Geofluids — Part IV

Introduction to Geochemical and Reactive Transport Modeling

Dr. Svetlana Kyas (extension of the lecture notes of Dr. Allan Leal)

May 17, 2021

Postdoc Associate
svetlana.kyas@erdw.ethz.ch
NO F 61

Table of contents i

Numerical method for chemical equilibrium calculation

Recap of a chemical equilibrium problem

- Calculate the amounts of the species in each phase at equilibrium at given:
    - **temperature** (e.g., 100 °C);
    - **pressure** (e.g, 300 bar); and
    - **amounts of elements H, O, C, Na, Cl**
    - **Note**: the latter is rather given as a recipe that determine the values of H, O, C, Na, Cl by formula matrix $b = A\, n$, e.g.,
        - 1 kg of H_2O,
        - 1 mol CO_2,
        - 0.1 mol of NaCl.

Aqueous Phase	Gaseous Phase	Mineral Phases
$H_2O(aq)$	$CO_2(g)$	NaCl(s, halite)
$H^+(aq)$	$H_2O(g)$	
$OH^-(aq)$		
$Na^+(aq)$		
$Cl^-(aq)$		
$CO_2(aq)$		
$HCO_3^-(aq)$		
$CO_3^{2-}(aq)$		

Fundamental condition for chemical equilibrium reads as follows:

Gibbs energy minimization (GEM) problem

Given temperature T, pressure P, and elements amounts $b = (b_1, \ldots, b_E)$, find the amounts of the species $n = (n_1, \ldots, n_N)$ that solve the **constrained minimization problem**:

$$\min_n G(n) \qquad \text{subject to (s.t.)} \qquad An = b \quad \text{and} \quad n \geq 0.$$

How do we transform it into a system of equations from which species amounts $n = (n_1, \ldots, n_N)$ can be calculated?

Numerical method for chemical equilibrium calculation

Unconstrained minimization

Chemical equilibrium equations, Finding the minimum

- Assume there were **no constraints** of mass conservation and non-negative bounds.
- Then, the **chemical equilibrium equations** would result from the condition

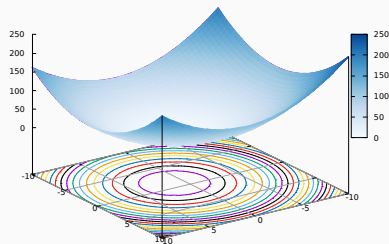$$\frac{\partial G}{\partial n_i} = 0 \qquad (i = 1, \ldots, \mathsf{N}),$$

  which corresponds to a state in which the **Gibbs energy does not change with infinitesimal changes in the amounts of the species**.

- **But!** some care would still be needed to ensure that a **minimum is found**, and not a **maximum** or **saddle point**.

Example of finding the minimum

- Consider the function f defined as:

$$f(x, y) = (x - 1)^2 + (y + 1)^2.$$

- The values of (x, y) that minimizes f can be found using:

$$\frac{\partial f}{\partial x} = 0 \implies 2(x - 1) = 0 \implies x = 1$$

$$\frac{\partial f}{\partial y} = 0 \implies 2(y + 1) = 0 \implies y = -1.$$

- **Question**: How can we tell, mathematically, that the solution above corresponds to a minimum?

  http://etc.ch/cKvh   or

- **Answer:** Check if the derivatives $\partial^2 f / \partial x^2$ and $\partial^2 f / \partial y^2$ at $(1, -1)$ are non-negative.

Distinguishing between minimum, maximum, saddle point values

- In general, one checks the **eigenvalues of Hessian matrix**

$$H = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x^2} & \dfrac{\partial^2 f}{\partial x \partial y} \\ \dfrac{\partial^2 f}{\partial y \partial x} & \dfrac{\partial^2 f}{\partial y^2} \end{bmatrix}$$

  at obtained stationary point $(x_\star, y_\star)$.

- If **the eigenvalues are non-negative** at the $(x_\star, y_\star)$ (or Hessian is a positive semi-definite matrix at $(x_\star, y_\star)$), this point is indeed a **local minimum**.

- If **the eigenvalues are negative** at the $(x_\star, y_\star)$ (or Hessian is negative-definite at $(x_\star, y_\star)$), this point is a **local maximum**.

- If **the eigenvalues are of different signs** at the $(x_\star, y_\star)$, this point is a **saddle point**.

Example of saddle point problem

- Consider the function f defined as:

$$f(x, y) = x^2 - y^2.$$
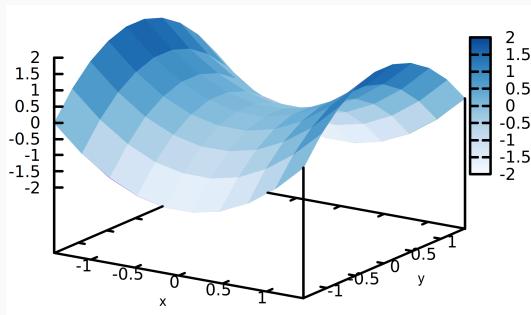
- At $(x, y) = (0, 0)$:

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial y} = 0,$$

  but $f(0, 0) = 0$ is **neither a minimum nor a maximum**.

- **Quiz:** What are the signs of $\partial^2 f / \partial x^2$ and $\partial^2 f / \partial y^2$ for this example?

  http://etc.ch/cKvh   or

- **Answer:** The values of $\partial^2 f / \partial x^2 > 0$ and $\partial^2 f / \partial y^2 < 0$ at $(0, 0)$, which implies a **saddle point**.

Numerical method for chemical equilibrium calculation

Constrained minimization

Lagrangian function of chemical equilibrium problem

- Because of the **mass conservation constraints**, $An = b$, and the **non-negative constraints**, $n \geq 0$, we need to write the **Lagrange function** L for the Gibbs energy minimization problem:

$$L(n, y, z) := G(n) - (An - b)^T y - n^T z,$$

where the unknowns n, y, and z correspond to
- the vector of *species amounts* $n = (n_1, \ldots, n_N)$,
- the vector of *Lagrange multipliers* $y = (y_1, \ldots, y_E)$, and
- the vector of *complementary variables* $z = (z_1, \ldots, z_N)$ that satisfy

$$n_i z_i = 0, \qquad n_i \geq 0, \qquad z_i \geq 0.$$

- **Note**: The above form of the Lagrangian function is called **matrix form** but one can write it explicitly for each unknown n_i, y_j, and z_i.

System of equation corresponding to minimization of Lagrangian

- Instead of finding only $n = (n_1, \ldots, n_N)$ that solves

$$\frac{\partial G}{\partial n_i} = 0 \qquad (i = 1, \ldots, N),$$

  we find $n = (n_1, \ldots, n_N)$, $y = (y_1, \ldots, y_E)$, and $z = (z_1, \ldots, z_N)$ that solves

$$\frac{\partial L}{\partial n_i} = 0 \qquad (i = 1, \ldots, N),$$

$$\frac{\partial L}{\partial y_j} = 0 \qquad (j = 1, \ldots, E),$$

$$n_i z_i = 0 \qquad (i = 1, \ldots, N),$$

$$n_i, z_i \geq 0 \qquad (i = 1, \ldots, N).$$

System of equation corresponding to minimization of Lagrangian   ii

We rewrite the system of equations

using the definition of $\mu_i := \partial G / \partial n_i$:

$$\frac{\partial L}{\partial n_i} = 0 \qquad (i = 1, \ldots, \mathsf{N}),$$

$$\mu_i - \sum_{j=1}^{\mathsf{E}} A_{ji} y_j - z_i = 0 \qquad (i = 1, \ldots, \mathsf{N}),$$

$$\frac{\partial L}{\partial y_j} = 0 \qquad (j = 1, \ldots, \mathsf{E}),$$

$$\sum_{i=1}^{\mathsf{N}} A_{ji} n_i - b_j = 0 \qquad (j = 1, \ldots, \mathsf{E}),$$

$$n_i z_i = 0 \qquad (i = 1, \ldots, \mathsf{N}),$$

$$n_i z_i = 0 \qquad (i = 1, \ldots, \mathsf{N}),$$

$$n_i, z_i \geq 0 \qquad (i = 1, \ldots, \mathsf{N}),$$

$$n_i, z_i \geq 0 \qquad (i = 1, \ldots, \mathsf{N}).$$

Note: Since μ_i is a function of unknown n_i, we obtain **non-linear system of equation**.

Interior-point method for constrained minimization

Abstract constrained minimization formulation

- Solve the following general **constrained minimization problem**:

$$\min_x f(x) \qquad \text{subject to} \qquad Ax = b \quad \text{and} \quad x \geq 0.$$

- **Quiz**: What would be $f(x)$ and x in a context of chemical equilibrium problem?

http://etc.ch/cKvh   or

- **Answer**: We will define f as our Gibbs energy function

$$f(x) = G(x; T, P),$$

where

  - $x = (x_1, \ldots, x_n)$ represents the unknown species amounts, $n = (n_1, \ldots, n_N)$,
  - A specifies the formula matrix of the system, and
  - b stands for the vector of element amounts.

Lagrangian of general minimization formulation

- First, we write the **Lagrange function for the general minimization problem**:

$$L(x, y, z) = f(x) - (Ax - b)^T y - x^T z.$$

- Then, we formulate its **first-order necessary conditions for the minimum**:

$$\partial f / \partial x - A^T y - z = 0,$$
$$Ax - b = 0,$$
$$x_i z_i = 0 \qquad (i = 1, \ldots, n),$$
$$x_i, z_i \geq 0 \qquad (i = 1, \ldots, n).$$

Complementarity conditions

- The **complementarity conditions**

$$x_i z_i = 0$$
$$x_i \geq 0$$
$$z_i \geq 0$$

  are **challenging to solve numerically**.

- They are sharp conditions, in which

$$x_i \geq 0 \qquad \text{and} \qquad z_i = 0$$

  or

$$x_i = 0 \qquad \text{and} \qquad z_i \geq 0.$$

# Interior-point method for constrained minimization

Interior-point perturbation approach

Interior-point perturbation approach

- Perturb $x_i z_i = 0$ with a **small and constant number**

$$x_i z_i = \tau.$$

- **Note:** the choice of τ affects the smallest amount a species that can exist in equilibrium.

- **Example:** the choice $\tau = 10^{-20}$ causes species in **unstable phases** to have amounts around this value.

Smoothing the condition $x_i z_i = 0$

- **Quiz**: What can we say about the amount of mineral dissolved in water?
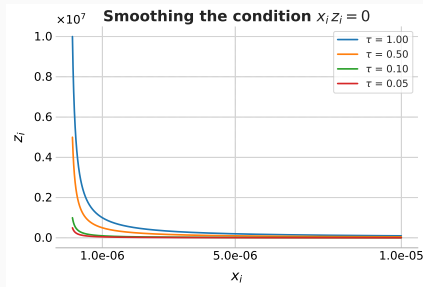
http://etc.ch/cKvh   or

- **Answer**: The amount will be of order τ.

Interior-point Newton method

- We'll use **Newton's method** to solve the obtained system of non-linear equations:

$$\frac{\partial f}{\partial x} - A^T y - z = 0,$$
$$Ax - b = 0,$$
$$x_i z_i = \tau \qquad (i = 1, \ldots, n),$$
$$x_i, z_i \geq 0 \qquad (i = 1, \ldots, n).$$

- Define the **residual function** F as follows:

$$F(x, y, z) := \begin{bmatrix} \frac{\partial f}{\partial x} - A^T y - z \\ Ax - b \\ XZe - \tau \end{bmatrix},$$

where
- $e = (1, \ldots, 1)^T$,
- $X = \mathsf{diag}(x)$, and
- $Z = \mathsf{diag}(z)$.

The resulting problem: find (x, y, z), with $x, z \geq 0$, so that:

$$F(x, y, z) = 0.$$

Jacobian matrix for the interior-point Newton method

- Newton's method require the Jacobian J of the residual function:

$$F(x, y, z) := \begin{bmatrix} \frac{\partial f}{\partial x} - A^T y - z \\ Ax - b \\ XZe - \tau \end{bmatrix},$$

which is defined as

$$J := \begin{bmatrix} \frac{\partial F_1}{\partial x} & \frac{\partial F_1}{\partial y} & \frac{\partial F_1}{\partial z} \\ \frac{\partial F_2}{\partial x} & \frac{\partial F_2}{\partial y} & \frac{\partial F_2}{\partial z} \\ \frac{\partial F_3}{\partial x} & \frac{\partial F_3}{\partial y} & \frac{\partial F_3}{\partial z} \end{bmatrix}.$$

- **Exercise**: derive the matrix blocks in Jacobian of J.

- **Answer**:

$$J = \begin{bmatrix} H & -A^T & -I \\ A & 0 & 0 \\ Z & 0 & X \end{bmatrix}.$$

where

- I be the identity matrix and
- H represent the Hessian matrix of f (second derivatives of f)

$$H := \frac{\partial^2 f}{\partial x^2}.$$

- **How can we calculate Hessian matrix H?**

Interior-point method for constrained minimization

Hessian matrix of the Gibbs energy function

Hessian matrix of the Gibbs energy function

- The **Hessian matrix of the Gibbs energy** is

$$H := \frac{\partial^2 G}{\partial n^2} \qquad \text{or in element-wise notation} \qquad H := \left\{ H_{ij} \right\}_{ij} := \left\{ \frac{\partial^2 G}{\partial n_i \partial n_j} \right\}_{ij}$$

- Recall that

$$\frac{\partial G}{\partial n_i} = \mu_i \qquad \text{and} \qquad \mu_i = \mu_i^o + RT \ln a_i.$$

- Thus,

$$H_{ij} = \frac{\partial \mu_i}{\partial n_j} = \frac{\partial}{\partial n_j} \left(\mu_i^o + RT \ln a_i \right) = RT \frac{\partial \ln a_i}{\partial n_j}.$$

- Expressions for $\partial \ln a_i / \partial n_j$ can be derived for all previous activity models.

Partial molar derivatives of activities

- Instead of exact calculation of partial molar derivatives, we **derive them for ideal models**.
- We use activity models to **correct for the non-ideal behavior** of the aqueous and gaseous solutions.
- We use the partial molar derivatives of ideal activity models as it is **simpler to derive**.
- So, in

$$\mu_i - \sum_{j=1}^{E} A_{ji} y_j - z_i = 0 \qquad (i = 1, \dots, N),$$

$$\sum_{i=1}^{N} A_{ji} n_i - b_j = 0 \qquad (j = 1, \dots, E),$$

$$n_i z_i = 0 \qquad (i = 1, \dots, N),$$

$$n_i, z_i \geq 0 \qquad (i = 1, \dots, N),$$

  μ_i **will be calculated exactly**. But the **Jacobian will be approximated**.
- Ideally, an **automatic differentiation** scheme should be used to compute these derivatives (beyond the scope of this course).

18

Partial molar derivatives of activities of aqueous solutes

- Recall that the **activity of an aqueous solute** is

$$a_i = \gamma_i m_i, \text{ where } m_i = 55.508 \frac{n_i}{n_w}.$$

- Its ideal activity model is $a_i = m_i$, with $\gamma_i = 1$.
- Thus,

$$\frac{\partial \ln a_i}{\partial n_j} \approx \frac{\partial \ln m_i}{\partial n_j} = \text{Exercise}$$

$$= \frac{\partial 55.508}{\partial n_j} + \frac{\partial \ln n_i}{\partial n_j} - \frac{\partial \ln n_w}{\partial n_j} = \begin{cases} \dfrac{1}{n_i} & j = i, \\ -\dfrac{1}{n_w} & j = w, \\ 0 & \text{otherwise.} \end{cases}$$

19

Partial molar derivatives of activities of aqueous solvent, $H_2O(aq)$

- Recall that the **ideal activity of water**, $H_2O(aq)$, is

$$a_w = -\frac{1 - x_w}{x_w}, \qquad \text{where} \qquad x_w = \frac{n_w}{\sum n_i}.$$

- Thus,

$$\frac{\partial \ln a_w}{\partial n_j} = \text{Exercise}$$

$$= \frac{\partial \ln(x_w - 1)}{\partial n_j} - \frac{\partial \ln x_w}{\partial n_j} = \begin{cases} -\dfrac{1}{n_w} & j = w, \\ -\dfrac{1}{n_w}\dfrac{x_w}{x_w - 1} & j \neq w. \end{cases}$$

Partial molar derivatives of activities of $CO_2(g)$

- Recall that the **activity of $CO_2(g)$** was calculated using

$$a_{CO_2(g)} = \varphi_{CO_2(g)} \frac{P}{P^o},$$

where the fugacity coefficient $\varphi_{CO_2(g)}$ was only a function of T and P, i.e.,

$$\varphi_{CO_2(g)} = \varphi_{CO_2(g)}(T, P).$$

- Thus, the value of the following derivative

$$\frac{\partial \ln a_{CO_2(g)}}{\partial n_j} = \text{Exercise}$$

$$= 0.$$

Partial molar derivatives of activities – pure minerals

- Recall that the **activities of pure minerals** (e.g., mineral phases with only one mineral end-member),

$$a_i = 1.$$

- Thus, for mineral species,

$$\frac{\partial \ln a_i}{\partial n_j} = 0.$$

- Have a look in the Python code of the project to see the implementation of these formulas for when calculating the Hessian matrix H with entry H_{ij} given by:

$$H_{ij} = RT\frac{\partial \ln a_i}{\partial n_j} \quad \text{used in} \quad J = \begin{bmatrix} H & -A^T & -I \\ A & 0 & 0 \\ Z & 0 & X \end{bmatrix}.$$

```python
# Define the function that calculates:
#  - G, the Gibbs energy of the system;
#  - u, the gradient of the Gibbs energy, or chemical potentials of the
    species
#  - H, the Hessian of the Gibbs energy, or partial molar derivatives of
    activities
# These quantities are normalized by RT for numerical reasons.
def gibbs_energy(T, P, n):
```

Calculation of the Hessian matrix of the Gibbs energy function in Python ii

```python
7    RT = R*T
8    u = chemical_potentials(T, P, n)/RT
9    H = chemical_potentials_ddn(T, P, n)/RT
10   G = n.dot(u)/RT
11   return G, u, H
12
13 # Define the function that calculates the partial molar derivatives of the
14 # chemical potentials of the species.
15 def chemical_potentials_ddn(T, P, n):
16   return R*T*ln_activities_ddn(T, P, n)
17
18 # Define the function that calculates the partial molar derivatives of the
19 # ln activities of all species
20 def ln_activities_ddn(T, P, n):
21   # Create an array with the entries in n corresponding to aqueous species
22   n_aqueous = n[slice_aqueous]
23
```

Calculation of the Hessian matrix of the Gibbs energy function in Python iii

```
24    # The matrix with partial molar derivatives of the activities
25    ln_a_ddn = zeros((num_species, num_species))
26
27    ln_a_ddn[slice_aqueous, slice_aqueous] = ln_activities_aqueous_species_ddn(T
        , P, n_aqueous)
28
29    return ln_a_ddn
30
31 # Define the function that calculates the activities of the aqueous species.
32 # Parameters:
33 #   - T is temperature in units of K
34 #   - P is pressure in units of Pa
35 #   - nphase is an array with the mole amounts of the aqueous species
36 # Return:
37 #   - an array with the ln activities of aqueous species
38 def ln_activities_aqueous_species_ddn(T, P, nphase):
39    # The molar amount of H20(l)
```

Calculation of the Hessian matrix of the Gibbs energy function in Python iv

```python
40   nH2O = nphase[iH2O]
41
42   # The mole fraction of H2O(l)
43   xH2O = nH2O/sum(nphase)
44
45   # Calculate the partial molar derivatives of the solute activities
46   ddn = diag(1.0/nphase)     #  1/ni, j = i
47   ddn[:, iH2O] = -1.0/nH2O   # -1/nw, j = w
48
49   # Calculate the partial molar derivatives of the solvent water activity
50   ddn[iH2O, :] = -1.0/nH2O * xH2O/(xH2O - 1.0)  # -1/nw, j = w
51   ddn[iH2O, iH2O] = -1.0/nH2O                       #   1/nw*xw/(xw -1), j<>w
52
53   return ddn
```

Listing 1: Calculating Jacobian using Python

Interior-point method for constrained minimization

Numerical solution by the interior-point Newton method

Numerical solution by the interior-point Newton method

Newton method algorithm at step 1:

- At the first Newton iteration, start with **initial guess**
  $x^0 = (x_1^0, \ldots, x_n^0)$, $y^0 = (y_1^0, \ldots, y_m^0)$ and $z^0 = (z_1^0, \ldots, z_n^0)$.

- Compute $x^1 = (x_1^1, \ldots, x_n^1)$, $y^1 = (y_1^1, \ldots, y_m^1)$ and $z^1 = (z_1^1, \ldots, z_n^1)$ using

$$x^1 = x^0 + \alpha_x^0 \Delta x^0$$
$$y^1 = y^0 + \Delta y^0$$
$$z^1 = z^0 + \alpha_z^0 \Delta z^0$$

  where α_x and α_z are introduced parameters to ensure $x_i^1 > 0$ and $z_i^1 > 0$.

- How do we calculate variations Δx^0, Δy^0, Δz^0?

Numerical solution by the interior-point Newton method   ii

- To calculate the Newton variations $\Delta x^0 = (\Delta x_1^0, \ldots, \Delta x_n^0)$, $\Delta y^0 = (\Delta y_1^0, \ldots, \Delta y_m^0)$ and $\Delta z^0 = (\Delta z_1^0, \ldots, \Delta z_n^0)$, we solve the matrix equation

$$\underbrace{\begin{bmatrix} H^0 & -A^T & -I \\ A & 0 & 0 \\ Z^0 & 0 & X^0 \end{bmatrix}}_{J^0} \begin{bmatrix} \Delta x^0 \\ \Delta y^0 \\ \Delta z^0 \end{bmatrix} = -\underbrace{\begin{bmatrix} g^0 - A^T y^0 - z^0 \\ Ax^0 - b \\ X^0 Z^0 e - \tau \end{bmatrix}}_{F^0}$$

where J^0, F^0, H^0, Z^0, X^0 represent to evaluation of respective matrices using the values of x, y, z at the 0th iteration, i.e.,
   - $J^0 := J(x^0, y^0, z^0)$,
   - $F^0 := F(x^0, y^0, z^0)$, etc.

Newton method algorithm at the step $k + 1$:

- The vector steps Δx^k, Δy^k, Δz^k at the iteration k is calculated by

$$
J^k \begin{bmatrix} \Delta x^k \\ \Delta y^k \\ \Delta z^k \end{bmatrix} = -F^k
$$

- New approximations x^{k+1}, y^{k+1}, z^{k+1} is obtained by

$$
\begin{aligned}
x^{k+1} &= x^k + \alpha_x^k \Delta x^k, \\
y^{k+1} &= y^k + \Delta y^k, \\
z^{k+1} &= z^k + \alpha_z^k \Delta z^k.
\end{aligned}
$$

- How do we calculate α_x^k and α_z^k reassuring positivity of x_i^{k+1} and z_i^{k+1}?

- Start with $\alpha_x^k \leftarrow 1$.
- Check if

$$x_i^k + \Delta x_i^k \leq 0 \quad \text{for} \quad i = 1, \ldots, n$$

  - if so, $\alpha_x^k \leftarrow (\delta - 1)\dfrac{x_i^k}{\Delta x_i^k}$, where $\delta \in (0,1)$, e.g., $\delta = 10^{-4}$;
  - otherwise, go to next component with index i.
- It results into either a **full Newton step** (for $\alpha_x^k = 1$) or a **partial Newton step** (for $\alpha_x^k < 1$).
- If for certain index i, parameter $\alpha_x^k < 1$, then

$$x_i^{k+1} = \delta x_i^k,$$

and for all other indices

$$x_j^{k+1} = x_j^k + \alpha_x^k \Delta x_j^k, \quad j = 1, \ldots, n, j \neq i.$$

30

Summary of the Newton method algorithm:

- With initial guess x^0, y^0, and z^0, repeat

$$x^{k+1} = x^k + \alpha_x^k \Delta x^k$$
$$y^{k+1} = y^k + \Delta y^k \qquad \text{iteration } k$$
$$z^{k+1} = z^k + \alpha_z^k \Delta z^k$$

until

$$\|F^k\|_\infty = \|F(x^k, y^k, z^k)\|_\infty < \epsilon_{\text{tol}},$$

where

- $\epsilon_{\text{tol}} = 10^{-6}$ is **selected tolerance** and
- the **norm of a vector** v is defined by $\|v\|_\infty = \max(|v_i|)$.

References

Peng, D.-Y. & Robinson, D. B. (1976). A New Two-Constant Equation of State. *Industrial & Engineering Chemistry Fundamentals*, 15(1), 59–64.

Redlich, O. & Kwong, J. N. S. (1949). On the Thermodynamics of Solutions. V. An Equation of State. Fugacities of Gaseous Solutions. *Chemical Reviews*, 44(1), 233–244.

Smith, J. M., Ness, H. C. V., & Abbott, M. M. (2005). *Introduction to chemical engineering thermodynamics*. McGraw-Hill.

Soave, G. (1972). Equilibrium constants from a modified Redlich-Kwong equation of state. *Chemical Engineering Science*, 27(6), 1197–1203.

van der Waals, J. D. (1873). *On the Continuity of the Gaseous and Liquid States*. PhD thesis, Universiteit Leiden.