)
  IF(IFLOW.LE.2)GO TO 300
C
C ANNULUS CIRCULATION-LIQUIDS
C
  CALL FPROP(DENF,PVF,YPF,SHF,CONF,ZKP,ZKM,2)
  VIS=VISC(TMA,0.,ROK,RI(2),VFR,PVF,YPF)
  H2=CONV(VFR,VIS,DENF,SHF,CONF,ROK,RI(2))
  GO TO 41
C
C SHUT-IN TUBING-LIQUIDS
C
100 CALL FPROP(DENF,PVF,YPF,SHF,CONF,ZKP,ZKM,1)
  VIS=VISC(TMT,TMM,0.,RIK,DENF,PVF,YPF)
  DZL=ZKP-ZKM
  H1=HTUBE(TMT,TMM,DZL,DENF,VIS,CONF,SHF,0)
  IF(IFLOW.LE.2)GO TO 300
C
C SHUT-IN ANNULUS-LIQUIDS
C
  CALL FPROP(DENF,PVF,YPF,SHF,CONF,ZKP,ZKM,2)
  VIS=VISC(TMA,TMM,ROK,R(2),DENF,PVF,YPF)
  CNA=CONVN(VIS,DENF,SHF,CONF,ROK,R(2),TMA,TMM)
  GO TO 42
C
C ANNULUS PRODUCTION-INJECTION CASES
C
300 CNA=CAN(1)
  GO TO 42
C
C COMPRESSIBLE FLOW-TUBING
C
200 CALL GPROP(DENT,SHT,SHTCV,CONT,VIST,VFT,K,1)
  CALL GPROP(DENA,SHA,SHACV,CONA,VISA,VFA,K,2)
  VFT=VFT*PI*RIK**2*3600.0
  VFA=VFA*PI*(RI(2)**2-ROK**2)*3600.0
  IF(VFT.LT.1.0)GO TO 210
  H1=CONV(VFT,VIST,DENT,SHT,CONT,0.,RIK)
C
C COMPRESSIBLE FLOW-ANNULUS
C
  H2=CONV(VFA,VISA,DENA,SHA,CONA,ROK,RI(2))
  GO TO 41
C
C COMPRESSIBLE CONVECTION
C
210 DZL=ZKP-ZKM
C
C TUBING CONVECTION
C
  H1=HTUBE(TMT,TMM,DZL,DENT,VIST,CONT,SHT,1)

```

C ANNULUS CONVECTION

C

CNA=CONVN(VISA,DENA,SHA,CONA,ROK,R(2),TMA,TMM)
GO TO 42

C

C TWO-PHASE CONVECTION

C

310 CALL GPROP(DENT,SHT,SHTCV,CONT,VIST,VFT,K,1)
VFT=VFT*PI*RIK**2*3600.
H1=CONV(VFT,VIST,DENT,SHT,CONT,0.,RIK)
GO TO 300

C

41 U(K,1)=2.0*PI/(1./(H1*RIK)+RT1+1./(H2*ROK))
U(K,2)=2.0*PI/(R23+1./(H2*ROK))
GO TO 22
42 U(K,1)=2.0*PI/(1./(H1*RIK)+RT1+RT2/CNA)
U(K,2)=2.0*PI/(R23+RT2/CNA)
22 CONTINUE

C

C SOIL CONDUCTANCES

C

R1 = 2.*PI/ALOG(R(2)/R(1))
R2 = 2.*PI/ALOG(R(3)/R(2))
K1 = KDR + 1
DO 30 K=K1,NZP1
U(K,1) = R1*CONS(K)
30 U(K,2) = R2*CONS(K)
DO 8 I=3,NR
RR = 2.*PI/ALOG(R(I+1)/R(I))
DO 8 K=1,NZP1
8 U(K,I) = RR*CONS(K)
RETURN
END

SUBROUTINE FPROP(DEN,PV,YP,SH,CON,ZKP,ZKM,J)

C

```
COMMON /BLK13/ DENFP(5),PVFP(5),YPFP(5),SHFP(5),CONFP(5),
@IPF,IAF,ISF
COMMON /BLK17/ ZI(10,2),NA(10,2),NI(2),NOF
NIJ=1
IF(J.LT.3)GO TO 4
I=IAF
GO TO 5
4 NIJ=NI(J)
2 IF(ZKP.LE.ZI(NIJ,J))GO TO 1
NIJ=NIJ-1
IF(NIJ.LE.0)NIJ=1
IF(NIJ.EQ.1)GO TO 1
GO TO 2
1 I=NA(NIJ,J)
5 DEN=DENFP(I)
PV=PVFP(I)
YP=YPFP(I)
SH=SHFP(I)
CON=CONFP(I)
IF(ZKM.LE.ZI(NIJ,J))GO TO 3
IF(NIJ.EQ.1.OR.J.GT.2)GO TO 3
IF((ZKM-ZI(NIJ,J))**2.LT.1.0)GO TO 3
ZINT=ZI(NIJ-1,J)
INT=NA(NIJ-1,J)
D1=(ZKP-ZINT)/(ZKP-ZKM)
D2=(ZINT-ZKM)/(ZKP-ZKM)
DEN=DEN*D1+DENFP(INT)*D2
PV=PV*D1+PVFP(INT)*D2
YP=YP*D1+YPFP(INT)*D2
SH=SH*D1+SHFP(INT)*D2
CON=CON*D1+CONFP(INT)*D2
3 RETURN
END
```

SUBROUTINE GPROP(DENF,SHF,SHFCV,CONF,VIS,VEL,K,ITA)

THIS SUBROUTINE RETURNS THE FOLLOWING GAS PROPERTIES

DENF = DENSITY, LBM/CF
SHF = SPECIFIC HEAT CAPACITY, CP , BTU/LBM-F
SHFCV = SPECIFIC HEAT CAPACITY, CV , BTU/LBM-F
CONF = THERMAL CONDUCTIVITY, BTU/HR-FT-F
VIS = VISCOSITY, LBM/FT-HR
VEL = VELOCITY, FT/SEC

COMMON /BLK1/ TW(151,20),TWJ(151,21)
COMMON /BLK3/ KDR,NZ,NZP1,TD,Z(151),ZC(15),DZ(15),TVD,DDVN
COMMON /BLK17/ ZI(10,2),NA(10,2),NI(2),NOF
COMMON /BLK18/ DENG(151,2),PRESS(151,2),VGAS(151,2)
COMMON /BLK20/ DCUT(151),VCUT(151),DLIQ(151,2),DVAP(151,2),DRT,
& VFR3
COMMON /BLK21/ IREGM(151,3),QUALS,PSTEM,I2PH,ISKIP(151,2)

KP1=K+1
IF(KP1.GT.KDR)KP1=KDR
IF(I2PH.GT.0)GO TO 2
IGAS=NA(1,1)
DENF=(DENG(K,ITA)+DENG(KP1,ITA))/2.
SHF=.2398
IF(IGAS.EQ.2)SHF=.2480
SHFCV=SHF/1.4
TCELL=TWJ(K,ITA)+TWJ(KP1,ITA)
TKEL=(TCELL/2.+459.67)/1.8
VIS=GVISC(TKEL,IGAS)*.671971*3600.
VEL=(VGAS(K,ITA)+VGAS(KP1,ITA))/2.

DETERMINE AVERAGE PROPERTIES FOR GAS-SOLIDS MIXTURE

IF(ITA.EQ.1)GO TO 1
IF(VEL.LE.0.1)GO TO 1
CS=0.30
D2=(DCUT(K)+DCUT(KP1))/2.
V2=(VCUT(K)+VCUT(KP1))/2.
D1=DENF+D2
G1=DENF*VEL+D2*V2
SHFCV=(SHFCV*DENF+CS*D2)/D1
SHF=(SHF*DENF*VEL+CS*D2*V2)/G1
VEL=(DENF*VEL+D2*V2)/D1
DENF=D1

1 CONTINUE

PR=.7368
VIS1=VIS
IF(IGAS.EQ.3)VIS1=GVISC(TKEL,1)*.671971*3600.
CONF=SHF*VIS1/PR
RETURN

C
C
C

TWO-PHASE STEAM PROPERTIES

```
2 D1=(DVAP(K,1)+DVAP(KP1,1))/2.
  D2=(DLIQ(K,1)+DLIQ(KP1,1))/2.
  DENF=D1+D2
  D1M=D1*16.0184625
  D2M=D2*16.0184625
  T1=(TWJ(K,1)+TWJ(KP1,1))/2.
  T1K=(T1+459.67)/1.8
  DE1=0.0
  DE2=0.0
  IF(D1M.GT.1.E-12.AND.D2M.GT.1.E-12)GO TO 3
  IF(D1M.GT.1.E-12)CALL STEAM(X1,X2,X3,X4,X5,DE1,D1M,T1K)
  IF(D2M.GT.1.E-12)CALL STEAM(X1,X2,X3,X4,X4,DE2,D2M,T1K)
  GO TO 4
3 CALL DVSAT(D1S,DD1S,T1K)
  CALL DLSAT(D2S,DD2S,T1K)
  CALL ESAT(X1,X2,DE1,DE2,D1S,D2S,DD1S,DD2S,T1K)
4 CONTINUE
  SHFCV=(D1*DE1+D2*DE2)/DENF*2.38845E-4
  SHF=(D1*DE1*1.329+D2*DE2)/DENF*2.38845E-4
  VIS=(SLVIS(T1K)*D2+SVVIS(T1K)*D1)/DENF*.671971*3600.
  PR=.7368
  CONF=SHF*VIS/PR
  VEL=(VGAS(K,1)+VGAS(KP1,1))/2.
  RETURN
  END
```

FUNCTION KTUBE(Z1)

C
C
C

DETERMINES TUBING INTERVAL GIVEN DEPTH Z1

COMMON /BLK3/ KDR,NZ,NZP1,TD,Z(151),ZC(15),DZ(15),TVD,DDVN

COMMON /BLK16/ NPF,NTS,KZP

K=6

2 IF(ZC(K).GE.Z1)GO TO 3

K=K+1

IF(K.LT.NTS+5)GO TO 2

K=NTS+5

3 KTUBE=K

RETURN

END

```

FUNCTION CONV(VFR,VIS,DEN,SH,CON,RI,RO)
C
C THIS FUNCTION COMPUTES A SURFACE COEFFICIENT FOR FORCED CONVECTION
C
C VFR = VOLUME FLOW RATE, FT3/HR
C VIS = FLUID VISCOSITY, LBM/FT-HR
C DEN = DENSITY, LBM/FT3
C SH = SPECIFIC HEAT CAPACITY, BTU/LBM-F
C CON = THERMAL CONDUCTIVITY, BTU/HR-FT-F
C RI, RO = INNER AND OUTER RADII OF ANNULUS, FT
C CONV = SURFACE CONVECTION COEFFICIENT, BTU/HR-FT2-F
C
C PRAND L NUMBER
C
C PR = SH*VIS/CON
C
C IF(PR.LT.0.0)WRITE(6,100)SH,VIS,CON
100 FORMAT('***CONV ERROR***/' SH=',E15.8,' VIS=',E15.8,
& ' CON=',E15.8)
C
C REYNOLDS NUMBER
C
C AREA = 3.1415927*(RO*RO - RI*RI)
C VEL = VFR/AREA
C RE = 2.*(RO - RI)*DEN*VEL/VIS
C IF(RE.LE.0.0)WRITE(6,200)RO,RI,DEN,VEL,VIS
200 FORMAT('***CONV ERROR***/' RO=',F8.3,' RI=',F8.3,' DEN=',F8.2,
& ' VEL=',F8.0,' VIS=',E15.8)
C
C LOW PRANDTL NUMBER CONVECTION COEFFICIENT
C
C IF (PR .GT. 50.) GO TO 2
C CONV = CON*0.023*(RE**0.8)*(PR**0.35)/(2.*(RO - RI))
C RETURN
C
C HIGH PRANDTL NUMBER CONVECTION COEFFICIENT
C
C 2 F = 64./RE
C IF (RE .LT. 2000.) GO TO 6
C IF (RE .GT. 4000.) GO TO 4
C F = 0.0077349
C F = (64. + F*(RE - 2000.))/2000.
C GO TO 6
C 4 F = 0.013
C IF (RE .LT. 349120.) F = 0.316/(RE**0.25)
C 6 H = 1.2 + 11.8*(PR - 1)*SQRT(F/8.)/(PR**0.333)
C H = F*RE*PR/(8.*H)
C CONV = CON*H/(2.*(RO - RI))
C RETURN
C END

```

```

FUNCTION CONVN(VIS,DEN,SH,CON,RI,RO,TI,TO)
C
C THIS FUNCTION COMPUTES AN OVERALL CONDUCTIVITY OF A NATURALLY
C CONVECTING FLUID INCLUDING EFFECTIVE CONDUCTIVITY OF FLUID
C AND SURFACE CONVECTION COEFFICIENT.
C
DATA GRAV/4.17E+8/,BETA/0.0005/
C GRAV = ACCELERATION OF GRAVITY, FT3/HR
C BETA = VOLUME COEFFICIENT OF THERMAL EXPANSION
C
BETA=.0005
IF(DEN.LT.5.0)BETA=1./(459.67+TI/2.0+TO/2.0)
C
C PRANDTL NUMBER
C
PR = SH*VIS/CON
C
IF(PR.LE.0.0)WRITE(6,100)SH,VIS,CON
100 FORMAT('***CONVN ERROR***/' SH=',E15.8,' VIS=',E15.8,
& ' CON=',E15.8)
C
C GRASHOF NUMBER
C
GR = ABS(TO - TI)*GRAV*BETA*DEN*DEN
GR = GR*((RO - RI)**3)/(VIS*VIS)
IF(GR.LT.1.E-10)GR=1.E-10
C
C EFFECTIVE FLUID CONDUCTIVITY
C
CONEFF = 0.049*CON*((GR*PR)**0.333)*(PR**0.074)
IF (CONEFF .LT. CON) CONEFF = CON
CONVN=CONEFF
RETURN
END

```

```

FUNCTION HTUBE(TF, TC, ZL, DEN, VIS, CON, SH, IND)
G=4.17E8
BETA=0.0005
IF(IND.EQ.1)BETA=1./(459.67+TF)
GR=G*BETA*DEN**2*ZL**3*ABS(TF-TC)/VIS**2
PR=VIS*SH/CON
GRPR=GR*PR
IF(GRPR.GT.1.E9)H=0.129*GRPR**.3333
IF(GRPR.GT.1.E4.AND.GRPR.LE.1.E9)H=0.59*GRPR**.25
IF(GRPR.GE.0.1.AND.GRPR.LE.1.E4)H=.678*(0.952+PR)**(-.25)*
& SQRT(PR)*GR**.25
IF(GRPR.LT.0.1)H=1.0
HTUBE=H*CON/ZL
RETURN
END

```

SUBROUTINE CONAN(J,K,ZKP,ZKM)

THIS SUBROUTINE COMPUTES THE CONDUCTIVITY OF ALL THE ANNULI
NEEDED TO DETERMINE AN OVERALL CONDUCTANCE AT DEPTH LEVEL K.

COMMON /BLK5/ CAN(5)
COMMON /BLK3/ KDR,NZ,NZP1,TD,Z(151),ZC(15),DZ(15),TVD,DDVN
COMMON /BLK1/ T(151,20),TWJ(151,21)
COMMON /BLK8/ NR,RMAX,R(21),DR(21),AR(20),RI(15),RO(15)
COMMON /BLK10/ DENST,SHST,CONST,DENC,SHC,CONC
COMMON /BLK13/ DENFP(5),PVFP(5),YPFP(5),SHFP(5),CONFP(5),
@IPF,IAF,ISF
COMMON /BLK16/ NPF,NTS,KZP

FACTORS FOR LINEAR INTERPOLATION OF TEMPERATURES FROM T(K,2) TO T(K,3)

TA = T(K,2)
TB = (T(K,3) - T(K,2))/(R(3) - R(2))

LOOP ON THE NUMBER OF ANNULI NEEDED, J, AT LEVEL K.

DO 2 I=1,J

INTERIOR AND EXTERIOR ANNULI TEMPERATURES.

ROC=RO(I)
IF(I.EQ.1)ROC=RO(KZP+5)
TI = TA + TB*(ROC - R(2))
IF (I .EQ. 1) TI = (T(K,1) + T(K,2))/2.
TO = T(K,3)
IF (I .LT. 5) TO = TA + TB*(RI(I+1) - R(2))

VISCOSITY OF ANNULUS FLUID

RR = R(3)
IF (I .LT. 5) RR = RI(I+1)
VIS = VISC(TI,TO,ROC,RR,DENFP(IAF),PVFP(IAF),YPFP(IAF))

CONDUCTIVITY OF ANNULUS FLUID AND OF CELL AT LEVEL K

CAU=CONVN(VIS,DENFP(IAF),SHFP(IAF),CONFP(IAF),ROC,RR,TI,TO)
ZCM=ZC(I)
DZM=DZ(I)
IF(I.GT.1)GO TO 3
ZCM=ZC(KZP+5)
DZM=DZ(KZP+5)
3 CONTINUE
2 CAN(I) = CA(CAU,CONC,ZCM,DZM,ZKP,ZKM)
RETURN
END

FUNCTION CA(CAU,CAL,ZC,DZ,ZKP,ZKM)

C
C
C
C

THIS FUNCTION CHECKS FOR THE TYPE MATERIAL IN AN ANNULUS AND
AVERAGES THE CONDUCTIVITY FOR A CELL IF AN INTERFACE IS FOUND.

IF ((ZC - DZ) .LT. ZKP) GO TO 2
CA = CAU
RETURN
2 IF ((ZC - DZ) .GT. ZKM) GO TO 4
CA = CAL
RETURN
4 CC = CAU*(ZC - DZ - ZKM) + CAL*(ZKP - ZC + DZ)
CA = CC/(ZKP - ZKM)
RETURN
END

```

FUNCTION VISC(TEM1,TEM2,R1,R2,FRD,PV,YP)
C
C THIS FUNCTION COMPUTES THE VISCOSITY OF A FLUID.
C TEM1 = FLUID TEMPERATURE, OR TEMPERATURE AT R1
C TEM2 = 0, OR TEMPERATURE AT R2
C R1 = INSIDE RADIUS OF ANNULUS, FT. R1 = 0 FOR INSIDE A PIPE.
C R2 = OUTSIDE RADIUS OF ANNULUS, FT.
C PV = PLASTIC VISCOSITY, CENTIPOISE.
C YP = YIELD POINT, LBF/100FT2.
C FRD = FLOW RATE, FT3/HR, OR DENSITY, LBM/FT3
C THE FIRST OPTIONS ARE FOR FORCED CONVECTION,
C AND THE SECOND OPTIONS ARE FOR NATURAL CONVECTION.
C GRAV = GRAVITY ACCELERATION, FT/HR2
C BETA = VOLUMETIC COEFFICIENT OF THERMAL EXPANSION, 1/F
C VISW = VISCOSITY OF WATER FROM 30 F TO 300 F, LBM/FT-HR
C
C DIMENSION VISW(28)
C DATA GRAV/4.17E+8/,BETA/0.0001/
C
C POWER LAW EXPONENT AND COEFFICIENT
C
C R600 = YP + 2.*PV
C R300 = YP + PV
C R6DR3=R600/R300
C IF(R1.GE.R2)WRITE(6,100)R1,R2
100 FORMAT('***VISC ERROR***/' R1=',E15.8, '.GT. R2=',E15.8)
C IF(R6DR3.LE.0.0)WRITE(6,200)PV,YP
200 FORMAT('***VISC ERROR***/' PV=',E15.8, ' YP=',E15.8)
C EXP = 3.322*ALOG10(R6DR3)
C COEF = 5.109*R300/(510.9**EXP)
C A = (3.*EXP + 1.)/(EXP*R2)
C IF (R1 .GT. 0.01) A = 2.*(2.*EXP + 1.)/(EXP*(R2 - R1))
C
C NATURAL CONVECTION
C
C IF (TEM2 .LT. 0.1) GO TO 12
C
C APPROXIMATE FLUID VELOCITY TIMES VISCOSITY, LBM/HR2
C VTV = 8550. CORRESPONDS TO WATER AT 1 FT/SEC
C
C VTV = GRAV*FRD*BETA*ABS(TEM2 - TEM1)*(R2 - R1)**2
C VTV = 8550.
C
C SHEAR RATE TIMES VISCOSITY, POISE/SEC
C
C SRTV = 1.148E-6*A*VTV
C
C VISCOSITY OF FLUID AT 70 F
C NOTE (241.92 LBM/FT-HR)/POISE
C
C VISC = COEF*(SRTV**(EXP - 1.))
C VISC = 241.92*(VISC**(1./EXP))
C GO TO 10
C
C FORCED CONVECTION

```

```

12 CONTINUE
C
C FLUID VELOCITY, FT/HR
C
      AREA = 3.1415927*(R2*R2 - R1*R1)
      VEL = FRD/AREA
C
C SHEAR RATE
C
      SR = VEL*A/3600.
C
C VISCOSITY OF FLUID AT 70 F
C NOTE (241.92 LBM/FT-HR)/POISE
C
      VISC = COEF*(SR**(EXP - 1.))
      VISC = 241.92*VISC
C
C WATER VISCOSITY AT TEMPERATURE TEM
C
10 TEM = TEM1
   IF (TEM2 .GT. 0.1) TEM = (TEM1 + TEM2)/2.
   TEMC=(TEM-32.0)/1.8
   VISWTR=.24192*241.4*10.** (247.6/(TEMC+133.15))
   IF (TEMC.GT.300.)VISWTR=.24192*(2843.02-6.4759*TEMC)
   IF (TEMC.GT.375.)VISWTR=.24192*(.407*TEMC+262.28)
   VISWTR=VISWTR/1000.
   VISW70=2.3528
C
C VISCOSITY OF FLUID AT TEMPERATURE TEM.
C
      VISC=VISC*VISWTR/VISW70
      IF (VISC.LE.0.0)WRITE(6,300)VISC
300 FORMAT('***VISC ERROR***/' VISC=',E15.8)
      RETURN
      END

```

SUBROUTINE COEF

C
C THIS SUBROUTINE COMPUTES THE COEFFICIENTS.
C

```

COMMON /BLK2/ A(151,20),B(151,20),C(151,20),D(151,20),E(151,20),
& QDOT(151,3)
COMMON /BLK3/ KDR,NZ,NZP1,TD,Z(151),ZC(15),DZ(15),TVD,DDVN
COMMON /BLK8/ NR,RMAX,R(21),DR(21),AR(20),RI(15),RO(15)
COMMON /BLK4/ U(151,20),Q(151,2)
COMMON /BLK6/ IFLOW,PI,DT,TIN1,TIN2,FR1,FR2,FR3,NIT
COMMON /BLK7/ F(151,3),G(151,3)
COMMON /BLK9/ DENS(150),SHS(150),CONS(150)
COMMON /BLK10/ DENST,SHST,CONST,DENC,SHC,CONC
COMMON /BLK11/ VFR,DDCHG,COSDVN
COMMON /BLK12/ VFR2
COMMON /BLK13/ DENFP(5),PVFP(5),YFP(5),SHFP(5),CONF(5),
@IPF,IAF,ISF
COMMON /BLK15/ ATA(10,2)
COMMON /BLK18/ DENG(151,2),PRESS(151,2),VGAS(151,2)
COMMON /BLK19/ DEPTH,BITR,BITA,DCLRL,DCLRI,DCLRO,PSTAND
COMMON /BLK20/ DCUT(151),VCUT(151),DLIQ(151,2),DVAP(151,2),DRT,
& VFR3
COMMON /BLK21/ IREGM(151,3),QUALS,PSTEM,I2PH,ISKIP(151,2)

```

C
C
C
C

```

DENF=DENFP(IPF)
SHF=SHFP(IPF)
SHFCV=SHF
VFRV=VFR
IF(IFLOW.LE.4.AND.I2PH.EQ.0)GO TO 9
IF(IFLOW.LE.4)GO TO 9
CALL GPROP(DENF,SHF,SHFCV,CONF,VIS,VEL,1,1)
VFRV=VEL*ATA(1,1)*3600.
9 CONTINUE
FRMAS=DENF*VFRV
FR1 = 2.*VFRV*DENF*SHF
FR2 = 2.*VFR2*DENFP(ISF)*SHFP(ISF)
FR3 = FR1 + FR2
FFR = 0.

```

C
C WELLBORE COEFFICIENTS
C

```

DO 3 I=1,3
IF(I.LE.2)AF=ATA(1,I)
IF(I.EQ.3)AF=PI*(RI(3)**2-RO(2)**2)
AP=PI*(RO(I)**2-RI(I)**2)
IF(I.EQ.1)AP=PI*(RO(6)**2-RI(6)**2)
IF(IFLOW.GT.4.AND.I.LT.3)GO TO 13
IF(I2PH.GT.0.AND.I.EQ.1)GO TO 13
CALL FPROP(DENF,PVF,YPF,SHF,CONF,10.,0.,I)
SHFCV=SHF
GO TO 14
13 CALL GPROP(DENF,SHF,SHFCV,CONF,VIS,VEL,1,I)
14 CONTINUE
HH=(DENF*SHFCV*AF+DENST*SHST*AP)/DT
IFL = 0
IF (I .EQ. 2 .AND. IFLOW .LE. 2) IFL = 1

```

IF (FRI .GT. 1. .AND. I .NE. 3 .AND. IFL .EQ. 0) GO TO 20

C
C
C

NO FLOW CONDITIONS

UKIM1 = 0.
KK = 2
IF (I .LE. 2) KK = 1
DO 22 K=KK, KDR
IF (I .GT. 2) GO TO 30
Z1=Z(K)
I1=KTUBE(Z1)
AP=PI*(RO(I1)**2-RI(I1)**2)
IF (IFLOW .GT. 4 .AND. I .LT. 3) GO TO 15
IF (I2PH .GT. 0 .AND. I .EQ. 1) GO TO 15
CALL FPROP (DENF, PVF, YPF, SHF, CONF, Z1+10., Z1-10., I)
SHFCV=SHF
GO TO 16
15 CALL GPROP (DENF, SHF, SHFCV, CONF, VIS, VEL, K, I)
16 CONTINUE
HH=(DENF*SHFCV*ATA(I1-5, I)+DENST*SHST*AP)/DT
30 IF (I .NE. 1) UKIM1 = U(K, I-1)
H = U(K, I) + UKIM1 + HH
C(K, I) = UKIM1/H
D(K, I) = U(K, I)/H
QDOT(K, I)=0.0
22 E(K, I) = HH/H
GO TO 3

C
C
C

FLOWING CONDITIONS

20 IF (I .EQ. 2) GO TO 4
FFR=FRI
IF (IFLOW .EQ. 2) FFR=-FRI
IF (IFLOW .EQ. 4) FFR=-FRI-FR2
UKIM1=0.
UK1IM1=0.
GO TO 12
4 FFR=FRI
IF (IFLOW .EQ. 3) FFR=-FRI-FR2
IF (IFLOW .EQ. 5) FFR=-FRI-FR2
IF (KDR .EQ. (NZ-1)) GO TO 12
JJ=0
IF (DEPTH .GT. ZC(5)) JJ=5
IF (DEPTH .GT. ZC(4)) JJ=4
IF (DEPTH .GT. ZC(3)) JJ=3
RR=R(3)
IF (JJ .NE. 0) RR=RI(JJ)
AF=PI*(RR**2-RO(6)**2)
AP=0.
IF (JJ .NE. 0) AP=PI*(RO(JJ)**2-RI(JJ)**2)
HH=(DENF*SHF*AF+DENST*SHST*AP)/DT
12 DO 2 K=2, KDR
Z1=Z(K)
I1=KTUBE(Z1)
AP=PI*(RO(I1)**2-RI(I1)**2)
DV2=0.0

```

DHVAP=0.0
IF (IFLOW.GT.4.AND.I.LT.3) GO TO 17
IF (I2PH.GT.0.AND.I.EQ.1) GO TO 17
CALL FPROP(DENF,PVF,YPF,SHF,CONF,Z1+10.,Z1-10.,I)
SHFCV=SHF
GO TO 18
17 CALL GPROP(DENF,SHF,SHFCV,CONF,VIS,VEL,K,I)
CALL GPROP(DKML,SKML,SKMLCV,CKML,VSKML,VKML,K-1,I)
IF (I2PH.GT.0) GO TO 18
G1KML=DENG(K-1,I)*VGAS(K-1,I)*3600.
G1K=DENG(K,I)*VGAS(K,I)*3600.
ARK=FRMAS/G1K
ARKML=FRMAS/G1KML
DV2=FRMAS*(VKML*VKML-VEL*VEL)/(Z(K)-Z(K-1))
DV2=DV2*3.9942E-5
DMSL=DLIQ(K-1,I)*VGAS(K-1,I)*ARKML-DLIQ(K,I)*VGAS(K,I)*ARK
DHVAP=2000.*DMSL/(Z(K)-Z(K-1))
IF (I.EQ.2) DV2=-DV2
IF (I.EQ.2) DHVAP=-DHVAP
18 CONTINUE
HH=(DENF*SHFCV*ATA(I1-5,I)+DENST*SHST*AP)/DT
HHKML=HH
IF (IFLOW.GT.4) HHKML=(DKML*SHFCV*ATA(I1-5,I)+DENST*SHST*AP)/DT
FR1=2.*FRMAS*SHF
FFR=FR1
IF (I.EQ.1.AND.IFLOW.EQ.2) FFR=-FR1
IF (I.EQ.1.AND.IFLOW.EQ.4) FFR=-FR1-FR2
IF (I.EQ.2.AND.IFLOW.EQ.3) FFR=-FR1-FR2
IF (I.EQ.2.AND.IFLOW.EQ.5) FFR=-FR1-FR2
FR = FFR/(Z(K) - Z(K-1))
IF (I.EQ.1) GO TO 6
UKIM1 = U(K,I-1)
UKLIM1 = U(K-1,I-1)
6 H = FR + U(K,I) + UKIM1 + HH
A(K,I) = (FR - U(K-1,I) - UKLIM1 - HHKML)/H
C(K,I) = UKIM1/H
D(K,I) = U(K,I)/H
E(K,I) = HH/H
F(K,I) = UKLIM1/H
QDOT(K,I)=(DV2+DHVAP)/H
2 G(K,I) = U(K-1,I)/H
3 CONTINUE

```

C
C
C SOIL COEFFICIENTS

```

DO 8 I=1,NR
AREA = AR(I)
UKIM1 = 0.
KK = 2
IF (I.LT.4) KK = KDR + 1
DO 8 K=KK,NZ
HH = DENS(K)*SHS(K)*AREA/DT
DDZ = Z(K+1) - Z(K)
CON1 = AREA*(CONS(K-1) + CONS(K))/(4.*(Z(K) - Z(K-1))**2)
CON2 = AREA*(CONS(K) + CONS(K+1))/(4.*DDZ*DDZ)
IF (I.GT.1) UKIM1 = U(K,I-1)

```

H = CON1 + CON2 + UKIM1 + U(K,I) + HH
A(K,I) = CON1/H
B(K,I) = CON2/H
C(K,I) = UKIM1/H
D(K,I) = U(K,I)/H
8 E(K,I) = HH/H

C

RETURN
END

SUBROUTINE WELL(DAYS)

```

C
C THIS SUBROUTINE COMPUTES THE TEMPERATURES IN THE WELLBORE
C
COMMON /BLK1/ TW(151,20),TWJ(151,21)
COMMON /BLK2/ A(151,20),B(151,20),C(151,20),D(151,20),E(151,20),
& QDOT(151,3)
COMMON /BLK3/ KDR,NZ,NZP1,TD,Z(151),ZC(15),DZ(15),TVD,DDVN
COMMON /BLK8/ NR,RMAX,R(21),DR(21),AR(20),RI(15),RO(15)
COMMON /BLK4/ U(151,20),Q(151,2)
COMMON /BLK7/ F(151,3),G(151,3)
COMMON /BLK6/ IFLOW,PI,DT,TIN1,TIN2,FR1,FR2,FR3,NIT
COMMON /BLK11/ VFR,DDCHG,COSDVN
COMMON /BLK13/ DENFP(5),PVFP(5),YPFP(5),SHFP(5),CONFP(5),
@IPF,IAF,ISF
COMMON /BLK15/ ATA(10,2)
COMMON /BLK21/ IREGM(151,3),QUALS,PSTEM,I2PH,ISKIP(151,2)

C
C INITIALIZE
C
NIT = 0
DO 10 I=1,NR
DO 10 K=1,NZP1
10 TW(K,I) = TWJ(K,I)

C
C ITERATE
C
8 NIT = NIT + 1

C
C COMPUTE NEW TEMPERATURES
C
BIG = 0.
DO 20 I=1,3

C
C CHECK FOR FLOWING STREAM
C
Z1=Z(KDR)
IF(IFLOW.GT.4.AND.I.LE.2)GO TO 33
IF(I2PH.GT.0.AND.I.EQ.1)GO TO 33
CALL FPROP(DENF,PVF,YPF,SHF,CONF,Z1+10.,Z1-10.,I)
VFRV=VFR
GO TO 34
33 CALL GPROP(DENF,SHF,SHFCV,CONF,VIS,VEL,KDR,I)
KB=KTUBE(Z(KDR))-5
VFRV=VEL*ATA(KB,I)*3600.
34 CONTINUE
FR1=2.*VFRV*DENF*SHF
FR3=FR1+FR2
IFL = 0
IF (I .EQ. 2 .AND. IFLOW .LE. 2) IFL = 1
IF(VFR.LT.1.0)GO TO 24
IF(I.EQ.3)GO TO 24
IF(IFL.EQ.1)GO TO 24
GO TO 26
24 CONTINUE

```

C
C
C

NO FLOW CONDITIONS

```
KK = 2
IF (IFLOW .LE. 2 .AND. I .EQ. 1) KK = 1
IF (IFLOW .GE. 3 .AND. I .LE. 2) KK = 1
DO 28 K=KK,KDR
TKIM1 = 0.
IF (I .GT. 1) TKIM1 = TWJ(K,I-1)
TOLD = TWJ(K,I)
IF(I.GT.1)GO TO 25
IF(I2PH.GT.0.AND.ISKIP(K,1).EQ.1)GO TO 27
25 TWJ(K,I) = C(K,I)*TKIM1 + D(K,I)*TWJ(K,I+1) + E(K,I)*TW(K,I)
27 DIFF = ABS(TOLD - TWJ(K,I))
28 IF (DIFF .GT. BIG) BIG = DIFF
GO TO 20
```

C
C
C

FLOWING STREAM CALCULATIONS

```
26 JFLOW = 0
IF (I .EQ. 2) GO TO 2
IF (IFLOW .EQ. 2 .OR. IFLOW .EQ. 4) JFLOW = 1
GO TO 4
2 IF (IFLOW .EQ. 3) JFLOW = 1
IF(IFLOW.EQ.5)JFLOW=1
4 CONTINUE
IF (JFLOW .EQ. 0) GO TO 6
```

C
C
C

COMPUTE UP FROM BOTTOM

```
IF (I .EQ. 2) GO TO 30
TWJ(KDR,1) = TIN1
IF (IFLOW .EQ. 4) TWJ(KDR,1) = (TWJ(KDR,2)*FR1 + TIN2*FR2)/FR3
GO TO 32
30 TWJ(KDR,2) = (TWJ(KDR,1)*FR1 + TIN2*FR2)/FR3
IF(IFLOW.LE.4)GO TO 32
CALL GPROP(DEN1,SHF1,SHFCV,CONF1,VIS1,VEL1,KDR,1)
CALL GPROP(DEN2,SHF2,SHFCV,CONF2,VIS2,VEL2,KDR,2)
TGAS=(TWJ(KDR,1)*SHF1-(VEL2**2-VEL1**2)*1.9971E-5)/SHF1
TROCK=TWJ(KDR+1,1)
TWJ(KDR,2)=(TGAS-TROCK)*DEN1*SHF1/DEN2/SHF2+TROCK
32 CONTINUE
DO 14 M=2,KDR
K = KDR - M + 2
TOLD=TWJ(K-1,I)
IF(I2PH.GT.0.AND.ISKIP(K-1,1).EQ.1)GO TO 64
TKIM1 = 0.
TT = G(K,I)*TWJ(K-1,I+1)
IF (I .EQ. 1) GO TO 5
TKIM1 = TWJ(K,I-1)*C(K,I)
TT = TT + F(K,I)*TWJ(K-1,I-1)
5 TOLD = TWJ(K-1,I)
IF (I .EQ. 1) GO TO 60
TDIF = TWJ(K,I) - TKIM1
IF (ABS(TDIF/TKIM1) .LT. 0.001) TDIF = 0.
TDIFF = TDIF - D(K,I)*TWJ(K,I+1)
```

```

GO TO 62
60 TDIF = TWJ(K,I) - D(K,I)*TWJ(K,I+1)
   IF (ABS(TDIF/TWJ(K,I)) .LT. 0.001) TDIF = 0.
   TDIF = TDIF - TKIM1
62 AT = TDIF - E(K,I)*(TW(K-1,I) + TW(K,I)) - TT
   AT=AT-QDOT(K,I)
   TWJ(K-1,I) = AT/A(K,I)
64 DIFF = ABS(TOLD - TWJ(K-1,I))
14 IF (DIFF .GT. BIG) BIG = DIFF
   GO TO 20

```

C
C
C

COMPUTE DOWN FROM TOP

```

6 IF (I .EQ. 1) TWJ(1,1) = TIN1
  IF (I .EQ. 2) TWJ(1,2) = TIN1
  DO 12 K=2,KDR
  TOLD=TWJ(K,I)
  IF(IFLOW.GT.1)GO TO 68
  IF(I2PH.GT.0.AND.ISKIP(K,1).EQ.1)GO TO 66
68 CONTINUE
  TKIM1 = 0.
  TT = G(K,I)*TWJ(K-1,I+1)
  IF (I .EQ. 1) GO TO 22
  TKIM1 = TWJ(K,I-1)*C(K,I)
  TT = TT + F(K,I)*TWJ(K-1,I-1)
22 TOLD = TWJ(K,I)
  TWJ(K,I) = A(K,I)*TWJ(K-1,I)
  & + TKIM1 + D(K,I)*TWJ(K,I+1)
  & + E(K,I)*(TW(K-1,I) + TW(K,I)) + TT + QDOT(K,I)
66 DIFF = ABS(TOLD - TWJ(K,I))
12 IF (DIFF .GT. BIG) BIG = DIFF
20 CONTINUE

```

C
C
C

SOIL CALCULATION

```

DO 3 I=1,NR
  KK = 2
  IF (I .LT. 4) KK = KDR + 1
  DO 3 K=KK,NZ
  TT = 0.
  IF (I .GT. 1) TT = C(K,I)*TWJ(K,I-1)
  TOLD = TWJ(K,I)
  TWJ(K,I) = A(K,I)*TWJ(K-1,I) + B(K,I)*TWJ(K+1,I)
  & + TT + D(K,I)*TWJ(K,I+1) + E(K,I)*TW(K,I)
  DIFF = ABS(TOLD - TWJ(K,I))
  3 IF (DIFF .GT. BIG) BIG = DIFF

```

C
C
C

CHECK FOR CONVERGENCE

```

IF (NIT .LT. 50) GO TO 16
WRITE (6,100) NIT,BIG
100 FORMAT (/2X,10('*'),2X,' TEMPERATURES NOT CONVERGED ',
  & 2X,10('*')/20X,'NIT =',I3,5X,'BIG =',E12.4/)
GO TO 18
16 IF (BIG .LE. 0.1) GO TO 18
C IF(I2PH.GT.0)CALL TUFAS

```

IF (IFLOW.LT.5) GO TO 8
CALL FLOW
CALL COEF
GO TO 8
18 CONTINUE

C
C UPDATE TWO-PHASE FLOW
C
C IF (I2PH.GT.0) CALL TUFAS
C
C RETURN
C END

6. REFERENCES

1. Mitchell, Robert F., Advanced Wellbore Thermal Simulator-GEOTEMP2-User Manual, submitted to Sandia Laboratories, (September 1981).
2. Zucrow, Maurice J. and J.D.Hoffman, Gas Dynamics, John Wiley and Sons, Inc., New York, N.Y., (1976).
3. Govier, G.W. and K.Aziz, The Flow of Complex Mixtures in Pipes, Robert E. Krieger Publishing Co., Huntington, N.Y., (1977), p. 166.
4. Swanson, V.F., "The Settling Velocity of Particles of Arbitrary Shape", Trans.Soc.Mining Engineers, 238, (1967), p.160.
5. Mitchell, Robert F., "The Simulation of Air and Mist Drilling for Geothermal Wells", presented at the Conference and Exhibition of the S.P.E of the A.I.M.E, paper number SPE 10234, (October 5-7, 1981).
6. Keenan, Joseph H. et.al., Steam Tables, Wiley-Interscience, New York, N.Y., (1969), pp. 134-159.
7. Orkiszewski, J., "Predicting Two-Phase Pressure Drops in Vertical Pipe", J.P.T., (June 1967), pp. 829-838.
8. DeGance, Anthony E., and R.W.Atherton, Chemical Engineering Aspects of Two-Phase Flow, reprinted from Chemical Engineering, various issues 1970-1971.
9. Chapman, Alan J., Heat Transfer, Second Edition, The Macmillan Co., New York, N.Y., (1967).
10. Wooley, G.R., Wellbore and Soil Thermal Simulation for Geothermal Wells-Development of Computer Model and Acquisition of Field Temperature Data, report submitted to Sandia Laboratories, (December 1978).

DO NOT
MICROFILM

Distribution:
TID-4500-R66-UC-66c (507)

Tom Anderson
Venture Innovations
P.O. Box 35845
Houston, Texas 77035

Ed Bingman
Shell Oil Company
Two Shell Plaza
P.O. Box 2099
Houston, Texas 77001

Larry Diamond
Dyna-Drill
P.O. Box C-19576
Irvine, California 92713

John E. Fontenot
NL Petroleum Services
P.O. Box 60087
Houston, Texas 77205

Dr. Melvin Friedman
Center for Tectonophysics
and Dept. of Geology
Texas A&M University
College Station, Texas 77843

Tom Turner
Phillips Petroleum Company
Geothermal Operations
655 East 4500 South
Salt Lake City, Utah 84107

Jim Kingsolver
Smith Tool
P.O. Box C-19511
Irvine, California 92713

James W. Langford
Security Division
Dresser Industries, Inc.
P.O. Box 24647
Dallas, Texas 75224

Harvey E. Mallory
P.O. Box 54696
Tulsa, Oklahoma 74155

Ed Martin
Superior Oil
Eastern Division
P.O. Box 51108 OCS
Lafayette, Louisiana 70505

Gene Polk
NL Baroid
6400 Uptown Blvd. N.E., 365W
Albuquerque, New Mexico 87110

Del Pyle
Union Geothermal Division
Union Oil Company of California
Union Oil Center
Los Angeles, California 90017

John C. Rowley
Los Alamos National Labs
Mail Stop 570
Los Alamos, New Mexico 87545

Bill Rumbaugh
Otis
P.O. Box 34380
Dallas, Texas 75234

Dwight Smith
Halliburton
Drawer 1431
Duncan, Oklahoma 73533

Tom Warren
Amoco Production Company
P.O. Box 591
Tulsa, Oklahoma 74102

E. L. D. Fooks
Ministry of Works & Development
Private Bag, Taupo
Wairakei, New Zealand

Marion Arnold
Core Laboratories, Inc.
Box 47547
Dallas, Texas 75247

Distribution cont.

Gary McClinton, 2600
Conoco, Inc.
Box 2197
Houston, Texas 77001

D. R. Davies
Koninklijke/Shell Exploratie
en Produktie Laboratorium
Postbus 60, 2280 AB
Rijswijk Z-H
The Netherlands

Michael A. Storms
Deep Sea Drilling Project
U. of California
Scripps Institution of
Oceanography
La Jolla, California 92037

Welby Sweet
Marathon Oil
539 South Main Street
Findley, Ohio 45840

Dr. David D. Balckwell
Professor of Geophysics
Institute for the Study
of Earth and Man
Geothermal Laboratory
Southern Methodist University
Dallas, Texas 75275

Rich Haut
Exxon Production Research
P.O. Box 2189
Houston, Texas 77001

Milton Marks
Dow Chemical Company
Bldg. 1707
Midland, Michigan 48640

John Stalder
Conoco, Inc.
Thermal Recovery Lab
Box 1267
Ponca City, Oklahoma 74601

Professor G. Paul Willhite
Dept. of Chemical and
Petroleum Engineering
University of Kansas
Lawrence, Kansas 66045

Sam Saren
Union Oil
Science & Technology Division
P.O. Box 76
Brea, California 92621

Sam Smith
Union Oil
Science & Technology Division
P.O. Box 76
Brea, California 92621

David McStravick
Baker Packers
P.O. Box 3048
Houston, Texas 77001

U.S. Department of Energy (6)
Geothermal Hydropower
Technologies Division
Forrestal Bldg., CE 324
1000 Independence Ave. S.W.
Washington, D.C. 20585
Attn: J. Bresee
D. Clements
R. Toms
D. Allen
G. Ellis

F. L. Goldsberry
U.S. Department of Energy
Geopressure Projects Office
Suite 8620, Federal Bldg.
515 Rusk Street
Houston, Texas 77002

B. J. Livesay
129 Liverpool
Cardiff, California 92007

Distribution cont.

Pat McGuire
Los Alamos National Labs
Mail Stop 985
Los Alamos, New Mexico 87545

Stephen Pye
Geothermal Division
Union Oil Co. of California
P.O. Box 6854
Santa Rosa, California 95406

Richard Koseluk
Drilling Systems Technology
12950 W. Little York
Houston, Texas 77041

Dennis Goldman
Biological & Earth Sciences
EG&G Idaho, Inc.
P.O. Box 1625
Idaho Falls, Idaho 83401

T. P. Dobbie
Kingston, Reynolds, Thom
& Allardice Limited
P.O. Box 5348, Auckland Ofc.
Auckland, New Zealand

Gene Cannon
Dresser Industries, Inc.
10201 Westheimer Blvd.
Houston, Texas 77042

Ken Green
Dresser Industries, Inc.
10201 Westheimer Blvd.
Houston, Texas 77042

Richard Barker
Dresser Industries, Inc.
10201 Westheimer Blvd.
Houston, Texas 77042

Dr. William L. Perry
Dept. of Mathematics
Texas A&M University
College Station, Texas 77843

Mario Zamora
Imco Mud
P.O. Box 22605
Houston, Texas 77027

Robert Beirute
Amoco Research Center
P.O. Box 591
Tulsa, Oklahoma 74102

Charles Martin
Research & Development Dept.
ARCO Oil & Gas Company
P.O. Box 2819
Dallas, Texas 75221

Mark Cranshaw
Research & Development Dept.
ARCO Oil & Gas Company
P.O. Box 2819
Dallas, Texas 75221

John E. Clarke
Hydril Company
P.O. Box 60458
Houston, Texas 77205

Robert Gaye
Hughes Tool Company
Production Research Dept.
P.O. Box 2539
Houston, Texas 77001

Ronald J. Crook
Chemical Research & Development
Halliburton Services
Drawer 1431
Duncan, Oklahoma 73536

Nancy Jensen
Union Oil Company
461 S. Boylston
Room M-222
Los Angeles, California 90017

Distribution cont.

3141 L. J. Erickson (5)
3151 W. L. Garner (3)
5512 L. A. Mondy
8214 M. A. Pound
9000 G. A. Fowler
9700 E. H. Beckner
9740 R. K. Traeger
9741 J. R. Kelsey (25)
9741 C. C. Carson
9743 H. C. Hardee
9746 B. Granoff
9747 P. J. Hommert
9750 V. L. Dugan
9756 D. Engi
9756 C. Hart