

A First Course on Kinetics and Reaction Engineering

Example 3.2

Problem Purpose

This problem illustrates the calculation of the equilibrium composition of a reacting mixture. In most cases, one would first need to calculate the equilibrium constant at the temperature of interest, but the necessary equilibrium constant for this example was calculated in Example 3.1 so that this example can focus on the calculation of the equilibrium composition.

Problem Statement

Suppose a gas mixture containing 25% CO, 25% CO₂ and 50% H₂ reacts according to reaction (1) with no other reactions taking place. The reaction takes place at a constant pressure of 5 MPa and at a constant temperature. Assuming that the gases can be treated as ideal, calculate the equilibrium mole fractions of all the species in the system if the reaction takes place at 500 K, where the equilibrium constant has a value of 6.153×10^{-3} .



Problem Solution

Equation (2) relates the equilibrium constant to the composition; upon expanding the continuous product, equation (3) results.

$$K_1(T) = \prod_{i=1}^{N_{\text{species}}} a_i^{v_{i,1}} \quad (2)$$

$$K_1(T) = a_{\text{CO}}^{v_{\text{CO},1}} a_{\text{CO}_2}^{v_{\text{CO}_2,1}} a_{\text{H}_2}^{v_{\text{H}_2,1}} a_{\text{CH}_3\text{OH}}^{v_{\text{CH}_3\text{OH},1}} \quad (3)$$

The problem states that the reagents may be treated as ideal gases, in which case the thermodynamic activities in equations (2) and (3) are related to the mole fractions by equation (4), where i may equal CO, CO₂, H₂ or CH₃OH.

$$a_i = \frac{y_i P}{1 \text{ atm}} \quad (4)$$

Substitution of equation (4) for each of the species into equation (3), along with the stoichiometric coefficients (remembering the sign convention), leads to equation (5), which contains three unknown mole fractions.

$$K_1(T) = a_{\text{CO}}^{-1} a_{\text{CO}_2}^0 a_{\text{H}_2}^{-2} a_{\text{CH}_3\text{OH}}^1 = \frac{a_{\text{CH}_3\text{OH}}}{a_{\text{CO}} a_{\text{H}_2}^2}$$

$$K_1(T) = \frac{y_{CH_3OH}}{y_{CO}y_{H_2}^2} \left(\frac{P}{1 \text{ atm}} \right)^{-2} \quad (5)$$

You can't solve one equation with three unknowns, so additional relationships need to be found. A mole table can be used to express each of the mole fractions in terms of the extent of reaction. In the present case only intensive quantities are provided (nothing fixes the size of the system) so we are free to choose a basis. A convenient basis here is one mole of the starting mixture, in which case the starting number of moles of each species is numerically equal to its mole fraction.

$$n_i^0 = y_i^0 n_{total}^0 = (1 \text{ mol}) y_i^0$$

Using this basis and substituting the stoichiometric coefficients, the mole table is as follows:

Species	Initial Number of Moles	Number of Moles after Reaction
i	n_i^0	$n_i = n_i^0 + \sum_{j=1}^{N_{ind}} \nu_{i,j} \xi_j$
CO	0.25	$n_{CO} = 0.25 - \xi_1$
CO ₂	0.25	$n_{CO_2} = 0.25$
H ₂	0.5	$n_{H_2} = 0.5 - 2\xi_1$
CH ₃ OH	0.0	$n_{CH_3OH} = \xi_1$
Total Moles	1.0	$n_{total} = 1 - 2\xi_1$

The mole fractions needed for equation (5) are then found using the definition of a mole fraction with substitution from the mole table.

$$y_{CO} = \frac{n_{CO}}{n_{total}} = \frac{0.25 - \xi_1}{1 - 2\xi_1} \quad (6)$$

$$y_{H_2} = \frac{n_{H_2}}{n_{total}} = \frac{0.5 - 2\xi_1}{1 - 2\xi_1} \quad (7)$$

$$y_{CH_3OH} = \frac{n_{CH_3OH}}{n_{total}} = \frac{\xi_1}{1 - 2\xi_1} \quad (8)$$

The equilibrium mole fractions are found by substituting equations (6) through (8) into equation (5) along with the corresponding value of the equilibrium constant and the pressure (in units of atm, 5 MPa = 49.3 atm). The result is given in equation (9).

$$K_1(T) = \frac{y_{CH_3OH}}{y_{CO}y_{H_2}^2} \left(\frac{P}{1 \text{ atm}} \right)^{-2}$$

$$6.153 \times 10^{-3} - \frac{\left(\frac{\xi_1}{1-2\xi_1} \right)}{\left(\frac{0.25-\xi_1}{1-2\xi_1} \right) \left(\frac{0.5-2\xi_1}{1-2\xi_1} \right)^2} \left(\frac{49.3 \text{ atm}}{1 \text{ atm}} \right)^{-2} = 0 \quad (9)$$

Upon expansion and simplification, equation (9) is a cubic polynomial equation. It does not contain integrals or derivatives. As such, one could attempt to solve it manually, using symbolic algebra software or using numerical methods software. In the latter case, no matter which software one chooses, one will generally need to do the following:

- provide the the equation to be solved in the form $0 = f(z)$ where z is the unknown to be calculated.
- provide a guess for the value of z .

Here, the equation is already in the proper form, so all that is needed is a guess for the value of the extent of reaction. Looking at the mole table, the extent of reaction is expected to lie in the range from 0 to 0.25 mol, because those values correspond to zero and 100% conversion of the CO. Thus, it is probably wise to choose a guess within this range.

Irrespective of whether equation (9) is solved manually or using software, a value of $\xi_1 = 0.143$ mol is obtained for the equilibrium extent of reaction. Substitution of the equilibrium extent of reaction into equations (6) through (8) gives the equilibrium mole fractions of CO, H₂ and CH₃OH. The equilibrium mole fraction of CO₂ is found similarly using the the definition of a mole fraction with substitution of the equations in the mole table and the value of the equilibrium extent of reaction.

$$y_{CO} = \frac{0.25 - 0.143}{1 - 2(0.143)} = 0.15$$

$$y_{H_2} = \frac{0.5 - 2(0.143)}{1 - 2(0.143)} = 0.299$$

$$y_{CH_3OH} = \frac{0.143}{1 - 2(0.143)} = 0.201$$

$$y_{CO_2} = \frac{n_{CO_2}}{n_{total}} = \frac{0.25}{1 - 2(0.143)} = 0.350$$

Calculation Details Using MATLAB

Supplemental Unit S2 describes how to solve sets of non-differential equations numerically using MATLAB, and it provides a template file named `SolvNonAlg.m` for doing so. Before it can be used to solve a problem, that template file must be modified in four places, each indicated by a comment that begins “% EDIT HERE”. In addition to those required modifications, I recommend that you work with a copy of the file that has been given a more meaningful name. In this case, I made a copy of the file and saved it as `Example_3_2.m`; a copy of that file accompanies this solution. If you do rename the file, a fifth modification is required; you must change the function name given on the first executable line of the file. The next required modification is to enter the values of all universal and problem specific constants at the point indicated. At the same time these are entered, they should be converted to a consistent set of units. In this problem the known constants are the stoichiometric coefficients, the initial mole fractions, the pressure, the equilibrium constant at 500 K and the initial total moles. These can be used to calculate the initial moles of each species, which are also constants. Listing 1 shows `Example_3_2.m` after these modifications have been made.

```
function Example_3_2
% Known quantities and constants (in mol atm)
% stoichiometric coefficients
nu_co = -1;
nu_h2 = -2;
nu_meoh = 1;
% initial mole fractions
y0_co = 0.25;
y0_h2 = 0.5;
y0_meoh = 0.0;
y0_co2 = 0.25;
P = 5.0/.10133; % converted to atm
K_500 = 0.006153 ;
n0_tot = 1.0; % basis

% Calculate the initial moles
n0_co = y0_co*n0_tot;
n0_h2 = y0_h2*n0_tot;
n0_meoh = y0_meoh*n0_tot;
n0_co2 = y0_co2*n0_tot;
```

Listing 1. Portion of the modified template file `SolvNonDif.m` showing the re-naming of the function and the entry of the problem-specific constants in consistent units.

The next required modification is to provide code that evaluates each of the functions, f_i , when the equations being solved are written in the form $0 = f_i(z_1, \dots, z_n)$. At this point, you need to decide which of the unknowns in the equations being solved is going to be represented as z_1 , which as z_2 , and so on. I recommend actually defining variables with names similar to those used in your solution and setting them equal to the corresponding z_i . Doing so will make your code a little less efficient, but it may also reduce the chances of coding errors where you used $z(2)$ when it should have been $z(3)$. Such errors are much easier to spot when the variables in the equations have meaningful names. Perhaps this is overkill in this particular example, where there is only one equation and one unknown, but nonetheless, Listing 2

shows how this was done in Example_3_2.m and how code to evaluate the function was entered as the only element in the vector f.

```
% Function that evaluates the equations
function f = evalEqns(z)
    extent = z(1);
    n_co = n0_co - extent;
    n_h2 = n0_h2 - 2*extent;
    n_meoh = n0_meoh + extent;
    n_co2 = n0_co2;
    n_tot = n_co + n_h2 + n_meoh + n_co2;
    y_co = n_co/n_tot;
    y_h2 = n_h2/n_tot;
    y_meoh = n_meoh/n_tot;
    f = [
        K_500*P^2*y_co*y_h2^2 - y_meoh;
    ];
end % of internal function evalEqns
```

Listing 2. Portion of the modified template file SolvNonDif.m showing the entry of the code to evaluate the functions being solved.

One note in passing, in order to generate equation (9), it was necessary to start with equation (3) and then substitute expressions for the activities in terms of mole fractions, then substitute expressions for the mole fractions in terms of the moles and then expressions for the moles in terms of the extent. Each of these substitutions affords opportunities for making algebra mistakes. When solving the problem numerically, this isn't necessary. At the point where you provide code to evaluate the function $f(z)$ in Listing 2, a value of the extent is known, so the moles can be calculated, and then the mole fractions can be calculated, and then the function can be evaluated. That is, since the mole fractions can be calculated at this point, the form of the equilibrium expression given in equation (5) can be used instead of equation (9). Note that equation (5) was re-written so there were no denominators because doing so often helps with the convergence of the numerical methods.

The third required modification is where guesses for the unknowns are provided. The guesses are entered in the array named z_guess. Again, there is only one unknown, so only one guess needs to be entered as shown in Listing 3.

```
% guesses for the solution
z_guess = [
    0.01
];
```

Listing 3. Portion of the modified template file SolvNonDif.m where guesses are provided for the unknowns.

The final modification only applies if you need to use the results from solving the set of equations to calculate other quantities that the problem asked you to find. Listing 4 shows how this was done in the present problem where the equilibrium mole fractions needed to be calculated. Notice that semicolons are not used at the end of these lines so that they will be shown in the MATLAB command window when

Example_3_2.m is executed. (An alternative approach is to eliminate this part of Example_3_2.m and calculate these quantities at the MATLAB command prompt after Example_3_2.m has been executed since Example_3_2.m returns the values of the unknowns when it is finished. I prefer including them in Example_3_2.m because doing so creates a permanent documentation of how the calculations were performed.)

```
% Calculate the final mole fractions
extent = z(1);
n_co = n0_co - extent;
n_h2 = n0_h2 - 2*extent;
n_meoh = n0_meoh + extent;
n_co2 = n0_co2;
n_tot = n_co + n_h2 + n_meoh + n_co2;
y_co = n_co/n_tot
y_h2 = n_h2/n_tot
y_meoh = n_meoh/n_tot
y_co2 = n_co2/n_tot
```

Listing 4. Portion of the modified template file SolvNonDif.m where the results of solving the set of non-differential equations are used to calculate additional quantities.

At this point, Example_3_2.m can be executed by typing Example_3_2 at the MATLAB command prompt. The numerical methods used in the template file may generate pages and pages of messages; see the MATLAB documentation if you wish to eliminate these. However, it is critically important to check that MATLAB reports successful solution of the equations because otherwise it might return a solution that is not fully converged. The output generated upon execution of Example_3_2.m is shown in Listing 5 where it can be seen that a converged solution was obtained.

```
>> Example_3_2

Equation solved.

fsolve completed because the vector of function values is near zero
as measured by the default value of the function tolerance, and
the problem appears regular as measured by the gradient.

<stopping criteria details>

y_co =
    0.1496

y_h2 =
    0.2993

y_meoh =
    0.2007

y_co2 =
    0.3504
```

Listing 5. Output from the modified template file SolvNonDif.m used to solve this problem.