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

C THIS PROGRAM PERFORMS FLASH CALCULATION USING PENG-ROBINSON EOS
C SIX METHODS HAVE BEEN APPLIED.
C
*********************************************************************
***
      IMPLICIT REAL*8(A-H,O-Z), INTEGER (I-N)
      COMMON/CR/TC,VC,PC,W,AMOLW,AM
      COMMON/TAP/T,P,RT,AOA,BOB,T1,P1
      COMMON/CONS/C1,C2,C3,IL,IV,CPS,R,EPS1,EPS2,KMAX,ISTATE
      COMMON/CID/COMP,DIJ,NUMB,APJ
      COMMON/NOS/K,IERR
      DIMENSION ZZ(40),X(40),Y(40),AK(40),ZC(40),AKO(40)
      DIMENSION TC(40),VC(40),PC(40),W(40),AMOLW(40),AM(40)
      DIMENSION FEED(40),NUMB(40),DIJ(40,40),COMP(40,2)
     1  ,APJ(40,40)

C*********************************************************************
C     INPUT AND OUTPUT FILE
C*********************************************************************
      open(unit=20,file='input.dat',status='old',ACCESS='SEQUENTIAL')
      OPEN(UNIT=21,FILE='OUTPUTN.DAT',STATUS='NEW')

      IL=1
      IV=0
            C1=1.0+DSQRT(2.0D0)
      C2=2.0-C1
      C3=C2-C1
      CPS=5.372697
      R=8.20597D-5
      EPS1=1.D-4
      EPS2=1.D-5
      KMAX=300

C       INPUT BLOCK
C*********************************************************************
C     NC=NUMBER OF COMPONENTS, P1= PRESSURE IN KPa, T1=TEMP. IN K
C     NUMB=IDENTIFICATION OF COMPONENT
C*********************************************************************
      READ(20,*) NC,p1,t1
            READ(20,*) (NUMB(I),I=1,NC)

      IERR=0

      TYPE*,'METHOD=1 , FOR DEM-NR'
      TYPE*,'METHOD=2 , FOR GDEM-NR'
      TYPE*,'METHOD=3 , FOR GUPTA ET AL DEM'
      TYPE*,'METHOD=4,  FOR GUPTA ET AL, GDEM'
      TYPE*,'METHOD=5,  FOR MEHRA ET AL.'
      TYPE*,'METHOD=6,  FOR RISNES ET AL'
      TYPE*,'METHOD'
```

```
      ACCEPT*, METHOD
      TYPE*,'FEED MOLE FRACTIONS'
      TYPE*,'TO TERMINATE PROGRAM TYPE 0 FOR FEED'
95    TYPE*,'feed(i),i=1,nc'
      ACCEPT*, (feed(i),i=1,nc)

      IF (feed(1).EQ.0)GO TO 120
      total=0.
      do 30 i=1,nc
30    total=total+feed(i)
      do 40 i=1,nc
40    zz(i)=feed(i)/total
            P=P1*9.8692327D-03

      T=T1

      call eos(nc)
      CALL APARA(NC)
      RT=R*T
      BOB=P/RT
      AOA=BOB/RT

      DO 90 I=1,NC
90    AK(I)=(DEXP(CPS*(1.+W(I))*(1.-TC(I)/T))*PC(I)/P)

      GO TO (1,2,3,4,5,6),METHOD
1     CALL PHASE1(NC,X,Y,ZZ,AK,VW)
      GO TO 100
2     CALL MICHELSEN(NC,X,Y,ZZ,AK,VW)
      GO TO 100
3     CALL GUPTA1(NC,X,Y,ZZ,AK,VW)
      GO TO 100
4     CALL GUPTA2(NC,X,Y,ZZ,AK,VW)
      GO TO 100
5     CALL MEHRA(NC,X,Y,ZZ,AK,VW)
      GO TO 100
6     CALL RISNES(NC,X,Y,ZZ,AK,VW)
100   IF((VW.EQ.0.0).OR.(VW.EQ.1.0))THEN
      CALL HANDS(NC,X,Y,ZZ,AK,VW)
      ENDIF

C*****************************************************************
C     SUBROUTINE FOR OUTPUT
C*****************************************************************
      CALL RITE(NC,X,Y,ZZ,AK,VW,METHOD)

      GO TO 95
120   STOP
      END

      SUBROUTINE HANDS(NC,X,Y,ZZ,AK,VW)
      IMPLICIT REAL*8(A-H,O-Z),INTEGER(I-N)
      COMMON/CONS/C1,C2,C3,IL,IV,CPS,R,EPS1,EPS2,KMAX,ISTATE
      COMMON/CR/TC,VC,PC,W,AMOLW,AM
```

```
      COMMON/TAP/T,P,RT,AOA,BOB,T1,P1
      COMMON/ENT/VL,VV,ZL,ZV
      DIMENSION TC(20),VC(20),PC(20),W(20),AMOLW(20),AM(20)
      DIMENSION ZZ(20),X(20),Y(20),AK(20),SAX(20)
      K=0.0
      IF (VW.EQ.0.)THEN
      ISTATE=1.0
      ELSE
            ISTATE=0.
      ENDIF
      CALL ABMIX(NC,ZZ,AMX,BMX,SAX)
      CALL VOLUME(AMX,BMX,V,Z,ISTATE)
      IMI=IL-ISTATE
      DO 1 I=1,NC
      AK(I)=1.0
      X(I)=ZZ(I)*ISTATE
1     Y(I)=ZZ(I)*IMI
      VW=IMI
      ZL=Z*ISTATE
      VL=V*ISTATE
      ZV=V*IMI
      VV=V*IMI
      RETURN
      END

      SUBROUTINE EOS(NC)
      IMPLICIT REAL*8(A-H,O-Z),INTEGER(I-N)
      COMMON/CR/TC,VC,PC,W,AMOLW,AM
      COMMON/PAR/B
      COMMON/INT/ACIJ
      COMMON/CONS/C1,C2,C3,IL,IV,CPS,R,EPS1,EPS2,KMAX,ISTATE
      COMMON/CID/COMP,DIJ,NUMB
      COMMON/NOS/K,IERR,K1
        DIMENSION COMP(40,2),TC(40),VC(40),PC(40),W(40),AMOLW(40),
     1  NUMB(40),AM(40),AIN(350),DIJ(40,40),ACIJ(40,40),AC(40),B(40)
      DIMENSION CN1(40),CN2(40),CP(40),CV(40),CT(40),CW(40),CM(40)
      DATA CN1(1),CN2(1),CP(1),CT(1),CV(1),CW(1),CM(1)/'METH',
     1  'ANE',45.80,190.7,0.0990,0.0130,16.043/
      DATA CN1(2),CN2(2),CP(2),CT(2),CV(2),CW(2),CM(2)/'PROP',
     1  'ANE',42.01,369.9,0.200,0.1524,44.097/
      DATA CN1(3),CN2(3),CP(3),CT(3),CV(3),CW(3),CM(3)/'N-BU',
     1  'TANE',37.47,425.2,0.255,0.2010,58.124/
      DATA CN1(4),CN2(4),CP(4),CT(4),CV(4),CW(4),CM(4)/
     1      'N-DE','CANE',20.80,617.6,0.602,0.4885,142.286/
      DATA (AIN(I),I=1,6)/0.012,0.013,0.001,0.041,0.018,0.017/
      IF(NC.EQ.0)GO TO 11
      DO 1 I=1,NC
      J=NUMB(I)
      COMP(I,1)=CN1(J)
      COMP(I,2)=CN2(J)
      PC(I)=CP(J)
      TC(I)=CT(J)
      VC(I)=CV(J)
      W(I)=CW(J)
```

```
      AMOLW(I)=CM(J)
1     CONTINUE
      NM=NC-1
      IF(NM.EQ.0) GO TO 11
      DO 3 II=1,NM
      I=NUMB(II)
      IP1=II+1.
      DO 3 JJ=IP1,NC
      J=NUMB(JJ)
      IF(J.LT.I) THEN
            KIJ=J+(I-1)*(I-2)/2
            DIJ(JJ,II)=AIN(KIJ)
      ELSE
              KIJ=I+(J-1)*(J-2)/2
            DIJ(JJ,II)=AIN(KIJ)
      ENDIF
3     CONTINUE
11    CONTINUE
      DO 6 I=1,NC
      RTC=R*TC(I)
      RTP=RTC/PC(I)
      AC(I)=0.457235*RTP*RTC
      B(I)=0.077796*RTP
      AM(I)=0.37464+(1.54226-0.26992*W(I))*W(I)
      ACIJ(I,I)=AC(I)
6     DIJ(I,I)=0.0
      IF(NC.EQ.1)RETURN
      DO 10 I=2,NC
      IM1=I-1
      DO 10 J=1,IM1
      ACIJ(I,J)=(1.-DIJ(I,J))*DSQRT(AC(I)*AC(J))
      DIJ(J,I)=DIJ(I,J)
10    ACIJ(J,I)=ACIJ(I,J)

      RETURN
      END




      SUBROUTINE APARA(NC)
      IMPLICIT REAL*8(A-H,O-Z),INTEGER(I-N)
      COMMON/TAP/T,P,RT,AOA,BOB,T1,P1
      COMMON/CR/TC,VC,PC,W,AMOLW,AM
      COMMON/INT/ACIJ
      COMMON/AS/AIJ,ALFSQ
      COMMON/NOS/K,IERR,K1
      DIMENSION TC(40),VC(40),PC(40),W(40),AMOLW(40),AM(40)
      DIMENSION ALFSQ(40),ACIJ(40,40),AIJ(40,40)
      DO 1 I=1,NC
      ALFSQ(I)=1.+AM(I)*(1.-DSQRT(T/TC(I)))
1     AIJ(I,I)=ACIJ(I,I)*ALFSQ(I)*ALFSQ(I)
```

```
      IF(NC.EQ.1.)RETURN
      DO 2 I=2,NC
      IM1=I-1
      DO 2 J=1,IM1
      AIJ(I,J)=ACIJ(I,J)*ALFSQ(I)*ALFSQ(J)
2     AIJ(J,I)=AIJ(I,J)
      RETURN
      END

      SUBROUTINE ABMIX(NC,X,AM,BM,SAM)
      IMPLICIT REAL*8(A-H,O-Z),INTEGER(I-N)
      COMMON/AS/AIJ,ALFSQ
      COMMON/NOS/K,IERR
      COMMON/PAR/B
      DIMENSION AIJ(40,40),X(40),SAM(40),ALFSQ(40),B(40)
      AM=0.0
      BM=0.0
      DO 2 I=1,NC
      BM=BM+X(I)*B(I)
      SAM(I)=0.0
      DO 1 J=1,NC
1     SAM(I)=SAM(I)+X(J)*AIJ(I,J)
      AM=AM+X(I)*SAM(I)
2     CONTINUE
      RETURN
      END

      SUBROUTINE VOLUME(AMX,BMX,V,Z,IFASE)
      IMPLICIT REAL*8(A-H,O-Z),INTEGER(I-N)
      COMMON/TAP/T,P,RT,AOA,BOB,T1,P1
      COMMON/AB/AA,BB
      DIMENSION ZR(3),CU(3)
      AA=AMX*AOA
      BB=BMX*BOB
      CU(1)=BB-1.
      CU(2)=AA-BB*(2.+3.*BB)
      CU(3)=BB*(BB*BB+BB-AA)
      CALL CUBEQ(IRT,ZR,CU)
      IF(IRT)1,1,6
1     IF(IFASE)2,2,3
2     Z=DMAX1(ZR(1),ZR(2),ZR(3))
      GO TO 99
3     ZDUM=100.
      DO 5 I=1,3
      IF (ZR(I))4,4,5
4     ZR(I)=ZDUM
5     CONTINUE
      Z=DMIN1(ZR(1),ZR(2),ZR(3))
      GO TO 99
6     Z=ZR(1)
99    V=Z/BOB
      RETURN
      END
```

```
      SUBROUTINE CUBEQ(IROOT,Z,B)
      IMPLICIT REAL*8(A-H,O-Z),INTEGER(I-N)
      DIMENSION B(3),Z(3)
      TRD=1./3.
      B13=B(1)*TRD
      BET=B(3)+B13*(2.*B13*B13-B(2))
      BO2=0.5*BET
      AO3=TRD*(B(2)-B(1)*B13)
      CUA=AO3*AO3*AO3
      SQB=BO2*BO2
      DEL=SQB+CUA
      IF(DEL)30,10,20
10    IROOT=0
      GAM=DSQRT(-AO3)
      IF(BET)12,12,11
11    Z(1)=-2.*GAM-B13
      Z(2)=GAM-B13
      Z(3)=Z(2)
      RETURN
12    Z(1)=2.*GAM-B13
      Z(2)=-GAM-B13
      Z(3)=Z(2)
      RETURN
20    IROOT=1
      EPS=DSQRT(DEL)
      TAU=-BO2
      RCU=TAU+EPS
      SCU=TAU-EPS
      SIR=1.0
      SIS=1.0
      IF(RCU)21,22,22
21    SIR=-1.0
22    IF(SCU)23,24,24
23    SIS=-1.0
24    R=SIR*(SIR*RCU)**TRD
      S=SIS*(SIS*SCU)**TRD
      Z(1)=R+S-B13
      Z(2)=-0.5*(R+S)-B13
      Z(3)=0.8660254*(R-S)
      RETURN
30    IROOT=-1
      ROOT=DSQRT(-SQB/CUA)
      IF(BET)32,31,31
31    PEI=(1.5707963+DATAN(ROOT/DSQRT(1.-ROOT*ROOT)))*TRD
      GO TO 33
32    PEI=DATAN(DSQRT(1.-ROOT*ROOT)/ROOT)*TRD
33    FACT=2.*DSQRT(-AO3)
      CZ=2.0943951
      DO 34 I=1,3
34    Z(I)=FACT*DCOS(PEI+CZ*(I-1))-B13
      RETURN
      END
C*****************************************************************
C     ACCELERATION BASED ON Ln K (DEM) coupled with Newton's methd
```

```
c*****************************************************************
      SUBROUTINE PHASE1(NC,X,Y,ZZ,AK,VW)
      IMPLICIT REAL*8(A-H,O-Z),INTEGER(I-N)
      COMMON/CONS/C1,C2,C3,IL,IV,CPS,R,EPS1,EPS2,KMAX,ISTATE
      COMMON/TAP/T,P,RT,AOA,BOB,T1,P1
      COMMON/RF/RATIO
      COMMON/CR/TC,VC,PC,W,AMOLW,AM
      COMMON/ENT/VL,VV,ZL,ZV
      COMMON/NOS/K,IERR
      COMMON/SWIT/KL,KV,NR
      DIMENSION X(40),Y(40),ZZ(40),AK(40),TC(40),PC(40),VC(40)
        DIMENSION RATIO(40),FL(40),FV(40),SAL(40),SAV(40),W(40),
     1   AMOLW(40),AM(40),AKO(40),SRO(0:200),AKL(40,0:200),
     1   EI(200),FX(40),FY(40)
      K=0.0
      STEP=0
      ksw=0
10    DO 60 I=1,NC
      AKL(I,K)=DLOG(AK(I))
60    CONTINUE
      CALL FLASH(NC,X,Y,ZZ,AK,VW,DGDV)
      if((vw.eq.0).or.(vw.eq.1.0))go to 93
      CALL ABMIX(NC,X,AML,BML,SAL)
      CALL VOLUME(AML,BML,VL,ZL,IL)
      IF(VL.LE.BML) GO TO 25
      CALL FUGA(NC,X,AML,BML,ZL,SAL,FL,FX)
      CALL ABMIX(NC,Y,AMV,BMV,SAV)
      CALL VOLUME(AMV,BMV,VV,ZV,IV)
      IF(VV.LE.BMV) GO TO 25
      CALL FUGA(NC,Y,AMV,BMV,ZV,SAV,FV,FY)
      SRO(K)=0.
      DO 5 I=1,NC
      RATIO(I)=FL(I)/FV(I)
      AK(I)=AK(I)*RATIO(I)
5     SRO(K)=SRO(K)+((1-RATIO(I))**2.)
      SRO(K)=SRO(K)/NC
      IF(SRO(K).Lt.1E-12)RETURN
      IF(K.LT.1)THEN
      K=K+1
      GO TO 10
      ENDIF
      DLTN=0.
      DLTD=0.
      DO 6 I=1,NC
      DLTN=DLTN+(DLOG(RATIO(I)))**2
      DLTD=DLTD+(AKL(I,K)-AKL(I,K-1))**2
6     CONTINUE
      EI(K)=(DLTN/DLTD)**0.5

      IF(K.GT.1)THEN
      CHANG=DABS((EI(K)-EI(K-1))/EI(K))*100.


      ENDIF
```

```
      if((k-ksw).ge.3)then
      IF((k.eq.3).or.(CHANG.LE.10.))then
      ksw=k
      WRITE(21,102)K
102     FORMAT(2X,'acc','k=',I3)
      STEP=STEP+1
      denn=0.
      dedn=0.
      DO 7 I=1,NC
      sign=dlog(ratio(i))/(akl(i,k)-akl(i,k-1))
      if(sign.lt.0)then
      rx=1./(1+ei(k))
      else
      rx=1./(1-ei(k))
      endif
      AK(I)=DEXP(AKL(I,K-1)+(AKL(I,K)-AKL(I,K-1))*RX)
      denn=denn+(dlog(ak(i))-akl(i,k))**2.
      dedn=dedn+akl(i,k)**2.
7     CONTINUE
      deln=(denn/dedn)**0.5
      ENDIF
      endif
      IF((EI(K).GE.0.8).AND.(STEP.GE.2.))THEN
      NR=K
      IF(EI(K).GT.1.)THEN
      DO 100 I=1,NC
      AK(I)=DEXP(AKL(I,K-2))
100   CONTINUE
      ENDIF
      go to 30

      ENDIF
      IF(K.GT.KMAX)GO TO 21
      k=k+1
      GO TO 10


21    IERR=1
25    IERR=2
30    CALL PHASE2(NC,X,Y,ZZ,AK,VW,SRO,K1)

93    RETURN
      END


      SUBROUTINE PHASE2(NC,X,Y,ZZ,AK,VW,SRO,K1)
      IMPLICIT REAL*8(A-H,O-Z),INTEGER(I-N)
      COMMON/CONS/C1,C2,C3,IL,IV,CPS,R,EPS1,EPS2,KMAX,ISTATE
      COMMON/TAP/T,P,RT,AOA,BOB,T1,P1
      COMMON/RF/RATIO
      COMMON/CR/TC,VC,PC,W,AMOLW,AM
      COMMON/ENT/VL,VV,ZL,ZV
      COMMON/NOS/K,IERR
      COMMON/AB/AA,BB
```

```
      COMMON/BZ/BLN,ZMBB
      COMMON/PTXY /ROXY
      COMMON/NT/NCP,NCM
      DIMENSION X(40),Y(40),ZZ(40),AK(40),TC(40),PC(40),VC(40)
        DIMENSION RATIO(40),FL(40),FV(40),SAL(40),SAV(40),W(40),
     1   AMOLW(40),AM(40),ROXY(40,41),ROXX(40,39),ROYY(40,39),DEL(40)
     1  ,XXZ(40),DXDK(39,40),DYDK(39,40),DVDK(40),SRO(200),FX(40),FY(40)
      NCP=NC+1
      NCM=NC-1
71    K1=K
      CALL FLASH(NC,X,Y,ZZ,AK,VW,DGDV)
      IF(VW.EQ.1.0.OR.VW.EQ.0.0)GO TO 93
      CALL ABMIX(NC,X,AML,BML,SAL)
      CALL VOLUME(AML,BML,VL,ZL,IL)
      IF(VL.LE.BML) GO TO 25
      CALL FUGA(NC,X,AML,BML,ZL,SAL,FL,FX)

      CALL DRVS(NC,X,AML,BML,ZL,SAL,IL)
      DO 26 I=1,NC
      DO 26 J=1,NCM
26    ROXX(I,J)=ROXY(I,J+1)
      CALL ABMIX(NC,Y,AMV,BMV,SAV)
      CALL VOLUME(AMV,BMV,VV,ZV,IV)
      IF(VV.LE.BMV) GO TO 25
      CALL FUGA(NC,Y,AMV,BMV,ZV,SAV,FV,FY)
      CALL DRVS(NC,Y,AMV,BMV,ZV,SAV,IV)
      SRO(K)=0.
      DO 35 I=1,NC
      DO 28 J=1,NCM
28    ROYY(I,J)=ROXY(I,J+1)
      RATIO(I)=FL(I)/FV(I)
      ROXY(I,NCP)=1.-RATIO(I)
35    SRO(K)=SRO(K)+DABS(ROXY(I,NCP)**2)
      SRO(K)=SRO(K)/NC

      EPSV=1.-VW
      IF(SRO(K).GT.1.E-12)GO TO 365
      DO 33 I=1,NC
      IF(DABS(AK(I)-1.).GE.EPS1)GO TO 93
33    CONTINUE
      VW=1.0
      GO TO 93
365   CONTINUE
      DO 37 I=1,NC
      XXZ(I)=X(I)*X(I)/ZZ(I)
      DVDK(I)=XXZ(I)/DGDV
37    CONTINUE
      DO 38 L=1,NCM
      DXK=-X(L)*(Y(L)-X(L))/ZZ(L)
      AKK=AK(L)
      DO 381 J=1,NC
      DXDK(L,J)=DXK*DVDK(J)
381   DYDK(L,J)=AKK*DXDK(L,J)
      DXDK(L,L)=DXDK(L,L)-XXZ(L)*VW
```

```
38    DYDK(L,L)=DYDK(L,L)+XXZ(L)*(1.-VW)
      DO 39 I=1,NC
      DO 391 J=1,NC
      RXK=0
      RYK=0.
      DO 392 L=1,NCM
      RXK=RXK+ROXX(I,L)*DXDK(L,J)
392   RYK=RYK+ROYY(I,L)*DYDK(L,J)
391   ROXY(I,J)=RATIO(I)*(RXK-RYK)*AK(J)
      ROXY(I,I)=ROXY(I,I)-RATIO(I)
39    CONTINUE
      CALL LINSYS(NC,ROXY,DEL)
      DO 800 I=1,NC
      IF(DABS(DEL(I)).GT.0.0)GO TO 900
800   CONTINUE
      GO TO 21
900   CONTINUE
      DO 41 I=1,NC
      AK(I)=AK(I)*DEXP(DEL(I))
41    CONTINUE
      K=K+1
      IF(K.GT.KMAX)GO TO 21
      GO TO 71
93    RETURN
21    IERR=1
      RETURN
25    IERR=2
      RETURN
      END

      SUBROUTINE DRVS(NC,X,AMX,BMX,Z,SA,IFASE)
      IMPLICIT REAL*8(A-H,O-Z),INTEGER(I-N)
      COMMON/CR/TC,VC,PC,W,AMOLW,AM
      COMMON/AS/AIJ,ALFSQ
      COMMON/TAP/T,P,RT,AOA,BOB,T1,P1
      COMMON/CONS/C1,C2,C3,IL,IV,CPS,R,EPS1,EPS2,KMAX,ISTATE
      COMMON/NOS/K,IERR,K1
      COMMON/AB/AA,BB
      COMMON/SST/SAB,TEMP
      COMMON/PAR/B
      COMMON/PTXY/ROXY
      COMMON/BZ/BLN,ZMBB
      COMMON/NT/NCP,NCM
      DIMENSION AM(40),TC(40),VC(40),PC(40),W(40),AMOLW(40)
      DIMENSION ALFSQ(40),ACIJ(40,40),AIJ(40,40),B(40),ROXY(40,41)
      DIMENSION X(40),SA(40),SAB(40),TEMP(40)
      U0=AA-(3*BB+2.)*BB
      U3=AA/BB/(Z+C1*BB)/(Z+C2*BB)
      U5=6.*BB+2.
      Z50=Z*(Z-U5)-U0
      DENO=Z*(3.*Z-2.*(1.-BB))+U0
      R3=2.*BLN/AMX
      DO 70 J=1,NCM
      AXY=2.*(SA(J)-SA(NC))*AOA
```

```
      BXY=(B(J)-B(NC))*BOB
      ZXY=AXY*ZMBB+BXY*Z50
      ZXY=-ZXY/DENO
      R1=BXY/BB
      R2=(ZXY-BXY)/ZMBB
      R4=U3*(Z*BXY-BB*ZXY)
      R5=((SAB(J)-SAB(NC))*BLN+ZXY)/BMX
      L=J+1
      DO 70 I=1,NC
      ROXY(I,L)=R1*TEMP(I)+R2+R3*(AIJ(J,I)-AIJ(NC,I))
     1  +R4*SAB(I)-R5*B(I)
70    ROXY(I,L)=-ROXY(I,L)
      RETURN
      END

C*********************************************************************
****


C THIS PROGRAM PERFORMS FLASH CALCULATION USING PENG-ROBINSON EOS
C GDEM(MICHELSEN'S) COUPLED WITH NEWTON RAPHSON HAS BEEN USED
C
*********************************************************************
***
      SUBROUTINE MICHELSEN(NC,X,Y,ZZ,AK,VW)
      IMPLICIT REAL*8(A-H,O-Z),INTEGER(I-N)
      COMMON/CONS/C1,C2,C3,IL,IV,CPS,R,EPS1,EPS2,KMAX,ISTATE
      COMMON/TAP/T,P,RT,AOA,BOB,T1,P1
      COMMON/RF/RATIO
      COMMON/CR/TC,VC,PC,W,AMOLW,AM
      COMMON/ENT/VL,VV,ZL,ZV
      COMMON/NOS/K,IERR
      COMMON/ SWIT/KL,KV,NR
      DIMENSION X(40),Y(40),ZZ(40),AK(40),TC(40),PC(40),VC(40)
        DIMENSION RATIO(40),FL(40),FV(40),SAL(40),SAV(40),W(40),
     1   AMOLW(40),AM(40),AKO(40),SRO(0:200),AKL(40,0:200),
     1   RAT(40,100),FX(40),FY(40)
      K=0.0
      KS=5
10    DO 60 I=1,NC
      AKL(I,K)=DLOG(AK(I))
60    CONTINUE
      CALL FLASH(NC,X,Y,ZZ,AK,VW,DGDV)
      CALL ABMIX(NC,X,AML,BML,SAL)
      CALL VOLUME(AML,BML,VL,ZL,IL)
      IF(VL.LE.BML) GO TO 25
      CALL FUGA(NC,X,AML,BML,ZL,SAL,FL,FX)
      CALL ABMIX(NC,Y,AMV,BMV,SAV)
      CALL VOLUME(AMV,BMV,VV,ZV,IV)
      IF(VV.LE.BMV) GO TO 25
      CALL FUGA(NC,Y,AMV,BMV,ZV,SAV,FV,FY)
      SRO(K)=0.
      DO 5 I=1,NC
      RATIO(I)=FL(I)/FV(I)
```

```
      AK(I)=AK(I)*RATIO(I)
5     SRO(K)=SRO(K)+((1-RATIO(I))**2.)
      SRO(K)=SRO(K)/NC
      IF(SRO(K).LE.1E-12)RETURN
      IF(K.GE.(KS-2))THEN
      DO 11 I=1,NC
      RAT(I,K)=DLOG(RATIO(I))
11    CONTINUE
      ENDIF
      IF(K.LT.1)THEN
      K=K+1
      GO TO 10
      ENDIF
      IF(K.EQ.KS)THEN
      B01=0.
      B02=0.
      B11=0.
      B12=0.
      B22=0.
      B33=0.
      DO 12 I=1,NC
      B01=B01+RAT(I,K)*RAT(I,K-1)
      B02=B02+RAT(I,K)*RAT(I,K-2)
      B11=B11+RAT(I,K-1)**2
      B12=B12+RAT(I,K-1)*RAT(I,K-2)
      B22=B22+RAT(I,K-2)**2.
      B33=B33+AKL(I,K)**2
12    CONTINUE

      DLTD=(B11*B22-B12**2)
      EMU1=(B02*B12-B01*B22)/DLTD
      EMU2=(B01*B12-B02*B11)/DLTD
      DENO=1.+EMU1+EMU2
      DENN=0.
      DO 13 I=1,NC
      AK(I)=DEXP(AKL(I,K)+((RAT(I,K)-EMU2*RAT(I,K-1))/DENO))
      DENN=DENN+((DLOG(AK(I)))-AKL(I,K))**2
13    CONTINUE
      DELN=(DENN/B33)**0.5

      KS=KS+5
      ENDIF

      IF(K.GT.KMAX)GO TO 21
      k=k+1
      if(k.gt.10)THEN
      NR=10
      GO TO 30
      ENDIF
      GO TO 10


21    IERR=1
25    IERR=2
```

```
30    CALL PHASE2(NC,X,Y,ZZ,AK,VW,SRO,K1)

      RETURN
      END



      SUBROUTINE LINSYS(N,AM,X)
      IMPLICIT REAL*8(A-H,O-Z),INTEGER(I-N)
      DIMENSION AM(40,41),IP(40),JP(40),X(40)
      NB=N+1
      DO 1 I=1,N
1     IP(I)=0.
      DO 6 M=1,N
      PIVOT=0.0
      DO 4 I=1,N
      DO 2 MM=1,M
      IF(I.EQ.IP(MM)) GO TO 4
2     CONTINUE
      DO 3 J=1,N
      IF(DABS(PIVOT).GT.DABS(AM(I,J))) GO TO 3
      PIVOT=AM(I,J)
      IP(M)=I
      JP(M)=J
3     CONTINUE
4     CONTINUE
      K=IP(M)
      L=JP(M)
      IF(PIVOT.EQ.0.0) GO TO 113
      DO 41 J=1,NB
41    AM(K,J)=AM(K,J)/PIVOT
      DO 43 I=1,N
      IF(I.EQ.K) GO TO 43
      AN=AM(I,L)
      DO 42 J=1,NB
42    AM(I,J)=AM(I,J)-AM(K,J)*AN
      AM(I,L)=0.
43    CONTINUE
6     CONTINUE
      DO 8 I=1,N
      DO 7 J=1,N
      IF(AM(I,J).NE.0.0) GO TO 8
7     CONTINUE
      GO TO 113
8     CONTINUE
      DO 9 M=1,N
      I=IP(M)
      J=JP(M)
9     X(J)=AM(I,NB)
      RETURN
113   DO 12 I=1,N
12    X(I)=0.0
      RETURN
      END
```

```
C*************************************************************************
****
C THIS PROGRAM PERFORMS FLASH CALCULATION USING PENG-ROBINSON EOS
C ACCELERATION SCHEME BASED ON GUPTA ET AL.(DEM)
C
*************************************************************************
***
      SUBROUTINE GUPTA1(NC,X,Y,ZZ,AK,VW)
      IMPLICIT REAL*8(A-H,O-Z),INTEGER(I-N)
      COMMON/CONS/C1,C2,C3,IL,IV,CPS,R,EPS1,EPS2,KMAX,ISTATE
      COMMON/TAP/T,P,RT,AOA,BOB,T1,P1
      COMMON/RF/RATIO
      COMMON/CR/TC,VC,PC,W,AMOLW,AM
      COMMON/ENT/VL,VV,ZL,ZV
      COMMON/NOS/K,IERR,k1
      COMMON/SWIT/KL,KV,NR
      DIMENSION X(40),Y(40),ZZ(40),AK(40),TC(40),PC(40),VC(40)
        DIMENSION RATIO(40),FL(40),FV(40),SAL(40),SAV(40),W(40),
     1   AMOLW(40),AM(40),XOLD(40),YOLD(40),sro(0:100)
     1   ,rat(40),fy(40),fx(40)

C     EPSU=TOLERANCE LIMIT

      EPSU=1.E-6

C     KL=ITERATION NO FOR LIQUID UPDATE
C     KV=ITERATION NO FOR VAPOR UPDATE

      K=0.0
      ERX2=0.
      ERY2=0.
10    CALL FLASH(NC,X,Y,ZZ,AK,VW,DGDV)
      KL=0.0
      KV=0.
      DO 120 I=1,NC
      XOLD(I)=X(I)
      YOLD(I)=Y(I)
120   CONTINUE


      CALL ABMIX(NC,Y,AMV,BMV,SAV)
      CALL VOLUME(AMV,BMV,VV,ZV,IV)
      IF(VV.LE.BMV) GO TO 25
      CALL FUGA(NC,Y,AMV,BMV,ZV,SAV,FV,FY)
      KV=KV+1
3     CALL ABMIX(NC,X,AML,BML,SAL)
      CALL VOLUME(AML,BML,VL,ZL,IL)
      IF(VL.LE.BML) GO TO 25
      CALL FUGA(NC,X,AML,BML,ZL,SAL,FL,FX)
      KL=KL+1
4     DO 5 I=1,NC
      RAT(I)=FX(I)/FY(I)
```

```
      AK(I)=RAT(I)
5     CONTINUE
      sro(k)=0.
      do 101 i=1,nc
      RATIO(I)=FL(I)/FV(I)
      sro(k)=sro(k)+(RATIO(I)-1.)**2.
101   continue
      sro(k)=sro(k)/nc
      if(sro(k).le.1.d-12)return
      K=K+1
      CALL FLASH(NC,X,Y,ZZ,AK,VW,DGDV)
      type*,k,kl,kv,vw
      ERY1=ERY2
      ERY2=0.
      DO 12 I=1,NC
      ERY2=ERY2+(YOLD(I)-Y(I))**2
12    continue
      ERY2=(ERY2)**0.5/float(NC)
      IF(KV.EQ.1)GO TO 190
      DO 100 I=1,NC
      IF(DABS(YOLD(I)-Y(I)).GT.1.D-5)GO TO 51
100   CONTINUE
      GO TO 50
51    IF(K.GT.KMAX) GO TO 21
      KVAC=MOD(KV,3)
      IF(KVAC.NE.0)GO TO 190
      EIY=ERY2/ERY1
      W1=1./(1.-EIY)
      SUMY=0.
      DO 180 I=1,NC
      Y(I)=YOLD(I)+W1*(Y(I)-YOLD(I))
180   SUMY=SUMY+Y(I)
      DO 185 I=1,NC
185   Y(I)=Y(I)/SUMY
190   DO 195 I=1,NC
195   YOLD(I)=Y(I)
      CALL ABMIX(NC,Y,AMV,BMV,SAV)
      CALL VOLUME(AMV,BMV,VV,ZV,IV)
      IF(VV.LE.BMV) GO TO 25
      CALL FUGA(NC,Y,AMV,BMV,ZV,SAV,FV,FY)
      KV=KV+1

      GO TO 4
50    ERX1=ERX2
      ERX2=0.
      DO 200 I=1,NC
      ERX2=ERX2+(XOLD(I)-X(I))**2
200   CONTINUE
      ERX2=(ERX2)**0.5/FLOAT(NC)
      DO 111 I=1,NC
      IF(DABS(XOLD(I)-X(I)).GT.1.D-5)GO TO 52
111   CONTINUE
      GO TO 13
52    IF(KL.EQ.1)GO TO 6
```

```
      IF(K.GE.KMAX) GO TO 21
      KLAC=MOD(KL,3)
      IF(KLAC.NE.0)GO TO 6
      EIX=ERX2/ERX1
      W2=1./(1.-EIX)
      SUMX=0.
      DO 210 I=1,NC
      X(I)=XOLD(I)+W2*(X(I)-XOLD(I))
210   SUMX=SUMX+X(I)
      DO 220 I=1,NC
220   X(I)=X(I)/SUMX
6     DO 225 I=1,NC
      XOLD(I)=X(I)
      YOLD(I)=Y(I)
225   CONTINUE
      GO TO 3
21    IERR=1
13    IF((VW.EQ.0).OR.(VW.EQ.1.0))RETURN
      RETURN
25    IERR=2
      RETURN
      END
C*********************************************************************
****
C THIS PROGRAM PERFORMS FLASH CALCULATION USING PENG-ROBINSON EOS
C ACCELERATION SCHEME BASED ON GUPTA ET AL.(GDEM)
C
*********************************************************************
***

      SUBROUTINE GUPTA2(NC,X,Y,ZZ,AK,VW)
      IMPLICIT REAL*8(A-H,O-Z),INTEGER(I-N)
      COMMON/CONS/C1,C2,C3,IL,IV,CPS,R,EPS1,EPS2,KMAX,ISTATE
      COMMON/TAP/T,P,RT,AOA,BOB,T1,P1
      COMMON/RF/RATio
      COMMON/CR/TC,VC,PC,W,AMOLW,AM
      COMMON/ENT/VL,VV,ZL,ZV
      COMMON/NOS/K,IERR,k1
      COMMON/SWIT/KL,KV,NR
      DIMENSION X(40),Y(40),ZZ(40),AK(40),TC(40),PC(40),VC(40)
        DIMENSION RATIO(40),FL(40),FV(40),SAL(40),SAV(40),W(40),
     1   AMOLW(40),AM(40),XOLD(40),YOLD(40),sro(0:100)
     1   ,rat(40),fy(40),fx(40),XK(40,100),YK(40,100)

C     EPSU=TOLERANCE LIMIT

      EPSU=1.E-6
      KS=5
      KS1=5
C     KL=ITERATION NO FOR LIQUID UPDATE
C     KV=ITERATION NO FOR VAPOR UPDATE

      K=0.0
10    CALL FLASH(NC,X,Y,ZZ,AK,VW,DGDV)
```

```
      KL=0
      KV=0

      do 132 i=1,nc
      yold(i)=y(i)
      xold(i)=x(i)
132   continue
      CALL ABMIX(NC,Y,AMV,BMV,SAV)
      CALL VOLUME(AMV,BMV,VV,ZV,IV)
      IF(VV.LE.BMV) GO TO 25
      CALL FUGA(NC,Y,AMV,BMV,ZV,SAV,FV,FY)
      KV=KV+1
3     CALL ABMIX(NC,X,AML,BML,SAL)
      CALL VOLUME(AML,BML,VL,ZL,IL)
      IF(VL.LE.BML) GO TO 25
      CALL FUGA(NC,X,AML,BML,ZL,SAL,FL,FX)
      KL=KL+1
      if(kL.ge.(ks1-2))then
      DO 17 I=1,NC
      Xk(I,kL)=X(I)
17    continue
      ENDIF

4     DO 5 I=1,NC
      RAT(I)=FX(I)/FY(I)
      AK(I)=RAT(I)
5     CONTINUE
      sro(k)=0.
      do 101 i=1,nc
      RATIO(I)=FL(I)/FV(I)
      sro(k)=sro(k)+(RATIO(I)-1.)**2.
101   continue
      sro(k)=sro(k)/nc
      if(sro(k).le.1.d-12)return
      K=K+1
      CALL FLASH(NC,X,Y,ZZ,AK,VW,DGDV)
      IF(KV.EQ.1)GO TO 190
      DO 100 I=1,NC
      IF(DABS(YOLD(I)-Y(I)).GT.epsu)GO TO 51
100   CONTINUE
      GO TO 50
51    IF(K.GT.KMAX) GO TO 21

      KVAC=MOD(KV,5)
      IF(KVAC.NE.0)GO TO 190
      B01=0.
      B02=0.
      B11=0.
      B12=0.
      B22=0.
      DO 12 I=1,NC
      B01=B01+(Y(I)-yk(i,kv))*(YK(I,KV)-yk(i,kv-1))
      B02=B02+(Y(I)-yk(i,kv))*(YK(I,KV-1)-yk(i,kv-2))
      B11=B11+(YK(I,KV)-yk(I,kv-1))**2
```

```
      B12=B12+(yk(i,kv)-YK(I,KV-1))*(yk(i,kv-1)-YK(I,KV-2))
      B22=B22+(YK(I,KV-1)-yk(i,kv-2))**2
12    CONTINUE
      DLTD=B11*B22-B12**2
      EMU1=(B02*B12-B01*B22)/DLTD
      EMU2=(B01*B12-B02*B11)/DLTD
      DENO=1.+EMU1+EMU2
      SUMY=0.
      DO 13 I=1,NC
      Y(I)=Y(I)+(Y(I)-YK(I,KV)-EMU2*(YK(I,KV)-
     1  YK(I,KV-1)))/DENO
      SUMY=SUMY+Y(I)
13    CONTINUE
      DO 185 I=1,NC
185   Y(I)=Y(I)/SUMY
      KS=KS+5
190   CALL ABMIX(NC,Y,AMV,BMV,SAV)
      CALL VOLUME(AMV,BMV,VV,ZV,IV)
      IF(VV.LE.BMV) GO TO 25
      CALL FUGA(NC,Y,AMV,BMV,ZV,SAV,FV,FY)
      KV=KV+1
      do 141 i=1,nc
      yold(i)=y(i)
141   continue
      if(kv.ge.(ks-2))then
      DO 11 I=1,NC
      yk(I,kv)=y(I)

11    continue
      endif
      GO TO 4
50    DO 111 I=1,NC
      IF(DABS(XOLD(I)-X(I)).GT.EPSU)GO TO 52
111   CONTINUE
      GO TO 130
52    IF(KL.EQ.1)GO TO 6
      IF(K.GE.KMAX) GO TO 21
      KLAC=MOD(KL,5)
      IF(KLAC.NE.0)GO TO 6
      B01=0.
      B02=0.
      B11=0.
      B12=0.
      B22=0.
      DO 16 I=1,NC
      B01=B01+(x(i)-XK(I,KL))*(xk(i,kl)-xk(i,kl-1))
      B02=B02+(x(i)-XK(I,KL))*(xk(i,kl-1)-XK(I,KL-2))
      B11=B11+(xk(i,kl)-XK(I,KL-1))**2
      B12=B12+(xk(i,kl)-XK(I,KL-1))*(xk(i,kl-1)-XK(I,KL-2))
      B22=B22+(xk(i,kl-1)-XK(I,KL-2))**2
16    CONTINUE
      DLTD=B11*B22-B12**2
      EMU1=(B02*B12-B01*B22)/DLTD
      EMU2=(B01*B12-B02*B11)/DLTD
```

```
      DENO=1.+EMU1+EMU2
      SUMX=0.
      DO 14 I=1,NC
      X(I)=X(I)+(X(I)-XK(I,KL)-EMU2*(XK(I,KL)-
     1  XK(I,KL-1)))/DENO
      SUMX=SUMX+X(I)
14    CONTINUE
      ks1=ks1+5
6     do 142 i=1,nc
142   xold(I)=x(i)

      GO TO 3
21    IERR=1
      RETURN
25    IERR=2
130   RETURN
      END

C*********************************************************************
****
C THIS PROGRAM PERFORMS FLASH CALCULATION USING PENG-ROBINSON EOS
C ACCELERATION SCHEME BASED ON MEHRA ET AL.
C
*********************************************************************
***

      SUBROUTINE MEHRA(NC,X,Y,ZZ,AK,VW)
      IMPLICIT REAL*8(A-H,O-Z),INTEGER(I-N)
      COMMON/CONS/C1,C2,C3,IL,IV,CPS,R,EPS1,EPS2,KMAX,ISTATE
      COMMON/TAP/T,P,RT,AOA,BOB,T1,P1
      COMMON/RF/RATIO
      COMMON/CR/TC,VC,PC,W,AMOLW,AM
      COMMON/ENT/VL,VV,ZL,ZV
      COMMON/NOS/K,IERR,k1
      COMMON/SWIT/KL,KV
      DIMENSION X(40),Y(40),ZZ(40),AK(40),TC(40),PC(40),VC(40)
        DIMENSION RATIO(40),FL(40),FV(40),SAL(40),SAV(40),W(40),
     1   AMOLW(40),AM(40),sro(0:100),gold(40),gnew(40),xyz(40),
     1  roxy(40,40),FX(40),FY(40)
      K=0.0
      step=1.
      ksw=1
10    CALL FLASH(NC,X,Y,ZZ,AK,VW,DGDV)
      CALL ABMIX(NC,X,AML,BML,SAL)
      CALL VOLUME(AML,BML,VL,ZL,IL)
      IF(VL.LE.BML) GO TO 25
      CALL FUGA(NC,X,AML,BML,ZL,SAL,FL,FX)

      CALL ABMIX(NC,Y,AMV,BMV,SAV)
      CALL VOLUME(AMV,BMV,VV,ZV,IV)
      IF(VV.LE.BMV) GO TO 25
      CALL FUGA(NC,Y,AMV,BMV,ZV,SAV,FV,FY)
      DO 5 I=1,NC
      RATIO(I)=FL(I)/FV(I)
```

```
      gold(i)=-dlog(ratio(i))
5     CONTINUE
      k=k+1
      sro(k)=0.
      do 101 i=1,nc
      sro(k)=sro(k)+(ratio(i)-1.)**2.
101   continue
      sro(k)=sro(k)/nc
      if(sro(k).le.1.d-12)return
      do 8 i=1,nc
      ak(i)=ak(i)*ratio(i)
8     continue
      if(k.lt.ksw)go to 10
70    CALL FLASH(NC,X,Y,ZZ,AK,VW,DGDV)
      CALL ABMIX(NC,X,AML,BML,SAL)
      CALL VOLUME(AML,BML,VL,ZL,IL)
      IF(VL.LE.BML) GO TO 25
      CALL FUGA(NC,X,AML,BML,ZL,SAL,FL,FX)

      CALL ABMIX(NC,Y,AMV,BMV,SAV)
      CALL VOLUME(AMV,BMV,VV,ZV,IV)
      IF(VV.LE.BMV) GO TO 25
      CALL FUGA(NC,Y,AMV,BMV,ZV,SAV,FV,FY)
      do 35 i=1,nc
      ratio(i)=fl(i)/fv(I)
      gnew(i)=-dlog(ratio(i))
      roxy(i,ncp)=1.-ratio(i)
35    continue
      sro(k)=0.
      do 102 i=1,nc
      sro(k)=sro(k)+(ratio(i)-1.)**2.
102   continue
      sro(k)=sro(k)/nc
      if(sro(k).le.1.d-12)return

      IF(K.GT.KMAX) GO TO 21
      s=0.
      do 37 i=1,nc
      xyz(i)=x(i)*y(i)/zz(i)
37    s=s+xyz(i)
      s=1.-s
      do 38 i=1,nc
      xyzi=xyz(i)
      do 381 j=1,nc
      roxy(i,j)=xyzi*xyz(j)
381   continue
      roxy(i,i)=roxy(i,i)+xyzi*s
38    continue
      sum3=0.
      sum2=0.
      do 391 j=1,nc
      sum1=0.
      do 392 i=1,nc
      sum1=sum1+gold(i)*roxy(i,j)
```

```
392   continue
      sum2=sum2+sum1*gold(j)
      sum3=sum3+sum1*gnew(j)
391   continue
      scale=sum2/(sum2-sum3)
      scale=dabs(scale)
      step=step*scale
      do 41 i=1,nc
      gold(i)=gnew(i)
      rak=-step*gnew(i)
      if(dabs(rak).gt.6.)then
      rak=dsign(6.d0,rak)
      endif
      ak(i)=ak(i)*dexp(rak)
41    continue
      k=k+1
      go to 70

21    ierr=1
      return
25    ierr=2
      return
      end

C**********************************************************************
****
C THIS PROGRAM PERFORMS FLASH CALCULATION USING PENG-ROBINSON EOS
C ACCELERATION SCHEME BASED ON RISNES
C
**********************************************************************
***
      SUBROUTINE RISNES(NC,X,Y,ZZ,AK,VW)
      IMPLICIT REAL*8(A-H,O-Z),INTEGER(I-N)
      COMMON/CONS/C1,C2,C3,IL,IV,CPS,R,EPS1,EPS2,KMAX,ISTATE
      COMMON/TAP/T,P,RT,AOA,BOB,T1,P1
      COMMON/RF/RATio
      COMMON/CR/TC,VC,PC,W,AMOLW,AM
      COMMON/ENT/VL,VV,ZL,ZV
      COMMON/NOS/K,IERR,k1
      DIMENSION X(40),Y(40),ZZ(40),AK(40),TC(40),PC(40),VC(40)
        DIMENSION RATIO(40),FL(40),FV(40),SAL(40),SAV(40),W(40),
     1   AMOLW(40),AM(40),sro(0:100),aold(40),rold(40),FX(40),FY(40)
      id=0
      K=0.0
      ksw=5
      type*,'ks'
      accept*,ks
      WRITE(21,*)'ACC INTERVAL=',KS
10    do 8 i=1,nc
      rold(i)=ratio(I)
8     continue
      CALL FLASH(NC,X,Y,ZZ,AK,VW,DGDV)
      CALL ABMIX(NC,X,AML,BML,SAL)
      CALL VOLUME(AML,BML,VL,ZL,IL)
```

```
      IF(VL.LE.BML) GO TO 25
      CALL FUGA(NC,X,AML,BML,ZL,SAL,FL,FX)

      CALL ABMIX(NC,Y,AMV,BMV,SAV)
      CALL VOLUME(AMV,BMV,VV,ZV,IV)
      IF(VV.LE.BMV) GO TO 25
      CALL FUGA(NC,Y,AMV,BMV,ZV,SAV,FV,FY)
      DO 5 I=1,NC
      RATIO(I)=FL(I)/FV(I)
5     CONTINUE

      sro(k)=0.
      do 101 i=1,nc
      sro(k)=sro(k)+(ratio(i)-1.)**2.
101   continue
      sro(k)=sro(k)/nc
      WRITE(22,111)K,SRO(K)
111   FORMAT(1X,I3,2X,E15.5)
      IF(ID.EQ.1)THEN
      RAT=SRO(K)/SRO(K-1)
      IF(RAT.GT.1)THEN
      DO 90 I=1,NC
      AK(I)=AOLD(I)
90    CONTINUE
      k=k+1
      go to 10
      endif
      endif
      do 91 i=1,nc
      ak(i)=ak(i)*ratio(i)
      aold(I)=ak(I)
91    continue

      if(sro(k).le.1.d-12)return
      k=k+1
      if((k-ksw).eq.0)then
      ksw=ksw+ks
      ID=1
      DO 9 I=1,NC
      sK=(RATIO(I)-1)/(ROLD(I)-1)
      step=1./(1.-sk)
9     ak(i)=ak(i)*(ratio(i)**step)
      id=0
      ELSE
      endif
      go to 10

21    ierr=1
      return
25    ierr=2
      return
      end
```

```
      SUBROUTINE FLASH(NC,X,Y,ZZ,AK,VW,DF)
      IMPLICIT REAL*8(A-H,O-Z), INTEGER(I-N)
      COMMON/CONS/C1,C2,C3,IL,IV,CPS,R,EPS1,EPS2,ISTATE
      COMMON/NOS/K,IERR
      DIMENSION X(40),Y(40),ZZ(40),AK(40)
      SUMX=0.0
      do 30 I=1,nc
      if(ak(i).gt.1.e4)then
      ums=0
      go to 271
      endif
30    continue
      DO 1 I=1,NC
      X(I)=ZZ(I)/AK(I)
1     SUMX=SUMX+X(I)
      UMS=1.-SUMX
271   IF(UMS.GE.0.0) THEN
            VW=1.0
c           DO 22 I=1,NC
c           Y(I)=ZZ(I)
c           X(I)=X(I)/SUMX
c 22        CONTINUE
            RETURN


      ENDIF
            SUMY=0.0
      DO 2 I=1,NC
      Y(I)=ZZ(I)*AK(I)
2     SUMY=SUMY+Y(I)
      SMU=SUMY-1.
      IF (SMU.LE.0.)THEN
            VW=0.0
            DO 32 I=1,NC
            X(I)=ZZ(I)
            Y(I)=Y(I)/SUMY
32          CONTINUE
            RETURN
      ENDIF
      VW=SMU/(SMU-UMS)
      AVO=VW
10    F=0.0
      DF=0.
C     SUMX=0.
C     SUMY=0.
      DO 11 I=1,NC
      X(I)=ZZ(I)/(1.+(AK(I)-1.)*VW)
C     SUMX=SUMX+X(I)
      Y(I)=AK(I)*X(I)
C     SUMY=SUMY+Y(I)
      YMX=Y(I)-X(I)
      F=F+YMX
      DF=DF+YMX*YMX/ZZ(I)
11    CONTINUE
```

```
      DLV=F/DF

      IF (DABS(DLV).LT.eps2)RETURN
21    VW=VW+DLV
      IF(DABS(VW).GE.1.0) VW=0.5*(1.0+AVO)
      AVO=VW
      GO TO 10
      END
      SUBROUTINE FUGA(NC,X,AMX,BMX,Z,SA,FUG,FG)
      IMPLICIT REAL*8(A-H,O-Z),INTEGER(I-N)
      COMMON/TAP/T,P,RT,AOA,BOB,T1,P1
      COMMON/NOS/K,IERR
      COMMON/CONS/C1,C2,C3,IL,IV,CPS,R,EPS1,EPS2,KMAX,ISTATE
      COMMON/PAR/B
      COMMON/BZ/BLN,ZMBB
      COMMON/AB/AA,BB
      common/sst/sab,temp
      DIMENSION SA(40), SAB(40),TEMP(40),FUG(40),B(40),X(40),FG(40)
      ZMBB=Z-BB
      ALN=-DLOG(ZMBB)
      BLN=AA*DLOG((Z+C2*BB)/(Z+C1*BB))/BB/C3
      DO 3 I=1,NC
      BIOB=B(I)/BMX
      SAB(I)=SA(I)*2./AMX-BIOB
      TEMP(I)=(Z-1.)*BIOB-BLN*SAB(I)
      FOXP=TEMP(I)+ALN
      FG(I)=DEXP(FOXP)
3     FUG(I)=X(I)*P*DEXP(FOXP)
      RETURN
      END

      SUBROUTINE RITE(NC,X,Y,ZZ,AK,VW,METHOD)
      IMPLICIT REAL*8(A-H,O-Z),INTEGER(I-N)
      COMMON/TAP/T,P,RT,AOA,BOB,T1,P1
      COMMON/CR/TC,VC,PC,W,AMOLW,AM
      COMMON/CID/COMP,DIJ,NUMB
      COMMON/RF/RATIO
      COMMON/ENT/VL,VV,ZL,ZV
      COMMON/NOS/K,IERR
      COMMON/SWIT/KL,KV,NR
      DIMENSION ZZ(40),X(40),Y(40),AK(40),COMP(40,2),RATIO(40)
      DIMENSION TC(40),VC(40),PC(40),W(40),AMOLW(40),AM(40)
            PSI=P*0.101325
      TF=T
      VL=VL*1000.
      VV=VV*1000.

      WRITE(21,61)PSI,TF
      IF(IERR.LE.0)THEN
            WRITE(21,1000)
1000  FORMAT(2X,'COMP.',9X,'Z',13X,'X',14X,'Y',13X,'Ki',12x,'Ri',/)
      WRITE(21,114)((COMP(I,J),J=1,2),ZZ(I),X(I),Y(I),AK(I),
     1  RATIO(I),I=1,NC)
```

```
      WRITE(23,2000)
2000  FORMAT(2X,'COMP.',10X,'PC',12X,'TC',13X,'W',13X,'MW',/)

      WRITE(23,121)((COMP(I,J),J=1,2),PC(I),TC(I),W(I),AMOLW(I),
     1      I=1,NC)
      WRITE(21,115)ZL,ZV,VL,VV
      WRITE(21,116)K
      IF((METHOD.EQ.3).OR.(METHOD.EQ.4))THEN

            WRITE(21,119)KL,KV
      ENDIF
            IF((METHOD.EQ.1).OR.(METHOD.EQ.2))THEN
      NR1=K-NR
            WRITE(21,118)NR1
      ENDIF

      WRITE(21,117)VW
      ELSEIF (IERR.EQ.1.)THEN
      WRITE(21,71)
      ELSEIF(IERR.EQ.2.)THEN
      WRITE(21,72)
      ELSEIF(IERR.EQ.3)THEN
      WRITE(21,73)
      ENDIF
61    FORMAT(1X,'FLASH CALCULATION AT',F10.2,'MPa',F10.2,'DEG. K',/)
71    FORMAT(1X,'NO. OF ITERATIONS REACHED MAXIMUM')
72    FORMAT(1X,'UNREALISTIC VOLUME')
73    FORMAT(1X,'POSSIBLE SINGLE PHASE')
114   FORMAT(1X,2A4,5E14.5)
121   FORMAT(1X,2A4,4F14.5)
115   FORMAT(/,1X,'ZL=',F10.5,2X,'ZV=',F10.5,2X,'VL=',F10.5,1x,
     1      'l/gm mole',2X,'VV=',F10.5,1x,'l/g mole')
119   FORMAT(1X,'LIQUID UPDATE=',I3,2X,'VAPOR UPDATE='I3)
116   FORMAT(//,1X,'NO. OF ITERATIONS=',I3)
118   FORMAT(//,1X,'NO. OF NEWTONS ITERATIONS=',I3)
117   FORMAT(//,1X,'VAPOR FRACTION=',F10.5)
      RETURN
      END
```