Chemical Composition at Equilibrium

Bruno Abreu Calfa

August 19, 2013

1 System of Nonlinear Equations Example

1.1 Problem Definition

Adapted from: Smith, J. M., van Ness, H. C., Abbott, M. M. (2005) *Introduction to Chemical Engineering Thermodynamics*. Seventh edition. The McGraw-Hill Companies, Inc.

Given two reactions in equilibrium

$$C + CO_2 \rightleftharpoons 2\,CO \qquad \text{(I)}$$
$$H_2O + C \rightleftharpoons H_2 + CO \qquad \text{(II)}$$

and the following table with values of each equilibrium constant for different values of temperature

Table 1: Equilibrium constant values for different temperatures.

T [K]	K_I	K_{II}
$1,000$	1.758	2.561
$1,100$	11.405	11.219
$1,200$	53.155	38.609
$1,300$	194.430	110.064
$1,400$	584.85	268.76
$1,500$	$1,514.12$	583.58

determine the equilibrium composition of the system for each temperature in Table 1. The feed stream contains the following composition: 1 mol of H_2, 0.5 mol of CO_2, and 1.88 mol of N_2.

1.2 Mathematical Modeling of the System

To model the chemical composition at equilibrium, we use the *extent of reaction* for each reaction. Let the indices i and j denote the chemical species and reactions, respectively. The general formula is given by:

$$n_i = n_{i0} + \sum_j \nu_{ij}\xi_j$$

Table 2 below shows the values of the stoichiometric coefficient for reactions (I) and (II).

Table 2: Stoichiometric coefficients, ν_{ij}, for the gaseous species in the reactions.

i	j	
	I	II
CO_2	-1	0
CO	2	1
H_2O	0	-1
H_2	0	1

The total number of moles in the system can be obtained by summing over all the species:

$$N = \sum_i n_i = \sum_i \left(n_{i0} + \sum_j \nu_{ij}\xi_j \right) = \sum_i n_{i0} + \sum_i \sum_j \nu_{ij}\xi_j$$

Therefore, expressions for the molar fractions can be written as follows:

$$y_i = \frac{n_i}{N} = \frac{n_{i0} + \sum_j \nu_{ij}\xi_j}{\sum_i n_{i0} + \sum_i \sum_j \nu_{ij}\xi_j} \tag{1}$$

Equation (1) can be used in the definition of the equilibrium constants for each reaction:

$$K_I(T) = \frac{y_{\mathrm{CO}}^2}{y_{\mathrm{CO_2}}} \tag{2}$$

$$K_{II}(T) = \frac{y_{\mathrm{H_2}} y_{\mathrm{CO}}}{y_{\mathrm{H_2O}}} \tag{3}$$

where it is explicit that the equilibrium constants are functions of temperature, T. The molar fractions for each component as functions of the extents of reaction are given by:

$$y_{\mathrm{H_2}} = \frac{\xi_{II}}{3.38 + \xi_I + \xi_{II}} \tag{4}$$

$$y_{\mathrm{CO}} = \frac{2\xi_I + \xi_{II}}{3.38 + \xi_I + \xi_{II}} \tag{5}$$

$$y_{\mathrm{H_2O}} = \frac{1 - \xi_{II}}{3.38 + \xi_I + \xi_{II}} \tag{6}$$

$$y_{\mathrm{CO_2}} = \frac{0.5 - \xi_I}{3.38 + \xi_I + \xi_{II}} \tag{7}$$

$$y_{\mathrm{N_2}} = \frac{1.88}{3.38 + \xi_I + \xi_{II}} \tag{8}$$

Because the values of y_i must lie between zero and one, we must have: $-0.5 \leq \xi_I \leq 0.5$ and $0 \leq \xi_{II} \leq 1$. Substituting equations (4) – (8) into equations (2) and (3) yields:

$$K_I(T) = \frac{(2\xi_I + \xi_{II})^2}{(0.5 - \xi_I)(3.38 + \xi_I + \xi_{II})} \tag{9}$$

$$K_{II}(T) = \frac{\xi_{II}(2\xi_I + \xi_{II})}{(1 - \xi_{II})(3.38 + \xi_I + \xi_{II})} \tag{10}$$

Equations (9) and (10) form the nonlinear system of algebraic equations to be simultaneously solved for the variables ξ_I and ξ_{II}.

1.3 Root-Finding

Solving an equation is also known as "root-finding". In other words, we want to find the values for the vector of variables x such that:

$$f(x) = 0$$

where $f(\cdot)$ is a vector of functions. The values of x that solve the above equation are called the roots of $f(\cdot)$.

In our example, the vector of variables has two components, that is $x = [\xi_I,\ \xi_{II}]$, and the vector of functions also has two components:

$$f_1(\xi_I, \xi_{II}) := K_I(T) - \frac{(2\xi_I + \xi_{II})^2}{(0.5 - \xi_I)(3.38 + \xi_I + \xi_{II})} = 0$$
$$f_2(\xi_I, \xi_{II}) := K_{II}(T) - \frac{\xi_{II}(2\xi_I + \xi_{II})}{(1 - \xi_{II})(3.38 + \xi_I + \xi_{II})} = 0$$

Notice that we put all the terms to the LHS in order to obtain the form $f(x) = 0$.

There are systematic methods available to solve systems of equations such as the one above (*e.g.*, Newton-Raphson) and more details will be covered in a later course.

1.4 Solving $f(x) = 0$ in MATLAB

Solving a general system of multiple equations for multiple variables in MATLAB requires the function `fsolve`. The basic usage of `fsolve` is given below:

```
[x,fval] = fsolve(fun,x0)
```

where `x` contains the roots of function `fun`, `fval` is the value of the function `fun` evaluated at `x` (should be as close to 0 as possible), and `x0` is an initial guess for `fsolve`. All methods that solve nonlinear problems require an initial guess that should be as close as possible to a solution.

The MATLAB script (`equilib_script.m`) to solve this problem accompanies this handout. Notice that the system equations was slightly modified. For numerical stability, it is recommended to avoid fractions involving variables. Therefore, the equations were rearranged as follows:

$$f_1(\xi_I, \xi_{II}) := K_I(T)(0.5 - \xi_I)(3.38 + \xi_I + \xi_{II}) - (2\xi_I + \xi_{II})^2 = 0$$
$$f_2(\xi_I, \xi_{II}) := K_{II}(T)(1 - \xi_{II})(3.38 + \xi_I + \xi_{II}) - \xi_{II}(2\xi_I + \xi_{II}) = 0$$

Table 3 shows the values of each extent of reactions and molar fractions for a given temperature. Figure 1 shows the system's composition at equilibrium for each temperature.

Table 3: Equilibrium constant values for different temperatures.

T [K]	ξ_I	ξ_{II}	y_{H_2}	y_{CO}	y_{H_2O}	y_{CO_2}	y_{N_2}
1,000	0.2538	0.8923	0.1971	0.3093	0.0238	0.0544	0.4154
1,100	0.4378	0.9668	0.2021	0.3851	0.0069	0.013	0.3929
1,200	0.4851	0.9897	0.2039	0.4037	0.0021	0.0031	0.3872
1,300	0.4958	0.9963	0.2045	0.408	0.0008	0.0009	0.3859
1,400	0.4986	0.9985	0.2047	0.4092	0.0003	0.0003	0.3855
1,500	0.4995	0.9993	0.2048	0.4096	0.0001	0.0001	0.3853

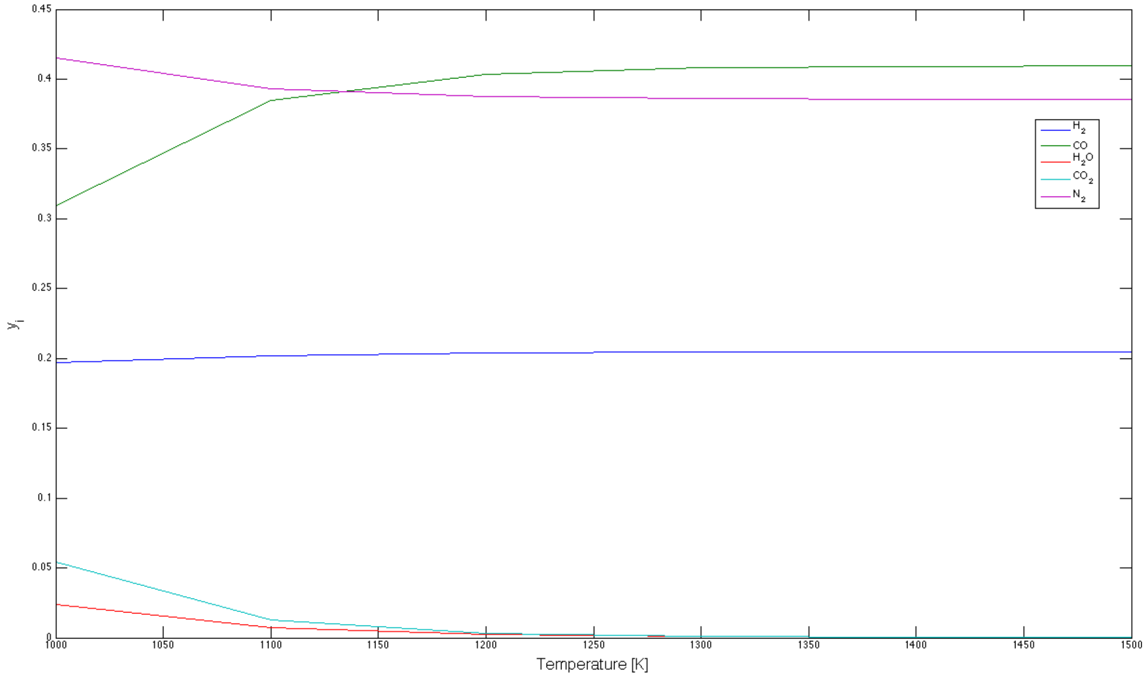

Figure 1: Equilibrium composition of the system at different temperatures.

Observe that at higher temperatures the values of extents of reaction become closer to their upper limiting values of 0.5 and 1.0, indicating that the reactions are proceeding to completion. Consequently, the composition of the reactants approaches zero.

Note that the method used to solve the system of nonlinear equations does not ensure that the solution obtained is the provably best one nor that it is unique. In fact, different initial guesses may lead to different numerical values for ξ_I and ξ_{II} and still make f_1 and f_2 close to zero. Moreover, the bounds on ξ_I and ξ_{II} were not explicitly accounted for. The function `fsolve` does not allow one to bound the variables. Alternatively, this problem could be solved as a nonlinear least squares problem with MATLAB's function `lsqnonlin`, which allows the user to bound the variables below and above.