Ion Activity, Ion Association and Solubility

CEE 373

Roadmap

SANDBOX	IMPLEMENTATION	IMPLEMENTATION
Modeling concepts, scales and approaches	Examination of Equilibrium-based Code	Visualization, Interface Design and Usability

SANDBOX	IMPLEMENTATION	READINESS
Programming languages, software engineering & numerical methods	Examination of Reaction Rate-based Code	Internal Testing and Code Freeze

DESIGN	IMPLEMENTATION	RELEASE
Project Proposal	Examination of Existing Models for Complex Systems	Final Presentations ("Rollout")

EQUILIBRIUM THERMODYNAMICS

OBJECTIVES
1. Build a modeling framework for equilibrium chemistry
2. Examine and understand computer code.
3. Produce model results and interpret critically.

EQUILIBRIUM THERMODYNAMICS

1. Ion activity
   - Review of general expressions
   - Activity coefficient calculations
   - Implementation in computer code
2. Ion pairing
   - Calculation of ion pairing
   - Implementation in computer code
   - Calculation of $CaSO_4^0$ formation
3. Solubility
   - Saturation Index (SI)
   - Interpretation of SI results
   - Model calculations (Wateq4f, Visual Minteq)

Equilibrium State and Ion Activity
A QUICK REVIEW OF RELEVANT EXPRESSIONS

Generalized chemical reaction : $aA + bB \Leftrightarrow cC + dD$

Activity, $a_i = m_i \gamma_i$

Ionic Strength, $I = \dfrac{1}{2} \displaystyle\sum_i m_i z_i^2$

$K_{eq} = \dfrac{a_C a_D}{a_A a_B} = \dfrac{m_C \gamma_C m_D \gamma_D}{m_A \gamma_A m_B \gamma_B}$

Mean activity coefficient, $\gamma_{\pm} = \sqrt{\gamma_+ \gamma_-}$

Equilibrium State and Ion Activity
A QUICK REVIEW OF RELEVANT EXPRESSIONS

Debye–Hückel equation

The chemical potential or activity of ions cannot be determined on a purely thermodynamic basis. This is due to the fact that the effects of an ion cannot be separated from the effects of the accompanying counter-ion, or in other terms, the electrochemical potential of the ion cannot be separated into the chemical and the electrical component. Such a separation must necessarily be based on a non-thermodynamic convention. The present convention is based on the assumption that the molal activity coefficient of the chloride ion in dilute aqueous solutions ($I < 0.10$ mol kg^{-1}) can be estimated by means of the Debye–Hückel equation:

$$- \lg \gamma_B = z_B{}^2 A I^{1/2} / (1 + \mathring{a} B I^{1/2})$$

where I is ionic strength, z is the charge number of the ion, $\mathring{a}$ is ion size parameter and A and B are temperature-dependent constants.
According to the Bates–Guggenheim convention $\mathring{a}B$ is taken to be 1.5 (mol kg^{-1})$^{1/2}$ at all temperatures and for all compositions of the solutions.
  See also *pH*.
  1984, *56*, 569

Equilibrium State and Ion Activity

A QUICK REVIEW OF ACTIVITY COEFFICIENT FORMULAE

Debye-Hückel Limiting Law
Very low ionic strength (I<10^{-2.3} M)

$$\log \gamma_i = -0.5 z_i^2 \sqrt{I}$$

Extended Debye-Hückel Equation
Low ionic strength (I<10^{-1} M)

$$\log \gamma_i = -A z_i^2 \frac{\sqrt{I}}{1 + B\mathring{a}\sqrt{I}}$$

Davies Equation
Low ionic strength (I<0.5 M)

$$\log \gamma_i = \frac{-A z_i^2 \sqrt{I}}{1 + \sqrt{I}} + 0.3 A z_i^2 I$$

B-dot Equation
Designed for NaCl solutions

$$\log \gamma_i = \frac{-A z_i^2 \sqrt{I}}{1 + B\mathring{a}_i \sqrt{I}} + \dot{B} I$$

Equilibrium State and Ion Activity

IMPLEMENTATION IN CODE

Davies Equation

Low ionic strength (I< 10^{-5} M)

$$\log \gamma_i = \frac{-Az_i^2 \sqrt{I}}{1 + \sqrt{I}} + 0.3Az_i^2 I$$

From
HYDRAQL.FOR ➝

```
C*********************
      ENTRY IONCOR(XMU)
      NC=NN(1)+NN(2)+NN(3)+NN(4)+NN(5)+NN(6)
      NX=NNN
C
C The A-factor in ACTCOF calculation is obtained
C using a regression fit of Helgeson/
C Kirkham data 25-225 deg and Harned/Owen
C data below 25 TO 0.
C
      DT=TEMP-25.
      AFACTR=0.50886+0.0008*DT+0.00001*DT*DT
      ET=-AFACTR
      XIS=SQRT (XMU)
      GF(1)=ET*(XIS/(1.0+XIS)-0.3*XMU)
      DO 2100 IZ=1,5
          GF(IZ)=GF(1)*IZ*IZ
          ACTCOF(IZ)=10.**GF(IZ)
 2100 CONTINUE
```

Equilibrium State and Ion Activity

IMPLEMENTATION IN CODE

From
phreeqe.f
Revision 1.14
(1993) ⟶

```
C               CALCULATE ACTIVITY COEFFICIENTS
      AMU=-A*MUHALF
      BMU=B*MUHALF
      CMU=-A*(MUHALF/(1.0D0+MUHALF)-0.3D0*MU)
      ZCHRG=0.1D0*MU
      LG(1)=AMU/(1.0D0+DHA(1)*BMU)
      IF (IOPT(6).EQ.1) LG(1)=CMU
      LG(2)=0.0D0
      LG(3)=0.0D0
      DO 70 I=4,LASTS
      IF (SFLAG(I).EQ.0) GO TO 70
      IF (DABS(ZSP(I)).LT.1.0D-40) GO TO 40
      IF (GFLAG(I).EQ.1) GO TO 50
      IF (DHA(I).LE.0.0D0) GOTO 60
      IF (IOPT(6).EQ.1) GO TO 60
C             EXTENDED DEBYE-HUCKEL WITH ION SIZE PARAMETER
      LG(I)=AMU*ZSP(I)*ZSP(I)/(1.0D0+DHA(I)*BMU)
      GO TO 65
C             GAMMA FOR UNCHARGED SPECIES
   40 LG(I)=ZCHRG
      GO TO 65
C             WATEQ DEBYE-HUCKEL
   50 LG(I)=AMU*ZSP(I)*ZSP(I)/(1.0D0+ADHSP(I,1)*BMU)+ADHSP(I,2)*MU
      GO TO 65
C             DAVIES GAMMA
   60 LG(I)=CMU*ZSP(I)*ZSP(I)
   65 CONTINUE
      IF(LG(I).LT.-1.0D1) LG(I)=-1.0D1
      IF(LG(I).GT.1.0D1) LG(I)=1.0D1
   70 CONTINUE
      RETURN
```

Generalized Steps for Computational Solution

1. Compose a balanced chemical reaction. If more than one reaction occurs simultaneously, write a reaction for each.
2. Invoke Law of Mass Action and write equilibrium constant expression(s), and relate to numeric values of K_{eq}.
3. Produce other pertinent expressions such as mass balance and charge balance. Need to have as many relations as unknowns.
4. Solve algebraically. Typically, successive approximations are needed to converge on a value.

Application to Ion Pairing

$CaSO_4^0$ in 0.01M $CaSO_4$ Solution

$$CaSO_4{}^0 \Leftrightarrow Ca^{2+} + SO_4{}^{2-}$$
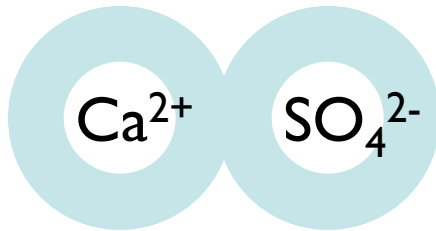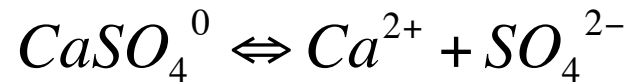
```
Private Sub Command1_Click()
pKCaSO4 = -2.274
IPFC = 10 ^ pKCaSO4
tolerance = 0.01
maxIterations = 30
loops = 0
oldIS = 0
conc = Val(Text1.Text)
CaTot = conc
SO4Tot = conc
Ca = CaTot * 0.5
SO4 = SO4Tot * 0.5
electron = 2 * (CaTot - SO4Tot)
IonStrength = (2 * (Ca + SO4)) + (0.5 * electron)
Do
oldIS = IonStrength
loops = loops + 1
RootIS = Sqr(IonStrength)
Gamma2 = 10 ^ (-1 * (2 * (RootIS / (1 + RootIS) - 0.3 * IonStrength)))
IPCaSO4 = Gamma2 * Gamma2 * Ca * SO4 / IPFC
Ca = CaTot - IPCaSO4
SO4 = SO4Tot - IPCaSO4
IonStrength = (2 * (Ca + SO4)) + (0.5 * electron)
ChangeIS = Abs(IonStrength - oldIS)
Loop While (ChangeIS > tolerance) Or (loops < maxIterations)
ActCa = Ca * Gamma2
Text2.Text = "{Ca2+} = " & ActCa
End Sub
```

Solubility Calculations

Solubility Reaction

Solid phase, dissociated species

Solubility Product, K_{sp}

Relationship to equilibrium constant

Ion Activity Product, IAP

Product of free ion species activities

Saturation Index, SI

$SI = \log (IAP/K_{sp})$

Solubility Calculations

$$SI = \log \frac{IAP}{K_{sp}}$$
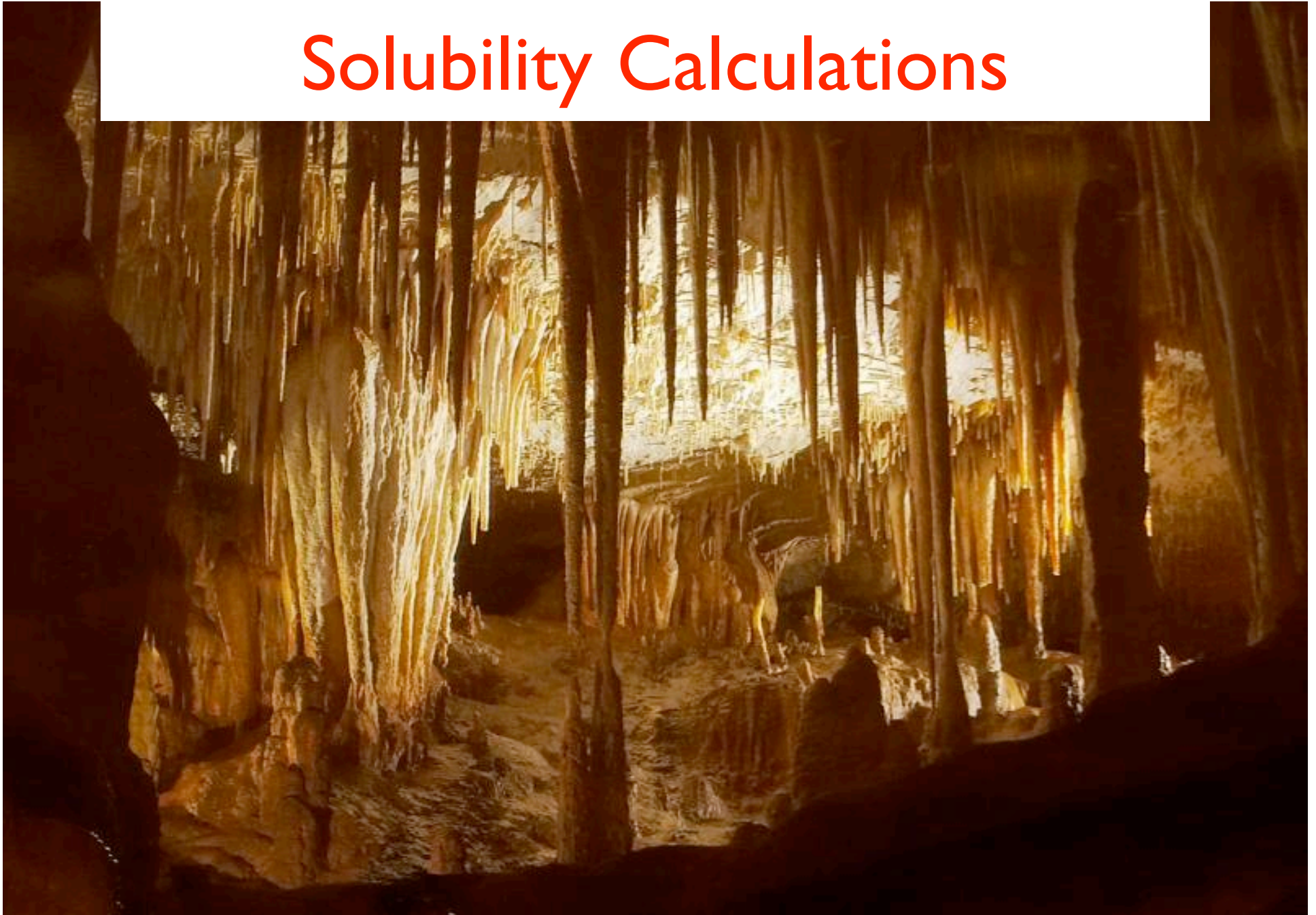
If IAP<K_{sp}, then SI is -ve. Mineral dissolves.

If IAP>K_{sp}, then SI is +ve. Mineral precipitates.

If IAP=K_{sp}, then SI is zero. Mineral equilibrium with solution.

How do you interpret saturation indices?

- Possibility vs Reality

- Ideal Equilibrium vs Local Equilibrium

- Supersaturation - why would it occur?

Solubility Calculations

Vadose Cave ("Ceiling Leaks") Water Example

Solubility Calculations

Vadose Cave ("Ceiling Leaks") Water Example

T (°C) = 12   pH = 7.56   DO (ppm) = 6.8

Concentrations (mg/l)

Ca = 92   Cl = 6.0

Mg = 24.9 SO_4 = 0.4

Na = 1.6 HCO_3 = 366.0

Sr = 0.02 SiO_2 = 8.0

- What do the modeling results tell you about the chemistry in the cave?

- What would you need to know to make a sound interpretation of the results?

Numeric Types: Visual BASIC

Visual Basic type	Common language runtime type structure	Nominal storage allocation	Value range
Boolean	System.Boolean	2 bytes	**True** or **False**.
Byte	System.Byte	1 byte	0 through 255 (unsigned).
Char	System.Char	2 bytes	0 through 65535 (unsigned).
Date	System.DateTime	8 bytes	0:00:00 on January 1, 0001 through 11:59:59 PM on December 31, 9999.
Decimal	System.Decimal	16 bytes	0 through +/-79,228,162,514,264,337,593,543,950,335 with no decimal point; 0 through +/-7.9228162514264337593543950335 with 28 places to the right of the decimal; smallest nonzero number is +/-0.0000000000000000000000000001 (+/-1E-28).
Double (double-precision floating-point)	System.Double	8 bytes	-1.79769313486231570E+308 through -4.94065645841246544E-324 for negative values; 4.94065645841246544E-324 through 1.79769313486231570E+308 for positive values.
Integer	System.Int32	4 bytes	-2,147,483,648 through 2,147,483,647.
Long (long integer)	System.Int64	8 bytes	-9,223,372,036,854,775,808 through 9,223,372,036,854,775,807.
Object	System.Object (class)	4 bytes	Any type can be stored in a variable of type **Object**.
Short	System.Int16	2 bytes	-32,768 through 32,767.
Single (single-precision floating-point)	System.Single	4 bytes	-3.4028235E+38 through -1.401298E-45 for negative values; 1.401298E-45 through 3.4028235E+38 for positive values.