Anexo 1. Métodos de estimación de las propiedades.

A. 1. 1. Cálculo de la densidad.

A. 1. 2. Cálculo de difusividad molecular.

A. 1. 3. Cálculo de la viscosidad.

A. 1. 1. Cálculo de la densidad.

El cálculo de la densidad se ha hecho a partir de la resolución de la ecuación de estado de Peng-Robinson en su forma polinomial en z (factor de compresibilidad), la cual es:

$$z^3 - (1-B)z^2 + \left(A - 3B^2 - 2B\right) z - \left(AB - B^2 - B^3\right) = 0$$

Donde:

$$A = \frac{a\,P}{R^2T^2},\ B = \frac{b\,P}{RT}$$

$$a = \sum_{i=1}\sum_{j=1} y_i y_j a_{ij},\ b = \sum_{i=1}^{3}\sum_{j=1}^{3} y_i y_j b_{ij}$$

$$a_{ij} = (1-k_{ij})\sqrt{a_i a_j},\ b_{ij} = \frac{(b_i + b_j)}{2}(1-\eta_{ij})$$

La resolución se ha hecho gracias al método de Newton, una vez tenemos z, la densidaddel fluido se ha calculado a partir de la ecuación:

$$\rho = \frac{P\ .PM}{zRT}$$

PM es la masa molecular.

En el anexo (A. 2. 5) se presenta el programa de cálculo utilizado para calcular la densidad de la mezcla CO_2 – tolueno.

A. 1. 2. Cálculo de difusividad molecular.

El cálculo de la difusividad molecular se ha hecho gracias a la correlación de Catchpole – King (1994):

$$D_{12} = 5.152 D_c T_r \left(\rho^{-2/3} - 0.4510\right) R/X$$

Con la condición de que la densidad reducida debe estar entre 1 y 2.5

$$X = \frac{\left(1+\left(V_{c2}/V_{c1}\right)^{1/3}\right)^2}{\left(1+PM_1/PM_2\right)^{0.5}}$$

Si $X < 2$, $R = 1 \pm 0.1$

Si $2 < X < 10$, $R = 0.664X^{0.17} \pm 0.1$

1. 1. 3. Cálculo de la viscosidad.

El cálculo de la viscosidad se ha hecho gracias a la correlación de Reichenberg, (1975):

$$\eta = \frac{aT}{\left[1+0.36(1+\frac{4}{T_c})T_r(T_r-1)\right]^{1/6}}$$

$$a = \frac{\eta_c}{T_c}$$

En el anexo (A. 2. 6) se presenta el programa del cálculo utilizado para calcular la viscosidad del CO_2 a diferentes condiciones de presión y temperatura.

Anexo 2. Programas de Fortran.

A. 2. 1. Programa para calcular el equilibrio sólido – vapor utilizando la teoría de la esfera rígida de van der Waals.

A. 2. 2. Programa para calcular el equilibrio sólido – vapor utilizando la ecuación de estado de Peng-Robenson

A. 2. 3. Programa para calcular el equilibrio líquido – vapor utilizando la ecuación de estado de Peng Robinson.

A. 2. 4. Programa para resolver el problema de extracción de β-naftol en un lecho de partículas cilíndricas, modelo del núcleo sin reaccionar junto con el método de las líneas.

A. 2. 5. Programa para calcular la densidad de una mezcla CO_2/tolueno a diferentes presiones y temperaturas.

A. 2. 6. Programa para calcular la viscosidad de CO_2 a diferentes presiones y temperaturas.

A. 2. 1. Programa para calcular el equilibrio sólido – vapor utilizando la teoría de la esfera rígida de van der Waals

```
      PARAMETER(N=2)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION Y(N),SIGMA(3)
      OPEN(UNIT=19,FILE='CAR.DAT',STATUS='OLD')
      OPEN(UNIT=18,FILE='CAR.OUT',STATUS='UNKNOWN')

C     ************************************************
C     ************************************************
      DO 10 I=1,N
   10 READ (19,*) SIGMA(I)
      READ(19,*) T
      RE.....AD(19,*) PLOW
      READ(19,*)PHIGH
      READ(19,*)PINC
      T=T+273.0
      READ(19,*)VM

      READ(19,*)a12
      WRITE(6,*)'a12',a12
      READ(19,*)a22
      WRITE(6,*)'a22',a22

      WRITE(6,*)'PLOW',PLOW, 'PHIGH',PHIGH
C     ************************************************
      WRITE(18,*)'T(C)=',T
      WRITE(6,11)
      WRITE(18,11)
   11 FORMAT(//,3X,'P(bar)',5X,'Y(2)',12X,'FP',6X,'Z',4X,'BM'8X,'-E-'
     +,6X,'HELM FR EN',3X,'DENS CO2',3X,'DENS RED')

C     ************************************************
```

```
C     SUPER CRITICAL FLUID CALCULATION
      GASR=83.14
      PHIGH=PHIGH
      PINC=PINC
      P=PLOW
      DP=PINC
      DO 5 IJ=1,2000

C     ************************************************
C     SUBLIMATION PRESSURE FOR THE SOLID
      WRITE(6,*)'T',T
      WRITE(6,*)'P=',P
C     ********************************************************************
C     ********************** NAPHTALENE*****************
      PS=10.**(13.575-(3729.3/(T)))/100000.
C     ********************** BENZOIC 55 C*****************
c     PS=2.16E-5
C     **********************PHENANTHRENE 55 °C****************
C     PS=0.423E-5
C     ********************1-NAPHTOL******39-50 C************
C     PS=10.**(11.526-(4389./T))/750.
C     ********************2-NAPHTOL******25-39 C************
C     PS=10.**(13.356-(5109./T))/750.
C     ********************2-NAPHTOL******39-58 C************
C     PS=10.**(11.660-(4579/T))/750.
C     ********************o-HBA******55 °C************
C     PS=6.65E-06
C     ********************p-HBA******55 °C************
C     PS=3.37E-08
c     ***************2,3-DIMETHYLNAPHTHALENE   55 °C***************
C     PS=9.01E-5
c     ***************2,6-DIMETHYLNAPHTHALENE   55 °C***************
C     PS=9.13E-5
C     ********************************************************************
```

```
      Y(2)=PS
      WRITE(6,*)'PS',PS
      WRITE(6,*)'Y(2)',Y(2)
      Y(1)=1.-Y(2)
       WRITE(6,*)VM
      ALPHA=(PS/P)*DEXP(VM*(P-PS)/83.14/T)
      WRITE(6,*)'ALPHA',ALPHA
C     ***************************************************
C     CALCUL DE LA FRACTION MOLAIRE DU COMPOSANT LOURD DANS
C     LE FLUIDE SUPERCRITIQUE
      DO 30 J=1,500
      CALL QBM(SIGMA,Y,OZETA,T,RO,P,VM,a12,a22,FP,Z,CO,ALPHA,HK
     +,HC,AN2PR,BM)
      IF(J.EQ.1.)THEN
      HC=2*CO*a12/83.14/T*FP+ALPHA/(Y(2)**2)

      ENDIF
      ALPHA=(PS/P)*DEXP(VM*(P-PS)/83.14/T)

      WRITE(6,*)'FP',FP

      YC=ALPHA/FP
      E=YC*PS/P
      DELY=(ABS(YC-Y(2)))/Y(2)

      Y(2)=(PS*HK-YC*HC)/(HK-HC)

      Y(1)=1.-Y(2)
      IF(DELY.LE.0.0005)GOTO 35
      IF(J.GE.500)GOTO 150

   30 CONTINUE

   35 T=T-273.0
```

```
      GOTO 25

C     ****************************************************
C     IMPRESSION DES DONNEES T et Y
 25   WRITE(18,118)P,YC,FP,Z,BM,E,AN2PR,CO,OZETA
      WRITE(6,118)P,Y(2)
      write(6,*)'j',j

C     LE CALCUL EST TERMINE

      IF(P.GE.PHIGH) GOTO 90
      P=P+DP
      T=T+273.0
 5    CONTINUE

C     ****************************************************
 150  WRITE(6,*)'CALCUL DE SOLUBILITE DIVERGE'
      GOTO 90
 100  WRITE(6,110)
 90   WRITE(18,*)'PRESSION DE SUBLIMATION'
      WRITE(18,117)PS
 117  FORMAT(E10.4)
      STOP
 110  FORMAT(5X,'NEAR MIXTURE CRITICAL POINT')
 118  FORMAT(2X,F5.1,3X,E10.5,3X,E10.4,3X,F5.2,3X,F5.2,3X,E10.4,3X,E10.3
     +,3X,E10.3,3X,E10.3)
      close(19)
      close(18)

      END
C     ************************************************************
C     SUBROUTINE DE CALCUL DU COEFFICIENT DE FUGACITE
C     CALCUL DU FACTEUR DE COMPRESSIBLOTE ZBM
```

```
      SUBROUTINE
QBM(SIGMA,Y,OZETA,T,RO,P,VM,a12,a22,FP,Z,CO,ALPHA,HK
     +,HC,AN2PR,BM)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION Y(2),SIGMA(3), CRO(20)
      DIMENSION COZETA(2),ZETA(20),ZBM(20)
      GASR=83.14
C     ESTIMATION DE RO KZETA ET ZBM

      IF(P.EQ.80.) THEN
      DEN=4.7736E-3
      ELSE
      IF(P.EQ.90.) THEN
      DEN=5.91E-3
      ELSE
      IF(P.EQ.100.) THEN
      DEN=7.2727E-3
      ELSE
      IF(P.EQ.110.) THEN
      DEN=9.091E-3

      ENDIF
      ENDIF
      ENDIF
      ENDIF
      IF(P.EQ.120) THEN
      den=1.136E-2
      ELSE
      IF(P.EQ.130) THEN
      den=1.295E-2
      ELSE
      IF(P.EQ.140.) THEN
      den=1.3977E-2
      ELSE
```

```
IF(P.EQ.150) THEN
den=1.466E-2
ELSE
IF(P.EQ.160.) THEN
den=1.523e-2
ELSE
IF(P.EQ.170) THEN
den=1.591E-2
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
IF(P.EQ.180.) THEN
den=1.636e-2
ELSE
IF(P.EQ.190.) THEN
den=1.66e-2
ELSE
IF(P.EQ.200.) THEN
den=1.70e-2
ELSE
IF(P.EQ.210.) THEN
den=1.7386e-2
ELSE
IF(P.EQ.220) THEN
den=1.773E-2
ELSE
IF(P.EQ.230.) THEN
den=1.795e-2
ELSE
IF(P.EQ.240.) THEN
den=1.8182E-2
```

```
ELSE
IF(P.EQ.250.) THEN
den=1.841e-2
ELSE
IF(P.EQ.260.) THEN
den=1.864E-2
ELSE
IF(P.EQ.270.) THEN
den=1.8864e-2
ELSE
IF(P.EQ.280.) THEN
den=1.89772E-2
ENDIF
ENDIF
ENDIF
endif
endif
endif
ENDIF
ENDIF
ENDIF
ENDIF
endif
endif

Q=1./VM

COZETA(1)=Y(1)*3.14*den*(6.022*10.**23)*(SIGMA(1)**3)/6.

COZETA(2)=Y(2)*3.14*Q*(6.022*10.**23)*(SIGMA(2)**3)/6.

SIGMA(3)=(SIGMA(1)+SIGMA(2))/2.
ZETA(0)=22/42.*6.022*10.**23*SIGMA(3)**3*(DEN*Y(1)+1./VM*Y(2))

D12=(((COZETA(1)*COZETA(2))**(1/2))/ZETA(0))*((SIGMA(1)-SIGMA(2)
```

```
     +)**2/(SIGMA(1)*SIGMA(2)))*(Y(1)*Y(2))**(1./2.)

      X1=D12*(SIGMA(1)+SIGMA(2))*(SIGMA(1)*SIGMA(2))**(-1./2.)

      X2=D12*((COZETA(1)/ZETA(0))*((SIGMA(1)*SIGMA(2))**(1./2.))/SIGMA(1
     +)+(COZETA(2)/ZETA(0))*((SIGMA(1)*SIGMA(2))**(1./2.))/SIGMA(2))

      X3=(((COZETA(1)/ZETA(0))**(2./3.))*Y(1)**(1./3.)+((COZETA(2)/ZETA(
     +0))**(2./3.))*Y(2)**(1./3.))**3.

C     CÁLCULO DE Z
      ZBM(0)=(1.+(1.-3.*X1)*ZETA(0)+(1.-3*X2)*ZETA(0)**2-X3*ZETA(0)**3)/
     +(1.-ZETA(0))**3.

      BM=ZBM(0)
      CO=den

      CRO(0)=DEN
C     ***************************************************************
        DO 1000 K=0,19
      GASR=83.14

      CRO(K+1)=P/ZBM(K)/83.14/T

      ZETA(K+1)=Y(1)*22./7*CRO(K+1)*(6.022*10.**23)*(SIGMA(3)**3)/6.
     ++Y(2)*(22./7)*Q*(6.022*10.**23)*(SIGMA(3)**3)/6.

      D12=(((COZETA(1)*COZETA(2))**(1./2.))/ZETA(K+1))*((SIGMA(1)-
     +SIGMA(2))**2/(SIGMA(1)*SIGMA(2)))*(Y(1)*Y(2))**(1./2.)

      X1=D12*(SIGMA(1)+SIGMA(2))*(SIGMA(1)*SIGMA(2))**(-1./2.)
```

```
      X2=D12*((COZETA(1)/ZETA(K+1))*((SIGMA(1)*SIGMA(2))**(1./2.))/SIGMA
     +(1)+(COZETA(2)/ZETA(K+1))*((SIGMA(1)*SIGMA(2))**(1./2.))/SIGMA(2))

      X3=(((COZETA(1)/ZETA(K+1))**(2./3.))*Y(1)**(1./3.)+((COZETA(2)/
     +ZETA(K+1))**(2./3.))*Y(2)**(1./3.))**3.

C     CALCULO DE Z
      ZBM(K+1)=(1.+(1.-3.*X1)*ZETA(K+1)+(1.-3*X2)*ZETA(K+1)**2-X3*
     +ZETA(K+1)**3)/((1.-ZETA(K+1))**3.)

      TEST=COZETA(K+1)-COZETA(K)
      IF(ABS(TEST).LE.0.0001) THEN
      Z=ZBM(K+1)
C     OZETA=ZETA(K+1)
      RO=CRO(K+1)
      GOTO 1500
      ENDIF
 1000 CONTINUE
 1500 CONTINUE
c     ********************************************************************
      COZETA(1)=Y(1)*3.14*den*(6.022*10.**23)*(SIGMA(1)**3)/6.

      COZETA(2)=Y(2)*3.14*Q*(6.022*10.**23)*(SIGMA(2)**3)/6.

      OZETA=22/42.*6.022*10.**23*SIGMA(3)**3*(DEN*Y(1)+1./VM*Y(2))

      S=P/Z/83.14/T
      RO=S
C     CALCULO DE DU COEFFICIENT DE LA FUGACITE
      GASR=83.14
      A=6.022*10.**23
      D12PR=(CO*Q)**(1./2.)*CO*SIGMA(3)**3.*Y(1)**2.*(SIGMA(1)-
     +SIGMA(2))**2.*SIGMA(1)**(1./2.)*SIGMA(2)**(1./2.)/(CO*SIGMA(3)
     +**3.*Y(1)+Q*SIGMA(3)**3*Y(2))**2
```

```
 Y1PR=((SIGMA(1)+SIGMA(2))/(SIGMA(1)*SIGMA(2))**(1./2.))*D12PR

  V=CO*SIGMA(1)**2.*(SIGMA(1)*SIGMA(2))**(1./2.)*Y(1)/(CO*SIGMA(3)
 +**3*Y(1)+Q*SIGMA(3)**3*Y(2))+Q*SIGMA(2)**2.*(SIGMA(1)*
 +SIGMA(2))**(1./2.)*Y(2)/(CO*SIGMA(3)**3*Y(1)+Q*SIGMA(3)**3*
 +Y(2))

  VPR=-CO*Q*SIGMA(1)**2.*(SIGMA(1)*SIGMA(2))**(1./2.)*Y(1)*
 +SIGMA(3)**3./((CO*SIGMA(3)**3*Y(1)+Q*SIGMA(3)**3*Y(2))**2.)+
 +Q*(SIGMA(2)**2.*(SIGMA(1)*SIGMA(2))**(1./2.)*(CO*SIGMA(3)**3*
 +Y(1)+Q*SIGMA(3)**3*Y(2))-SIGMA(2)**2.*(SIGMA(1)*SIGMA(2))**
 +(1./2.)*Y(2)*Q*SIGMA(3)**3.)/(CO*SIGMA(3)**3*Y(1)+Q*
 +SIGMA(3)**3*Y(2))**2

 Y2PR=D12*VPR+V*D12PR

   WPR=-(SIGMA(1)**2.*CO**(2./3)*Y(1)*(2./3.)*Q*SIGMA(3)**3.*
 +(CO*SIGMA(3)**3*Y(1)+Q*SIGMA(3)**3*Y(2))**(-1./3.)/(CO*
 +SIGMA(3)**3*Y(1)+Q*SIGMA(3)**3*Y(2))**(4./3.))+(CO**(2./3.)*
 +SIGMA(2)**2.*(CO*SIGMA(3)**3*Y(1)+Q*SIGMA(3)**3*Y(2))**(2./3.)-
 +(1./VM)**(2./3.)*SIGMA(2)**2.*Y(2)*(2./3.)*SIGMA(3)**3.*(CO*SIGMA
 +(3)**3*Y(1)+Q*SIGMA(3)**3*Y(2))**(-1./3.))/(CO*SIGMA(3)**3*
 +Y(1)+Q*SIGMA(3)**3*Y(2))**(4./3.)

   W=((CO*SIGMA(1)**3.*Y(1)/(CO*SIGMA(3)**3*Y(1)+Q*SIGMA(3)**
 +3*Y(2)))**(2./3.)*Y(1)**(1./3.)+(SIGMA(2)**3.*Y(2)*Q/(CO*
 +SIGMA(3)**3*Y(1)+Q*SIGMA(3)**3*Y(2)))**(2./3.)*Y(2)**(1./3.)
 +)**3.
 Y3PR=3.*WPR*W**2.

 AN2PR= ((-3./2.)*(-Y1PR+Y2PR+Y3PR)+((3.*Y2PR+2.*Y3PR)*(1.-ozeta)+
 +(3.*x2+2.*x3)*22./42.*A*RO*SIGMA(3)**3.)/(1.-OZETA)**2.
 ++(3./2.)*((-Y1PR-Y2PR-(1./3.)*Y3PR)*((1.-OZETA)**2.)-(1.-X1-X2-
 +(1./3.)*X3)*2.*(-22./42.*A*RO)*SIGMA(3)**3.*(1.-OZETA))/
```

```
+(1.-OZETA)**4.+Y3PR*LOG(1-OZETA)-(22./42.*RO*A*SIGMA(3)**3.)/
+(1.-OZETA)*(X3-1))

FP=EXP(AN2PR-2*CO*(Y(1)*a12+Y(2)*a22)/83.14/T-LOG(BM))

HK=FP-ALPHA/Y(2)

RETURN
END
```

A. 2. 2. Programa para calcular el equilibrio sólido – vapor utilizando la ecuación de estado de Peng-Robenson

```
      PARAMETER(N=2)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(N),Y(N),FP(N)
      DIMENSION AMW(N),W(N),TC(N),PC(N)
      COMMON/PENG/AMW,W,TC,PC,PR,ETA
      REAL*8 LASDEV
      OPEN(UNIT=15,FILE='SOLID.DAT',STATUS='OLD')
      OPEN(UNIT=16,FILE='SOLID.OUT',STATUS='UNKNOWN')
C     ************************************************
C     ************************************************
      DO 10 I=1,N
      READ (15,*) AMW(I)
   10 READ (15,*) TC(I),PC(I),W(I)
      READ (15,*) PR,ETA
      READ(15,*) T
      WRITE(6,*) T
      T = T + 273.15
      WRITE(*,*) 'T=', T-273.5
      READ(15,*) PLOW,PHIGH,PINC
C     ************************************************
      READ(15,*)VS
      WRITE(6,*)VS
      X(1)=0.005
      X(2)=1.-X(1)
C     ************************************************
      IFIRST=1
      ITYPE = 1
      WRITE(16,11)
      WRITE(6,11)
   11 FORMAT(//,3X,'P(bar)',5X,'Y(2)',6X,'LIQUID(g/cc)')
C     ************************************************
```

```
C      SUPER CRITICAL FLUID CALCULATION
       GASR=83.14
       DT=1.5
       P=PLOW
       DP=PINC
       DO 5 IJ=1,2000
       P=P+DP
       LASDEV=0.0
C      ************************************************
C        SUBLIMATION PRESSURE FOR THE SOLID
C      NAPHTALENE
c      PS=10**(10.0896-2926.61/((T-273.15)+237.32))/760

C      ******************1-NAPHTOL******39-50 C************
c      PS=10.**(11.526-(4389./T))/750.
C      ******************2-NAPHTOL******39-58 C************
       PS(2)=10.**(11.660-(4579/T))/750.
C      ********************* BENZOIC 45 C*****************
c      PS(3)=0.78E-5
C      **********************PHENANTHRENE 45 °C***************
c      PS(4)=0.155E-5
C      ******************o-HBA******45 °C************
c      PS(5)=2.25E-06
C      ******************p-HBA******45 °C************
c      PS(6)=1.37E-08
C    ***************2,3-DIMETHYLNAPHTHALENE   45 °C***************
c    PS(7)=3.49E-5
C      ***************2,6-DIMETHYLNAPHTHALENE   45 °C*****************
c      PS(8)=3.45E-5
C      ***************************ANTHRACENE*************************
c      PS(9)=10.**(12.147-(4397.6/T))/100000.
C      ***************************PYRENE
c      PS(10)=10.**(11.27-(4904/T))/750.
C      ***************************HEXAMETHYLBENZENE
```

```
c     PS(11)=10.**(13.134-(3855/T))/100000.
C     ********************************FLUORENE
c     PS(12)=10.**(14.205-(4561.8/T))/100000.
C     ****************************************************************
      IF(IFIRST.EQ.1)THEN
      Y(2)=PS/P
      Y(1)=1.-Y(2)
      ENDIF
      ALPHA=(PS/P)*DEXP(VS*(P-PS)/(GASR*T))
      WRITE(6,*)'PS',PS
C     ************************************************
C     CALCUL DE LA FRACTION MOLAIRE DU COMPOSANT LOURD DANS
C     LE FLUIDE SUPERCRITIQUE
      Do 30 J=1,500
      CALL PHIPR(T,P,Y,FP,DEN,0,N)
      WRITE(6,*)'FP(2)',FP(2)
      YC=ALPHA/FP(2)
      DELY=(ABS(YC-Y(2)))/Y(2)
      Y(2)=YC
      Y(1)=1.-Y(2)
      IF(DELY.LE.0.001)GOTO 35
      IF(J.GE.500)GOTO 150
   30 CONTINUE
   35 IF(ITYPE.EQ.1)GOTO 25
C     ******************************************************
C     IMPRESSION DES DONNEES T et P
   25 WRITE(16,118)P,Y(2),DEN
      WRITE(6,118)P,Y(2),DEN
c
C     LE CALCUL EST TERMINE POUR FSC
     IF((ITYPE.EQ.1).AND.(P.GE.PHIGH)) GOTO 90
C     TESTES POUR FINIR LES CALCULS PRES DU POINT CRITIQUE DU
MELANGE. LA
C     PRESSION EST ARBITRAIREMENT PRISE 50 bar
```

```
c
      IF(P.GE.50) THEN
      TEST=ABS(X(2)-Y(2))
      IF(TEST.LT.1.0D-3) GOTO 100
      ENDIF
C     ENDIF
      IFIRST=0
 5    CONTINUE
C     ************************************************
 150  WRITE(6,*)'WARNING:ITERATIONS EXCEED (SVE PART)'
      WRITE(6,*)P,Y(1),YC
      GOTO 90
 100  WRITE(6,110)
 90   STOP
 110  FORMAT(5X,'NEAR MIXTURE CRITICAL POINT')
 118  FORMAT(3X,F10.3,3X,F7.5,3X,F5.3)
      close(15)
      close(16)
      END
c
C
**********************************************************************
C       Subroutine PHIPR calculates the fugacity coefficient, and the phase
C       density from the PREOS and the mole fractions.
C       NNN=0 for the liquid phase
C       NNN=1 for the gas phase
C       See also McHugh-Krukonis, 2nd Ed. (1994)
C       ************************************************************
C
        SUBROUTINE PHIPR(T,P,Y,FP,DEN,NNN,N)
        IMPLICIT REAL*8(A-H,O-Z)
        DIMENSION Y(2),FP(2),A(2,2),B(2,2),AMW(2),W(2),TC(2),PC(2),
     +TR(2),TERM1(2),TERM2(2),TERM3(2),ROOT(3),G(2)
        COMMON/PENG/AMW,W,TC,PC,PR,ETA
```

```
      GASR=83.14
      BM=0.0
      AM=0.0
      SQ2=2.0**0.5
      Q1=1.0+SQ2
      Q2=SQ2-1.0
      Q3=2.0*SQ2
C
C     ****************************************************************
C     These are the mixture terms A and B for the PREOS. The SUM term
C     will be used later to calculate the mixture density.
C     ****************************************************************
C
      SUM=0.0
      DO 10 I=1,N
            SUM=SUM+Y(I)*AMW(I)
            B(I,I)=0.0778*GASR*TC(I)/PC(I)
            TERM3(I)=0.0
            TR(I)=T/TC(I)
            G(I)=(1.0+(0.37464+1.54226*W(I)-0.26992*W(I)**2)*(1.0-TR(I)
     +           **0.5))**2
 10   A(I,I)=0.45724*(GASR**2)*(TC(I)**2)/PC(I)*G(I)
      J=N-1
      DO 20 I=1,J
            DO 21 K=I,J
                  L=K+1
                  A(I,L)=(A(I,I)*A(L,L))**0.5*(1.0-PR)
                  A(L,I)=A(I,L)
                  B(I,L)=(B(I,I)+B(L,L))/2.0*(1.0-ETA)
                  B(L,I)=B(I,L)
 21         CONTINUE
 20   CONTINUE
      DO 30 I=1,N
        DO 31 J=1,N
```

```
            BM=BM+Y(I)*Y(J)*B(I,J)
            AM=AM+Y(I)*Y(J)*A(I,J)
 31       CONTINUE
 30    CONTINUE
       AA=AM*P/GASR**2/T**2
       BB=BM*P/GASR/T
C
C     ****************************************************************
C     The PREOS is written as a cubic equation in Z=PV/RT. The equation
C     becomes: RA1*Z3 + RA2*Z2 + RA3*Z1 + RA4 = 0
C     Only one of the 3 roots is valid. The following routine calculates
C     the valid root, using the trigonometric solution of the cubic equation
C     ****************************************************************
C
       RA1=1.0
       RA2=BB-1.0
       RA3=AA-2.0*BB-3.0*BB**2
       RA4=BB**3+BB**2-AA*BB
       A1=(3.0*RA3-RA2**2)/3.0
       B1=(2.0*RA2**3-9.0*RA2*RA3+27.0*RA4)/27
       TEST1=DABS((A1**3)/27.0)
       TEST2=(B1**2)/4.0
       PIE=3.141559265
       IF(A1.LT.0.0.AND.TEST1.GT.TEST2)THEN
            CO=2.0*(((-A1)/3)**0.5)
            THETA=(DACOS((3.0*B1)/(A1*CO)))/3.0
            ROOT(1)=CO*DCOS(THETA)-(RA2/3.0)
            ROOT(2)=CO*DCOS(THETA+(2.0*PIE)/3.0)-(RA2/3.0)
            ROOT(3)=CO*DCOS(THETA+(4.0*PIE)/3.0)-(RA2/3.0)
       DO 300 J=1,3
            IF(ROOT(J).LT.0.0.AND.NNN.EQ.1) THEN
                 ROOT(J)=1.E+10
            ENDIF
300    CONTINUE
```

```
C
C     ************************************************************
C     The largest root is the vapor's z, the smallest is the liquid's.
C     Depending upon the value of NNN (NNN=0 for the vapor, NNN=1 for the
C     liquid) either the largest or the smallest root is used to determine
C     the fugacity coefficients of the 2 components in a particular phase
C     ************************************************************
C
      IF(NNN.EQ.0) ZM=DMAX1(ROOT(1),ROOT(2),ROOT(3))
      IF(NNN.EQ.1) ZM=DMIN1(ROOT(1),ROOT(2),ROOT(3))
C
C     ************************************************************
C     But A1 may not be greater than zero, or TEST1 may not be greater
C     than TEST2:
C     ************************************************************
C
      ELSE
            DD=DSQRT(TEST2+(A1**3)/27.0)
            AL=1.0
            ALL=1.0
            TEST3=(-B1)/2.0+DD
                  IF(TEST3.LT.0.0) THEN
                  AL=-1.0
                  ENDIF
            TEST3=DABS(TEST3)
            A2=AL*((TEST3)**0.3333334)
            TEST4=(-B1)/2.-DD
                  IF(TEST4.LT.0.0) THEN
                  ALL=-1.0
                  ENDIF
            TEST4=DABS((-B1)/2.-DD)
            B2=ALL*((TEST4)**0.3333334)
            ZM=A2+B2-(RA2/3.0)
                  IF(TEST4.LT.1.E-04) GOTO 40
```

```
            TEST5=DABS(1.0-DABS(A2/B2))
      IF(TEST5.LT.5.E-04) THEN
      ZM=-1.0*((A2+B2)/2.0)-(RA2/3.)
                  ENDIF
 40   ENDIF
      VM=ZM*GASR*T/P
      DEN=(1/VM)*SUM
C
C     ****************************************************************
C     The fugacity coefficients for each component are now calculated
C     ****************************************************************
C
      BRPRIME=0.0
      DO 60 I=1,N
      IF(I.EQ.1)BJJ=B(2,2)
      IF(I.EQ.2)BJJ=B(1,1)
      BPRIME=Y(I)*B(I,I)*(2.-Y(I))+((1.-Y(I))**2)*(2*B(1,2)-BJJ)
            TERM1(I)=BPRIME*(ZM-1.0)/BM-DLOG(ZM-BB)
            TERM2(I)=BPRIME*AA*DLOG((ZM+Q1*BB)/(ZM-Q2*BB))
     +              /(BM*BB*Q3)
            DO 62 J=1,N
 62               TERM3(I)=TERM3(I)+2.0*Y(J)*A(J,I)
                  TERM3(I)=AA*DLOG((ZM+Q1*BB)/(ZM-Q2*BB))*
     +                  TERM3(I)/(BB*AM*Q3)
            FP(I)=DEXP(TERM1(I)+TERM2(I)-TERM3(I))
 60   CONTINUE
      RETURN
      END
```

A. 2. 3. Programa para calcular el equilibrio líquido – vapor utilizando la ecuación de estado de Peng Robinson

```
      PROGRAM PR2
      PARAMETER (N=2)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION H(N,1000),Q(N,1000),FPV(N),FPL(N),TEST(N),PR(N+1,N+1),
     +  ETA(N+1,N+1)
      DIMENSION X(N),Y(N),AMW(N),W(N),TC(N),PC(N),U(N),V(N)
      COMMON/PENG/AMW,W,TC,PC,PR,ETA
      OPEN (Unit=15,File='PR2.DAT',Status='OLD')
     OPEN (Unit=16, File='PR2.OUT', Status='UNKNOWN')
     CLOSE (16 ,Status='DELETE')
     OPEN (Unit=16, File='PR2.OUT', Status='NEW')
C     ****************************************************************
C     Subscripts:
C     Subscript 1 = the volatile component, light (CO2)
C     Subscript 2 = the non-volatile component, heavy (solute)
C     The second subscript in double-subscript variables is the iteration
C     number: for example, H(2,1) is the distribution K = y/x for component
C     2 during the first iteration. The first part of the program is set up
C     to read molecular weights, Tc, Pc and acentric factor (w) from an
C     external data file, PR2.DAT.
C     Variables which correspond to the mixture parameters:
C     kij = PR
C     nij = ETA
C     And to the pressure range:
C     PHIGH =high pressure limit
C     PLOW =low pressure limit
C     PINC= pressure increments
C     ****************************************************************
      DO 10 I=1,N
            READ (15,*) AMW(I)
10          READ (15,*) TC(I),PC(I),W(I)
```

```
                    DO 50 I=1,N-1
                        DO 60,J=1,N-1
                        READ (15,*) PR(I,J+1),ETA(I,j+1)
                        PR(J+1,I)=PR(I,J+1)
                        ETA(J+1,I)=ETA(I,J+1)
 60                     CONTINUE
 50                 CONTINUE
      WRITE(6,*)'kij=',PR(1,2),'nij=',ETA(1,2)
      READ(15,*)T,PLOW,PHIGH,PINC
      WRITE(6,*)T
      T=T+273.15
c   WRITE(*,*)'T=',T-273.15
      WRITE(*,13)
c   WRITE(6,*)"T=",T-273.15
      WRITE(16,13)
 13   FORMAT (24X,'(bar)',8X,'(g/cm3)',4X,'(g/cm3)')
      WRITE(*,12)
 12   FORMAT(3X,'Xheavy',2X,'Y(gas)heavy',3X,'Press',8X,'DENL',4X,'DENV',
     +     5X,'it')
C     ***************************************************************
C     The pressure is initialized for the first iteration and initial
C     estimates of the K = y/x are given.
C     ***************************************************************
      H(1,1)=20.0
      H(2,1)=0.05
      P=PLOW
      DP=PINC
C     ***************************************************************
C     Variables used in the calculation of the P-x-y isotherm:
C     ICOUNT= to flag the method and update the Ks, H(i,j)
C     JFLAG= number of iterations per tie line
C     Loops:
C
C     The Do 20 LOOP is used to calculate the entire P-x isotherm.
```

```
C     A very high number of iterations is set, so that the program is
C     expected to reach the mixture's critical pressure (PHIGH) before
C     reaching the maximum number of iterations in this loop.
C
C     The DO 30 LOOP is used to calculate a single tie line. Again the
C     number of iterations is set high so that the tie line is expected to
C     converge or a warning statement will be printed to the screen before
C     250 iterations can be performed.
C     ****************************************************************
      DO 20 ICOUNT=1,15000
            DO 30 JFLAG=1,5000
                  X(2)=(1.-H(1,JFLAG))/(H(2,JFLAG)-H(1,JFLAG))
                  X(1)= 1.-X(2)
                  IF((X(1).LT.0.0).OR.(X(1).GT.1.0)) GOTO 115
                  Y(1)=H(1,JFLAG)*X(1)
                  Y(2)=1.-Y(1)
C     ****************************************************************
C     The vapor-phase fugacity coefficient is calculated first, followed
C     by the liquid-phase fugacity coefficient:
C     ****************************************************************
            CALL PHIPR(T,P,Y,FPV,DENV,0,N)
            CALL PHIPR(T,P,X,FPL,DENL,1,N)
C     ****************************************************************
C     The following lines test the equilibrium, i.e., if the fugacity
C     of each component is equal in each of the phases.
C     ****************************************************************
            Q(1,JFLAG)=(X(1)*FPL(1))/(Y(1)*FPV(1))
            Q(2,JFLAG)=(X(2)*FPL(2))/(Y(2)*FPV(2))
            TEST(1)=ABS(Q(1,JFLAG)-1.)
            TEST(2)=ABS(Q(2,JFLAG)-1.)
C     ****************************************************************
C     If TEST(1) or TEST(2) are not within the adjustable tolerance,
C     the program will do another iteration at this temperature and
C     pressure with a      new estimate of the distribution coefficients:
```

```
C     ***************************************************************

            IF(TEST(1).GE.0.001.OR.TEST(2).GE.0.001) THEN
                  H(1,JFLAG+1)=H(1,JFLAG)*Q(1,JFLAG)
                  H(2,JFLAG+1)=H(2,JFLAG)*Q(2,JFLAG)
            ELSE
                  GOTO 35
            ENDIF
 30   CONTINUE
      WRITE (*,*) "THE TIE LINE DID NOT CONVERGE"
      GOTO 115
C     ***************************************************************
C     Equlibrium has been reached if TEST1 and TEST2 are within the
C     adjustable tolerance. Now the program guesses new xs and ys for
C     each component at the next pressure, P+DP. The pressure increment
C     is decreased as the mixture's critical point is approached, which
C     is seen by determining how close H(2,JFLAG) is to unity.
C     The method for calculating the Hs at P+DP can cause the program to
C     crash if the pressure increment is adjusted at low pressures, far
C     from the maximum pressure of the P-x loop. Here we use an arbitrary
C     value of 50 bar to avoid this problem.
C     ***************************************************************
 35   IF(P.LE.50.0) THEN
         DP=2.
      ELSE
     IF (H(2,JFLAG).GT.0.40) DP=1.0
        IF (H(2,JFLAG).GT.0.60) DP=1.0
        IF (H(2,JFLAG).GT.0.70) DP=0.5
        IF (H(2,JFLAG).GT.0.80) DP=0.5
        IF (H(2,JFLAG).GT.0.95) GOTO 75
C     ***************************************************************
C     This last statement stops the program very near the mixture
C     critical point.  Otherwise the program continues until PHIGH is
C     reached.
```

```
C     ****************************************************************

      ENDIF
C     ****************************************************************
C     Now a first guess for the values of x and y at this new pressure
C     is attained by fitting the straight K-line through the calculated
C     data and extrapolating it to the new pressure:
C     ****************************************************************
      POLD=P-DP
      PNEW=P+DP
      SS2=X(2)+(PNEW-P)*((X(2)-U(2))/(P-POLD))
            IF(ICOUNT.EQ.1) SS2=X(2)
      SS1=1.-SS2
      TT2=Y(2)+(PNEW-P)*((Y(2)-V(2))/(P-POLD))
            IF(ICOUNT.EQ.1) TT2=Y(2)
      TT1=1.0-TT2
      H(1,1)=TT1/SS1
      H(2,1)=TT2/SS2
      U(1)=X(1)
      U(2)=X(2)
      V(1)=Y(1)
      V(2)=Y(2)
C     ****************************************************************
C     The program prints the results on the screen and to a datafile
C     named PR2.OUT. See OPEN(Unit=16...) at the beginning.
C     ****************************************************************
            WRITE(*,85) X(2),Y(2),P,DENL,DENV,JFLAG
            WRITE(16,85) X(2),Y(2),P,DENL,DENV,JFLAG
 85         FORMAT (1X,F8.6,2X,F8.6,4X,F9.3,2X,F9.3,1X,F7.4,2X,I4)
C     ****************************************************************
C     The program will continue calculating a P-x isotherm as long as
C     the upper limit has not been reached and H(2,JFLAG) is less than
C     0.95
C     ****************************************************************
```

```
            IF(P.GE.PHIGH)THEN
                WRITE(*,*)'UPPER PRESSURE BOUND REACHED'
                GOTO 115
            ENDIF
            P=P+DP
 20     CONTINUE
 75     WRITE(*,*)'NEAR THE MIXTURE CRITICAL POINT'
115     WRITE(16,116)
        WRITE(*,116)
116     FORMAT(////,8X,'THE PROGRAM IS COMPLETED')
        CLOSE(16)
        END
C       ****************************************************************
C       Subroutine PHIPR calculates the fugacity coefficient, and the phase
C       density from the PREOS and the mole fractions.
C       NNN=0 for the liquid phase
C       NNN=1 for the gas phase
C       See also McHugh-Krukonis, 2nd Ed. (1994)
C       ****************************************************************
        SUBROUTINE PHIPR(T,P,Y,FP,DEN,NNN,N)
        IMPLICIT REAL*8(A-H,O-Z)
        DIMENSION Y(2),FP(2),A(2,2),B(2,2),AMW(2),W(2),TC(2),PC(2),
     +  PR(3,3),ETA(3,3),TR(2),TERM1(2),TERM2(2),TERM3(2),ROOT(3),G(2)
        COMMON/PENG/AMW,W,TC,PC,PR,ETA
        GASR=83.14
        BM=0.0
        AM=0.0
        SQ2=2.0**0.5
        Q1=1.0+SQ2
        Q2=SQ2-1.0
        Q3=2.0*SQ2
C       ****************************************************************
C       These are the mixture terms A and B for the PREOS. The SUM term
C       will be used later to calculate the mixture density.
```

```
C     ****************************************************************
      SUM=0.0
      DO 10 I=1,N
            SUM=SUM+Y(I)*AMW(I)
            B(I,I)=0.0778*GASR*TC(I)/PC(I)
            TERM3(I)=0.0
            TR(I)=T/TC(I)
            G(I)=(1.0+(0.37464+1.54226*W(I)-0.26992*W(I)**2)*(1.0-TR(I)
     +           **0.5))**2
 10   A(I,I)=0.45724*(GASR**2)*(TC(I)**2)/PC(I)*G(I)
      J=N-1
      DO 20 I=1,J
            DO 21 K=I,J
                  L=K+1
                  A(I,L)=(A(I,I)*A(L,L))**0.5*(1.0-PR(I,L))
                  A(L,I)=A(I,L)
                  B(I,L)=(B(I,I)+B(L,L))/2.0*(1.0-ETA(I,l))
                  B(L,I)=B(I,L)
 21         CONTINUE
 20   CONTINUE
      DO 30 I=1,N
         DO 31 J=1,N
             BM=BM+Y(I)*Y(J)*B(I,J)
             AM=AM+Y(I)*Y(J)*A(I,J)
 31      CONTINUE
 30   CONTINUE
      AA=AM*P/GASR**2/T**2
      BB=BM*P/GASR/T
C     ****************************************************************
C     The PREOS is written as a cubic equation in Z=PV/RT. The equation
C     becomes: RA1*Z3 + RA2*Z2 + RA3*Z1 + RA4 = 0
C     Only one of the 3 roots is valid. The following routine calculates
C     the valid root, using the trigonometric solution of the cubic equation
C     ****************************************************************
```

```
      RA1=1.0
      RA2=BB-1.0
      RA3=AA-2.0*BB-3.0*BB**2
      RA4=BB**3+BB**2-AA*BB
      A1=(3.0*RA3-RA2**2)/3.0
      B1=(2.0*RA2**3-9.0*RA2*RA3+27.0*RA4)/27
      TEST1=DABS((A1**3)/27.0)
      TEST2=(B1**2)/4.0
      PIE=3.141559265
      IF(A1.LT.0.0.AND.TEST1.GT.TEST2)THEN
            CO=2.0*(((-A1)/3)**0.5)
            THETA=(DACOS((3.0*B1)/(A1*CO)))/3.0
            ROOT(1)=CO*DCOS(THETA)-(RA2/3.0)
            ROOT(2)=CO*DCOS(THETA+(2.0*PIE)/3.0)-(RA2/3.0)
            ROOT(3)=CO*DCOS(THETA+(4.0*PIE)/3.0)-(RA2/3.0)
      DO 300 J=1,3
            IF(ROOT(J).LT.0.0.AND.NNN.EQ.1) THEN
                  ROOT(J)=1.E+10
            ENDIF
300   CONTINUE
C     ***************************************************************
C     The largest root is the vapor's z, the smallest is the liquid's.
C     Depending upon the value of NNN (NNN=0 for the vapor, NNN=1 for the
C     liquid) either the largest or the smallest root is used to determine
C     the fugacity coefficients of the 2 components in a particular phase
C     ***************************************************************
      IF(NNN.EQ.0) ZM=DMAX1(ROOT(1),ROOT(2),ROOT(3))
      IF(NNN.EQ.1) ZM=DMIN1(ROOT(1),ROOT(2),ROOT(3))
C     ***************************************************************
C     But A1 may not be greater than zero, or TEST1 may not be greater
C     than TEST2:
C     ***************************************************************
      ELSE
            DD=DSQRT(TEST2+(A1**3)/27.0)
```

```
      AL=1.0
      ALL=1.0
      TEST3=(-B1)/2.0+DD
            IF(TEST3.LT.0.0) THEN
            AL=-1.0
            ENDIF
      TEST3=DABS(TEST3)
      A2=AL*((TEST3)**0.3333334)
      TEST4=(-B1)/2.-DD
            IF(TEST4.LT.0.0) THEN
            ALL=-1.0
            ENDIF
      TEST4=DABS((-B1)/2.-DD)
      B2=ALL*((TEST4)**0.3333334)
      ZM=A2+B2-(RA2/3.0)
            IF(TEST4.LT.1.E-04) GOTO 40
      TEST5=DABS(1.0-DABS(A2/B2))
            IF(TEST5.LT.5.E-04) THEN
                  ZM=-1.0*((A2+B2)/2.0)-(RA2/3.)
            ENDIF
40    ENDIF
      VM=ZM*GASR*T/P
      DEN=(1/VM)*SUM
C     ****************************************************************
C     The fugacity coefficients for each component are now calculated
C     ****************************************************************
      BRPRIME=0.0
      DO 60 I=1,N
            BP2=0.0
            DO 61 INN=1,N
61                BP2=2.0*Y(INN)*B(I,INN)+BP2
            BPRIME=BP2-BM
            TERM1(I)=BPRIME*(ZM-1.0)/BM-DLOG(ZM-BB)
            TERM2(I)=BPRIME*AA*DLOG((ZM+Q1*BB)/(ZM-Q2*BB))
```

```
     +                /(BM*BB*Q3)
              DO 62 J=1,N
62                TERM3(I)=TERM3(I)+2.0*Y(J)*A(J,I)
                  TERM3(I)=AA*DLOG((ZM+Q1*BB)/(ZM-Q2*BB))*
     +                TERM3(I)/(BB*AM*Q3)
              FP(I)=DEXP(TERM1(I)+TERM2(I)-TERM3(I))
60    CONTINUE
      RETURN

      END
```

B. 2. 4. Programa para resolver el problema de extracción de β-naftol en un lecho de partículas cilíndricas, modelo del núcleo sin reaccionar junto con el método de las líneas.

```
C     ************************PROGRAMME PRINCIPAL*******************
C     ******************************************************************
C     ***********************Specification des variables locales***************
C     ******************************************************************
      INTEGER LDY, NPDES, NX, NEX
      PARAMETER (NPDES=3, NX=3, LDY=NPDES,NEX=9)
C     ***********NEX est le nombre de point experimentaux dont on tient*********
C     ********************compte pour calculer AARD**********************
      INTEGER I,IDO,IEND,K,INK
      REAL FCNBC,FCNUT,HINIT,T,TEND,TOL,XBREAK(NX),
     +Y(LDY,NX)
      Dimension zc(10),ze(10)
C
      EXTERNAL  FCNUT,FCNBC, MOLCH

      OPEN(UNIT=15,FILE='OUTPUT.OUT',STATUS='UNKNOWN')
      OPEN(UNIT=16,FILE='DON.DAT',STATUS='OLD')
      OPEN(UNIT=30,FILE='fr.OUT',STATUS='UNKNOWN')
      OPEN(UNIT=31,FILE='exp.DAT',STATUS='OLD')
      OPEN(UNIT=32,FILE='zcal.OUT',STATUS='UNKNOWN')
C     ************** definir les 'breakpoints' et les conditions initiales*************
c
      DO 10 I=1,NX
      XBREAK(I)=1./(NX-1)*(I-1)
   10 CONTINUE
C     ***********************definir les parametres de MOLCH***************
      TOL=1.E-7
      HINIT=0.01
      T=0.000000
      IDO=1
```

```
      DO 2 K=1,NX
      Y(1,K)=0.0000000
      Y(2,K)=0.0000000
      Y(3,K)=0.0000000
    2 CONTINUE
C     ************************Parametres de l'extraction*********************
C     *******************************Lecture*******************************
      READ(16,*)DM,R,OL,K,INK,EP,Bi,a
      READ(16,*)q,Csat,ROF,VIS,EB
      READ(16,*)V,HAUTL,ROC,VISC
      Tau=EP*DM*a*HAUTL/V/(R**2)
      DEFF=DM*EP/TAU

      CLOSE(16)
      DO 3 J=1,NX
      Y(1,J)=0.000000000000
      Y(2,J)=1.000000000000
      Y(3,J)=1.000000000000
    3 CONTINUE

      nn=0
      DO 22 J=0,nex
   22 Read(31,*)ze(j)

C     ***************************Calcul*********************************
      DO 20 IEND=1,K+1,INK
      M=IEND
      DEFF=DM*EP/TAU
C     DEFF=DM*EP**2
      TEND=M*DEFF/(R**2)
C                       resolution du probleme
C     ******************************************************************
```

```
      CALL
MOLCH(IDO,FCNUT,FCNBC,NPDES,T,TEND,NX,XBREAK,TOL,HINIT,Y,
     +LDY)
C     ****************************************************************
C     impression des resultats
      WRITE(15,100) T,M
      sum=0
      frac=0
      DO 25 L=1,NX
      WRITE(15,200)XBREAK(L),Y(1,L),Y(2,L),Y(3,L)
      sum=sum+1./3*((y(2,L))**2)*y(3,L)
   25 CONTINUE
c     **********************Calcul de la fraction massique extraite*************
c     write(6,*)'sum=',sum
c     Frac2=1-((y(2,1)+y(2,2)+y(3,3))/3)**2*(y(3,1)+y(3,2)+y(3,3))/3

      frac=1.-sum
      exp=ze(nn)
      write(32,*)frac
      Write(30,1500)M,frac,exp
c      Write(30,1500)M,frac
      nn=nn+1

   20 CONTINUE

      IDO=3
      CALL
MOLCH(IDO,FCNUT,FCNBC,NPDES,T,TEND,NX,XBREAK,TOL,HINIT,Y,
     +LDY)
C     ************************Calcul de AARD%************************
      close(32)
      OPEN(UNIT=32,FILE='ZCAL.OUT',STATUS='UNKNOWN')
      zc(0)=0.000000
      DO 500 K=0,nex
```

```
 500  READ(32,*)zc(K)
      asum=0
      dsum=0
      Do 33 i=0,nex
      if(i.eq.0.0)then
      sum=0.0
      goto 33
      endif
      sum=sum+abs(zc(i)-ze(i))/ze(i)
 33   continue

      AARD=sum*100/(nex+1)
      write(30,*)'AARD=',AARD
      write(6,*)'AARD=',AARD
C     ************************************************************************
 100  FORMAT(3X,'SOLUTION A F=',F7.4,//,3X,'temps en secondes=',I7,//,6X
     +,'Z',4X,'Concentration X',5X,'Rayon du noyau Zeta',3X,'hauteur l
     +')
 200  FORMAT(3X,F4.2,7X,F5.3,17X,F5.3,15X,F5.3)

 1500 Format(7x,I7,10x,F9.5,12x,F9.5)
      CLOSE(15)
      close(30)
      close(31)
      close(32)
      END
C     ************************************************************************
C     **********Sous-programme de definition des equations differentielles*********
C     ************************************************************************
C
      SUBROUTINE FCNUT(NPDES,X,T,U,UX,UXX,UT)
      REAL U(2),UX(2),UXX(2),UT(2)
C     ************************************************************************
      OPEN(UNIT=16,FILE='DON.DAT',STATUS='OLD')
```

```
C    **********Lecture des donnees
     READ(16,*)DM,R,OL,K,INK,EP,Bi,a
     READ(16,*)q,Csat,ROF,VIS,EB
     READ(16,*)V,HAUTL,ROC,VISC
C    **********Calcul des constantes et des nombres adimentionnels
     CLOSE(16)
     Tau=EP*DM*a*HAUTL/V/(R**2)
     DEFF=DM*EP/TAU
c    **********Renolds con velocidad superficial**********
     Re1=2*R*V*EB*ROF/VIS
c    **********Renolds con velocidad intersticial**********
     Re2=2*R*V*ROF/VIS
     Sc=VIS/ROF/Dm
c    DAX=(DM/EP)*(20+0.5*RE1*SC)
     Dax8=0.085*V**0.914*(2*R)**0.388*(ROF/ROC)**0.725*
    +(VIS/VISC)**0.676

C              **********kexp*********
     SKexp=V*R*Bi/HAUTL/a
c              **********
c    Sh=0.269*Re1**(0.88)*Sc**(0.3)
c    SKF=Sh*Dm/2/R
c    a=V*R**2/Deff/HAUTL
c    Bi=R*skf/Deff
     if (l.eq.1) then
     WRITE(6,*)'Tau',Tau
     WRITE(6,*)'Re',Re1
     endif
c
c    write(6,*)'Bi=',Bi
c    WRITE(6,*)'K',SKF
     if (l.eq.1) then
     WRITE(6,*)'Kexp',SKexp
     endif
```

```
c   **********Cálculo del Péclet de initialización **********
c   **del subprograma Regula-falsi con la formula propuesta por Tan & Liou**
    Pe0=1.634*(Re2**(0.265))*(Sc**(-0.919))
c   ************************************************************
    Skg=SKexp
    if (l.lt.1) then
    Call mab(Skg,Pep,Pe0)
    Dax20=V*HAUTL/Pep
    Bo=2*R*V/Dax20
    WRITE(6,*) 'Pe',PeP
    WRITE(30,*) 'Pe=',PeP
    WRITE(30,*)'Kexp',SKg
    Write(30,1600)
    endif
c   WRITE(6,*)'Dax20',Dax20
c   WRITE(6,*)'Dax8',Dax8
c   WRITE(6,*)'Re1=',Re1
c   WRITE(6,*)'Pe',PeP
c   WRITE(6,*)'Bo, dp',Bo
c   WRITE(6,*)'a=',a
C   **********Definition de PDE**********
C   **********Cylindre ouvert**********

    UT(1)=a/PeP*UXX(1)-a*UX(1)+2*Bi*EP*(1-EB)/EB*(1-U(1))*(1/(1-EP
   +*Bi*log(U(2)))+R/2/OL*1/(1+Bi*EP/(U(2)**2)*OL/R*(1-U(3))))

    UT(3)=-Bi*Csat/q*R/OL*(1-U(1))/(1+Bi*EP/(U(2)**2)*OL/R*(1-U(3)))
   +*EB/(1-EB)
    UT(2)=-Bi*Csat/q/U(2)*(1-U(1))/(1-Bi*EP*LOG(U(2)))*EB/(1-EB)
    l=l+1
1600 Format(3x,'t d extraxtion',3x,'Frac massique ext cal',5x,
   +'Frac massique ext exp' )
    RETURN
    END
```

```
C  ***********************************************************************
C  ****************Sous-programme de lectures des conditions limites*********
C  ***********************************************************************

   SUBROUTINE FCNBC(NPDES,X,T,ALPHA,BETA,GAMP)
   REAL ALPHA(2),BETA(2),GAMP(2)
C  ***********************************************************************
   OPEN(UNIT=16,FILE='DON.DAT',STATUS='OLD')
C  ********************Lecture des donnees
   READ(16,*)DM,R,OL,K,INK,EP,Bi,a
   READ(16,*)q,Csat,ROF,VIS,EB
   READ(16,*)V,HAUTL,ROC,VISC
   CLOSE(16)
C  ************************Conditions aux limites**********************

   IF(X.LT.0.1) THEN
   ALPHA(1)=1
   BETA(1)=0
   GAMP(1)=0.0
   ALPHA(2)=0.0
   BETA(2)=0.0
   GAMP(2)=0.0
   ALPHA(3)=0.0
   BETA(3)=0.0
   GAMP(3)=0.0
C  ***********************************************************************
   ELSE
   IF(X.GT.0.9) THEN
   ALPHA(1)=0.0
   BETA(1)=1.0
   GAMP(1)=0.0
   ALPHA(2)=0.0
   BETA(2)=0.0
   GAMP(2)=0.0
```

```
      ALPHA(3)=0.0
      BETA(3)=0.0
      GAMP(3)=0.0
      ENDIF
      ENDIF
      RETURN
      END

c     ********************Regula falsi,  f(X)=0*********************
      Subroutine mab(skg,Pep,pe0)

      dimension Pe(200)

      Pec=Pe0+2
      X=Skg
      k=0
      Pe(K)=PeC+PeC/10

      j=0
c     open (unit=25, file='kg.out', status='unknown')
c     write(25,110)
 20   continue
      j=j+1

      if(j.gt.100) then
      write(6,*)'1=200'
      goto 40
      endif

      Pe(k+1)=(PeC*f(x,Pe(k))-Pe(k)*f(x,Pec))/(f(x,Pe(k))-f(x,Pec))

      erreur=100*abs((Pe(k+1)-Pe(k))/Pe(k))
      d= f(x,Pe(k+1))
```

```
      if(erreur.lt.0.01.and.d.lt.1e-5) then
      Pep= Pe(k+1)
      goto 40
      else
      Pe(k)=Pe(k+1)
      k=k+1
      goto 20

      endif

 40  continue
      return
      end

      Function F(X,Pe)
      open (unit=24, file='Pe.dat', status='old')
      Read(24,*)Csat,qsc,sm,t
      F=sm/t-(1+(2*X*Pe*7465*EXP((Pe-SQRT(Pe**2+4*X*Pe*7465
     +))*0.02))/(Pe*SQRT(Pe**2+4*X*Pe*7465)-Pe-4*X*Pe*7465)
     +-((2*X*Pe*7465*EXP(Pe+SQRT(Pe**2+4*X*Pe*7465))*0.02
     +)/(Pe*SQRT(Pe**2+4*X*Pe*7465)+Pe**2+4*X*Pe*7465)))*Csat*qsc
       close(24)
      Return
      End
```

B. 2. 5. Programa para calcular la densidad de una mezcla CO_2/tolueno a diferentes presiones y temperaturas.

```
      PARAMETER (N=2)
      IMPLICIT REAL*8 (A-H,O-Z)

      DIMENSION Y(2),A(2,2),B(2,2),AMW(2),W(2),TC(2),PC(2),
     +  PR(3,3),ETA(3,3),TR(2),ROOT(3),G(2)

      OPEN (Unit=15,File='PR2.DAT',Status='OLD')
     OPEN (Unit=16, File='PR2.OUT', Status='UNKNOWN')

C     ************************************************************
C     Subscripts:
C     Subscript 1 = the volatile component, light (CO2)
C     Subscript 2 = the non-volatile component, heavy (solute)
C     The second subscript in double-subscript variables is the iteration
C     number.
C   The first part of the program is set up
C     to read molecular weights, Tc, Pc and acentric factor (w) from an
C     external data file, PR2.DAT.
C     Variables which correspond to the mixture parameters:
C     kij = PR
C     nij = ETA
C     And to the pressure range:
C     PHIGH =high pressure limit
C     PLOW =low pressure limit
C     PINC= pressure increments
C     ************************************************************
C
     NNN=0.0
      DO 600 I=1,N
            READ (15,*) AMW(I)
  600 READ (15,*) TC(I),PC(I),W(I)
```

```
                DO 50 I=1,N-1
                    DO 60,J=1,N-1
                    READ (15,*) PR(I,J+1),ETA(I,j+1)
                    PR(J+1,I)=PR(I,J+1)
                    ETA(J+1,I)=ETA(I,J+1)
60                  CONTINUE
50              CONTINUE
      WRITE(6,*)'kij=',PR(1,2),'nij=',ETA(1,2)
      READ(15,*)PLOW,PHIGH,PINC
      WRITE(6,*)'Temperatura de extraccion C'
      READ(5,*)T
      WRITE(6,*)'Fraction molar de tolueno Y(2)='
      READ(5,*)Y(2)
      WRITE(6,*)T,PLOW,PHIGH,PINC

      WRITE(16,150)T
      T=T+273.15

      WRITE(*,13)

      WRITE(16,160)Y(2)
      WRITE(16,13)
13  FORMAT (7X,'P(bar)',4X,'Dmezcla(g/cm3)')

150 FORMAT(5X,'T= ',2X,F7.1,1X,'C')
160 format(5X,'Fraction molar de tolueno Y(2)=',2X,E9.2)
   P=PLOW

500  CONTINUE

   Y(1)=1.-Y(2)
      GASR=83.14
      BM=0.0
      AM=0.0
```

```
      SQ2=2.0**0.5
      Q1=1.0+SQ2
      Q2=SQ2-1.0
      Q3=2.0*SQ2
C
C     ****************************************************************
C     These are the mixture terms A and B for the PREOS. The SUM term
C     will be used later to calculate the mixture density.
C     ****************************************************************
C
      SUM=0.0
      DO 10 I=1,N
            SUM=SUM+Y(I)*AMW(I)
            B(I,I)=0.0778*GASR*TC(I)/PC(I)

            TR(I)=T/TC(I)
            G(I)=(1.0+(0.37464+1.54226*W(I)-0.26992*W(I)**2)*(1.0-TR(I)
     +            **0.5))**2
        A(I,I)=0.45724*(GASR**2)*(TC(I)**2)/PC(I)*G(I)
 10   CONTINUE
      J=N-1
      DO 20 I=1,J
            DO 21 K=I,J
                  L=K+1
                  A(I,L)=(A(I,I)*A(L,L))**0.5*(1.0-PR(I,L))
                  A(L,I)=A(I,L)
                  B(I,L)=(B(I,I)+B(L,L))/2.0*(1.0-ETA(I,I))
                  B(L,I)=B(I,L)
 21   CONTINUE
 20   CONTINUE
      DO 30 I=1,N
        DO 31 J=1,N
            BM=BM+Y(I)*Y(J)*B(I,J)
            AM=AM+Y(I)*Y(J)*A(I,J)
```

```
 31   CONTINUE
 30 CONTINUE
      AA=AM*P/GASR**2/T**2
      BB=BM*P/GASR/T
C
C     ************************************************************
C     The PR EOS is written as a cubic equation in Z=PV/RT. The equation
C     becomes: RA1*Z3 + RA2*Z2 + RA3*Z1 + RA4 = 0
C     Only one of the 3 roots is valid. The following routine calculates
C     the valid root, using the trigonometric solution of the cubic equation
C     ************************************************************
C
      RA1=1.0
      RA2=BB-1.0
      RA3=AA-2.0*BB-3.0*BB**2
      RA4=BB**3+BB**2-AA*BB
      A1=(3.0*RA3-RA2**2)/3.0
      B1=(2.0*RA2**3-9.0*RA2*RA3+27.0*RA4)/27
      TEST1=DABS((A1**3)/27.0)
      TEST2=(B1**2)/4.0
      PIE=3.141559265
      IF(A1.LT.0.0.AND.TEST1.GT.TEST2)THEN
            CO=2.0*(((-A1)/3)**0.5)
            THETA=(DACOS((3.0*B1)/(A1*CO)))/3.0
            ROOT(1)=CO*DCOS(THETA)-(RA2/3.0)
            ROOT(2)=CO*DCOS(THETA+(2.0*PIE)/3.0)-(RA2/3.0)
            ROOT(3)=CO*DCOS(THETA+(4.0*PIE)/3.0)-(RA2/3.0)
      DO 300 J=1,3
            IF(ROOT(J).LT.0.0.AND.NNN.EQ.1) THEN
                  ROOT(J)=1.E+10
            ENDIF
 300  CONTINUE
C
C     ************************************************************
```

```
C     The largest root is the vapor's z, the smallest is the liquid's.
C     Depending upon the value of NNN (NNN=0 for the vapor, NNN=1 for the
C     liquid) either the largest or the smallest root is used to determine
C     the fugacity coefficients of the 2 components in a particular phase
C     ****************************************************************
C
      IF(NNN.EQ.0) ZM=DMAX1(ROOT(1),ROOT(2),ROOT(3))
      IF(NNN.EQ.1) ZM=DMIN1(ROOT(1),ROOT(2),ROOT(3))
C
C     ****************************************************************
C     But A1 may not be greater than zero, or TEST1 may not be greater
C     than TEST2:
C     ****************************************************************
C
      ELSE
            DD=DSQRT(TEST2+(A1**3)/27.0)
            AL=1.0
            ALL=1.0
            TEST3=(-B1)/2.0+DD
                  IF(TEST3.LT.0.0) THEN
                  AL=-1.0
                  ENDIF
            TEST3=DABS(TEST3)
            A2=AL*((TEST3)**0.3333334)
            TEST4=(-B1)/2.-DD
                  IF(TEST4.LT.0.0) THEN
                  ALL=-1.0
                  ENDIF
            TEST4=DABS((-B1)/2.-DD)
            B2=ALL*((TEST4)**0.3333334)
            ZM=A2+B2-(RA2/3.0)
                  IF(TEST4.LT.1.E-04) GOTO 40
            TEST5=DABS(1.0-DABS(A2/B2))
                  IF(TEST5.LT.5.E-04) THEN
```

```
                    ZM=-1.0*((A2+B2)/2.0)-(RA2/3.)
                ENDIF
 40     ENDIF
C     Cálculo de la densidad
       VM=ZM*GASR*T/P
       DEN=(1/VM)*SUM
       WRITE(16,100)P,DEN
       WRITE(6,100)P,DEN
       P=P+PINC
       IF(P.LE.PHIGH) GOTO 500
 100  FORMAT(5X,F7.1,5X,F9.6)
    close(15)
    close(16)
       END
```

B. 2. 6. Programa para calcular la viscosidad de CO_2 a diferentes presiones y temperaturas.

```
      OPEN (Unit=21, File='VIS.OUT', Status='UNKNOWN')
      WRITE(6,*)'PRECION DE TRABAJO MIN Y MAX INK'
      READ(5,*) PMIN,PMAX,INK
      WRITE(6,*)'TEMPERATURA DE TRABAJO C'
      READ(5,*) T
      WRITE(21,300)T
      WRITE(21,100)
      P=PMIN
      T=T+273.15
      TC=304.3
      PC=73
      M=44
5     continue
      ALPHA1=1.9824E-3
      ALPHA2=5.2683
      ALPHA3=0.5767
      BETA1=1.6552
      BETA2=1.2760
      GAMA1=0.1319
      GAMA2=3.7035
      GAMA3=79.8678
      DELTA1=2.9496
      DELTA2=2.9190
      DELTA3=16.6169
      TR=T/TC
      PR=P/PC
      A=(ALPHA1/TR)*EXP(ALPHA2*TR-ALPHA3)
      B=A*(BETA1*TR-BETA2)
      C=(GAMA1/TR)*EXP(GAMA2*TR-GAMA3)
      D=(DELTA1/TR)*EXP(DELTA2*TR-DELTA3)
      O=3.941E-10
```

```
      EK=195.2
      TET=T/EK
      OMEGA=1.16145/TET**(0.14874)+0.52487/EXP(0.77320*TET)+2.16178/
     +EXP(2.43787*TET)
      UM0=(26.69E-27*(M*T)**(0.5))/(O**2*OMEGA)
      UM=UM0*(1+(A*PR**(1.5))/(B*PR+(1+C*PR**D)**(-1.)))

      WRITE(21,200)P,UM
      P=P+INK
      IF(P.GT.PMAX) GOTO 10
      GOTO 5
   10 CONTINUE
  100 FORMAT(6X,'PRESION',8X,'VISCOSIDAD')
  200 FORMAT(4X,F9.2,7X,E9.4)
  300 FORMAT(3X,'TEMPERATURA DE TRABAJO' , F9.2,1X,'C')
      close(21)
      END
```