```
SUBROUTINE pitzer(mNa,mCl,mCa,mSO4,gamma_Ca,gamma_SO4,aH2O)

!Uses the Pitzer equations to calculate the activity coefficients of Ca and SO4
!and the activity of H2O in Ca-Na-Cl-SO4 solutions. Illustrates the use of the Pitzer equations.
!Equation numbers from Harvie & Weare (1980) are indicated by HW().

IMPLICIT none

REAL,INTENT(in) ::  mNa, mCl, mSO4, mCa
REAL,INTENT(out)::  gamma_Ca, gamma_SO4, aH2O
REAL            ::  v, w, y, f_gamma, Z, F, IS, ISp
REAL            ::  phi_1 NaCa, phip_NaCa, phiphi_NaCa, Etheta, Ethetap
REAL            ::  phi_ClSO4, phip_ClSO4, phiphi_ClSO4
REAL            ::  term1, term1a, term1b, term2, term3
REAL            ::  term4, term5, term6, term7, term8, term9
REAL            ::  terma, termb, termc, termd
REAL            ::  CNaCl, CNaSO4, CCaCl, CCaSO4
REAL            ::  ln_gamma_Ca, ln_gamma_SO4, osmotic, sum_m
!parameter data are from Harvier & Weare (1980).
REAL,PARAMETER  ::  Aphi = 0.392, b = 1.2
REAL,PARAMETER  ::  alpha = 2.0, alpha1 = 1.4, alpha2 = 12.0
REAL,PARAMETER  ::  theta_NaCa = 0.07, theta_ClSO4 = 0.02
REAL,PARAMETER  ::  psi_NaCaCl = -0.014, psi_NaCaSO4 = -0.023
REAL,PARAMETER  ::  psi_ClSO4Na = 0.0014, psi_ClSO4Ca = 0.0
REAL,PARAMETER  ::  CphiNaCl=0.00127, CphiNaSO4=0.00497, CphiCaCl=-0.00034, CphiCaSO4=0.0
REAL,DIMENSION(3),PARAMETER :: BNaCl  = (/ 0.07650, 0.2664, 0.0 /)
REAL,DIMENSION(3),PARAMETER :: BNaSO4 = (/ 0.01958, 1.1130, 0.0 /)
REAL,DIMENSION(3),PARAMETER :: BCaCl  = (/ 0.31590, 1.6140, 0.0 /)
REAL,DIMENSION(3),PARAMETER :: BCaSO4 = (/ 0.20000, 2.6500, -57.70 /)

EXTERNAL mixing

!SOME PRELIMINARIES:
!===================

!sum of all m terms
sum_m = mNa+mCa+mCl+mSO4

!Ionic strength functions
IS = (mNa+mCl+4*mCa+4*mSO4)/2
ISp = sqrt(IS)

!some commonly used parameters
v = alpha *ISp
w = alpha1*ISp
y = alpha2*ISp
Z = mNa+mCl+2*mCa+2*mSO4

!convert Cphi to C                                                                    HW(9)
CNaCl  = CphiNaCl /2
CNaSO4 = CphiNaSO4/(2*sqrt(2.0))
CCaCl  = CphiCaCl /(2*sqrt(2.0))
CCaSO4 = CphiCaSO4/4

!calculate unsymmetrical mixing terms Etheta, Ethetap
```

```
CALL mixing(IS,Etheta,Ethetap)

!calculate phi terms: interactions between ions of like sign
phiphi_NaCa = theta_NaCa + Etheta + IS*Ethetap                                  HW(10)
phi_NaCa = theta_NaCa + Etheta
phip_NaCa = Ethetap

phiphi_ClSO4 = theta_ClSO4 + Etheta + IS*Ethetap
phi_ClSO4 = theta_ClSO4 + Etheta
phip_ClSO4 = Ethetap

!D-H TERM:
!=========

term1 = mNa*mCl *(BNaCl(2) *gp(v)/IS                    )           &
      + mNa*mSO4*(BNaSO4(2)*gp(v)/IS                    )           &
      + mCa*mCl *(BCaCl(2) *gp(v)/IS                    )           &
      + mCa*mSO4*(BCaSO4(2)*gp(w)/IS + BCaSO4(3)*gp(y)/IS)

term1a = mNa*mCa *phip_NaCa
term1b = mCl*mSO4*phip_ClSO4

f_gamma = -Aphi*((ISp/(1+b*ISp)) + (2/b)*(log(1+b*ISp)))
F = f_gamma + term1 + term1a + term1b                                           HW(3)

!ACTIVITY COEFFICIENT OF Ca:
!===========================

term2 = mCl *(2*(BCaCl(1) + BCaCl(2) *g(v)                    ) + Z*CCaCl)    &
        + mSO4*(2*(BCaSO4(1)+ BCaSO4(2)*g(w) + BCaSO4(3)*g(y)) + Z*CCaSO4)

term3 = mNa*(2*phi_NaCa + mCl*psi_NaCaCl + mSO4*psi_NaCaSO4)

term4 = mCl*mSO4*psi_ClSO4Ca

term5 = 2*(mNa*mCl*CNaCl + mNa*mSO4*CNaSO4                                    &
        + mCa*mCl*CCaCl + mCa*mSO4*CCaSO4)

ln_gamma_Ca = 4*F + term2 + term3 + term4 + term5                               HW(2b)
gamma_Ca = exp(ln_gamma_Ca)

!ACTIVITY COEFFICIENT OF SO4:
!============================

term6 = mNa*(2*(BNaSO4(1) + BNaSO4(2)*g(v)                    ) + Z*CNaSO4)    &
        + mCa*(2*(BCaSO4(1) + BCaSO4(2)*g(w) + BCaSO4(3)*g(y)) + Z*CCaSO4)

term7 = mCl*(2*phi_ClSO4 + mNa*psi_ClSO4Na + mCa*psi_ClSO4Ca)

term8 = mNa*mCa*psi_NaCaSO4

term9 = term5

ln_gamma_SO4 = 4*F + term6 + term7 + term8 + term9
gamma_SO4 = exp(ln_gamma_SO4)                                                   HW(2c)
```

```
!OSMOTIC COEFFICIENT:
!===================

terma = -Aphi*IS**(1.5)/(1+b*ISp)

termb = mNa*mCl   * (BNaCl(1) + BNaCl(2) *exp(-v)                  + Z*CNaCl ) &
       + mNa*mSO4 * (BNaSO4(1)+ BNaSO4(2)*exp(-v)                  + Z*CNaSO4) &
       + mCa*mCl  * (BCaCl(1) + BCaCl(2) *exp(-v)                  + Z*CCaCl ) &
       + mCa*mSO4 * (BCaSO4(1)+ BCaSO4(2)*exp(-w) + BCaSO4(3)*exp(-y) + Z*CCaSO4)

termc = mNa*mCa*(phiphi_NaCa + mCl*psi_NaCaCl + mSO4*psi_NaCaSO4)

termd = mCl*mSO4*(phiphi_ClSO4 + mNa*psi_ClSO4Na + mCa*psi_ClSO4Ca)

osmotic = 1+(2/sum_m)*(terma + termb + termc + termd)                      HW(2a)
aH2O = exp(-osmotic*sum_m/55.51)

CONTAINS

function g(x)
     IMPLICIT none
     REAL :: g,x
     g = 2*(1-(1+x)*exp(-x))/x**2
END function g

function gp(x)
     IMPLICIT none
     REAL :: gp,x
     gp = -2*(1-(1+x+x**2/2)*exp(-x))/x**2
END function gp

END SUBROUTINE pitzer
```

E.1.2 Subroutine Mixing

Subroutine Mixing is called by subroutine Pitzer, and calculates the variables ${}^E\theta_{ij}$ and ${}^E\theta'_{ij}$, given the ionic strength. These variables were defined by Pitzer (1975, 1983), and account for the interaction of ions of like but different charge, that is, where i and j are both cations or both anions (such as Na^+ and Ca^{2+}, or Cl^- and SO_4^{2-} in this case). The term ${}^E\theta_{ij}$ was derived by Pitzer from the statistical mechanics theory of Friedman (1962). Its evaluation involves integrals [functions J_0, J_1 in the notation of Harvie and Weare (1980)] which have no analytical solution, and Pitzer (1975, 1995) gives approximate methods. Harvie (1981) and Harvie and Weare (1980) used a more accurate scaled Chebyshev expansion method, also described in Pitzer (1987, 1991), which is used in this subroutine.

Harvie (1981) fit numerically derived values of J_0 with two Chebyshev polynomial approximations, one for $x \leq 1$ (lines 47–53) and one for $x > 1$ (lines 55–60), in which arrays *bk* and *dk* are calculated. Then using these, J_0 and its x-derivative J_1 are calculated from the formulas in lines 63–72. There are three varieties of each *J*-function, because they depend on the charges on the ions as well as the ionic strength. The differences are established in lines 37–39. The subroutine uses double precision throughout, using the symbol dp, established in line 11.

```
SUBROUTINE mixing(I,Etheta,Ethetap)                                                          1
                                                                                             2
!Evaluates unsymmetrical mixing terms Etheta, Ethetap, using the Chebyshev approximation     3
!mentioned in Harvie & Weare (1980), Appendix.                                               4
                                                                                             5
IMPLICIT none                                                                                6
                                                                                             7
REAL,INTENT(in)              :: I                                                            8
REAL,INTENT(out)             :: Etheta,Ethetap                                               9
INTEGER                      :: m,k                                                         10
INTEGER,PARAMETER            :: dp = kind(1.0D0)                                            11
REAL(DP)                     :: x,xMN,xMM,xNN,z,dzdx,JMN,JMM,JNN,JpMN,JpMM,JpNN             12
REAL(DP),PARAMETER           :: Aphi = 0.392                                                13
REAL(DP),DIMENSION(0:20,2)   :: ak                                                          14
REAL(DP),DIMENSION(0:22)     :: bk,dk                                                       15
                                                                                            16
!array ak values are from Pitzer (1991) Table B-1, and copied from file phrqpitz.for        17
!in the USGS phrqptz distribution                                                           18
ak(0:20,1) = (/1.925154014814667_dp, -0.060076477753119_dp, -0.029779077456514_dp,      &   19
             -0.007299499690937_dp,  0.000388260636404_dp,  0.000636874599598_dp,       &   20
              0.000036583601823_dp, -0.000045036975204_dp, -0.000004537895710_dp,       &   21
              0.000002937706971_dp,  0.000000396566462_dp, -0.000000202099617_dp,       &   22
             -0.000000025267769_dp,  0.000000013522610_dp,  0.000000001229405_dp,       &   23
             -0.000000000821969_dp, -0.000000000050847_dp,  0.000000000046333_dp,       &   24
              0.000000000001943_dp, -0.000000000002563_dp, -0.000000000010991_dp/)          25
                                                                                            26
ak(0:20,2) = (/0.628023320520852_dp,  0.462762985338493_dp,  0.150044637187895_dp,      &   27
             -0.028796057604906_dp, -0.036552745910311_dp, -0.001668087945272_dp,       &   28
              0.006519840398744_dp,  0.001130378079086_dp, -0.000887171310131_dp,       &   29
             -0.000242107641309_dp,  0.000087294451594_dp,  0.000034682122751_dp,       &   30
             -0.000004583768938_dp, -0.000003548684306_dp, -0.000000250453880_dp,       &   31
              0.000000216991779_dp,  0.000000080779570_dp,  0.000000004558555_dp,       &   32
             -0.000000006944757_dp, -0.000000002849257_dp,  0.000000000237816_dp/)          33
                                                                                            34
!zCa = +2; zNa = +1 and zSO4 = -2; zCl = 1                                                  35
!so if M is Ca or SO4 and N is Na or Cl,                                                    36
xMN = 6*2*Aphi*sqrt(I)     ! i.e., 6*zCa*zNa*0.0392*sqrt(I); 6*2*zCl*zSO4*0.0392*sqrt(I)    HW(A1)
xMM = 6*4*Aphi*sqrt(I)     ! i.e., 6*zCa*zCa*0.0392*sqrt(I); 6*4*zSO4*zSO4*0.0392*sqrt(I)     38
xNN = 6*1*Aphi*sqrt(I)     ! i.e., 6*zNa*zNa*0.0392*sqrt(I); 6*1*zCl*zCl*0.0392*sqrt(I)       39
                                                                                            40
!this DO loop evaluates the J functions as described in Pitzer (1991) appendix B.           41
DO k=1,3                                                                                    42
     IF (k==1) x=xMN                                                                        43
     IF (k==2) x=xMM                                                                        44
     IF (k==3) x=xNN                                                                        45
                                                                                            46
     IF (x <= 1)           THEN                                                             47
               z = 4.0_dp * x**(0.2_dp) - 2.0_dp                                            48
               dzdx = (0.8_dp) * x**(-0.8_dp)                                               49
               DO m=20,0,-1                                                                 50
```

```
            bk(m) = z*bk(m+1) - bk(m+2) + ak(m,1)
            dk(m) = bk(m+1) + z*dk(m+1) - dk(m+2)
      END DO
ELSE
      z = (40.0_dp/9.0_dp) * x**(-0.1_dp) - 22.0_dp/9.0_dp
      dzdx = -(40.0_dp/90.0_dp) * x**(-1.1_dp)
      DO m=20,0,-1
            bk(m) = z*bk(m+1) - bk(m+2) + ak(m,2)
            dk(m) = bk(m+1) + z*dk(m+1) - dk(m+2)
      END DO
END IF

IF (k==1) THEN
      JMN = 0.25_dp*x -1.0_dp + 0.5_dp*(bk(0) - bk(2))
      JpMN = 0.25_dp + 0.5_dp*(dzdx)*(dk(0) - dk(2))
ELSE IF (k==2) THEN
      JMM = 0.25_dp*x -1.0_dp + 0.5_dp*(bk(0) - bk(2))
      JpMM = 0.25_dp + 0.5_dp*(dzdx)*(dk(0) - dk(2))
ELSE
      JNN = 0.25_dp*x -1_dp + 0.5_dp*(bk(0) - bk(2))
      JpNN = 0.25_dp + 0.5_dp*(dzdx)*(dk(0) - dk(2))
END IF
END DO

!finally, calculation of the Etheta terms.
Etheta = (2.0_dp/(4.0_dp*I)) * (JMN - 0.5_dp*JMM - 0.5_dp*JNN)                 ! HW(A2)
Ethetap=-(Etheta/I)+(2.0_dp/(8.0_dp*I**2))*(xMN*JpMN-0.5_dp*xMM*JpMM-0.5_dp*xNN*JpNN)! HW(A3)

END SUBROUTINE mixing
```

E.2 MATLAB

E.2.1 Program SPECIES

This is essentially the program EQBRM (Anderson and Crerar, 1993, Appendix E) written in MATLAB®. It uses input prepared by a spreadsheet and is easily modified to handle multiple inputs.

```
% SPECIES.M
% speciation program, patterned after EQBRM.
% like EQBRM, can only operate on one composition at a time.
% all input data (except species labels) are in a single matrix.
% m is number of species; n is number of rxns; (m-n-1) is number of components.

% input matrix size is (m,m+5)
% matrix format:
% ==============
% row 1   : m,n,all other columns zero.
% row 2   : stoichiometric coeffs for rxn 1.
%  :
% row n+1 : stoichiometric coeffs for rxn n.
% row n+2 : mass balance  for component 1.
%  :
% row m   : mass balance for component (m-n-1).
% row m+1 : (m-n-1) compositions, all other columns zero.
```