APPENDIX I-A
PROGRAM LISTINGS

CONTENTS

Section		Page
I-A	PROGRAM LISTINGS	
	Simulator Source Code (SIM.FOR).....	I-A-1
	Program for Estimation of Kinetic Parameters (FIT.FOR).....	I-A-37
	Program to Input Data to Simulator (SIMDAT. FOR).....	I-A-65
	Program to Input Data to FIT.FOR (FITDAT.FOR).....	I-A-72

SIMULATOR SOURCE CODE (SIM.FOR)

```
C     ------------------------------------------------------
C          SIMULATOR FOR PREHEATER AND REACTOR/REACTORS
C               FOR A COAL LIQUEFACTION PLANT

C     ------------------------------------------------------


C     List of variables defined in program:
C     -------------------------------------

C     AG    - Concentration of hydrogen in gas phase [-]
C     AGPI  - Inlet concentration of hydrogen in the gas [-]
C     AL    - Concentration of hydrogen in liquid phase [-]
C     ALPI  - Inlet concentration of hydrogen in the liquid [-]
C     BO    - Bond number
C     CGI   - Inlet hydrogen concentration in the gas phase [gmole/cc]
C     CPG   - Specific heat of gas [cal/gm.deg C]
C     CPBAR - Weighted average specific heat of liquid [cal/gm.deg C]
C     CPI   - Initial concentration of each species [gm/gm]
C     D     - Diameter of preheater/reactors [cm]
C     DHR   - Heat of Reaction [cal/gm]
C     DHS   - Heat of Dissolution [cal/gm]
C     DHSTAR- Dimensionless Heat of Dissolution [-]
C     DI    - Diffusivity of gas [sq.cm/sec]
C     E     - Activation energy for Hydrogen consumption kinetics [cal/gmole]
C     ECP   - Activation energy for reactions in preheater [cal/gmole]
C     ECR   - Activation energy for reactions in reactor [cal/gmole]
C     ESTAR - Dimensionless activation energy [-]
C     FR    - Froude number
C     FRAC  - Mole fraction of Hydrogen in inlet gas to preheater [-]
C     FGAS  - Mass flow rate of gas [kg/hr]
C     FSLUR - Mass flow rate of slurry [kg/hr]
C     G     - Acceleration due to gravity [cm/sec/sec]
C     H     - Henry's law constant [gm/cc(liq)/gm/cc(gas)]
C     HBAR  - Preexponential factor
C     HIL   - Solubility of hydrogen at inlet conditions
C     KBARP - Arrhenius factor for rate constant in preheater [1/sec]
C     KBARR - Arrhenius factor for rate constant in reactor [1/sec]
C     L     - Length of preheater/reactors [cm]
C     NAME  - Name of Species
C     NKP   - Number of non-zero rate constants for preheater
C     NKR   - Number of non-zero rate constants for reactor
C     NSPH  - Number of species for preheater
C     NSPR  - Number of species for reactor
C     P     - Operating pressure [atm]
```

```
C       PEH   - Peclet number for heat
C       PEL   - Peclet number for mass
C       R     - Gas constant, cal/gmole.deg K
C       RR    - Gas constant, cc.atm/gmole.deg K
C       SC    - Schimdt number
C       RL    - Dimensionless parameter
C       RP    - Dimensionless parameter
C       TIL   - Temperature at inlet to preheater [deg C]
C       TEMP  - Dimensionless temperature [-]
C       TEMPI - Inlet temperature [deg C]
C       TEMW  - Dimensionless furnace temperature [-]
C       TH    - Temperature of furnace [deg C]
C       TIME  - Residence time of slurry in preheater/reactors [sec]
C       VG    - Superficial Gas Velocity in preheater/reactors [cm/sec]
C       VL    - Superficial Liquid Velocity in preheater/reactors [cm/sec]
C       VOLG  - Volumetric flow rate of gas to preheater [cc/sec]
C       VOLL  - Volumetric flow rate of slurry to preheater [cc/sec]
C       X     - Dimensionless axial distance [-]
C       Z     - Axial distance [cm]
C       ZKLA  - Mass transfer coefficient [1/sec]
C       ZKBAR - Preexponential factor [1/sec]
C       ZNO   - Galileo Number
C
C       Greek Symbols:
C       --------------
C
C
C       EG    - Gas holdup in preheater/reactors [-]
C       ALFA  - Dimensionless parameter
C       BETA  - Dimensionless parameter
C       GAMR  - Dimensionless parameter for preheater/reactors
C       GAMS  - Dimensionless parameter for preheater/reactors
C       ZMEW  - Viscosity [poise]
C       RHOG  - Density of gas [gm/cc]
C       RHOA  - Weighted average density of slurry [gm/cc]
C       SIG   - Surface Tension of slurry [dyne/cm]
C
C       ------------------------------------------------------------------

        IMPLICIT REAL *8(A-H,O-Z)
        COMMON /B/ TIL,HIL,CGI,AGPI,ALPI,TEMPI,TEMW
        COMMON /AN/N

        OPEN (UNIT=1,FILE='SIM1.DAT')
        OPEN (UNIT=11,FILE='SIM2.DAT')
        OPEN (UNIT=3,FILE='SUMRY.DAT')
        OPEN (UNIT=21,FILE='RHYGAS.DAT')
        OPEN (UNIT=31,FILE='RHYLIQ.DAT')
```

```
      OPEN (UNIT=41,FILE='RTEMP.DAT')
      OPEN (UNIT=23,FILE='RCONC.DAT')

      CALL INPUT
      CALL INPUT1(TIL)

      IDUM=1

      CALL PHEATR
      CALL CHEM1
      CALL OUTPUT(IDUM,1)

      IDUM=2

      DO 10 I=2,N
      NR=I
      CALL REATEM(NR)
      CALL CHEM2(NR)
      CALL OUTPUT(IDUM,NR)
10     CONTINUE

      IDUM=3

      DO 20 I=1,N
      NR=I
      CALL COST(NR)
      CALL OUTPUT(IDUM,NR)
20     CONTINUE

      CLOSE (UNIT=23)
      CLOSE (UNIT=41)
      CLOSE (UNIT=31)
      CLOSE (UNIT=21)
      CLOSE (UNIT=3)
      CLOSE (UNIT=11)
      CLOSE (UNIT=1)

      WRITE(6,100)
100    FORMAT(70('-'),/,
     *  25X,'SIMULATION COMPLETE',/,
     *  70('-'),/,
     *  10X,'OUTPUT IS CONTAINED IN THE FOLLOWING FILES',/,
     *  10X,'Output Summary',T50,'SUMRY.DAT',/,
     *  10X,'Hydrodynamic Parameters',T50,'HYDRO.DAT',/,
     *  10X,'Preheater:',/,
     *  15X,'Temperature Profile',T50,'PTEMP.DAT',/,
     *  15X,'Hydrogen Profile (Gas Phase)',T50,'PHYGAS.DAT',/,
     *  15X,'Hydrogen Profile (Liquid Phase)',T50,'PHYLIQ.DAT',/,
```

```
     *  15X,'Viscosity Profile',T50,'PVISCO.DAT',/,
     *  15X,'Species Concentration Profile',T50,'PCONC.DAT',/,
     *  10X,'Reactor/Reactors :',/,
     *  15X,'Temperature Profile',T50,'RTEMP.DAT',/,
     *  15X,'Hydrogen Profile (Gas Phase)',T50,'RHYGAS.DAT',/,
     *  15X,'Hydrogen Profile (Liquid Phase)',T50,'RHYLIQ.DAT',/,
     *  15X,'Species Concentration Profile',T50,'RCONC.DAT',/,
     *  70('-'))
         STOP
         END



         SUBROUTINE PHEATR

C            THIS SUBROUTINE CALCULATES THE HYDROGEN CONCENTRATION IN THE
C        GAS AND LIQUID PHASES AND TEMPERATURE DISTRIBUTION WITHIN THE
C        PREHEATER. THIS IS DONE BY SOLVING THE FIRST ORDER DIFFERENTIAL
C        EQUATIONS WHICH DESCRIBE THE HEAT AND MASS BALANCES FOR THE
C        PREHEATER.

         IMPLICIT REAL *8(A-H,O-Z)
         INTEGER RUNGE
         DOUBLE PRECISION L

         DIMENSION VG(5),VL(5),L(5),D(5)
         DIMENSION GAMMA(5),GAMS(5),GAMR(5),
     *   FLUX(5),HT(5),ALFA(5)
         DIMENSION RL(5),RP(5),PEL(5),PEH(5)
         DIMENSION AG(5),AL(5),TEMP(5)
         DIMENSION Y(3),F(3)
         DIMENSION PAG(101),PAL(101),PTEMP(101)

         COMMON /A/ VG,VL,L,D
         COMMON /B/ TIL,HIL,CGI,AGPI,ALPI,TEMPI,TEMW
         COMMON/DD/ HBAR
         COMMON /DDD/ DHSTAR,ESTAR,GAMMA,GAMS,GAMR,RL,RP
     *  ,FLUX,HT,ALFA
         COMMON /FF/ AG,AL,TEMP
         COMMON/Z9/ PEL,PEH
         COMMON/Z15/ TEMPA,AGPA,ALPA,TEMPO,AGPO,ALPO
         COMMON/QT/ QT



         OPEN (UNIT=10,FILE='PVISCO.DAT')
         OPEN (UNIT=20,FILE='PHYGAS.DAT')
```

```
      OPEN (UNIT=30,FILE='PHYLIQ.DAT')
      OPEN (UNIT=40,FILE='PTEMP.DAT')

      WRITE(6,300)
300    FORMAT(10X,'SIMULATING PREHEATER',/)
      WRITE(10,496)
496    FORMAT(T21,'VISCOSITY PROFILE IN PREHEATER',/,
     * 70('-'),/,
     * T7,'DISTANCE ALONG PREHEATER',18X,'VISCOSITY',/,
     * T5,'DIMENSIONLESS',4X,'CENTIMETERS',18X,'POISE',/,
     * 70('-'),/)
      WRITE(20,497)
497    FORMAT(T21,'HYDROGEN PROFILE IN PREHEATER',/,
     * T30,'(GAS PHASE)',/,
     * 70('-'),/,
     * T7,'DISTANCE ALONG PREHEATER',12X,'HYDROGEN CONCENTRATION',/,
     * T5,'DIMENSIONLESS',4X,'CENTIMETERS',8X,'DIMENSIONLESS',
     * 4X,'GM/CM**3',/,
     * 70('-'),/)

      WRITE(30,498)
498    FORMAT(T21,'HYDROGEN PROFILE IN PREHEATER',/,
     * T28,'(LIQUID PHASE)',/,
     * 70('-'),/,
     * T7,'DISTANCE ALONG PREHEATER',12X,'HYDROGEN CONCENTRATION',/,
     * T5,'DIMENSIONLESS',4X,'CENTIMETERS',8X,'DIMENSIONLESS',
     * 4X,'GM/CM**3',/,
     * 70('-'),/)

      WRITE(40,499)
499    FORMAT(T20,'TEMPERATURE PROFILE IN PREHEATER',//,
     * 70('-'),/,
     * T7,'DISTANCE ALONG PREHEATER',18X,'TEMPERATURE',/,
     * T5,'DIMENSIONLESS',4X,'CENTIMETERS',8X,'DIMENSIONLESS',
     * 4X,'DEGREES C',/,
     * 70('-'),/)

      X=0.0
      QT=0.0
      Y(1)=AGPI
      Y(2)=ALPI
      Y(3)=TEMPI

      TEMPA=TIL*(1.0+TEMPI)
      AGPA=CGI*AGPI
      ALPA=CGI*HIL*ALPI
      ICOUNT=1
      PAG(ICOUNT)=Y(1)
```

```
      PAL(ICOUNT)=Y(2)
      PTEMP(ICOUNT)=Y(3)
      ZMEW=VISCOS(TEMPI)
      ZX=X*L(1)
      ZPAG=PAG(ICOUNT)*CGI
      ZPAL=PAL(ICOUNT)*CGI*HIL
      ZPTEMP=PTEMP(ICOUNT)*TIL+TIL-273.0
      WRITE(10,502) X,ZX,ZMEW
      WRITE(20,500) X,ZX,PAG(ICOUNT),ZPAG
      WRITE(30,500) X,ZX,PAL(ICOUNT),ZPAL
      WRITE(40,500) X,ZX,PTEMP(ICOUNT),ZPTEMP
500    FORMAT(8X,F5.2,8X,F8.1,12X,F8.4,6X,E11.4)
502    FORMAT(8X,F5.2,8X,F8.1,18X,E11.4)
      HH=0.01

10     K=RUNGE(3,Y,F,X,HH)
      IF(K.NE.1) GO TO 15
      H=HBAR*EXP(-DHSTAR/(1+Y(3)))
      TEMPP=Y(3)
      ZMEW=VISCOS(TEMPP)
      F(1)=-RL(1)*(H*Y(1)/HIL-Y(2))
      F(2)=-F(1)/(GAMMA(1)*HIL)-RP(1)*(EXP(-ESTAR/(1.0+Y(3))))*Y(2)
     *  /GAMMA(1)
      F(3)=ALFA(1)*ZMEW**(-0.47)*
     * (TEMW-Y(3))+HIL*RP(1)*GAMR(1)*EXP(-ESTAR/(1+Y(3)))*Y(2)
     * +RL(1)*GAMS(1)*(H*Y(1)/HIL-Y(2))
      Q=ALFA(1)*ZMEW**(-0.47)*(TEMW-Y(3))*HH
      QT=QT+Q
      GO TO 10
15     CONTINUE
      ICOUNT=ICOUNT+1
      PAG(ICOUNT)=Y(1)
      PAL(ICOUNT)=Y(2)
      PTEMP(ICOUNT)=Y(3)
      ZX=X*L(1)
      ZPAG=PAG(ICOUNT)*CGI
      ZPAL=PAL(ICOUNT)*CGI*HIL
      ZPTEMP=PTEMP(ICOUNT)*TIL+TIL-273.0
      WRITE(10,502) X,ZX,ZMEW
      WRITE(20,500) X,ZX,PAG(ICOUNT),ZPAG
      WRITE(30,500) X,ZX,PAL(ICOUNT),ZPAL
      WRITE(40,500) X,ZX,PTEMP(ICOUNT),ZPTEMP
      IF(X.GE.1.0) GO TO 16
      GO TO 10
16     CONTINUE
      AG(1)=Y(1)
      AL(1)=Y(2)
      TEMP(1)=Y(3)
```

```
      WRITE(10,501)
      WRITE(20,501)
      WRITE(30,501)
      WRITE(40,501)
501    FORMAT(/,70('-'))
      TEMPO=TIL*(1.0+Y(3))
      AGPO=CGI*Y(1)
      ALPO=CGI*HIL*Y(2)
      CLOSE (UNIT=10)
      CLOSE (UNIT=20)
      CLOSE (UNIT=30)
      CLOSE (UNIT=40)
      RETURN
      END



      FUNCTION REACTR(ALR)

C          THIS FUNCTION SUBPROGRAM CALCULATES THE HYDROGEN CONCENTRATION
C     IN THE GAS AND LIQUID PHASES AND THE TEMPERATURE DISTRIBUTION WITHIN
C     THE REACTOR. THIS IS DONE BY SOLVING A SERIES OF COUPLED FIRST ORDER
C     DIFFERENTIAL EQUATIONS WHICH DESCRIBE THE HEAT AND MASS BALANCES FOR
C     THE REACTOR.

      IMPLICIT REAL *8(A-H,O-Z)
      INTEGER RUNGE

      DIMENSION Y(5),F(5)
      DIMENSION XX(101),RAG(101),RAL(101),RTEMP(101)
      DIMENSION GAMMA(5),GAMS(5),GAMR(5),
     *  FLUX(5),HT(5),ALFA(5)
      DIMENSION RL(5),RP(5),PEL(5),PEH(5)
      DIMENSION AG(5),AL(5),TEMP(5)

      COMMON /B/ TIL,HIL,CGI,AGPI,ALPI,TEMPI,TEMW
      COMMON /DUM/ XX,RAG,RAL,RTEMP
      COMMON /NUM/ ICOUNT
      COMMON/DD/ HBAR
      COMMON /DDD/ DHSTAR,ESTAR,GAMMA,GAMS,GAMR,RL,RP
     *  ,FLUX,HT,ALFA
      COMMON /FF/ AG,AL,TEMP
      COMMON /REACT/ AGRI,ALRI,DALRI,TEMPRI,DTEMPI
      COMMON /REACT1/ DTEMP,DALR
      COMMON /REACT2/ MM,J
      COMMON/Z9/PEL,PEH
```

```
      ICOUNT=1
      X=0.0
      ALRI=ALR
      DALRI=(ALR-AL(J))*PEL(MM)
      Y(1)=AGRI
      Y(2)=ALR
      Y(3)=DALRI
      Y(4)=TEMPRI
      Y(5)=DTEMPI
      XX(ICOUNT)=X
      RAG(ICOUNT)=Y(1)
      RAL(ICOUNT)=Y(2)
      RTEMP(ICOUNT)=Y(4)
      HH=0.01
10     K=RUNGE(5,Y,F,X,HH)
      IF(K.NE.1) GO TO 15
      H=HBAR*EXP(-DHSTAR/(1+Y(4)))
      F(1)=-RL(MM)*(H*Y(1)/HIL-Y(2))
      F(2)=Y(3)
      F(3)=(Y(3)+F(1)/(GAMMA(MM)*HIL)+RP(MM)/GAMMA(MM)*(EXP(-ESTAR/
     * (1.0+Y(4))))*Y(2))*PEL(MM)
      F(4)=Y(5)
      F(5)=(Y(5)-(RL(MM)*GAMS(MM)*(H/HIL*Y(1)-Y(2)))
     * -HIL*RP(MM)*GAMR(MM)*EXP(-ESTAR/(1.0+Y(4)))*Y(2))*PEH(MM)
      GO TO 10
15     CONTINUE
      ICOUNT=ICOUNT+1
      XX(ICOUNT)=X
      RAG(ICOUNT)=Y(1)
      RAL(ICOUNT)=Y(2)
      RTEMP(ICOUNT)=Y(4)
      IF(X.GE.1.0) GO TO 16
      GO TO 10
16     CONTINUE
      AG(MM)=Y(1)
      AL(MM)=Y(2)
      DALR=Y(3)
      TEMP(MM)=Y(4)
      DTEMP=Y(5)
      REACTR=DALR
      RETURN
      END



      FUNCTION RUNGE(N,Y,F,X,H)
```

```
C       THE FUNCTION RUNGE EMPLOYS THE FOURTH ORDER RUNGE-KUTTA METHOD
C     WITH GILL'S COEFFICIENTS TO INTEGRATE A SYSTEM OF N SIMULTANEOUS
C     FIRST ORDER DIFFERENTIAL EQUATIONS ACROSS ONE STEP OF LENGTH H IN
C     THE INDEPENDENT VARIABLE X, SUBJECT TO THE INITIAL CONDITIONS.

      IMPLICIT REAL *8(A-H,O-Z)
      INTEGER RUNGE

      DIMENSION PHI(50),SAVEY(50),Y(20),F(20)

      DATA M /0/

      M=M+1
      GO TO (1,2,3,4,5),M
1     RUNGE=1
      RETURN
2     DO 22 J=1,N
      SAVEY(J)=Y(J)
      PHI(J)=F(J)
22    Y(J)=SAVEY(J)+0.5*H*F(J)
      X=X+0.5*H
      RUNGE=1
      RETURN
3     DO 33 J=1,N
      PHI(J)=PHI(J)+2.0*F(J)
33    Y(J)=SAVEY(J)+0.5*H*F(J)
      RUNGE=1
      RETURN
4     DO 44 J=1,N
      PHI(J)=PHI(J)+2.0*F(J)
44    Y(J)=SAVEY(J)+H*F(J)
      X=X+0.5*H
      RUNGE=1
      RETURN
5     DO 55 J=1,N
55    Y(J)=SAVEY(J)+(PHI(J)+F(J))*H/6.0
      M=0
      RUNGE=0
      RETURN
      END



      SUBROUTINE HYDRO(N)
```

```
C       THIS SUBROUTINE EVALUATES THE HYDRODYNAMIC PARAMETERS FOR THE
C     PREHEATER AND REACTORS. CORRELATIONS FOR PECLET NUMBER, MASS
C     TRANSFER COEFFICIENT, AND GAS HOLDUP ARE CALCULATED ACCORDING TO THE
C     CORRELATIONS PRESENTED IN THE REPORT. THESE PARAMETERS CAN BE
C     MODIFIED ACCORDINGLY. NOTE THAT SINCE THE CORRELATION FOR THE GAS
C     HOLDUP IS AN IMPLICIT EQUATION IN EG, AN INTERVAL HALVING TECHNIQUE
C     IS USED TO CALCULATE EG. IF EXPLICIT CORRELATIONS FOR EG ARE
C     AVAILABLE , THEN IT IS NOT NECESSARY TO CALL INTHAL.

      IMPLICIT REAL*8(A-H,O-Z)
      DOUBLE PRECISION L

      EXTERNAL FEG

      DIMENSION VG(5),VL(5),L(5),D(5)
      DIMENSION ZKLA(5),EG(5),PL(5),PH(5)

      COMMON /A/ VG,VL,L,D
      COMMON /B/ TIL,HIL,CGI,AGPI,ALPI,TEMPI,TEMW
      COMMON /C/ RHOA,RHOG,DI,SIG,R,G
      COMMON /D/ CPG,CPBAR,E,DHR,DHS,ZKBAR
      COMMON /E/ ZKLA,EG
      COMMON /INT/ EEG,DEG
      COMMON /INT1/ ERR
      COMMON /HYD/BO,ZNO,FR
      COMMON /Z9/ PL,PH

      OPEN (UNIT=29,FILE='HYDRO.DAT')

      DO 10 I=1,N
      V=VG(I)
      DC=D(I)
      ZMEW=VISCOS(TEMPI)
      BO=((DC**2.0)*G*RHOA)/SIG
      FR=V/(G*DC)**0.5
      SC=ZMEW/RHOA/DI
      ZNO=(G*DC**3.0)*((RHOA/ZMEW)**2.0)
      EEG=0.2
      DEG=0.025
      ERR=1.0E-04
      CALL INTHAL(FEG)
      EG(I)=EEG
      ZKLA(I)=(0.6*DI/DC**2.0)*(SC**0.5)*(BO**0.62)*
     * (ZNO**0.31)*(EG(I)**1.1)
      PL(I)=(13.0*FR*(L(I)/D(I))*(VL(I)/VG(I)))
     * /(1.0+8.0*FR**0.85)
      PH(I)=(RHOG*VG(I)*CPG+RHOA*CPBAR*VL(I))
```

```
     *  /(RHOA*CPBAR*VL(I)/PL(I))

        IF(I.NE.1) GO TO 21
        WRITE(29,100)
        GO TO 31
21      K=I-1
        WRITE(29,101) K
31      WRITE(29,102) EG(I),ZKLA(I),PL(I),PH(I)
10      CONTINUE

100     FORMAT(//////5X,'HYDRODYNAMIC PARAMETERS FOR PREHEATER',///)
101     FORMAT(//5X,'HYDRODYNAMIC PARAMETERS FOR REACTOR # ',I3,///)
102     FORMAT(5X,'Gas Holdup ',T50,E12.5,//,
     *  5X,'Volumetric mass transfer coefficient ',T50,E12.5,//,
     *  5X,'Peclet Number for mass dispersion ',T50,E12.5,//,
     *  5X,'Peclet Number for heat dispersion ',T50,E12.5,/)

        CLOSE (UNIT=29)
        RETURN
        END



        FUNCTION VISCOS(TEMP)

C           THIS FUNCTION EVALUATES THE VISCOSITY OF THE COAL SLURRY IN THE
C       PREHEATER AS A FUNCTION OF TEMPERATURE. THE VISCOSITY IS CALCULATED
C       FROM AN EQUATION FITTED TO DATA FOR 35 WT % KENTUCKY NO. 9 COAL
C       WITH WILSONVILLE RECYCLE SOLVENT. THE USER HAS TO CHANGE THE
C       FUNCTION STATEMENTS TO OBTAIN VISCOSITY-TEMPERATURE RELATION FOR
C       THE COAL-SOLVENT SYSTEM USED.


        IMPLICIT REAL *8(A-H,O-Z)
        DOUBLE PRECISION M

        COMMON/B/ TIL

        A=-1.608
        B=9.8578E+11
        M=4.773
        ATEMP=TEMP*TIL+TIL
        IF(ATEMP.GE.550.0.AND.ATEMP.LE.675.0) GO TO 2
        ZMU=10.0**((A+B/ATEMP**M))
        GO TO 3
2       ZMU=0.14
3       CONTINUE
        VISCOS=ZMU
```

```
      RETURN
      END



      SUBROUTINE INTHAL(FUNC)

C     THIS SUBROUTINE IS USED TO SOLVE IMPLICIT ALGEBRAIC EQUATIONS.

      IMPLICIT REAL*8(A-H,O-Z)

      EXTERNAL FEG,REACTR

      COMMON /INT/ X,DX
      COMMON /INT1/ ERR

      U=-1.0
      V=-1.0
1     F=FUNC(X)
      IF(ABS(F).LE.ERR) GO TO 2
      IF(F.GE.0.0) GO TO 3
      IF(V.LE.0.0) GO TO 4
      DX=DX/2.0
4     X=X+DX
      U=1.0
      GO TO 1
3     IF(U.LE.0.0) GO TO 5
      DX=DX/2.0
5     CONTINUE
      X=X-DX
      GO TO 1
2     CONTINUE
      RETURN
      END



      SUBROUTINE INPUT

C     THIS SUBROUTINE READS THE INPUT DATA ON THE PHYSICAL DIMENSIONS
C     OF THE UNITS, PHYSICAL AND THERMAL PROPERTIES AND HYDROGEN
C     CONSUMPTION KINETICS.
```

```
      IMPLICIT REAL *8(A-H,O-Z)
      DOUBLE PRECISION L

      DIMENSION VG(5),VL(5),L(5),D(5)
      DIMENSION GAMMA(5),GAMS(5),GAMR(5),
     * FLUX(5),HT(5),ALFA(5),TIME(5)
      DIMENSION RL(5),RP(5)
      DIMENSION ZKLA(5),EG(5),PL(5),PH(5)

      COMMON/Z9/ PL,PH
      COMMON /A/ VG,VL,L,D
      COMMON /B/ TIL,HIL,CGI,AGPI,ALPI,TEMPI,TEMW
      COMMON /C/ RHOA,RHOG,DI,SIG,R,G
      COMMON /D/ CPG,CPBAR,E,DHR,DHS,ZKBAR
      COMMON/DD/ HBAR
      COMMON /DDD/ DHSTAR,ESTAR,GAMMA,GAMS,GAMR,RL,RP
     *  ,FLUX,HT,ALFA
      COMMON /E/ ZKLA,EG
      COMMON/Z5/ TIME
      COMMON/Z20/ FSLUR,FGAS
      COMMON /AN/ N

      READ(01,*) N
      READ(01,*) VOLG,VOLL,(L(I),I=1,N),(D(I),I=1,N),TIL,P,FRAC,TH
      TIL=TIL+273.0
      TH=TH+273.0


      READ(01,*) CPG,CPBAR,ZKBAR,E,DHR,DHS,HBAR,DI,SIG,RHOA
      RR=82.057
      R=1.98
      G=981.0
      DO 10 I=1,N
      VG(I)=VOLG/(D(I)**2.0*0.786)
      VL(I)=VOLL/(D(I)**2.0*0.786)
10     CONTINUE

      CGI=P/(RR*TIL)*FRAC*2.0
      RHOG=CGI
      AGPI=1.0
      ALPI=0.0
      TEMPI=0.0
      TEMW=(TH-TIL)/TIL
      DHSTAR=DHS/(R*TIL)
      ESTAR=E/(R*TIL)
      HIL=HBAR*(EXP(-DHSTAR))
      CALL HYDRO(N)
```

```
      FSLUR=VOLL*RHOA*3600.0/1000.0
      FGAS=VOLG*RHOG*3600.0/1000.0
      DO 11 I=1,N
      RL(I)=ZKLA(I)*L(I)*HIL/VG(I)
      GAMMA(I)=VL(I)/VG(I)
      RP(I)=ZKBAR*(1-EG(I))*L(I)/VG(I)
      TIME(I)=L(I)*(1.-EG(I))/VL(I)
      FLUX(I)=VG(I)*RHOG*CPG+VL(I)*RHOA*CP8AR
      GAMS(I)=(CGI*VG(I)*DHS)/TIL/FLUX(I)
      GAMR(I)=(CGI*VG(I)*DHR)/TIL/FLUX(I)
      BETA=0.0045
      HT(I)=8ETA*VL(I)**0.8/D(I)**0.2
      ALFA(I)=HT(I)/D(I)*L(I)/FLUX(I)
11    CONTINUE
      RETURN
      END


      FUNCTION FEG(X)

C       THIS FUNCTION CALCULATES THE GAS HOLDUP ACCORDING TO THE
C     CORRELATION OF AKITA AND YOSHIDA.

      IMPLICIT REAL *8(A-H,O-Z)

      COMMON /HYD/ BO,ZNO,FR

      F1=X/((1.0-X)**4.0)
      F2=0.2*(BO**0.125)*(ZNO**0.083)*FR
      FEG=F1-F2
      RETURN
      END


      SUBROUTINE REATEM(NR)

C       THIS SUBROUTINE SIMULATES THE THERMAL BEHAVIOR OF THE REACTOR.

      IMPLICIT REAL *8(A-H,O-Z)
      DOUBLE PRECISION L
```

```
      EXTERNAL REACTR

      DIMENSION AG(5),AL(5),TEMP(5),L(5),D(5),VG(5),VL(5)
      DIMENSION XX(101),RAG(101),RAL(101),RTEMP(101)
      DIMENSION ZKLA(5),EG(5),PL(5),PH(5),RRTEMP(101)

      COMMON /A/ VG,VL,L,D
      COMMON /B/ TIL,HIL,CGI,AGPI,ALPI,TEMPI,TEMW
      COMMON /REACT/AGRI,ALRI,DALRI,TEMPRI,DTEMPI
      COMMON /DUM/ XX,RAG,RAL,RTEMP
      COMMON /NUM/ ICOUNT
      COMMON /REACT1/ DTEMP,DALR
      COMMON /REACT2/ M,J
      COMMON /INT/ ALR,DAL
      COMMON /INT1/ ERR
      COMMON /FF/ AG,AL,TEMP
      COMMON /E/ ZKLA,EG
      COMMON/Z8/ RRTEMP
      COMMON/Z9/ PL,PH
      COMMON/Z16/ ZRTEMP,ZRAG,ZRAL

      J=NR-1
      WRITE(6,300) J
300    FORMAT(10X,'SIMULATING REACTOR # ',I3,/)
      WRITE(21,497) J
497    FORMAT('1',T22,'HYDROGEN PROFILE IN REACTOR # ',I3,/,
     * T30,'(GAS PHASE)',/,
     * 70('-'),/,
     * T8,'DISTANCE ALONG REACTOR',13X,'HYDROGEN CONCENTRATION',/,
     * T5,'DIMENSIONLESS',4X,'CENTIMETERS',8X,'DIMENSIONLESS',
     * 4X,'GM/CM**3',/,
     * 70('-'),/)

      WRITE(31,498) J
498    FORMAT('1',T22,'HYDROGEN PROFILE IN REACTOR # ',I3,/,
     * T28,'(LIQUID PHASE)',/,
     * 70('-'),/,
     * T8,'DISTANCE ALONG REACTOR',13X,'HYDROGEN CONCENTRATION',/,
     * T5,'DIMENSIONLESS',4X,'CENTIMETERS',8X,'DIMENSIONLESS',
     * 4X,'GM/CM**3',/,
     * 70('-'),/)

      WRITE(41,499) J
499    FORMAT('1',T21,'TEMPERATURE PROFILE IN REACTOR # ',I3,//,
     * 70('-'),/,
     * T8,'DISTANCE ALONG REACTOR',19X,'TEMPERATURE',/,
     * T5,'DIMENSIONLESS',4X,'CENTIMETERS',8X,'DIMENSIONLESS',
     * 4X,'DEGREES C',/,
```

```
      *  70('-'),/)

         ERR= 0.001
         M=NR
         U=-1.0
         V=-1.0
         TEMPRI=TEMP(J)*1.5
         DTEMPI=(TEMPRI-TEMP(J))*PH(NR)
         DTMPI=0.1
         AGRI=AG(J)
         ALRI=AL(J)*0.9
         ALR=ALRI
         DAL=0.2
    1    CONTINUE
         CALL INTHAL(REACTR)
         DAL=0.01
         IF(ABS(DTEMP).LE.0.001) GO TO 2
         IF(DTEMP.GE.0.0)GO TO 3
         IF(V.LE.0.0)GO TO 4
         ERR=0.001
         DTMPI=DTMPI/2.0
    4    TEMPRI=TEMPRI+DTMPI
         DTEMPI=(TEMPRI-TEMP(J))*PH(NR)
         U=1.0
         GO TO 1
    3    IF(U.LE.0.0)GO TO 5
         ERR=0.001
         DTMPI=DTMPI/2.0
    5    TEMPRI=TEMPRI-DTMPI
         DTEMPI=(TEMPRI-TEMP(J))*PH(NR)
         GO TO 1
    2    CONTINUE
         DO 23 II=1,ICOUNT
         RRTEMP(II)=RTEMP(NR)
         ZX=XX(II)*L(NR)
         ZRAG=RAG(II)*CGI
         ZRAL=RAL(II)*CGI*HIL
         ZRTEMP=RTEMP(II)*TIL+TIL-273.0

         WRITE(21,500) XX(II),ZX,RAG(II),ZRAG
         WRITE(31,500) XX(II),ZX,RAL(II),ZRAL
         WRITE(41,500) XX(II),ZX,RTEMP(II),ZRTEMP
   500    FORMAT(8X,F5.2,8X,F8.1,12X,F8.4,6X,E11.4)
    23    CONTINUE
    10    CONTINUE
         WRITE(21,501)
         WRITE(31,501)
         WRITE(41,501)
```

```
501    FORMAT(/,70('-'))

       RETURN
       END


       SUBROUTINE INPUT1(TIL)

C          THIS SUBROUTINE READS THE INPUT DATA ON COAL CONVERSION KINETICS.


       IMPLICIT REAL *8(A-H,O-Z)
       DOUBLE PRECISION KBARP,KBARR
       CHARACTER*20 NAME(10,2)

       DIMENSION KBARP(20,20),KBARR(20,20),COEFFP(20,20)
       DIMENSION COEFFR(20,20),ECP(20,20),ECR(20,20),CPI(10)

       COMMON/Z1/ KBARP,ECP,COEFFP
       COMMON/Z11/ KBARR,ECR,COEFFR
       COMMON/Z2/ CPI
       COMMON/Z6/ NSPH,NSPR

       DATA R/1.98/

       DO 1 I=1,20
       DO 1 J=1,20
       KBARP(I,J)=0.0
       KBARR(I,J)=0.0
       ECP(I,J)=0.0
       ECR(I,J)=0.0
       COEFFP(I,J)=0.0
       COEFFR(I,J)=0.0
1      CONTINUE


       READ(11,*) NSPH
       DO 2 I=1,NSPH
       READ(11,102) NAME(I,1)
102    FORMAT(A20)
2      CONTINUE
       READ(11,*) NKP
       DO 4 I=1,NKP
       READ(11,*) II
       READ(11,*) JJ
       READ(11,*) KBARP(II,JJ)
       READ(11,*) ECP(II,JJ)
```

```
      ECP(II,JJ)=ECP(II,JJ)/(R*TIL)
      COEFFP(II,JJ)=1.0
4     CONTINUE

      READ(11,*) NSPR
      DO 6 I=1,NSPR
      READ(11,102) NAME(I,2)
6     CONTINUE
      READ(11,*) NKR
      DO 7 I=1,NKR
      READ(11,*) II
      READ(11,*) JJ
      READ(11,*) KBARR(II,JJ)
      READ(11,*) ECR(II,JJ)
      ECR(II,JJ)=ECR(II,JJ)/(R*TIL)
      COEFFR(II,JJ)=1.0
7     CONTINUE


      DO 5 I=1,NSPR
      READ(11,*) CPI(I)
5     CONTINUE
      RETURN
      END



      SUBROUTINE EQTNS(TEMP,ECSTAR,KBAR,COEFF,NSP)

C       THIS SUBROUTINE CALCULATES THE RATES OF CHEMICAL SPECIES REACTION.


      IMPLICIT REAL *8(A-H,O-Z)
      DOUBLE PRECISION KBAR,K

      DIMENSION KBAR(20,20),ECSTAR(20,20),RATE(10)
      DIMENSION COEFF(20,20),TEMP(101),K(20,20)
      DIMENSION C(10,101)

      COMMON/Z3/ C,RATE,ICOUNT


      DO 1 IJ=1,NSP
      RATE(IJ)=0.0
      DO 1 I=1,NSP
      DO 1 J=1,NSP
      IF(COEFF(I,J).EQ.0.0) GO TO 1
```

```
      IF((IJ.NE.I).AND.(IJ.NE.J)) GO TO 1
      K(I,J)=KBAR(I,J)*EXP(-ECSTAR(I,J)/(1.0+TEMP(ICOUNT)))
      IF(IJ.EQ.I) RATE(IJ)=RATE(IJ)-COEFF(I,J)*K(I,J)*C(I,ICOUNT)
      IF(IJ.EQ.J) RATE(IJ)=RATE(IJ)+COEFF(I,J)*K(I,J)*C(I,ICOUNT)
1     CONTINUE
      RETURN
      END


      SUBROUTINE PHEATC

C          THIS SUBROUTINE SIMULATES THE CONVERSION OF THE CHEMICAL SPECIES
C     IN THE PREHEATER.


      IMPLICIT REAL *8(A-H,O-Z)
      DOUBLE PRECISION KBARP
      INTEGER RUNGE

      DIMENSION CPI(10),CPO(10),RATE(10),PTEMP(101),C(10,101),TIME(5)
      DIMENSION Y(20),F(20),XX(101)
      DIMENSION KBARP(20,20),ECP(20,20),COEFFP(20,20)

      COMMON/Z1/ KBARP,ECP,COEFFP
      COMMON/Z2/ CPI
      COMMON/Z3/ C,RATE,ICOUNT
      COMMON/Z4/ PTEMP
      COMMON/Z5/ TIME
      COMMON/Z6/ NSPH,NSPR
      COMMON/Z7/ CPO
      COMMON/Z13/ XX


      X=0.0
      ICOUNT=1
      NSP=NSPH
      XX(ICOUNT)=X
      DO 25 I=1,NSP
      Y(I)=CPI(I)
      C(I,ICOUNT)=Y(I)
25    CONTINUE
      HH=0.01

10    K=RUNGE(NSP,Y,F,X,HH)
      IF(K.NE.1) GO TO 15
      CALL EQTNS(PTEMP,ECP,KBARP,COEFFP,NSPH)
      DO 26 I=1,NSP
      F(I)=RATE(I)*TIME(1)
```

```
      C(I,ICOUNT)=Y(I)
26     CONTINUE
      GO TO 10
15     CONTINUE
      ICOUNT=ICOUNT+1
      XX(ICOUNT)=X
      DO 27 I=1,NSP
      C(I,ICOUNT)=Y(I)
27     CONTINUE
      IF(X.GE.1.0) GO TO 16
      GO TO 10
16     CONTINUE
      DO 28 I=1,NSP
      CPO(I)=Y(I)
28     CONTINUE
      IF(NSPH.EQ.NSPR) RETURN
      DO 29 I=NSPH,NSPR
      CPO(I)=CPI(I)
29     CONTINUE

      RETURN
      END



      SUBROUTINE CHEM1

C          THIS SUBROUTINE INITIALIZES SIMULATION OF CHEMICAL SPECIES
C     CONVERSION IN THE PREHEATER.

      IMPLICIT REAL *8(A-H,O-Z)
      DOUBLE PRECISION L

      DIMENSION C(10,101),RATE(10),XX(101),VG(5),VL(5),L(5),D(5)

      COMMON /A/ VG,VL,L,D
      COMMON/Z3/ C,RATE,ICOUNT
      COMMON/Z6/ NSPH,NSPR
      COMMON/Z13/ XX

      CALL PHEATC
      OPEN (UNIT=22,FILE='PCONC.DAT')
      NSP=NSPH
      WRITE(22,499)
499    FORMAT(T30,'CONCENTRATION PROFILE IN PREHEATER',/)
      IF(NSP.EQ.2) WRITE(22,502)
      IF(NSP.EQ.3) WRITE(22,503)
```

```
      IF(NSP.EQ.4) WRITE(22,504)
      IF(NSP.EQ.5) WRITE(22,505)
      IF(NSP.EQ.6) WRITE(22,506)
      IF(NSP.EQ.7) WRITE(22,507)
      IF(NSP.EQ.8) WRITE(22,508)
      IF(NSP.EQ.9) WRITE(22,509)
      IF(NSP.EQ.10) WRITE(22,510)
502    FORMAT(T30,70('-'),/,
     * T37,'DISTANCE ALONG PREHEATER',12X,'CHEMICAL SPECIES',/,
     * T35,'DIMENSIONLESS',4X,'CENTIMETERS',12X,'1',10X,'2',/,
     * T30,70('-'))
503    FORMAT(T5,80('-'),/,
     * T12,'DISTANCE ALONG PREHEATER',19X,'CHEMICAL SPECIES',/,
     * T10,'DIMENSIONLESS',4X,'CENTIMETERS',12X,'1',11X,'2',11X,'3',/,
     * T5,80('-'))
504    FORMAT(T20,90('-'),/,
     * T27,'DISTANCE ALONG PREHEATER',24X,'CHEMICAL SPECIES',/,
     * 24X,'DIMENSIONLESS',4X,'CENTIMETERS',12X,'1',11X,'2',11X,'3',
     * 11X,'4',/,
     * T20,90('-'))
505    FORMAT(T15,100('-'),/,
     * T22,'DISTANCE ALONG PREHEATER',29X,'CHEMICAL SPECIES',/,
     * 19X,'DIMENSIONLESS',4X,'CENTIMETERS',12X,'1',11X,'2',11X,'3',
     * 11X,'4',11X,'5',/,
     * T15,100('-'))
506    FORMAT(T10,110('-'),/,
     * T17,'DISTANCE ALONG PREHEATER',34X,'CHEMICAL SPECIES',/,
     * 14X,'DIMENSIONLESS',4X,'CENTIMETERS',12X,'1',11X,'2',11X,'3',
     * 11X,'4',11X,'5',11X,'6',/,
     * T10,110('-'))
507    FORMAT(T5,125('-'),/,
     * T12,'DISTANCE ALONG PREHEATER',39X,'CHEMICAL SPECIES',/,
     * 9X,'DIMENSIONLESS',4X,'CENTIMETERS',12X,'1',11X,'2',11X,'3',
     * 11X,'4',11X,'5',11X,'6',11X,'7',/,
     * T5,125('-'))
508    FORMAT(132('-'),/,
     * T7,'DISTANCE ALONG PREHEATER',44X,'CHEMICAL SPECIES',/,
     * 4X,'DIMENSIONLESS',4X,'CENTIMETERS',12X,'1',11X,'2',11X,'3',
     * 11X,'4',11X,'5',11X,'6',11X,'7',11X,'8',/,
     * 132('-'))
509    FORMAT(132('-'),/,
     * T4,'DISTANCE ALONG PREHEATER',47X,'CHEMICAL SPECIES',/,
     * 1X,'DIMENSIONLESS',3X,'CENTIMETERS',11X,'1',10X,'2',10X,'3',
     * 10X,'4',10X,'5',10X,'6',10X,'7',10X,'8',10X,'9',/,
     * 132('-'))
510    FORMAT(132('-'),/,
     * T3,'DISTANCE ALONG PREHEATER',49X,'CHEMICAL SPECIES',/,
     * 1X,'DIMENSIONLESS',2X,'CENTIMETERS',10X,'1',9X,'2',9X,'3',
```

```
     *  9X,'4',9X,'5',9X,'6',9X,'7',9X,'8',9X,'9',9X,'10',/,
     *  132('-'))
        DO 1 J=1,ICOUNT
        ZX=XX(J)*L(1)
        IF(NSP.EQ.2) WRITE(22,602) XX(J),ZX,(C(I,J),I=1,NSP)
        IF(NSP.EQ.3) WRITE(22,603) XX(J),ZX,(C(I,J),I=1,NSP)
        IF(NSP.EQ.4) WRITE(22,604) XX(J),ZX,(C(I,J),I=1,NSP)
        IF(NSP.EQ.5) WRITE(22,605) XX(J),ZX,(C(I,J),I=1,NSP)
        IF(NSP.EQ.6) WRITE(22,606) XX(J),ZX,(C(I,J),I=1,NSP)
        IF(NSP.EQ.7) WRITE(22,607) XX(J),ZX,(C(I,J),I=1,NSP)
        IF(NSP.EQ.8) WRITE(22,608) XX(J),ZX,(C(I,J),I=1,NSP)
        IF(NSP.EQ.9) WRITE(22,609) XX(J),ZX,(C(I,J),I=1,NSP)
        IF(NSP.EQ.10) WRITE(22,610) XX(J),ZX,(C(I,J),I=1,NSP)
1       CONTINUE
        IF(NSP.EQ.2) WRITE(22,702)
        IF(NSP.EQ.3) WRITE(22,703)
        IF(NSP.EQ.4) WRITE(22,704)
        IF(NSP.EQ.5) WRITE(22,705)
        IF(NSP.EQ.6) WRITE(22,706)
        IF(NSP.EQ.7) WRITE(22,707)
        IF(NSP.EQ.8) WRITE(22,708)
        IF(NSP.EQ.9) WRITE(22,708)
        IF(NSP.EQ.10) WRITE(22,708)
602     FORMAT(38X,F5.2,8X,F8.1,10X,E9.3,2X,E9.3)
603     FORMAT(13X,F5.2,8X,F8.1,10X,E9.3,2(3X,E9.3))
604     FORMAT(28X,F5.2,8X,F8.1,10X,E9.3,3(3X,E9.3))
605     FORMAT(23X,F5.2,8X,F8.1,10X,E9.3,4(3X,E9.3))
606     FORMAT(18X,F5.2,8X,F8.1,10X,E9.3,5(3X,E9.3))
607     FORMAT(13X,F5.2,8X,F8.1,10X,E9.3,6(3X,E9.3))
608     FORMAT(8X,F5.2,8X,F8.1,10X,E9.3,7(3X,E9.3))
609     FORMAT(5X,F5.2,7X,F8.1,9X,E9.3,8(2X,E9.3))
610     FORMAT(5X,F5.2,6X,F8.1,9X,E9.3,9(1X,E9.3))
702     FORMAT(T30,70('-'))
703     FORMAT(T5,80('-'))
704     FORMAT(T20,90('-'))
705     FORMAT(T15,100('-'))
706     FORMAT(T10,110('-'))
707     FORMAT(T5,125('-'))
708     FORMAT(132('-'))
        CLOSE (UNIT=22)
        RETURN
        END



        SUBROUTINE REACTC(NR)
```

```
C          THIS SUBROUTINE SIMULATES THE CHEMICAL SPECIES CONVERSION IN
C       THE PREHEATER.

      IMPLICIT REAL *8(A-H,O-Z)
      DOUBLE PRECISION KBARR
      INTEGER RUNGE,ORDER

      DIMENSION CPO(10),RATE(10),RTEMP(101),C(10,101),TIME(5)
      DIMENSION Y(20),F(20),A(20,20),PEL(5),B(20),ZETA(20),PEH(5)
      DIMENSION ORDER(20),GUESS(20,20),FACTOR(50),RRATE(10)
      DIMENSION AA(20,20),XX(101)
      DIMENSION KBARR(20,20),ECR(20,20),COEFFR(20,20)

      COMMON/Z11/ KBARR,ECR,COEFFR
      COMMON/Z7/ CPO
      COMMON/Z13/ XX
      COMMON/Z9/ PEL,PEH
      COMMON/Z3/ C,RATE,ICOUNT
      COMMON/Z8/ RTEMP
      COMMON/Z5/ TIME
      COMMON/Z6/ NSPH,NSPR

      ITEST=0
      ICOUNT=1
      NSP=NSPR
      DO 8 I=1,NSP
      B(I)=CPO(I)
      C(I,ICOUNT)=CPO(I)
      B(I+NSP)=0.0
8     CONTINUE
      CALL EQTNS(RTEMP,ECR,KBARR,COEFFR,NSPR)
      DO 90 J=1,NSP
      RRATE(J)=RATE(J)
90    CONTINUE
      HH=0.01
      NNSP=2*NSP
      SEED=1
      DO 1 I=1,NNSP
      ICOUNT=1
      X=0.0
      XX(ICOUNT)=X
      DO 20 J=1,NNSP
      FACTOR(J)=RAN(SEED)
20    CONTINUE
      DO 2 N=1,NSP
      Y(N)=CPO(N)*(1+(-1)**I*FACTOR(N)*0.2)
      Y(N+NSP)=RRATE(N)/ABS(RRATE(N))*(0.5*FACTOR(N+NSP))
2     CONTINUE
```

```
      DO 43 N=1,NSP
      A(N,I)=Y(N)-Y(N+NSP)/PEL(NR)
      C(N,ICOUNT)=Y(N)
43    CONTINUE
      DO 6 J=1,NNSP
      GUESS(J,I)=Y(J)
6     CONTINUE

10    CONTINUE
      K=RUNGE(NNSP,Y,F,X,HH)
      IF(K.NE.1) GO TO 15
      CALL EQTNS(RTEMP,ECR,KBARR,COEFFR,NSPR)
      DO 26 J=1,NSP
      F(J)=Y(J+NSP)
      F(J+NSP)=PEL(NR)*(Y(J+NSP)-RATE(J)*TIME(NR))
      C(J,ICOUNT)=Y(J)
26    CONTINUE
      GO TO 10
15    CONTINUE
      ICOUNT=ICOUNT+1
      XX(ICOUNT)=X
      DO 17 J=1,NSP
      C(J,ICOUNT)=Y(J)
17    CONTINUE
      IF(X.GE.1.0) GO TO 16
      GO TO 10
16    CONTINUE
      IF(ITEST.NE.0) GO TO 12
      DO 55 J=1,NSP
      A(NSP+J,I)=Y(NSP+J)
55    CONTINUE
1     CONTINUE
      DO 45 I1=1,NNSP
45    CONTINUE
C     DO 68 I1=NSP+1,NNSP
C     DO 68 I2=1,NNSP
C     A(I1,I2)=A(I1,I2)/A(I1,1)
68    CONTINUE
      DO 69 I1=1,NNSP
      DO 69 I2=1,NNSP
      AA(I1,I2)=A(I1,I2)
69    CONTINUE
      CALL LUDCMP(A,NNSP,NNSP,ORDER)
      CALL SOLVLU(A,B,ZETA,NNSP,NNSP,ORDER)
C
C
      DO 70 I1=1,NNSP
      BTOT=0.0
```

```
      DO 71 I2=1,NNSP
      BTOT=BTOT+AA(I1,I2)*ZETA(I2)
71    CONTINUE
70    CONTINUE
      ITEST=1
      ICOUNT=1
      X=0.0
      DO 7 J=1,NNSP
      Y(J)=0.0
      DO 7 I=1,NNSP
      Y(J)=Y(J)+ZETA(I)*GUESS(J,I)
7     CONTINUE
      DO 18 I=1,NSP
      C(I,ICOUNT)=Y(I)
18    CONTINUE
      GO TO 10
12    CONTINUE
      DO 28 I=1,NSP
28    CPO(I)=Y(I)

      RETURN
      END



      SUBROUTINE CHEM2(NR)

C          THIS SUBROUTINE INITIALIZES THE SIMULATION OF THE CHEMICAL SPECIES
C     CONVERSION IN THE REACTOR.

      IMPLICIT REAL *8(A-H,O-Z)
      DOUBLE PRECISION L

      DIMENSION C(10,101),RATE(10),XX(101),VG(5),VL(5),L(5),D(5)

      COMMON/A/ VG,VL,L,D
      COMMON/Z3/ C,RATE,ICOUNT
      COMMON/Z6/ NSPH,NSPR
      COMMON/Z13/ XX

      CALL REACTC(NR)
      K=NR-1
      NSP=NSPR
      WRITE(23,499) K
499   FORMAT('1',T31,'CONCENTRATION PROFILE IN REACTOR #  ',I3,/)
      IF(NSP.EQ.2) WRITE(23,502)
      IF(NSP.EQ.3) WRITE(23,503)
```

```
      IF(NSP.EQ.4) WRITE(23,504)
      IF(NSP.EQ.5) WRITE(23,505)
      IF(NSP.EQ.6) WRITE(23,506)
      IF(NSP.EQ.7) WRITE(23,507)
      IF(NSP.EQ.8) WRITE(23,508)
      IF(NSP.EQ.9) WRITE(23,509)
      IF(NSP.EQ.10) WRITE(23,510)
502    FORMAT(T30,70('-'),/,
     * T38,'DISTANCE ALONG REACTOR',13X,'CHEMICAL SPECIES',/,
     * T35,'DIMENSIONLESS',4X,'CENTIMETERS',12X,'1',10X,'2',/,
     * T30,70('-'))
503    FORMAT(T5,80('-'),/,
     * T13,'DISTANCE ALONG REACTOR',20X,'CHEMICAL SPECIES',/,
     * T10,'DIMENSIONLESS',4X,'CENTIMETERS',12X,'1',11X,'2',11X,'3',/,
     * T5,80('-'))
504    FORMAT(T20,90('-'),/,
     * T28,'DISTANCE ALONG REACTOR',25X,'CHEMICAL SPECIES',/,
     * 24X,'DIMENSIONLESS',4X,'CENTIMETERS',12X,'1',11X,'2',11X,'3',
     * 11X,'4',/,
     * T20,90('-'))
505    FORMAT(T15,100('-'),/,
     * T23,'DISTANCE ALONG REACTOR',30X,'CHEMICAL SPECIES',/,
     * 19X,'DIMENSIONLESS',4X,'CENTIMETERS',12X,'1',11X,'2',11X,'3',
     * 11X,'4',11X,'5',/,
     * T15,100('-'))
506    FORMAT(T10,110('-'),/,
     * T18,'DISTANCE ALONG REACTOR',35X,'CHEMICAL SPECIES',/,
     * 14X,'DIMENSIONLESS',4X,'CENTIMETERS',12X,'1',11X,'2',11X,'3',
     * 11X,'4',11X,'5',11X,'6',/,
     * T10,110('-'))
507    FORMAT(T5,125('-'),/,
     * T13,'DISTANCE ALONG REACTOR',40X,'CHEMICAL SPECIES',/,
     * 9X,'DIMENSIONLESS',4X,'CENTIMETERS',12X,'1',11X,'2',11X,'3',
     * 11X,'4',11X,'5',11X,'6',11X,'7',/,
     * T5,125('-'))
508    FORMAT(132('-'),/,
     * T8,'DISTANCE ALONG REACTOR',45X,'CHEMICAL SPECIES',/,
     * 4X,'DIMENSIONLESS',4X,'CENTIMETERS',12X,'1',11X,'2',11X,'3',
     * 11X,'4',11X,'5',11X,'6',11X,'7',11X,'8',/,
     * 132('-'))
509    FORMAT(132('-'),/,
     * T5,'DISTANCE ALONG REACTOR',48X,'CHEMICAL SPECIES',/,
     * 1X,'DIMENSIONLESS',3X,'CENTIMETERS',11X,'1',10X,'2',10X,'3',
     * 10X,'4',10X,'5',10X,'6',10X,'7',10X,'8',10X,'9',/,
     * 132('-'))
510    FORMAT(132('-'),/,
     * T4,'DISTANCE ALONG REACTOR',50X,'CHEMICAL SPECIES',/,
     * 1X,'DIMENSIONLESS',2X,'CENTIMETERS',10X,'1',9X,'2',9X,'3',
```

```
     *  9X,'4',9X,'5',9X,'6',9X,'7',9X,'8',9X,'9',9X,'10',/,
     *  132('-'))
        DO 1 J=1,ICOUNT
        ZX=XX(J)*L(NR)
        IF(NSP.EQ.2) WRITE(23,602) XX(J),ZX,(C(I,J),I=1,NSP)
        IF(NSP.EQ.3) WRITE(23,603) XX(J),ZX,(C(I,J),I=1,NSP)
        IF(NSP.EQ.4) WRITE(23,604) XX(J),ZX,(C(I,J),I=1,NSP)
        IF(NSP.EQ.5) WRITE(23,605) XX(J),ZX,(C(I,J),I=1,NSP)
        IF(NSP.EQ.6) WRITE(23,606) XX(J),ZX,(C(I,J),I=1,NSP)
        IF(NSP.EQ.7) WRITE(23,607) XX(J),ZX,(C(I,J),I=1,NSP)
        IF(NSP.EQ.8) WRITE(23,608) XX(J),ZX,(C(I,J),I=1,NSP)
        IF(NSP.EQ.9) WRITE(23,609) XX(J),ZX,(C(I,J),I=1,NSP)
        IF(NSP.EQ.10) WRITE(23,610) XX(J),ZX,(C(I,J),I=1,NSP)
1       CONTINUE
        IF(NSP.EQ.2) WRITE(23,702)
        IF(NSP.EQ.3) WRITE(23,703)
        IF(NSP.EQ.4) WRITE(23,704)
        IF(NSP.EQ.5) WRITE(23,705)
        IF(NSP.EQ.6) WRITE(23,706)
        IF(NSP.EQ.7) WRITE(23,707)
        IF(NSP.EQ.8) WRITE(23,708)
        IF(NSP.EQ.9) WRITE(23,708)
        IF(NSP.EQ.10) WRITE(23,708)
602     FORMAT(38X,F5.2,8X,F8.1,10X,E9.3,2X,E9.3)
603     FORMAT(13X,F5.2,8X,F8.1,10X,E9.3,2(3X,E9.3))
604     FORMAT(28X,F5.2,8X,F8.1,10X,E9.3,3(3X,E9.3))
605     FORMAT(23X,F5.2,8X,F8.1,10X,E9.3,4(3X,E9.3))
606     FORMAT(18X,F5.2,8X,F8.1,10X,E9.3,5(3X,E9.3))
607     FORMAT(13X,F5.2,8X,F8.1,10X,E9.3,6(3X,E9.3))
608     FORMAT(8X,F5.2,8X,F8.1,10X,E9.3,7(3X,E9.3))
609     FORMAT(5X,F5.2,7X,F8.1,9X,E9.3,8(2X,E9.3))
610     FORMAT(5X,F5.2,6X,F8.1,9X,E9.3,9(1X,E9.3))
702     FORMAT(T30,70('-'))
703     FORMAT(T5,80('-'))
704     FORMAT(T20,90('-'))
705     FORMAT(T15,100('-'))
706     FORMAT(T10,110('-'))
707     FORMAT(T5,125('-'))
708     FORMAT(132('-'))
        CLOSE (UNIT=22)
        RETURN
        END


      SUBROUTINE SOLVLU(LU,B,X,N,NDIM,ORDER)
```

```
C  THIS SUBROUTINE IS USED TO FIND THE SOLUTION TO A SYSTEM
C  OF EQUATIONS, AX = B, AFTER THE LU EQUIVALENT OF A HAS BEEN FOUND.
C  BEFORE USING THIS ROUTINE, THE VECTOR B SHOULD BE SCALED IF MATRIX A WAS
C  SCALED, USING THE SAME SCALE FACTORS. WITHIN THIS ROUTINE, THE ELEMENTS
C  OF B ARE REARRANGED IN THE SAME WAY THAT THE ROWS OF A WERE
C  INTERCHANGED, USING THE ORDER VECTOR WHICH HOLDS THE ROW ORDERINGS.
C  THE SOLUTION IS RETURNED IN X.

C  PARAMETERS ARE -
C   LU   THE LU EQUIVALENT OF THE COEFFICIENT MATRIX
C   B    THE VECTOR OF RIGHT HAND SIDES
C   X    SOLUTION VECTOR
C   N    NUMBER OF EQUATIONS
C   NDIM FIRST DIMENSION OF A IN THE MAIN PROGRAM
C   ORDER INTEGER ARRAY OF ROW ORDER AS REARRANGED DURING PIVOTING

      IMPLICIT REAL*8(A-H,O-Z)
      DOUBLE PRECISION LU(20,20)
      INTEGER ORDER(20)

     DIMENSION B(20),X(20)

     DO 10 I=1,N
     J=ORDER(I)
     X(I)=B(J)
10   CONTINUE

     X(1)=X(1)/LU(1,1)
     DO 50 IROW=2,N
     IM1=IROW-1
     SUM=0.
     DO 40 JCOL=1,IM1
     SUM=SUM+LU(IROW,JCOL)*X(JCOL)
40   CONTINUE
     X(IROW)=(X(IROW)-SUM)/LU(IROW,IROW)
50   CONTINUE

     DO 70 IROW=2,N
     NVBL=N-IROW+1
     SUM=0.
     NP1=NVBL+1
     DO 60 JCOL=NP1,N
     SUM=SUM+LU(NVBL,JCOL)*X(JCOL)
60   CONTINUE
     X(NVBL)=X(NVBL)-SUM
70   CONTINUE
     RETURN
     END
```

```
      SUBROUTINE LUDCMP(A,N,NDIM,ORDER)


C     THIS SUBROUTINE COMPUTES THE L AND U TRIANGULAR MATRICES
C     EQUIVALENT TO THE A MATRIX, SUCH THAT LU=A. THESE MATRICES
C     ARE RETURNED IN THE SPACE OF A, IN COMPACT FORM. THE U MATRIX
C     HAS ONES ON ITS DIAGONAL.
C     PARTIAL PIVOTING IS USED TO GIVE MAXIMUM VALUED ELEMENTS ON THE
C     DIAGONAL OF L. THE ORDER OF THE ROWS AFTER PIVOTING IS RETURNED IN
C     THE INTEGER VECTOR ORDER.
C     VECTORS BEFORE SOLVING THE SYSTEM AX=B

C     PARAMETERS ARE-
C      A     THE N X N MATRIX OF COEFFICIENTS
C      A      THE NUMBER OF EQUATRIONS
C      NDIM THE FIRST DIMENSION OF A IN THE CALLING PROGRAM
C      ORDER INTEGER VECTOR ROW ORDERE AFTER PIVOTING

C  THIS ROUTINE CALLS A SUBROUTINE APVT TO LOCATE THE PIVOT ROW AND
C  MAKE INTERCHANGES

       IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(20,20)
      INTEGER ORDER(20)

      DO 10 I=1,N
      ORDER (I)=I
 10   CONTINUE

      CALL APVT(A, N, NDIM, ORDER,1)

      IF (ABS(A(1,1)) .LT. 1.E-20) GO TO 99
      DO 20 KCOL=2,N
      A(1,KCOL)=A(1,KCOL)/A(1,1)
 20   CONTINUE

      NM1=N-1
      DO 80 JCOL=2,NM1
      JCOL1=JCOL

      JM1=JCOL-1
      DO 50 IROW=JCOL,N
      SUM=0.
      DO 40 KCOL=1,JM1
      SUM=SUM+A(IROW,KCOL)*A(KCOL,JCOL)
```

```
40   CONTINUE
     A(IROW,JCOL)= A(IROW,JCOL)-SUM
50   CONTINUE

     CALL APVT (A,N, NDIM,ORDER, JCOL1)
     IF (ABS(A(JCOL,JCOL)).LT. 1.E-20) GO TO 99

     JP1=JCOL+1
     DO 70 KCOL=JP1,N
     SUM=0.
     DO 60 IROW=1,JM1
     SUM=SUM+A(JCOL,IROW)*A(IROW,KCOL)
60     CONTINUE
     A(JCOL,KCOL) =(A(JCOL,KCOL)-SUM)/A(JCOL,JCOL)
70     CONTINUE
80   CONTINUE

     SUM=0
     DO 90 KCOL=1,NM1
     SUM=SUM+A(N,KCOL)*A(KCOL,N)
90   CONTINUE
       A(N,N)=A(N,N)-SUM
     RETURN

99     WRITE (6,100)
100  FORMAT (1HO,60H VERY SMALL PIVOT ELEMENT INDICATES A NEARLY SINGU
    +LAR MATRIX)
     RETURN
     END



     SUBROUTINE APVT (A,N,NDIM,ORDER,JCOL)


C  THIS SUBROUTINE FINDS THE LARGEST ELEMENT FOR PIVOT IN JCOL OF
C    MATRIX A, PERFORMS INTERCHANGES THE ELEMENTS IN A AND ALSO
C  INTERCHANGES THE ELEMENTS IN THE ORDER VECTOR.

C  PARAMETERS ARE-
C    A  MATRIX OF COEFFICIENTS WHOSE ROWS ARE TO BE INTERCHANGED
C    N  NUMBER OF EQUATIONS
C    NDIM  FIRST DIMENSION OF A IN THE MAIN PROGRAM
C    ORDER INTEGER VECTOR TO HOLD ROW ORDERING
C    JCOL COLUMN OF A BEING SEARCHED FOR PIVOT ELEMENT

       IMPLICIT REAL*8(A-H,O-Z)
```

```
      DIMENSION A(20,20)
        INTEGER ORDER(20)

      IPVT=JCOL
      BIG=ABS(A(JCOL,JCOL))
      JP1=JCOL+1
      DO 10 IROW=JP1,N
      ANEXT=ABS(A(IROW,JCOL))
      IF (ANEXT .LE. BIG) GO TO 10
      BIG=ANEXT
      IPVT=IROW
   10  CONTINUE

      IF (IPVT .EQ. JCOL) GO TO 25
      DO 20 KCOL=1,N
      SAVE =A(JCOL,KCOL)
      A(JCOL,KCOL)=A(IPVT,KCOL)
      A(IPVT,KCOL)=SAVE
   20  CONTINUE

      ISAVE=ORDER(JCOL)
      ORDER(JCOL)=ORDER(IPVT)
      ORDER(IPVT)=ISAVE
   25  RETURN
      END



        SUBROUTINE OUTPUT(IDUM,NR)

C       .....PRINTS OUTPUT.

        IMPLICIT REAL*8(A-H,O-Z)

        DIMENSION CPI(10),CPO(10),C(10,101),RATE(10)

        COMMON/Z15/ TEMPA,AGPA,ALPA,TEMPO,AGPO,ALPO
        COMMON/Z16/ ZRTEMP,ZRAG,ZRAL
        COMMON/Z2/ CPI
        COMMON/Z3/C,RATE,ICOUNT
        COMMON/Z6/ NSPH,NSPR
        COMMON/Z7/ CPO
        COMMON/COST1/ PCCOST,POCOST,PMCOST,PFCOST,PVCOST,RCCOST,ROCOST
     * ,RMCOST,RFCOST,RVCOST
        COMMON/Z20/ FSLUR,FGAS
        COMMON/AN/ N
C
        IF(IDUM-2) 32,33,34
```

```
 32     WRITE(3,200)
 200    FORMAT(80('-'))
        WRITE(3,300)
 300    FORMAT(T33'OUTPUT SUMMARY',/,80('-'),/)
        WRITE(3,99) NSPH,NSPR,FSLUR,FGAS
 99     FORMAT(5X,'Number of Species in Preheater ',T60,I3,/,
    *  5X,'Number of Species in Reactor ',T60,I3,/,
    1  5X,'Flow Rate of Slurry <kg/hr>',T60,E12.4,/,
    2  5X,'Flow Rate of Gas <kg/hr>',T60,E12.4,/)
        TEMPA=TEMPA-273.0
        WRITE(3,100) TEMPA,AGPA,ALPA
 100    FORMAT(/,15X,'Conditions at Preheater Inlet',/,
    *  5X,'Temperature <Degrees C>',T60,E12.4,/,
    *  5X,'Hydrogen Concentration (Gas Phase) <gm/cm**3>',T60,E12.4,/,
    *  5X,'Hydrogen Concentration (Liquid Phase) <gm/cm**3>',T60,E12.4
    *  ,/)
        DO 1 I=1,NSPH
        WRITE(3,102) I,CPI(I)
 102    FORMAT(5X,'Concentration of Species',I3,'  <gm/gm>',T60,E12.4)
 1      CONTINUE
        TEMPO=TEMPO-273.0
        WRITE(3,101) TEMPO,AGPO,ALPO
 101    FORMAT(/,15X,'Conditions at Preheater Outlet (Reactor Inlet)',/,
    *  5X,'Temperature <Degrees C>',T60,E12.4,/,
    *  5X,'Hydrogen Concentration (Gas Phase) <gm/cm**3>',T60,E12.4,/,
    *  5X,'Hydrogen Concentration (Liquid Phase) <gm/cm**3>',T60,E12.4
    *  ,/)

        DO 2 I=1,NSPH
        WRITE(3,103) I,CPO(I)
 103    FORMAT(5X,'Concentration of Species',I3,'  <gm/gm>',T60,E12.4)
 2      CONTINUE
        RETURN

 33     CONTINUE


        K=NR-1
        WRITE(3,104) K,ZRTEMP,ZRAG,ZRAL
 104    FORMAT(/,15X,'Conditions at Outlet of Reactor # ',I3,/,
    *  5X,'Temperature <Degrees C>',T60,E12.4,/,
    *  5X,'Hydrogen Concentration (Gas Phase) <gm/cm**3>',T60,E12.4,/,
    *  5X,'Hydrogen Concentration (Liquid Phase) <gm/cm**3>',T60,E12.4
    *  ,/)
        DO 3 I=1,NSPR
        WRITE(3,105) I,CPO(I)
 105    FORMAT(5X,'Concentration of Species',I3,'  <gm/gm>',T60,E12.4)
 3      CONTINUE
```

```
      RETURN

34    CONTINUE
      IF(NR.NE.1) GO TO 35
      WRITE(3,106)
106   FORMAT('1',25X,'COST ESTIMATES',/)
      WRITE(3,107) PCCOST,POCOST,PMCOST,PFCOST,PVCOST
107   FORMAT(15X,'PREHEATER',/,
     * 5X,'Capital Cost <$>',T60,E12.4,/,
     * 5X,'Operating Cost <$/YR>',T60,E12.4,/,
     * 5X,'Maintenance Cost <$/YR>',T60,E12.4,/,
     * 5X,'Fixed Cost <$/YR>',T60,E12.4,/,
     * 5X,'Variable Cost <$/YR>',T60,E12.4,/)
      RETURN

35    K=NR-1
      WRITE(3,108) K,RCCOST,ROCOST,RMCOST,RFCOST,RVCOST
108   FORMAT(15X,'REACTOR # ',I3,/,
     * 5X,'Capital Cost <$>',T60,E12.4,/,
     * 5X,'Operating Cost <$/YR>',T60,E12.4,/,
     * 5X,'Maintenance Cost <$/YR>',T60,E12.4,/,
     * 5X,'Fixed Cost <$/YR>',T60,E12.4,/,
     * 5X,'Variable Cost <$/YR>',T60,E12.4,/)
      IF(NR.EQ.N) WRITE(3,200)
      CALL STEADY
      RETURN
      END



      SUBROUTINE COST(NR)

C     ....CALCULATE COSTS OF PREHEATER/REACTORS

      IMPLICIT REAL*8(A-H,O-Z)
      DOUBLE PRECISION L

      DIMENSION VG(5),VL(5),L(5),D(5)
      DIMENSION GAMMA(5),GAMS(5),GAMR(5),RL(5),RP(5)
      DIMENSION FLUX(5),HT(5),ALFA(5)

      COMMON/A/ VG,VL,L,D
      COMMON/B/ TIL,HIL,CGI,AGPI,ALPI,TEMPI,TEMW
      COMMON/DDD/ DHSTAR,ESTAR,GAMMA,GAMS,GAMR,RL,RP,
     * FLUX,HT,ALFA
      COMMON/QT/ QT
      COMMON/COST1/ PCCOST,POCOST,PMCOST,PFCOST,PVCOST,RCCOST,ROCOST
     * ,RMCOST,RFCOST,RVCOST
```

```
      IF(NR.NE.1) GO TO 33
      PCCOST=5.0E+05*(3.142/4.0*D(1)**2*L(1)/13.0)**0.6
      PMCOST=0.03*PCCOST*(0.5+0.5*(300.0/365.0))
      PFCOST=0.03*PCCOST*0.5
      PVCOST=(QT*FLUX(1)*TIL*3.142*D(1)*L(1))/0.8*3600.0*24.0*300.0
     * /(252.0*1.0E+06)*5.0+0.03*PCCOST*0.5*(300.0/365.0)
      POCOST=PFCOST+PVCOST
      RETURN
33    CONTINUE

      RCCOST=2.0E+06*(3.142*D(NR)**2*L(NR)/6400.0)**0.6
      RMCOST=0.03*RCCOST*(0.5+0.5*(300.0/365.0))
      RFCOST=0.03*RCCOST*0.5
      RVCOST=0.03*RCCOST*0.5*(300.0/365.0)
      ROCOST=RFCOST+RVCOST
      RETURN
      END

      SUBROUTINE STEADY

C     ... This subroutine determines whether the steady state in the
C     reactor is stable or not. The analysis assumes the liquid phase
C     in the reactor to be completely mixed.

      IMPLICIT REAL*8(A-H,O-Z)

      DIMENSION GAMMA(5),GAMS(5),GAMR(5),RL(5),RP(5)
      DIMENSION AG(5),AL(5),TEMP(5),FLUX(5),HT(5),ALFA(5)

      COMMON/B/ TIL,HIL,CGI,AGPI,ALPI,TEMPI,TEMW
      COMMON/DD/ H8AR
      COMMON/DDD/ DHSTAR,ESTAR,GAMMA,GAMS,GAMR,RL,RP,
     *          FLUX,HT,ALFA
      COMMON/FF/ AG,AL,TEMP

      H=H8AR*EXP(-DHSTAR/(1.0+TEMP(2)))
      HPM=H*DHSTAR/(1.0+TEMP(2))**2.
      TER=1.-EXP(-H*RL(2)/HIL)

      Y1=ESTAR*EXP(-ESTAR/(1.+TEMP(2)))/(1.+TEMP(2))**2
      Y2=H*RL(2)*EXP(-H*RL(2)/HIL)*HPM/HIL
```

```
      T1=1.0+RP(2)*EXP(-ESTAR/(1.0+TEMP(2)))/GAMMA(2)
      T2=TER/(H*GAMMA(2))
      T3=(T1+T2)*AG(1)*Y2/(GAMMA(2)*HIL)

      T4=AL(1)+AG(1)*TER/(GAMMA(2)*HIL)
      T5=Y1*RP(2)/GAMMA(2)+Y2/(H*GAMMA(2))-TER/(GAMMA(2)*H**2.)
      Y4=HBAR*DHSTAR*EXP(-DHSTAR/(1.0+TEMP(2)))/(1.0+TEMP(2))**3
      T6=T4*T5*Y4

      T7=T3-T6
      T8=1.0+RP(2)*EXP(-ESTAR/(1.0+TEMP(2)))/GAMMA(2)+TER/(H*GAMMA(2))
      T8=T8**(-2.)
      Y3=T7*T8

      T9=RP(2)*GAMR(2)*AL(2)*Y1+GAMS(2)*Y2*(H*AG(1)/HIL-AL(2))/H+
     *  (RP(2)*GAMR(2)*EXP(-ESTAR/(1.0+TEMP(2)))+GAMS(2)*TER/H)*Y3-
     *  Y4*GAMS(2)*AL(2)/H**2.

      IF(T9.LT.1.0) RETURN

      WRITE(6,100)
  100  FORMAT(//,5X,'WARNING!! OPERATING CONDITIONS ARE SUCH THAT IF'
     *  ,/,5X,'LIQUID PHASE IN THE REACTOR WAS COMPLETELY MIXED
     *  ,/,5X,'THEN THE STEADY STATE LIES IN A REGION OF MULTIPLE
     *  ,/,5X,'STEADY STATES AND IS AN UNSTABLE STEADY STATE',//)

      RETURN
      END
```

PROGRAM FOR ESTIMATION OF KINETIC
PARAMETERS (FIT.FOR)

```
C     ------------------------------------------------------------
C           PROGRAM TO OPTIMIZE KINETIC CONSTANTS FROM
C                 EXISTING DATA

C     ------------------------------------------------------------



      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4 ZXMN(20),ZXMX(20),ZXCALC(20)
      REAL*4 WORK,FMIN,H,G,W
      EXTERNAL SIMUL
      EXTERNAL SIMUL1
      DIMENSION CPOUT(10),XCALC(20),XMIN(20),XMAX(20)
      DIMENSION WORK(430),IW(20),H(210),G(20),W(60)
      COMMON/FIT/XCALC,XMIN,XMAX,NTRIAL,NDATA,NVAR,CPOUT
      COMMON/B/TIL,HIL,CGI,AGPI,ALPI,TEMPI,TEMW
      COMMON/IND/IND
      CALL INPUT(2,1)
      CALL INPUT1(TIL,1)
      DO 2 I=1,NVAR
      ZXMN(I)=XMIN(I)
      ZXMX(I)=XMAX(I)
 2    ZXCALC(I)=XCALC(I)
      IF(IND.EQ.1) GO TO 20
      GO TO 30
 20    CALL ZXMIN(SIMUL,NVAR,6,100,0,ZXCALC,H,G,FMIN,W,IER)

      GO TO 40
 30    CALL ZXMIN(SIMUL1,NVAR,6,100,0,ZXCALC,H,G,FMIN,W,IER)

 40    CONTINUE
      WRITE(6,200)
200    FORMAT('  JOB DONE')
      CALL EXIT
      END


      SUBROUTINE SIMUL1(N,X,FMIN)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4 X,FMIN
      DIMENSION C(10,101),RATE(10),CPOUT(10)
      DIMENSION XCALC(20),XMIN(20),XMAX(20)
      DIMENSION X(20)
```

```
      COMMON/B/TIL,HIL,CGI,AGPI,ALPI,TEMPI,TEMW
      COMMON/Z6/NSP
      COMMON/FIT/XCALC,XMIN,XMAX,NTRIAL,NDATA,NVAR,CPOUT
      COMMON/Z3/C,RATE,ICOUNT
      DO 3 I=1,N
3     XCALC(I)=X(I)
      NVAR=N
      WRITE(6,50)
50    FORMAT(//,5X,'PARAMETER VALUES',/)
      WRITE(6,51)(X(I),I=1,N)
51    FORMAT(5X,E12.5)

      FMIN=0.
      OPEN(UNIT=2,FILE='FIT02.DAT')
      DO 1 I=1,NDATA
      CALL INPUT(2,2)
      CALL INPUT1(TIL,2)
      CALL PHEATR
      CALL PHEATC
      CALL REACTC
      DO 2 J=1,NSP
      WRITE(6,*) C(J,ICOUNT)
2     FMIN=FMIN+(C(J,ICOUNT)-CPOUT(J))**2
1     CONTINUE
      CLOSE(UNIT=2)
      WRITE(6,52)FMIN
52    FORMAT(' FMIN=', E12.5)
      RETURN
      END



      SUBROUTINE SIMUL(N,X,FMIN)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4 X,FMIN
      EXTERNAL FCN
      DIMENSION C(10,101),RATE(10),CPOUT(10),CTEMP(1)
      DIMENSION XCALC(20),XMIN(20),XMAX(20)
      DIMENSION X(20),Y(3),F(3),AG(2),AL(2),TEMP(2)
      COMMON/B/TIL,HIL,CGI,AGPI,ALPI,TEMPI,TEMW
      COMMON/Z6/NSP
      COMMON/FIT/XCALC,XMIN,XMAX,NTRIAL,NDATA,NVAR,CPOUT
      COMMON/Z3/C,RATE,ICOUNT
      COMMON/TEMP/CTEMP
      COMMON/FF/AG,AL,TEMP

      DO 31 I=1,N
31    XCALC(I)=X(I)
```

```
      NVAR=N
      WRITE(6,50)
50     FORMAT(//,5X,'PARAMETER VALUES',/)
      WRITE(6,51)(X(i),I=1,N)
51     FORMAT(5X,E12.5)

      FMIN=0.
      OPEN(UNIT=2,FILE='FIT02.DAT')
      DO 1 K=1,NDATA
      CALL INPUT(2,2)
      CALL INPUT1(TIL,2)
      CALL PHEATR
      CALL PHEATC
      Y(1)=AL(1)*.6
      Y(2)=TEMP(1)*2.
      CALL NLSYST(FCN,2,100,Y,F,.01,1.0E-4,1.0E-4,I)
      CTEMP(1)=Y(2)
      CALL CSTR
      DO 2 J=1,NSP
      WRITE(6,*) C(J,1)
2      FMIN=FMIN+(C(J,1)-CPOUT(J))**2
      WRITE(6,52) FMIN
1      CONTINUE
      CLOSE(UNIT=2)
52     FORMAT(' FMIN=', E12.5)
      RETURN
      END


      SUBROUTINE PHEATR

      IMPLICIT REAL *8(A-H,O-Z)
      INTEGER RUNGE
      DOUBLE PRECISION L
      DIMENSION VG(2),VL(2),L(2),D(2)
      DIMENSION GAMMA(2),GAMS(2),GAMR(2),
     *  FLUX(2),HT(2),ALFA(2)
      DIMENSION RL(2),RP(2),PEL(2),PEH(2)
      DIMENSION AG(2),AL(2),TEMP(2)
      DIMENSION Y(3),F(3)
      DIMENSION PAG(101),PAL(101),PTEMP(101)

      COMMON /A/ VG,VL,L,D
      COMMON /B/ TIL,HIL,CGI,AGPI,ALPI,TEMPI,TEMW
      COMMON/DD/ HBAR
      COMMON /DDD/ DHSTAR,ESTAR, GAMMA,GAMS,GAMR,RL,RP
     *,FLUX,HT,ALFA
      COMMON /FF/ AG,AL,TEMP
```

```
      COMMON/Z15/ TEMPA,AGPA,ALPA,TEMPO,AGPO,ALPO



      X=0.0
      Y(1)=AGPI
      Y(2)=ALPI
      Y(3)=TEMPI
C
      TEMPA=TIL*(1.0+TEMPI)
      AGPA=CGI*AGPI
      ALPA=CGI*HIL*ALPI
      ICOUNT=1
      PAG(ICOUNT)=Y(1)
      PAL(ICOUNT)=Y(2)
      PTEMP(ICOUNT)=Y(3)
      ZX=X*L(1)
      ZPAG=PAG(ICOUNT)*CGI
      ZPAL=PAL(ICOUNT)*CGI*HIL
      ZPTEMP=PTEMP(ICOUNT)*TIL+TIL-273.0
      HH=0.01

10     K=RUNGE(3,Y,F,X,HH)
      IF(K.NE.1) GO TO 15
      H=HBAR*EXP(-DHSTAR/(1+Y(3)))
      TEMPP=Y(3)
      ZMEW=VISCOS(TEMPP)
      F(1)=-RL(1)*(H*Y(1)/HIL-Y(2))
      F(2)=-F(1)/(GAMMA(1)*HIL)-RP(1)*(EXP(-ESTAR/(1.0+Y(3))))*Y(2)
     * /GAMMA(1)
      F(3)=ALFA(1)*ZMEW**(-0.47)*
     * (TEMW-Y(3))+HIL*RP(1)*GAMR(1)*EXP(-ESTAR/(1+Y(3)))*Y(2)
     * +RL(1)*GAMS(1)*(H*Y(1)/HIL-Y(2))
      GO TO 10
15     CONTINUE
      ICOUNT=ICOUNT+1
      PAG(ICOUNT)=Y(1)
      PAL(ICOUNT)=Y(2)
      PTEMP(ICOUNT)=Y(3)
      ZX=X*L(1)
      ZPAG=PAG(ICOUNT)*CGI
      ZPAL=PAL(ICOUNT)*CGI*HIL
      ZPTEMP=PTEMP(ICOUNT)*TIL+TIL-273.0
      IF(X.GE.1.0) GO TO 16
      GO TO 10
16     CONTINUE
      AG(1)=Y(1)
      AL(1)=Y(2)
```

```
      TEMP(1)=Y(3)
      TEMPO=TIL*(1.0+Y(3))
      AGPO=CGI*Y(1)
      ALPO=CGI*HIL*Y(2)
      RETURN
      END



      FUNCTION REACTR(ALR)



      IMPLICIT REAL *8(A-H,O-Z)
      INTEGER RUNGE
      DIMENSION Y(5),F(5)
      DIMENSION XX(101),RAG(101),RAL(101),RTEMP(101)
      DIMENSION GAMMA(2),GAMS(2),GAMR(2),
     * FLUX(2),HT(2),ALFA(2)
      DIMENSION RL(2),RP(2),PEL(2),PEH(2)
      DIMENSION AG(2),AL(2),TEMP(2)

      COMMON /B/ TIL,HIL,CGI,AGPI,ALPI,TEMPI,TEMW
      COMMON /DUM/ XX,RAG,RAL,RTEMP
      COMMON /NUM/ ICOUNT
      COMMON/DD/ HBAR
      COMMON /DDD/ DHSTAR,ESTAR, GAMMA,GAMS,GAMR,RL,RP
     *  ,FLUX,HT,ALFA
      COMMON /FF/ AG,AL,TEMP
      COMMON /REACT/ AGRI,ALRI,DALRI,TEMPRI,DTEMPI
      COMMON /REACT1/ DTEMP,DALR
      COMMON /REACT2/ MM,J
      COMMON/Z9/PEL,PEH


      ICOUNT=1
      X=0.0
      ALRI=ALR
      DALRI=(ALR-AL(J))*PEL(MM)
      Y(1)=AGRI
      Y(2)=ALR
      Y(3)=DALRI
      Y(4)=TEMPRI
      Y(5)=DTEMPI
      XX(ICOUNT)=X
      RAG(ICOUNT)=Y(1)
      RAL(ICOUNT)=Y(2)
```

```
      RTEMP(ICOUNT)=Y(4)
      HH=0.01
10     K=RUNGE(5,Y,F,X,HH)
      IF(K.NE.1) GO TO 15
      H=HBAR*EXP(-DHSTAR/(1+Y(4)))
      F(1)=-RL(MM)*(H*Y(1)/HIL-Y(2))
      F(2)=Y(3)
      F(3)=(Y(3)+F(1)/(GAMMA(MM)*HIL)+RP(MM)/GAMMA(MM)*(EXP(-ESTAR/
     * (1.0+Y(4))))*Y(2))*PEL(MM)
      F(4)=Y(5)
      F(5)=(Y(5)-(RL(MM)*GAMS(MM)*(H/HIL*Y(1)-Y(2)))
     * -HIL*RP(MM)*GAMR(MM)*EXP(-ESTAR/(1.0+Y(4)))*Y(2))*PEH(MM)
      GO TO 10
15     CONTINUE
      ICOUNT=ICOUNT+1
      XX(ICOUNT)=X
      RAG(ICOUNT)=Y(1)
      RAL(ICOUNT)=Y(2)
      RTEMP(ICOUNT)=Y(4)
      IF(X.GE.1.0) GO TO 16
      GO TO 10
16     CONTINUE
      AG(MM)=Y(1)
      AL(MM)=Y(2)
      DALR=Y(3)
      TEMP(MM)=Y(4)
      DTEMP=Y(5)
      REACTR=DALR
      RETURN
      END



      FUNCTION RUNGE(N,Y,F,X,H)


      IMPLICIT REAL *8(A-H,O-Z)
      INTEGER RUNGE
      DIMENSION PHI(50),SAVEY(50),Y(20),F(20)
      DATA M /0/
      M=M+1
      GO TO (1,2,3,4,5),M
1     RUNGE=1
      RETURN
2     DO 22 J=1,N
      SAVEY(J)=Y(J)
      PHI(J)=F(J)
```

```
22     Y(J)=SAVEY(J)+0.5*H*F(J)
       X=X+0.5*H
       RUNGE=1
       RETURN
3      DO 33 J=1,N
       PHI(J)=PHI(J)+2.0*F(J)
33     Y(J)=SAVEY(J)+0.5*H*F(J)
       RUNGE=1
       RETURN
4      DO 44 J=1,N
       PHI(J)=PHI(J)+2.0*F(J)
44     Y(J)=SAVEY(J)+H*F(J)
       X=X+0.5*H
       RUNGE=1
       RETURN
5      DO 55 J=1,N
55     Y(J)=SAVEY(J)+(PHI(J)+F(J))*H/6.0
       M=0
       RUNGE=0
       RETURN
       END
```

```
       SUBROUTINE HYDRO(N)
       IMPLICIT REAL *8(A-H,O-Z)
       EXTERNAL FEG
       DOUBLE PRECISION L
       DIMENSION VG(2),VL(2),L(2),D(2)
       DIMENSION ZKLA(2),EG(2),PL(2),PH(2)
       COMMON /A/ VG,VL,L,D
       COMMON /B/ TIL,HIL,CGI,AGPI,ALPI,TEMPI,TEMW
       COMMON /C/ RHO,RHOA,RHOG,DI,SIG,R,G
       COMMON /D/ CPG,CPBAR,E,DHR,DHS,ZKBAR
       COMMON /E/ ZKLA,EG
       COMMON /INT/ EEG,DEG
       COMMON /INT1/ ERR
       COMMON /HYD/BO,ZNO,FR
       COMMON/Z9/PL,PH

       DO 10 I=1,N
       V=VG(I)
       DC=D(I)
       ZMEW=VISCOS(TEMPI)
       BO=((DC**2.0)*G*RHOA)/SIG
       FR=V/(G*DC)**0.5
```

```
      SC=ZMEW/RHOA/DI
      ZNO=(G*DC**3.0)*((RHOA/ZMEW)**2.0)
      EEG=0.2
      DEG=0.025
      ERR=1.0E-04
      CALL INTHAL(FEG)
      EG(I)=EEG
      ZKLA(I)=(0.6*DI/DC**2.0)*(SC**0.5)*(BO**0.62)*
     * (ZNO**0.31)*(EG(I)**1.1)
      PL(I)=(13.0*FR*(L(I)/D(I))*(VL(I)/VG(I)))
     * /(1.0+8.0*FR**0.85)
      PH(I)=(RHO*VG(I)*CPG+RHOA*CPBAR*VL(I))
     * /(RHOA*CPBAR*VL(I)/PL(I))
10    CONTINUE
      RETURN
      END



      FUNCTION VISCOS(TEMP)

      IMPLICIT REAL *8(A-H,O-Z)
      DOUBLE PRECISION M
      COMMON/B/ TIL


      A=-1.608
      B=9.8578E+11
      M=4.773
      ATEMP=TEMP*TIL+TIL
      IF(ATEMP.GE.550.0.AND.ATEMP.LE.675.0) GO TO 2
      ZMU=10.0**((A+B/ATEMP**M))
      GO TO 3
2     ZMU=0.14
3     CONTINUE
      VISCOS=ZMU
      RETURN
      END




      SUBROUTINE INTHAL(FUNC)
      IMPLICIT REAL *8(A-H,O-Z)
      EXTERNAL FEG,REACTR
      COMMON /INT/ X,DX
      COMMON /INT1/ ERR
```

```
      U=-1.0
      V=-1.0
1     F=FUNC(X)
      IF(ABS(F).LE.ERR) GO TO 2
      IF(F.GE.0.0) GO TO 3
      IF(V.LE.0.0) GO TO 4
      DX=DX/2.0
4     X=X+DX
      U=1.0
      GO TO 1
3     IF(U.LE.0.0) GO TO 5
      DX=DX/2.0
5     CONTINUE
      X=X-DX
      GO TO 1
2     CONTINUE
      RETURN
      END



      SUBROUTINE INPUT(N,NTIME)
      IMPLICIT REAL *8(A-H,O-Z)
      DOUBLE PRECISION L
      DIMENSION VG(2),VL(2),L(2),D(2)
      DIMENSION GAMMA(2),GAMS(2),GAMR(2),
     * FLUX(2),HT(2),ALFA(2),TIME(2)
      DIMENSION RL(2),RP(2)
      DIMENSION ZKLA(2),EG(2),PL(2),PH(2)

      COMMON/Z9/ PL,PH
      COMMON /A/ VG,VL,L,D
      COMMON /B/ TIL,HIL,CGI,AGPI,ALPI,TEMPI,TEMW
      COMMON /C/ RHO,RHOA,RHOG,DI,SIG,R,G
      COMMON /D/ CPG,CPBAR,E,DHR,DHS,ZKBAR
      COMMON/DD/ HBAR
      COMMON /DDD/ DHSTAR,ESTAR, GAMMA,GAMS,GAMR,RL,RP
     *  ,FLUX,HT,ALFA
      COMMON /E/ ZKLA,EG
      COMMON/Z5/ TIME
      COMMON/IND/IND

      IF(NTIME.EQ.2)GO TO 100
      OPEN(UNIT=1,FILE='FIT01.DAT')
```

```
      READ(1,*) IND
      READ(01,*) (L(I),I=1,N),(D(I),I=1,N)
      READ(01,*)CPG,CPBAR,ZKBAR,E,DHR,DHS,HBAR,DI,SIG,RHOA
      RETURN
100    CONTINUE
      READ(2,*)VOLG,VOLL,TIL,P,FRAC,TH
      TIL=TIL+273.0
      TH=TH+273.0
      RR=82.057
      R=1.98
      G=981.0
      DO 10 I=1,N
      VG(I)=VOLG/(D(I)**2.0*0.786)
      VL(I)=VOLL/(D(I)**2.0*0.786)
10     CONTINUE

      CGI=P/(RR*TIL)*FRAC*2.0
      RHOG=CGI
      AGPI=1.0
      ALPI=0.0
      TEMPI=0.0
      TEMW=(TH-TIL)/TIL
      DHSTAR=DHS/(R*TIL)
      ESTAR=E/(R*TIL)
      HIL=HBAR*(EXP(-DHSTAR))
      CALL HYDRO(N)

      DO 11 I=1,N
      TIME(I)=L(I)*(1.-EG(I))/VL(I)
      RL(I)=ZKLA(I)*L(I)*HIL/VG(I)
      GAMMA(I)=VL(I)/VG(I)
      RP(I)=ZKBAR*(1-EG(I))*L(I)/VG(I)
      FLUX(I)=VG(I)*RHOG*CPG+VL(I)*RHOA*CPBAR
      GAMS(I)=(CGI*VG(I)*DHS)/TIL/FLUX(I)
      GAMR(I)=(CGI*VG(I)*DHR)/TIL/FLUX(I)
      BETA=0.0045
      HT(I)=BETA*VL(I)**0.8/D(I)**0.2
      ALFA(I)=HT(I)/D(I)*L(I)/FLUX(I)
11     CONTINUE
      RETURN
      END




      FUNCTION FEG(X)
      IMPLICIT REAL *8(A-H,O-Z)
```

```
      COMMON /HYD/ BO,ZNO,FR
      F1=X/((1.0-X)**4.0)
      F2=0.2*(BO**0.125)*(ZNO**0.083)*FR
      FEG=F1-F2
      RETURN
      END



      SUBROUTINE REATEM(N)
C
      IMPLICIT REAL *8(A-H,O-Z)
      DOUBLE PRECISION L
      EXTERNAL REACTR
      DIMENSION AG(2),AL(2),TEMP(2),L(2),D(2),VG(2),VL(2)
      DIMENSION XX(101),RAG(101),RAL(101),RTEMP(101)
      DIMENSION ZKLA(2),EG(2),PL(2),PH(2),RRTEMP(101)
      COMMON /A/ VG,VL,L,D
      COMMON /B/ TIL,HIL,CGI,AGPI,ALPI,TEMPI,TEMW
      COMMON /REACT/AGRI,ALRI,DALRI,TEMPRI,DTEMPI
      COMMON /DUM/ XX,RAG,RAL,RTEMP
      COMMON /NUM/ ICOUNT
      COMMON /REACT1/ DTEMP,DALR
      COMMON /REACT2/ M,J
      COMMON /INT/ ALR,DAL
      COMMON /INT1/ ERR
      COMMON /FF/ AG,AL,TEMP
      COMMON /E/ ZKLA,EG
      COMMON/Z8/ RRTEMP
      COMMON/Z16/ ZRTEMP,ZRAG,ZRAL
      COMMON/Z9/PL,PH
      ERR= 0.01
      DO 10 I=2,N
      M=I
      U=-1.0
      V=-1.0
      J=I-1
      TEMPRI=TEMP(J)*1.5
      DTEMPI=(TEMPRI-TEMP(J))*PH(I)
      DTMPI=0.1
      AGRI=AG(J)
      ALRI=AL(J)*0.9
      ALR=ALRI
      DAL=0.2
 1    CONTINUE
      CALL INTHAL(REACTR)
```

```
      DAL=0.01
      IF(ABS(DTEMP).LE.0.001) GO TO 2
      IF(DTEMP.GE.0.0)GO TO 3
      IF(V.LE.0.0)GO TO 4
      ERR=0.0001
      DTMPI=DTMPI/2.0
4     TEMPRI=TEMPRI+DTMPI
      DTEMPI=(TEMPRI-TEMP(J))*PH(I)
      U=1.0
      GO TO 1
3     IF(U.LE.0.0)GO TO 5
      ERR=0.0001
      DTMPI=DTMPI/2.0
5     TEMPRI=TEMPRI-DTMPI
      DTEMPI=(TEMPRI-TEMP(J))*PH(I)
      GO TO 1
2     CONTINUE
      DO 23 II=1,ICOUNT
      RRTEMP(II)=RTEMP(I)
      ZX=XX(II)*L(2)
      ZRAG=RAG(II)*CGI
      ZRAL=RAL(II)*CGI*HIL
      ZRTEMP=RTEMP(II)*TIL+TIL-273.0
23    CONTINUE
10    CONTINUE
      RETURN
      END


      SUBROUTINE INPUT1(TIL,NTIME)
      IMPLICIT REAL *8(A-H,O-Z)
      DOUBLE PRECISION K,KBAR,L,KBAR1,KBAR2
      DIMENSION K(20,20),KBAR(20,20),EC(20,20),COEFF(20,20),CPI(10)
      DIMENSION KBAR1(20,20),KBAR2(20,20),EC1(20,20),EC2(20,20)
      DIMENSION CPOUT(10)
      DIMENSION II(20),JJ(20),XCALC(20),XMIN(20),XMAX(20)
      DIMENSION ECSTAR(20,20),PL(2),PH(2),VL(2),VG(2)
      DIMENSION D(2),L(2),X(20),Y(20),TABLE(20,20)
      CHARACTER*20 NAME(10)
      COMMON/Z1/ KBAR,ECSTAR,COEFF
      COMMON/Z2/ CPI
      COMMON/Z6/ NSP
      COMMON/FIT/XCALC,XMIN,XMAX,NTRIAL,NDATA,NVAR,CPOUT
      COMMON/Z9/PL,PH
      COMMON/A/VG,VL,L,D
      COMMON/C/ RHO,RHOA,RHOG,DI,SIG,R,G
      COMMON/D/CPG,CPBAR,E,DHR,DHS,ZKBAR
```

```
      COMMON/PINT/ X,Y,TABLE,NDT,MNDT
      COMMON/IND/IND
      DATA R/1.98/
C
C
      IF(NTIME.EQ.2)GO TO 100
      DO 1 I=1,20
      DO 1 J=1,20
      K(I,J)=0.0
      KBAR(I,J)=0.0
      EC(I,J)=0.0
      COEFF(I,J)=0.0
 1    CONTINUE
C
C
      READ(01,*) NSP
      DO 2 I=1,NSP
      READ(01,102) NAME(I)
 102   FORMAT(A20)
 2    CONTINUE
      READ(01,*) NK
      DO 4 I=1,NK
      READ(01,*) II(I)
      READ(01,*) JJ(I)
      READ(1,*)KBAR1(II(I),JJ(I)),KBAR2(II(I),JJ(I))
      XMIN(2*I-1)=KBAR1(II(I),JJ(I))
      XMAX(2*I-1)=KBAR2(II(I),JJ(I))
      READ(1,*)EC1(II(I),JJ(I)),EC2(II(I),JJ(I))
      XMIN(2*I)=EC1(II(I),JJ(I))
      XMAX(2*I)=EC2(II(I),JJ(I))
      XCALC(2*I-1)=(XMIN(2*I-1)+XMAX(2*I-1))/2.
      XCALC(2*I)=(XMIN(2*I)+XMAX(2*I))/2.
      COEFF(II(I),JJ(I))=1.0
 4    CONTINUE
      NVAR=2*NK

      READ(1,*)NDATA
      CLOSE(UNIT=1)
      RETURN

 100   CONTINUE

      DO 250 I=1,NK
      EC(II(I),JJ(I))=XCALC(2*I)
      KBAR(II(I),JJ(I))=XCALC(2*I-1)
      ECSTAR(II(I),JJ(I))=EC(II(I),JJ(I))/(R*TIL)
250   CONTINUE
      DO 5 I=1,NSP
```

```
      READ(02,*) CPI(I)
5     CONTINUE
      DO 6 I=1,NSP
      READ(02,*) CPOUT(I)
6     CONTINUE
      IF(IND.EQ.1) RETURN
      READ(02,*) NDT
      DO 10 I=1,NDT
      READ(02,*) X(I),Y(I)
      X(I)=X(I)/L(2)
      Y(I)=Y(I)+273.
      Y(I)=(Y(I)-TIL)/TIL
10    CONTINUE
      MNDT=NDT-1
      CALL DTABLE(X,Y,TABLE,NDT,MNDT,TRUBL,20)
      IF(TRUBL.NE.0.0) CALL EXIT

      RETURN
      END


      SUBROUTINE EQTNS(TEMP)
      IMPLICIT REAL *8(A-H,O-Z)
      DOUBLE PRECISION K,KBAR
      DIMENSION K(20,20),KBAR(20,20),ECSTAR(20,20),RATE(10)
      DIMENSION COEFF(20,20),TEMP(101)
      DIMENSION C(10,101)
      COMMON/Z1/ KBAR,ECSTAR,COEFF
      COMMON/Z3/ C,RATE,ICOUNT
      COMMON/Z6/ NSP
C
C
      DO 1 IJ=1,NSP
      RATE(IJ)=0.0
      DO 1 I=1,NSP
      DO 1 J=1,NSP
      IF(COEFF(I,J).EQ.0.0) GO TO 1
      IF((IJ.NE.I).AND.(IJ.NE.J)) GO TO 1
      K(I,J)=KBAR(I,J)*EXP(-ECSTAR(I,J)/(1.0+TEMP(ICOUNT)))
      IF(IJ.EQ.I) RATE(IJ)=RATE(IJ)-COEFF(I,J)*K(I,J)*C(I,ICOUNT)
      IF(IJ.EQ.J) RATE(IJ)=RATE(IJ)+COEFF(I,J)*K(I,J)*C(I,ICOUNT)
1     CONTINUE
      RETURN
      END


      SUBROUTINE PHEATC
```

```
      IMPLICIT REAL *8(A-H,O-Z)
      INTEGER RUNGE
      DIMENSION CPI(10),CPO(20),RATE(10),PTEMP(101),C(10,101),TIME(2)
      DIMENSION Y(20),F(20),XX(101)
      COMMON/Z2/ CPI
      COMMON/Z3/ C,RATE,ICOUNT
      COMMON/Z4/ PTEMP
      COMMON/Z5/ TIME
      COMMON/Z6/ NSP
      COMMON/Z7/ CPO
      COMMON/Z13/ XX


      X=0.0
      ICOUNT=1
      XX(ICOUNT)=X
      DO 25 I=1,NSP
      Y(I)=CPI(I)
      C(I,ICOUNT)=Y(I)
25     CONTINUE
      HH=0.01
       •
10     K=RUNGE(NSP,Y,F,X,HH)
      IF(K.NE.1) GO TO 15
      CALL EQTNS(PTEMP)
      DO 26 I=1,NSP
      F(I)=RATE(I)*TIME(1)
      C(I,ICOUNT)=Y(I)
26     CONTINUE
      GO TO 10
15     CONTINUE
      ICOUNT=ICOUNT+1
      XX(ICOUNT)=X
      DO 27 I=1,NSP
      C(I,ICOUNT)=Y(I)
27     CONTINUE
      IF(X.GE.1.0) GO TO 16
      GO TO 10
16     CONTINUE
      DO 28 I=1,NSP
      CPO(I)=Y(I)
      CPO(I+NSP)=F(I)
28     CONTINUE
      RETURN
      END

      SUBROUTINE REACTC
```

```
      IMPLICIT REAL *8(A-H,O-Z)
      INTEGER RUNGE
      INTEGER ORDER
      DIMENSION CPO(20),RATE(10),RTEMP(101),C(10,101),TIME(2)
      DIMENSION Y(20),F(20),A(20,20),PEL(2),B(20),ZETA(20)
      DIMENSION ORDER(20),GUESS(20,20),FACTOR(50),RRATE(10)
      DIMENSION AA(20,20),XX(101),XD(20),YD(20),TABLE(20,20)
      COMMON/Z7/ CPO
      COMMON/Z13/ XX
      COMMON/Z9/ PEL,PEH
      COMMON/Z3/ C,RATE,ICOUNT
      COMMON/Z8/ RTEMP
      COMMON/Z5/ TIME
      COMMON/Z6/ NSP
      COMMON/PINT/XD,YD,TABLE,NDT,MNDT

      ITEST=0
      ICOUNT=1
      X=0.0
      XX(ICOUNT)=X
      XARG=X
      RTEMP(ICOUNT)=FNEWT(XD,YD,TABLE,NDT,MNDT,7,XARG,TRUBL,20)

      DO 8 I=1,NSP
      B(I)=CPO(I)
      C(I,ICOUNT)=CPO(I)
      B(I+NSP)=0.0
8     CONTINUE
      CALL EQTNS(RTEMP)
      DO 90 J=1,NSP
      RRATE(J)=RATE(J)
90    CONTINUE
      HH=0.1
      NNSP=2*NSP
      SEED=1
      DO 1 I=1,NNSP
      ICOUNT=1
      X=0.0
      XX(ICOUNT)=X
      DO 20 J=1,NNSP
      FACTOR(J)=RAN(SEED)
20    CONTINUE
      DO 2 N=1,NSP
      Y(N)=CPO(N)*(1+(-1)**I*FACTOR(N)*0.2)
      Y(N+NSP)=RRATE(N)/ABS(RRATE(N))*(0.5*FACTOR(N+NSP))
2     CONTINUE
      DO 43 N=1,NSP
```

```
      A(N,I)=Y(N)-Y(N+NSP)/PEL(2)
      C(N,ICOUNT)=Y(N)
43     CONTINUE
      DO 6 J=1,NNSP
      GUESS(J,I)=Y(J)
6      CONTINUE

10     CONTINUE
      K=RUNGE(NNSP,Y,F,X,HH)
      IF(K.NE.1) GO TO 15
      XARG=X
      RTEMP(ICOUNT)=FNEWT(XD,YD,TABLE,NDT,MNDT,7,XARG,TRUBL,20)
      CALL EQTNS(RTEMP)
      DO 26 J=1,NSP
      F(J)=Y(J+NSP)
      F(J+NSP)=PEL(2)*(Y(J+NSP)-RATE(J)*TIME(2))
      C(J,ICOUNT)=Y(J)
26     CONTINUE
      GO TO 10
15     CONTINUE
      ICOUNT=ICOUNT+1
      XX(ICOUNT)=X
      DO 17 J=1,NSP
      C(J,ICOUNT)=Y(J)
17     CONTINUE
      IF(X.GE.1.0) GO TO 16
      GO TO 10
16     CONTINUE
      IF(ITEST.NE.0) GO TO 12
      DO 55 J=1,NSP
      A(NSP+J,I)=Y(NSP+J)
55     CONTINUE
1      CONTINUE
      DO 69 I1=1,NNSP
      DO 69 I2=1,NNSP
      AA(I1,I2)=A(I1,I2)
69     CONTINUE
      CALL LUDCMP(A,NNSP,NNSP,ORDER)
      CALL SOLVLU(A,B,ZETA,NNSP,NNSP,ORDER)
C
C
      DO 70 I1=1,NNSP
      BTOT=0.0
      DO 71 I2=1,NNSP
      BTOT=BTOT+AA(I1,I2)*ZETA(I2)
71     CONTINUE
70     CONTINUE
      ITEST=1
```

```
      ICOUNT=1
      X=0.0
      DO 7 J=1,NNSP
      Y(J)=0.0
      DO 7 I=1,NNSP
      Y(J)=Y(J)+ZETA(I)*GUESS(J,I)
 7    CONTINUE
      DO 18 I=1,NSP
      C(I,ICOUNT)=Y(I)
 18   CONTINUE
      GO TO 10
 12   CONTINUE
      RETURN
      END


      SUBROUTINE SOLVLU(LU,B,X,N,NDIM,ORDER)
        IMPLICIT REAL*8(A-H,O-Z)
      DOUBLE PRECISION LU(20,20)
      DIMENSION B(20),X(20)
      INTEGER ORDER(20)
      DO 10 I=1,N
      J=ORDER(I)
      X(I)=B(J)
 10   CONTINUE
      X(1)=X(1)/LU(1,1)
      DO 50 IROW=2,N
      IM1=IROW-1
      SUM=0.
      DO 40 JCOL=1,IM1
      SUM=SUM+LU(IROW,JCOL)*X(JCOL)
 40   CONTINUE
      X(IROW)=(X(IROW)-SUM)/LU(IROW,IROW)
 50   CONTINUE
      DO 70 IROW=2,N
      NVBL=N-IROW+1
      SUM=0.
      NP1=NVBL+1
      DO 60 JCOL=NP1,N
      SUM=SUM+LU(NVBL,JCOL)*X(JCOL)
 60   CONTINUE
      X(NVBL)=X(NVBL)-SUM
 70   CONTINUE
      RETURN
      END

      SUBROUTINE LUDCMP(A,N,NDIM,ORDER)
```

```
      IMPLICIT REAL*8(A-H,O-Z)
     DIMENSION A(20,20)
     INTEGER ORDER(20)
     DO 10 I=1,N
     ORDER (I)=I
10   CONTINUE
     CALL APVT(A, N, NDIM, ORDER,1)
     IF (ABS(A(1,1)) .LT. 1.E-20) GO TO 99
     DO 20 KCOL=2,N
     A(1,KCOL)=A(1,KCOL)/A(1,1)
20   CONTINUE
     NM1=N-1
     DO 80 JCOL=2,NM1
     JCOL1=JCOL
     JM1=JCOL-1
     DO 50 IROW=JCOL,N
     SUM=0.
     DO 40 KCOL=1,JM1
     SUM=SUM+A(IROW,KCOL)*A(KCOL,JCOL)
40   CONTINUE
     A(IROW,JCOL)= A(IROW,JCOL)-SUM
50   CONTINUE
     CALL APVT (A,N, NDIM,ORDER, JCOL1)
     IF (ABS(A(JCOL,JCOL)).LT. 1.E-20) GO TO 99
     JP1=JCOL+1
     DO 70 KCOL=JP1,N
     SUM=0.
     DO 60 IROW=1,JM1
     SUM=SUM+A(JCOL,IROW)*A(IROW,KCOL)
60     CONTINUE
     A(JCOL,KCOL) =(A(JCOL,KCOL)-SUM)/A(JCOL,JCOL)
70     CONTINUE
80   CONTINUE
     SUM=0
     DO 90 KCOL=1,NM1
     SUM=SUM+A(N,KCOL)*A(KCOL,N)
90   CONTINUE
       A(N,N)=A(N,N)-SUM
     RETURN
99     write (6,100)
100  FORMAT (1HO,60H VERY SMALL PIVOT ELEMENT INDICATES A NEARLY SINGU
    +LAR MATRIX)
     RETURN
     END

     SUBROUTINE APVT (A,N,NDIM,ORDER,JCOL)

       IMPLICIT REAL*8(A-H,O-Z)
```

```
      DIMENSION A(20,20)
        INTEGER ORDER(20)
      IPVT=JCOL
      BIG=ABS(A(JCOL,JCOL))
      JP1=JCOL+1
      DO 10 IROW=JP1,N
      ANEXT=ABS(A(IROW,JCOL))
      IF (ANEXT .LE. BIG) GO TO 10
      BIG=ANEXT
      IPVT=IROW
   10  CONTINUE
      IF (IPVT .EQ. JCOL) GO TO 25
      DO 20 KCOL=1,N
      SAVE =A(JCOL,KCOL)
      A(JCOL,KCOL)=A(IPVT,KCOL)
      A(IPVT,KCOL)=SAVE
   20  CONTINUE
      ISAVE=ORDER(JCOL)
      ORDER(JCOL)=ORDER(IPVT)
      ORDER(IPVT)=ISAVE
   25  RETURN
      END

       SUBROUTINE NLSYST (FCN, N, MAXIT, X, F, DELTA, XTOL, FTOL, I)

       IMPLICIT REAL*8(A-H,O-Z)

C  THIS SUBROUTINE SOLVES A SYSTEM OF N NON-LINEAR EQUATIONS BY
C  NEWTON'S METHOD.  THE PARTIAL DERIVATIVES OF THE FUNCTIONS
C  ARE ESTIMATED BY DIFFERENCE QUOTIENTS WHEN A VARIABLE IS PERTURBED
C  BY AN AMOUNT EQUAL TO DELTA (DELTA IS ADDED.)  THIS IS DONE FOR EACH
C  VARIABLE IN EACH FUNCTION.  INCREMENTS TO IMPROVE THE ESTIMATES
C  FOR THE X-VALUES ARE COMPUTED FROM A SYSTEM OF EQUATIONS USING
C  SUBROUTINE ELIM.
C  PARAMETERS ARE
C       FCN     SUBROUTINE THAT COMPUTES VALUES OF THE FUNCTIONS.  MUST
C               BE DECLARED EXTERNAL IN THE CALLING PROGRAM
C       N       THE NUMBER OF EQUATIONS
C       MAXIT   LIMIT TO THE NUMBER OF ITERATIONS THAT WILL BE USED
C       X       ARRAY TO HOLD  X VALUES.  INITIALLY THIS ARRAY HOLDS THE
C               INITIAL GUESSES.  IT RETURNS THE FINAL VALUES.
C       F       AN ARRAY THAT HOLDS VALUES OF THE FUNCTIONS
C       DELTA   A SMALL VALUE USED TO PERTURB THE X VALUES SO PARTIAL
C               DERIVATIVES CAN BE COMPUTED BY DIFFERENCE QUOTIENT
C       XTOL    TOLERANCE VALUE FOR CHANGE IN X VALUES TO STOP
C               ITERATIONS.  WHEN THE LARGEST CHANGE IN ANY X MEETS
C               XTOL, THE SUBROUTINE TERMINATES.
C       FTOL    TOLERANCE VALUE ON F TO TERMINATE.  WHEN THE LARGEST F
```

```
C                VALUE IS LESS THAN FTOL, SUBROUTINE TERMINATES.
C       I      RETURNS VALUES TO INDICATE HOW THE ROUTINE TERMINATED.
C                I = 1    XTOL WAS MET
C                I = 2    FTOL WAS MET
C                I = -1   MAXIT EXCEEDED BUT TOLERANCES NOT MET
C                I = -2   VERY SMALL PIVOT ENCOUNTERED IN GAUSSIAN
C                          ELIMINATION STEP.  NO RESULTS OBTAINED
C                I = -3   INCORRECT VALUE OF N WAS SUPPLIED.  N MUST BE
C                          BETWEEN 2 AND 10

      DIMENSION X(N), F(N), A(10,11), XSAVE(10), FSAVE(10)
      LOGICAL PRINT
C  CHECK VALIDITY OF VALUE OF N
      IF (N .LT. 2 .OR. N .GT. 10) GO TO 999
      PRINT = .TRUE.
      IF (I.NE. 0) PRINT = .FALSE.
C  BEGIN ITERATIONS
C  SAVE X VALUES, THEN GET F VALUES
      NP = N + 1
      IDT=0
      DO 100 IT = 1,MAXIT
        DO 10 IVBL = 1, N
            XSAVE(IVBL) = X(IVBL)
10       CONTINUE
        CALL FCN (X, F)
C  TEST F VALUES AND SAVE THEM
        ITEST = 0
        DO 20 IFCN = 1,N
            IF (ABS(F(IFCN)) .GT. FTOL)  ITEST = ITEST + 1
            FSAVE(IFCN) = F(IFCN)
20       CONTINUE
C  PRINT CURRENT VALUES IF PRINT IS .TRUE.
        IF (.NOT. PRINT)  GO TO 30
            WRITE(6,1000) IT, X
1000         FORMAT (1H0, 23HAFTER ITERATION NUMBER , I3,
     +                  19H X AND F VALUES ARE  / 1H0, 10F13.5)
            WRITE (6,1001) F
1001         FORMAT (1H0, 10F13.5)
C  SEE IF FTOL IS MET.  IF NOT, CONTINUE.  IF SO, SET I = 2 AND RETURN.
30       IF (ITEST .NE. 0) GO TO 35
        I = 2
        RETURN
C  THIS DOUBLE LOOP COMPUTES THE PARTIAL DERIVATIVES OF EACH FUNCTION
C  FOR EACH VARIABLE AND STORES THEM IN A COEFFICIENT ARRAY
35      DO 50 JCOL = 1,N
            X(JCOL) = XSAVE(JCOL) + DELTA
            CALL FCN (X, F)
            DO 40 IROW = 1,N
```

```
                A(IROW,JCOL) = (F(IROW) - FSAVE(IROW))/DELTA
40              CONTINUE
C  RESET X VALUES FOR NEXT COLUMN OF PARTIALS
                X(JCOL) = XSAVE(JCOL)
50      CONTINUE
C  NOW WE PUT NEGATIVE OF F VALUES AS RIGHT HAND SIDES AND CALL ELIM
                DO 60 IROW = 1,N
                  A(IROW,NP) = -FSAVE(IROW)
60      CONTINUE
        CALL ELIM ( A,N,NP,10 )
C  BE SURE THAT COEFFICEINT MATRIX NOT TOO ILL-CONDITIONED
        DO 70 IROW = 1,N
                IF (ABS(A(IROW,IROW)) .LE. 1.E-5) GO TO 998
70      CONTINUE
C  APPLY THE CORRECTIONS TO THE X VALUES, ALSO SEE IF XTOL IS MET.
                ITEST = 0
                DO 80 IVBL = 1,N
                X(IVBL) = XSAVE(IVBL) + A(IVBL,NP)
        IF (ABS(A(IVBL,NP)) .GT. XTOL)  ITEST = ITEST + 1
80      CONTINUE
C  IF XTOL IS MET, PRINT LAST VALUES AND RETURN, ELSE DO
C  ANOTHER ITERATION
                IF (ITEST .EQ. 0) GO TO 997
100     CONTINUE
C  WHEN WE HAVE DONE MAXIT ITERATIONS, SET I = -1 AND RETURN
        I = -1
        RETURN
C  WE COME HERE WHEN XTOL IS MET.  PRINT LAST VALUES, SET I = 1, RETURN
997     I = 1
        IF (.NOT. PRINT) GO TO 110
110     RETURN
C  WE COME HERE WHEN PARTIALS FORM A NEARLY SINGULAR MATRIX.  SET I = -2
C  PRINT MESSAGE, RETURN
998     I = -2
        WRITE (6,1003)
1003  FORMAT (1H0,  44HCANNOT SOLVE SYSTEM. MATRIX NEARLY SINGULAR )
        RETURN
C  WE COME HERE WHEN NUMBER OF EQUATIONS PASSED TO NLSYST IS INVALID.
C  PRINT MESSAGE, RETURN
999     I = -3
        WRITE (6,1004) N
1004    FORMAT  (1H0, 81HNUMBER OF EQUATIONS PASSED TO NLSYST IS INVALID.
     + MUST BE 1 < N < 11.  VALUE WAS , I3)
        RETURN
        END

      SUBROUTINE ELIM (AB,N,NP,NDIM)
```

```
      IMPLICIT REAL*8(A-H,O-Z)

     DIMENSION AB(NDIM,NP)
C THIS SUBROUTINE SOLVES A SET OF LINEAR EQUATIONS
C THE GUASS ELIMINATION METHOD IS USED
C MULTIPLE RIGHT HAND SIDES ARE PERMITTED,THEY SHOULD BE SUPPLIED
C AS COLUMNS THAT AUGMENT THE COEFFICIENT MATRIX.
C PARAMETERS ARE -
C  AB     COEFFICIENT MATRIX AUGMENTED WITH R.H.S. VECTORS
C  N      NUMBER OF EQUATIONS
C  NP     TOTAL NUMBER OF COLUMNS IN THE AUGMENTED MATRIX
C  NDIM   FIRST DIMENSION OF MATRIX AB IN THE CALLING PROGRAM.
C THE SOLUTION VECTOR(S) ARE RETURNED IN THE AUGMENTATION
C COLUMNS OF AB.
C BEGIN THE REDUCTION
     NM1=N-1
     DO 35 I=1,NM1
C FIND THE ROW NUMBER OF THE PIVOT ROW, WE WILL THEN
C INTERCHANGE ROWS TO PUT THE PIVOT ELEMENT ON THE DIAGONAL.
     IPVT=I
     IP1=I+1
     DO 10 J=IP1,N
       IF(ABS(AB(IPVT,I)) .LT. ABS(AB(J,I)))  IPVT=J
10    CONTINUE
C CHECK THE PIVOT ELEMENT TO BE SURE THAT IT IS NOT TO SMALL, IF SO
C PRINT A MESSAGE AND RETURN
     IF ((ABS(AB(IPVT,I)) .LT. 1.E-5))  GO TO 99
C NOW INTERCHANGE, EXCEPT IF THE PIVOT ELEMENT IS ALREADY ON
C THE DIAGONAL, DON'T NEED TO.
     IF (IPVT.EQ.I)  GO TO 25
     DO 20 JCOL=I,NP
       SAVE=AB(I,JCOL)
       AB(I,JCOL)=AB(IPVT,JCOL)
       AB(IPVT,JCOL)=SAVE
20    CONTINUE
C NOW REDUCE ALL ELEMENTS BELOW THE DIAGONAL IN THE I-TH ROW.
C CHECK FIRST TO SEE IF A ZERO IS ALREADY PRESENT.  IF SO
C CAN SKIP REDUCTION FOR THAT ROW
25    DO 32 JROW=IP1,N
       IF (AB(JROW,I) .EQ. 0) GO TO 32
       RATIO=AB(JROW,I)/AB(I,I)
       DO 30 KCOL=IP1,NP
         AB(JROW,KCOL)=AB(JROW,KCOL) - RATIO*AB(I,KCOL)
30      CONTINUE
32    CONTINUE
35    CONTINUE
C WE STILL NEED TO CHECK A(N,N) FOR SIZE.
     IF (ABS(AB(N,N)) .LT. 1.E-5)  GO TO 99
```

```
C NOW WE BACK SUBSTITUTE.
      NP1=N+1
      DO 50 KCOL=NP1,NP
        AB(N,KCOL)=AB(N,KCOL)/AB(N,N)
        DO 45 J=2,N
          NVBL=NP1-J
          L=NVBL+1
          VALUE=AB(NVBL,KCOL)
          DO 40 K=L,N
            VALUE=VALUE-AB(NVBL,K)*AB(K,KCOL)
40        CONTINUE
         AB(NVBL,KCOL)=VALUE/AB(NVBL,NVBL)
45        CONTINUE
50        CONTINUE
      RETURN
C MESSAGE FOR A SINGULAR MATRIX
99     WRITE (6,100)
100    FORMAT(1H0, 57HSOLUTION NOT FEASIBLE.  A NEAR ZERO PIVOT WAS ENCOUN
     +TRED. )
      RETURN
      END


        SUBROUTINE FCN(X,F)

        IMPLICIT REAL*8(A-H,O-Z)
        DIMENSION X(10),F(10),GAMMA(2),GAMS(2),GAMR(2),FLUX(2),HT(2),
     *          ALFA(2),RL(2),RP(2)
        DIMENSION AG(2),AL(2),TEMP(2)

        COMMON/DD/HBAR
        COMMON/FF/AG,AL,TEMP
        COMMON/DDD/ DHSTAR,ESTAR,GAMMA,GAMS,GAMR,RL,RP,
     *           FLUX,HT,ALFA
        COMMON/B/TIL,HIL,CGI,AGPI,ALPI,TEMPI,TEMW
        COMMON/IDT/IDT

        H=HBAR*EXP(-DHSTAR/(1.+X(2)))
        T1=(H*AG(1)/HIL-X(1))*(1.0-EXP(-RL(2)*H/HIL))
        T2=EXP(-ESTAR/(1.+X(2)))

        F(1)=AL(1)-X(1)-RP(2)*T2*X(1)/GAMMA(2)
     *  +T1/(H*GAMMA(2))
        F(2)=X(2)-TEMP(1)-GAMR(2)*RP(2)*T2*X(1)*HIL-
     *  GAMS(2)*T1*HIL/H

C10      H=HBAR*EXP(-DHSTAR/(1.+X(3)))
C        F(1)=X(1)-AG(1)+RL(2)*(H*X(1)/HIL-X(2))
```

```
C       F(2)=AL(1)-X(2)+RL(2)*(H*X(1)/HIL-X(2))/(GAMMA(2)*HIL)
C     *          -RP(2)*EXP(-ESTAR/(1.+X(3)))*X(2)/GAMMA(2)
C       F(3)=TEMP(1)-X(3)+RL(2)*GAMS(2)*(H*X(1)/HIL-X(2))
C     *          +RP(2)*GAMR(2)*HIL*EXP(-ESTAR/(1.+X(3)))*X(2)


      RETURN
      END

      SUBROUTINE CSTR

      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION CPO(20),RATE(10),C(10,101),TIME(2),TEMP(1)
      DIMENSION XOLD(10),FOLD(10)
      COMMON/Z3/ C,RATE,ICOUNT
      COMMON/Z5/TIME
      COMMON/Z6/NSP
      COMMON/Z7/CPO
      COMMON/TEMP/TEMP

      ICOUNT=1
C     DELC=.0001
      SEED=1.
      DO 10 I=1,NSP
10    C(I,ICOUNT)=CPO(I)*RAN(SEED)

      DO 11 I=1,NSP
      XL=1.0E-6
      XR=1.0
      C(I,ICOUNT)=XL
      CALL EQTNS(TEMP)
      FXL=CPO(I)-XL+RATE(I)*TIME(2)
      C(I,ICOUNT)=XR
      CALL EQTNS(TEMP)
      FXR=CPO(I)-XR+RATE(I)*TIME(2)

      IF (FXL*FXR) 5,3,2
2     WRITE(6,201)
      RETURN

3     ITER=1
      IF(FXL.NE.0.) GO TO 4
      C(I,ICOUNT)=XL
      FX2=0.0
      GO TO 11
4     C(I,ICOUNT)=XR
      FX2=0.0
      GO TO 11
```

```
5       DO 7 ITER=1,100
        C(I,ICOUNT)=(XL*FXR-XR*FXL)/(FXR-FXL)
        CALL EQTNS(TEMP)
        FX2=CPO(I)-C(I,ICOUNT)+RATE(I)*TIME(2)

        IF(ABS(FX2).LE.1.0E-4) GO TO 11

        IF(FX2*FXL.LT.0.) GO TO 6
        XL=X2
        FXL=FX2

        GO TO 7
6       XR=X2
        FXR=FX2
7       CONTINUE
        WRITE(6,202)

11      CONTINUE

201     FORMAT(5X,'NO POSSIBLE ROOT ON STARTING INTERVAL')
202     FORMAT(5X,'NO CONVERGENCE OBTAINED IN CSTR')

        RETURN
        END

        SUBROUTINE DTABLE (X,Y,TABLE,N,M,TRUBL,K)
        IMPLICIT REAL*8(A-H,O-Z)
        DIMENSION X(N),Y(N),TABLE(K,K)
        IF(M.LT.N) GO TO 2
        TRUBL=1.0
        RETURN

2       NM1=N-1
        DO 3 I=1,NM1
3       TABLE(I,1)=(Y(I+1)-Y(I))/(X(I+1)-X(I))
        IF(M.LE.1) GO TO 6

        DO 5 J=2,M
        DO 5 I=J,NM1
        ISUB=I+1-J
5       TABLE(I,J)=(TABLE(I,J-1)-TABLE(I-1,J-1))/(X(I+1)-X(ISUB))

6       TRUBL=0.0
        RETURN
        END

        FUNCTION FNEWT(X,Y,TABLE,N,M,IDEG,XARG,TRUBL,K)
```

```
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(N),Y(N),TABLE(K,K)
      IF(IDEG.LE.M) GO TO 2
      TRUBL=1.0
      FNEWT=0.0
      RETURN

2     DO 4 I=1,N
      IF(I.EQ.N.OR.XARG.LE.X(I)) GO TO 5
4     CONTINUE
5     MAX=I+IDEG/2

      IF(MAX.LE.IDEG) MAX=IDEG+1
      IF(MAX.GT.N) MAX=N

      YEST=TABLE(MAX-1,IDEG)
      IF(IDEG.LE.1) GO TO 13
      IDEGM1=IDEG-1
      DO 12 I=1,IDEGM1
      ISUB1=MAX-I
      ISUB2=IDEG-I
12    YEST=YEST*(XARG-X(ISUB1))+TABLE(ISUB1-1,ISUB2)
13    ISUB1=MAX-IDEG
      TRUBL=0.0
      FNEWT=YEST*(XARG-X(ISUB1))+Y(ISUB1)

      RETURN
      END
```

PROGRAM TO INPUT DATA TO
SIMULATOR (SIMDAT.FOR)

```
C     ------------------------------------------------------------------
C           PROGRAM FOR CREATING INPUT DATA FILE FOR SIMULATOR
C     ------------------------------------------------------------------
C       This is an interactive program by which the user can supply
C     the necessary data to run the simulator.
C     ------------------------------------------------------------------

      IMPLICIT REAL*8(A-H,O-Z)
      DOUBLE PRECISION L,KBARP,KBARR
      DIMENSION L(5),D(5),CPI(10),KBARP(20,20),KBARR(20,20)
      DIMENSION ECP(20,20),ECR(20,20)
      DIMENSION II(10,2),JJ(10,2)
      CHARACTER*20 NAME(10,2)


      OPEN (UNIT=11,FILE='SIM2.DAT')

      WRITE(6,899)
  899  FORMAT(5X,'IF THE USER WISHES TO TEST DIFFERENT REACTION ',/,
     * 5X,'NETWORKS FOR THE SAME SET OF INPUT CONDITIONS ,THEN',/,
     * 5X,'AFTER "SIM1.DAT" HAS BEEN CREATED , THE USER SHOULD',/,
     * 5X,'SPECIFY "IDUM = 1" IN SUBSEQUENT RUNS TO SKIP READING',/,
     * 5X,'THE SAME INPUT CONDITIONS. OTHERWISE SET IDUM = 2',/,
     * 5X,'WHAT IS THE VALUE OF IDUM ? '$)
      READ(6,*) IDUM

      IF(IDUM.EQ.1) GO TO 33

      OPEN (UNIT=1,FILE='SIM1.DAT')
      WRITE(06,100)
  100  FORMAT(////20X,'SIMULATION OF DIRECT COAL LIQUEFACTION',///,
     * 2X,'IF ONE REACTOR : ENTER 1',/,
     * 2X,'IF TWO REACTORS IN SERIES : ENTER 2, AND SO ON'.
     * '(MAXIMUM OF 4) '$)
      READ(5,101) NN
  101  FORMAT(I)
      N=NN+1
      WRITE(01,*) N

      WRITE(06,200)
  200  FORMAT(//20X,'ENTER INLET CONDITIONS:',/,
```

```
     * 2X,'Flowrate of gas to preheater (cm**3/sec)',/,
     * 2X,'Flowrate of slurry to preheater (cm**3/sec)',/,
     * 2X,'Length of preheater coil (cm)',/,
     * 2X,'Length of reactor/reactors  (cm)',/,
     * 2X,'Diameter of preheater coil (cm)',/,
     * 2X,'Diameter of reactor/reactors  (cm)',/,
     * 2X,'Inlet temperature of the slurry to preheater (degrees C)',/,
     * 2X,'Operating pressure (atm)',/,
     * 2X,'Mole fraction of hydrogen in inlet gas to preheater',/,
     * 2X,'Temperature of furnace (degrees C)',/,
     * 2X,'ALL ENTRIES TO BE MADE IN FREE FORMAT',//)
       READ(06,*) VOLG,VOLL,(L(I),I=1,N),(D(I),I=1,N),TIL,P,FRAC,TH
       WRITE(01,*) VOLG,VOLL,(L(I),I=1,N),(D(I),I=1,N),TIL,P,FRAC,TH

       WRITE(06,201)
 201   FORMAT(//20X,'ENTER VALUES OF VARIABLES LISTED BELOW:',/,
     * 2X,'Heat capacity of gas (cal/gm C)',/,
     * 2X,'Heat capacity of slurry (cal/gm C)',/,
     * 2X,'Arrhenius frequency factor for hydrogen consumption ',
     * 'kinetics (1/sec)',/,
     * 2X,'Activation energy for hydrogen consumption kinetics ',
     * '(cal/gmole)',/,
     * 2X,'Heat of reaction (cal/gm)',/,
     * 2X,'Heat of dissolution (cal/gm)',/,
     * 2X,'Solubility constant (<gm/cm**3>)liq/<gm/cm**3>gas)',/,
     * 2X,'Diffusivity of the gas (cm**2/sec)',/,
     * 2X,'Surface tension of the slurry (dynes/cm)',/,
     * 2X,'Density of the slurry (gm/cm**3)',/,
     * 2X,'ALL ENTRIES TO BE MADE IN FREE FORMAT',//)
       READ(06,*) CPG,CPBAR,ZKBAR,E,DHR,DHS,HBAR,DI,SIG,RHOA
       WRITE(01,*) CPG,CPBAR,ZKBAR,E,DHR,DHS,HBAR,DI,SIG,RHOA

       CLOSE(UNIT=1)

  33   WRITE(6,990)
 990   FORMAT(////5X,'Enter total number of species present in',
     * ' preheater '$)
       READ(6,*) NSP
       WRITE(11,*) NSP
       DO 2 I=1,NSP
       WRITE(6,991) I
 991   FORMAT(/,5X,'Enter Name of Species Number',I2,3X,$)
       READ(6,102) NAME(I,1)
       WRITE(11,102) NAME(I,1)
 102   FORMAT(A20)
   2   CONTINUE
       WRITE(6,203)
 203   FORMAT(/,5X,'The Species have been Coded as:',/)
```

```
      DO 3 I=1,NSP
      WRITE(6,103) I,NAME(I,1)
103    FORMAT(10X,I2,2X,'=',2X,A20)
3     CONTINUE
      WRITE(6,105)
105    FORMAT(///,5X,'Enter Total Number of Non-Zero Rate Constants',/,
     * 5X,' for reaction network in preheater  '$)
      READ(6,*) NKP
      WRITE(11,*) NKP
      WRITE(6,106)
106    FORMAT(5X,//,5X,'Enter the Following Data for Each Rate Constant')
      DO 4 I=1,NKP
      WRITE(6,107) I
107    FORMAT(///,5X,'For Rate Constant Number',I2,' Enter')
      WRITE(6,108)
108    FORMAT(5X,'Code of Reacting Species  '$)
      READ(6,*) II(I,1)
      WRITE(11,*) II(I,1)
      WRITE(6,109)
109    FORMAT(5X,'Code of Product Species  '$)
      READ(6,*) JJ(I,1)
      WRITE(11,*) JJ(I,1)
      WRITE(6,110)
110    FORMAT(5X,'Arrhenius Factor for Rate Constant (1/sec)  '$)
      READ(6,*) KBARP(II(I,1),JJ(I,1))
      WRITE(11,*) KBARP(II(I,1),JJ(I,1))
      WRITE(6,111)
111    FORMAT(5X,'Activation Energy (cal/gmol)  '$)
      READ(6,*) ECP(II(I,1),JJ(I,1))
      WRITE(11,*) ECP(II(I,1),JJ(I,1))
4     CONTINUE

      WRITE(6,400)
400    FORMAT(///5X,'Enter total number of species present in',
     * ' reactor '$)
      READ(6,*) NSPR
      WRITE(11,*) NSPR
      DO 6 I=1,NSPR
      WRITE(6,991) I
      READ(6,102) NAME(I,2)
      WRITE(11,102) NAME(I,2)
6     CONTINUE
      WRITE(6,203)
      DO 7 I=1,NSPR
      WRITE(6,103) I,NAME(I,2)
7     CONTINUE
      WRITE(6,115)
115    FORMAT(///5X,'Enter total number of non-zero rate constants',/,
```

```
     *  5X,' for reaction network in reactor '$)
        READ(6,*) NKR
        WRITE(11,*) NKR
        WRITE(6,106)
        DO 8 I=1,NKR
        WRITE(6,107) I
        WRITE(6,108)
        READ(6,*) II(I,2)
        WRITE(11,*) II(I,2)
        WRITE(6,109)
        READ(6,*) JJ(I,2)
        WRITE(11,*) JJ(I,2)
        WRITE(6,110)
        READ(6,*) KBARR(II(I,2),JJ(I,2))
        WRITE(11,*) KBARR(II(I,2),JJ(I,2))
        WRITE(6,111)
        READ(6,*) ECR(II(I,2),JJ(I,2))
        WRITE(11,*) ECR(II(I,2),JJ(I,2))
    8   CONTINUE

        WRITE(6,112)
  112    FORMAT(///5X,'Enter Inital Concentration of Each Species',
     *  '(gm/gm)')
        DO 5 I=1,NSPR
        WRITE(6,113) I
  113    FORMAT(5X,'Species Number ',I2,2X,$)
        READ(6,*) CPI(I)
        WRITE(11,*) CPI(I)
    5   CONTINUE

        OPEN(UNIT=2,NAME='INPUT.DAT')

        IF(IDUM.EQ.1) GO TO 34
        WRITE(2,222) VOLG,VOLL,L(1),(L(I),I=2,N),D(1),(D(I),I=2,N),TIL,
     *                  P,FRAC,TH
  222    FORMAT(/////20X,'LIST OF INPUT DATA :',/////////,
     *  2X,'Flowrate of gas to preheater (cm**3/sec)',T70,E12.4,//,
     *  2X,'Flowrate of slurry to preheater (cm**3/sec)',T70,E12.4,//,
     *  2X,'Length of preheater coil (cm)',T70,E12.4,//,
     *  2X,'Length of reactor/reactors  (cm)',T70,E12.4,//,
     *  2X,'Diameter of preheater coil (cm)',T70,E12.4,//,
     *  2X,'Diameter of reactor/reactors  (cm)',T70,E12.4,//,
     *  2X,'Inlet temperature of slurry to preheater (deg. C)',T70,E12.4
     *  ,//,2X,'Operating pressure (atm)',T70,E12.4,//,
     *  2X,'Mole fraction of hydrogen in inlet gas to preheater',T70,
     *  E12.4,//,2X,'Temperature of furnace (deg. C)',T70,E12.4,/)

        WRITE(2,223) CPG,CPBAR,ZKBAR,E,DHR,DHS,HBAR,DI,SIG,RHOA
```

```
223   FORMAT(2X,'Heat capacity of gas (cal/gm C)',T70,E12.4,//,
     * 2X,'Heat capacity of slurry (cal/gm C)',T70,E12.4,//,
     * 2X,'Arrhenius frequency factor for hydrogen consumption ',
     * 'kinetics (1/sec)',T70,E12.4,//,
     * 2X,'Activation energy for hydrogen consumption kinetics ',
     * '(cal/gmole)',T70,E12.4,//,
     * 2X,'Heat of reaction (cal/gm)',T70,E12.4,//,
     * 2X,'Heat of dissolution (cal/gm)',T70,E12.4,//,
     * 2X,'Solubility constant (<gm/cm**3>)liq/<gm/cm**3>gas)',T70,
     * E12.4,//,2X,'Diffusivity of the gas (cm**2/sec)',T70,E12.4,//,
     * 2X,'Surface tension of the slurry (dynes/cm)',T70,E12.4,//,
     * 2X,'Density of the slurry (gm/cm**3)',T70,E12.4,/)


34    WRITE(2,224) NSP
224   FORMAT('1',5X,'Total number of species present in',
     * 'preheater =',I3,/)
      WRITE(2,225) NKP
225   FORMAT(5X,'Total Number of Non-Zero Rate Constants',
     * ' for reaction network in preheater =',I3,/)

      WRITE(2,230)
230   FORMAT(5X,'The Reaction Pathways for preheater are: '//)

      DO 10 I=1,NKP
      WRITE(2,231) II(I,1),JJ(I,1),KBARP(II(I,1),JJ(I,1)),
     *             ECP(II(I,1),JJ(I,1))
231   FORMAT(5X,'For the reaction path ',I3,' to ',I3,/,
     * 5X, 'k0 = 'E12.4,2X,' E =',E12.4,//)
10    CONTINUE


      WRITE(2,226) NSPR
226   FORMAT('1',5X,'Total number of species present in reactor =',I3,/)
      WRITE(2,227) NKR
227   FORMAT(5X,'Total number of non-zero rate constants for reaction',
     *  network in reactor =',I3,/)

      WRITE(2,232)
232   FORMAT(5X,'The Reaction pathways for reactor are: ',//)

      DO 11 I=1,NKR
      WRITE(2,231) II(I,2),JJ(I,2),KBARR(II(I,2),JJ(I,2)),
     *             ECR(II(I,2),JJ(I,2))
11    CONTINUE
      WRITE(2,228)
228   FORMAT('1',5X,'Initial Concentration of Each Species (gm/gm)',///)
      DO 47 I=1,NSPR
```

```
      WRITE(2,229) I,CPI(I)
229    FORMAT(5X,'Concentration of species # ',I2,' = ',E12.4,//)
47     CONTINUE
      CLOSE (UNIT=2)
      WRITE(6,300)
300    FORMAT(/////70('*'),/,10X,'DATA FILES HAVE BEEN CREATED',/,
     * 10X,'THE INPUT DATA ARE LISTED IN THE FILE "INPUT.DAT"',/,
     * 5X,'THE SIMULATOR CAN NOW BE RUN IN BATCH OR TIME SHARING MODE',
     * /,70('*'))
      STOP
      END
```

PROGRAM TO INPUT DATA TO
FIT.FOR (FITDAT.FOR)

```
C     ------------------------------------------------------------
C        PROGRAM FOR CREATING INPUT DATA FOR PARAMETER ESTIMATION
C     ------------------------------------------------------------
C           This is an interactive program by which the user can
C     input data required for parameter estimation.

      IMPLICIT REAL*8(A-H,O-Z)
      DOUBLE PRECISION L
      DOUBLE PRECISION K,KBAR1,KBAR2
      DIMENSION K(20,20),KBAR1(20,20),EC1(20,20),COEFF(20,20),CPI(10)
      DIMENSION KBAR2(20,20),EC2(20,20)
      DIMENSION ECSTAR(20,20)
      DIMENSION L(2),D(2),X(20),T(20)
      CHARACTER*20 NAME(10)
      OPEN (UNIT=1,FILE='FIT01.DAT')
      OPEN (UNIT=2,FILE='FIT02.DAT')

      WRITE(6,99)
 99    FORMAT(///,5X,'IF ISOTHERMAL REACTOR SPECIFY IND=1',/,
     * 5X,'FOR NON-ISOTHERMAL REACTOR SPECIFY IND=2',/,
     * 5X,'WHAT IS THE VALUE OF IND ? ',$)
      READ(6,*) IND
      WRITE(01,*) IND
      WRITE(06,200)
200    FORMAT(//20X,'ENTER EQUIPMENT SIZES',/,
     * 2X,'Length of preheater (cm)',/,
     * 2X,'Length of reactor (cm)',/,
     * 2X,'Diameter of preheater (cm)',/,
     * 2X,'Diameter of reactor (cm)',/,
     * 2X,'ALL ENTRIES TO BE MADE IN FREE FORMAT',//)
      READ(06,*) (L(I),I=1,2),(D(I),I=1,2)
      WRITE(01,*) (L(I),I=1,2),(D(I),I=1,2)
      WRITE(06,201)
201    FORMAT(//20X,'ENTER PHYSICO-CHEMICAL PARAMETERS',/,
     * 2X,'Heat capacity of gas (cal/gm C)',/,
     * 2X,'Heat capacity of slurry (cal/gm C)',/,
     * 2X,'Arrhenius frequency factor for hydrogen consumption '
     * 'kinetics (1/sec)',/,
     * 2X,'Activation energy for hydrogen consumption kinetics ',
     * '(cal/gmole)',/,
     * 2X,'Heat of reaction (cal/gm)',/,
     * 2X,'Heat of dissolution (cal/gm)',/,
     * 2X,'Solubility constant (<gm/cm**3>)liq/<gm/cm**3>gas)',/,
```

```
     *  2X,'Diffusivity of the gas (cm**2/sec)',/,
     *  2X,'Surface tension of the slurry (dynes/cm)',/,
     *  2X,'Density of the slurry (gm/cm**3)',/,
     *  2X,'ALL ENTRIES TO BE MADE IN FREE FORMAT',//)


        READ(06,*) CPG,CPBAR,ZKBAR,E,DHR,DHS,HBAR,DI,SIG,RHOA
        WRITE(01,*) CPG,CPBAR,ZKBAR,E,DHR,DHS,HBAR,DI,SIG,RHOA
        WRITE(6,100)
100      FORMAT(//,5X,'Enter Total Number of Species' $)
        READ(6,*) NSP
        WRITE(1,*) NSP
        DO 2 I=1,NSP
        WRITE(6,101) I
101      FORMAT(/,5X,'Enter Name of Species Number',I2, $)
        READ(6,102) NAME(I)
        WRITE(1,102) NAME(I)
102      FORMAT(A20)
2       CONTINUE
        DO 3 I=1,NSP
        WRITE(6,103) I,NAME(I)
103      FORMAT(10X,I2,2X,'=',2X,A20)
3       CONTINUE
        WRITE(6,105)
105      FORMAT(/,5X,'Enter Total Number of Non-Zero Rate Constants' $)
        READ(6,*) NK
        WRITE(1,*) NK
        WRITE(6,106)
106      FORMAT(/,5X,'Enter the Following Data for Each Rate Constant' $)
        DO 4 I=1,NK
        WRITE(6,107) I
107      FORMAT(/,5X,'For Rate Constant Number',I2,' Enter')
        WRITE(6,108)
108      FORMAT(5X,'Code of Reacting Species' $)
        READ(6,*) II
        WRITE(1,*) II
        WRITE(6,109)
109      FORMAT(5X,'Code of Product Species' $)
        READ(6,*) JJ
        WRITE(1,*) JJ
        WRITE(6,110)
110      FORMAT(5X,'Lower',
     *  ' & Upper bounds on Arrhenius Factor for Rate Constant (1/sec)')
        READ(6,*) KBAR1(II,JJ),KBAR2(II,JJ)
        WRITE(1,*) KBAR1(II,JJ),KBAR2(II,JJ)
        WRITE(6,111)
111      FORMAT(5x,'Lower',
     *  ' & Upper Bounds on Activation Energy (cal/gmol)')
```

```
      READ(6,*) EC1(II,JJ),EC2(II,JJ)
      WRITE(1,*) EC1(II,JJ),EC2(II,JJ)
 4    CONTINUE
      WRITE(6,500)
500    FORMAT(//,5X,'Enter no. of data points' $)
      READ(6,*)NDATA
      WRITE(1,*)NDATA
      CLOSE(UNIT=1)
      WRITE(6,502)
502    FORMAT(/,5X,'  FOR EACH DATA POINT, ENTER THE FOLLOWING')
      DO 501 IQ=1,NDATA
      WRITE(6,503)IQ
503    FORMAT('        FOR DATA PT NO', I3)
      WRITE(6,504)
504    FORMAT('          Gas flowrate (cm**3/sec)',/,
     * 2x,'Slurry flowrate (cm**3/sec)',/,
     * 2x,'Inlet temp of slurry to preheater (deg C)',/,
     * 2x,'Operating pressure (atm)',/,
     * 2x,'Mole frac of hydrogen in inlet to preheater',/,
     * 2X,'Temp of heating medium',/,
     * 2x,'ENTER IN FREE FORMAT',//)
      READ(6,*)VOLG,VOLL,TIL,P,FRAC,TH
      WRITE(2,*)VOLG,VOLL,TIL,P,FRAC,TH
      WRITE(6,507)
507    FORMAT(//,5X,'  ENTER INLET CONC OF EACH SPECIES, GM/GM')
      DO 508 IIQ=1,NSP
      WRITE(6,113)IIQ
      READ(6,*)CPI(IIQ)
      WRITE(2,*)CPI(IIQ)
508    CONTINUE
      WRITE(6,505)
505    FORMAT(//,5X,'  ENTER OUTLET CONC OF EACH SPECIES, GM/GM')
      DO 506 IIQ=1,NSP
      WRITE(6,113)IIQ
113    FORMAT(5X,'Species Number ',I2, $)
      READ(6,*) CPI(IIQ)
      WRITE(2,*) CPI(IIQ)
506    CONTINUE
501    CONTINUE


      IF(IND.EQ.1) GO TO 31
      WRITE(6,204)
204    FORMAT(5X,'ENTER DATA FOR TEMPERATURE PROFILE IN REACTOR',/)
      WRITE(6,202)
202    FORMAT(5X,'ENTER NUMBER OF POSITIONS AT WHICH TEMPERATURE WAS '
     * 'MEASURED ',$)
      READ(6,*) NDT
```

```
      WRITE(2,*) NDT
      WRITE(6,203)
203    FORMAT(5X,'ENTER PAIRS OF POSITION-TEMPERATURE DATA',/,
     * 5X,'POSITION IN CM, TEMPERATURE IN DEG. C',/)
      DO 11 I=1,NDT
      READ(6,*) X(I),T(I)
      WRITE(2,*) X(I),T(I)
11     CONTINUE
31     CLOSE(UNIT=2)
      WRITE(6,300)
300    FORMAT(16X,'DATA FILES "FIT01.DAT" & "FIT02.DAT" HAVE' ,/,
     * 'BEEN CREATED. THE SIMULATOR MAY NOW BE RUN IN BATCH MODE',
     * /,70('*'))
      STOP
      END
```