

A First Course on Kinetics and Reaction Engineering

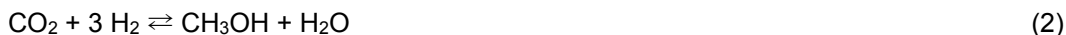
Example 1.4

Problem Purpose

This example illustrates the process of identifying reactions that are linear combinations of other reactions in a set and eliminating them until a mathematically independent set remains. It also illustrates how the initial composition and selected final composition variables can be used to calculate the remaining final composition variables.

Problem Statement

The charge (starting composition) to a reactor consists of 66 mol H₂ and 34 mol CO. Reactions (1) through (6) take place within the reactor, after which 40% of the CO has been converted. The final ratio of H₂ to CO is 1.5, and the methanol selectivity, defined as the ratio of the moles of methanol to the moles of CO₂, is 8.3. Calculate the number of moles of each species at the end of the process.



Problem Analysis

This problem clearly involves chemical reactions. While a reactor is mentioned in the problem statement, no details are offered as to its type, size or other characteristics. Examining the problem statement more closely, we see that it provides information about the initial composition of the system along with values for selected reaction progress variables. It never mentions reaction kinetics or reaction thermodynamics (equilibrium). This suggests that this is a stoichiometry problem.

Problem Solution

The following quantities are given in the problem statement:

$$n_{\text{CO}}^0 = 34 \text{ mol}$$

$$n_{\text{H}_2}^0 = 66 \text{ mol}$$

$$f_{\text{CO}} = 0.4$$

$$R_{\text{H}_2/\text{CO}} = 1.5$$

$$S_{CH_3OH/CO_2} = 8.3$$

A very common approach to stoichiometry problems is to use the information given in the problem statement to calculate the extent(s) of reaction, and then use the resulting extent(s) of reaction to calculate the quantities asked for in the problem. As such, mole table is often useful, but before one can be constructed for this problem, it will be necessary to identify a complete mathematically independent sub-set of the reactions taking place. As described in Supplemental Unit S1, a set of linear equations or chemical reactions can be checked to determine whether they are mathematically independent manually or using software that can determine the rank of a matrix. In either case you will need to do the following:

- create a reaction/equation matrix containing the coefficients in the reactions/equations where each column corresponds to one species/independent variable and each row corresponds to one reaction/equation. In the case of equations, the last column corresponds to the intercept.
- process the matrix as follows:
 - step 1: determine its rank
 - if the rank equals the number of reactions/equations, the reactions/equations are mathematically independent and you are done
 - otherwise proceed to the step 2
 - step 2: remove one row from the matrix and determine the rank of the resulting matrix
 - if the rank remains the same, the reaction/equation corresponding to the removed equation was a linear combination of the remaining reactions/equations, so leave it out of the matrix
 - If the rank decreases by one, the removed row corresponds to a mathematically independent reaction, so put it back into the matrix
 - step 3: if the rank now equals the number of equations remaining in the matrix, the corresponding reactions/equations are mathematically independent and you are done, otherwise repeat step 2 removing a different row

Here, the reaction matrix will be represented as \underline{C} . The columns of \underline{C} , in order from left to right will correspond, to CO, H₂, CH₃OH, CO₂, H₂O and CH₄. Rows of \underline{C} , in order from top to bottom, will correspond to reactions (1) through (6). The reaction matrix, \underline{C} , can then be constructed so that the entry at row j and column i is the stoichiometric coefficient ($v_{i,j}$) of the corresponding species in the corresponding reaction. The result is given in equation (7).

$$C = \begin{bmatrix} -1 & -2 & 1 & 0 & 0 & 0 \\ 0 & -3 & 1 & -1 & 1 & 0 \\ -1 & 1 & 0 & 1 & -1 & 0 \\ -1 & -3 & 0 & 0 & 1 & 1 \\ 0 & -4 & 0 & -1 & 2 & 1 \\ 0 & 1 & 1 & 0 & -1 & -1 \end{bmatrix} \quad (7)$$

Using this matrix, one finds that reactions (1), (2) and (4) constitute a mathematically independent subset of the reactions that are occurring. That is, reactions (3), (5) and (6) represent linear combinations of reactions (1), (2), and (4). With the knowledge that reactions (1), (2) and (4) are a complete mathematically independent subset of the reactions, the moles of any species can be expressed in terms of the extents of those three reactions. This is shown for an arbitrary species i in equation (8) where the summation over j would include three terms corresponding to $j = 1, 2$ and 4 .

$$n_i = n_i^0 + \sum_{j=1}^{N_{ind}} \nu_{i,j} \xi_j \quad (8)$$

With this knowledge, a mole table can be constructed as shown in Table 1.

Table 1. Mole table.

Species	Initial Moles	Moles after Reaction
CO	n_{CO}^0	$n_{CO}^0 - \xi_1 - \xi_4$
H ₂	$n_{H_2}^0$	$n_{H_2}^0 - 2\xi_1 - 3\xi_2 - 3\xi_4$
CH ₃ OH	0	$\xi_1 + \xi_2$
CO ₂	0	$-\xi_2$
H ₂ O	0	$\xi_2 + \xi_4$
CH ₄	0	ξ_4
Total	$n_{CO}^0 + n_{H_2}^0$	$n_{CO}^0 + n_{H_2}^0 - 2\xi_1 - 2\xi_2 - 2\xi_4$

The problem statement provides three additional pieces of information, namely the CO conversion, the final H₂ to CO ratio and the methanol selectivity. These defining equations for these quantities can be written in terms of the final moles of the species as listed in the mole table leading to equations (9), (10) and (11).

$$f_{CO} = \frac{n_{CO}^0 - n_{CO}}{n_{CO}^0} = \frac{n_{CO}^0 - (n_{CO}^0 - \xi_1 - \xi_4)}{n_{CO}^0} = \frac{\xi_1 + \xi_4}{n_{CO}^0} \quad (9)$$

$$R_{H_2/CO} = \frac{n_{H_2}}{n_{CO}} = \frac{n_{H_2}^0 - 2\xi_1 - 3\xi_2 - 3\xi_4}{n_{CO}^0 - \xi_1 - \xi_4} \quad (10)$$

$$S_{CH_3OH/CO_2} = \frac{n_{CH_3OH}}{n_{CO_2}} = \frac{\xi_1 + \xi_2}{-\xi_2} \quad (11)$$

At this point we have three equations ((9) through (11)) in three unknowns (ξ_1 , ξ_2 and ξ_4). These equations do not contain integrals or derivatives. As such, one could attempt to solve them manually (which, in this case is fairly easy to do), 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 each of the n equations to be solved in the form $0 = f_i(z_1, \dots, z_n)$ where i ranges from 1 to n , and z_1 through z_n are the unknowns to be calculated.
- provide a guess for the value of each unknown, z_i .

As already noted, in this problem the three unknowns are ξ_1 , ξ_2 and ξ_4 , and it is trivial to re-write the equations (9) through (11) in the required form, equations (12) through (14). In doing so, I also multiplied through by the denominator, as that often improves convergence when solving non-differential equations numerically. The extents of the reactions are expected to have values between 0 and 66 mol, but any arbitrary value, say 0.1 mol, can be used as a guess. Of course, if one elects to solve these equations manually, none of these steps are needed.

$$0 = f_{CO} n_{CO}^0 - \xi_1 - \xi_4 \quad (12)$$

$$0 = R_{H_2/CO} (n_{CO}^0 - \xi_1 - \xi_4) - (n_{H_2}^0 - 2\xi_1 - 3\xi_2 - 3\xi_4) \quad (13)$$

$$0 = \xi_1 + \xi_2 + S_{CH_3OH/CO_2} \xi_2 \quad (14)$$

Irrespective of whether the equations are solved manually or using a computer, one finds $\xi_1 = 4.08$, $\xi_2 = -0.44$ and $\xi_4 = 9.52$. These values can then be used to calculate the final moles of each species using the formulae given in the third column of the mole table.

$$n_{CO} = n_{CO}^0 - \xi_1 - \xi_4 = 20.4 \text{ mol} \quad (15)$$

$$n_{H_2} = n_{H_2}^0 - 2\xi_1 - 3\xi_2 - 3\xi_4 = 30.6 \text{ mol} \quad (16)$$

$$n_{CH_3OH} = \xi_1 + \xi_2 = 3.64 \text{ mol} \quad (17)$$

$$n_{CO_2} = -\xi_2 = 0.44 \text{ mol} \quad (18)$$

$$n_{H_2O} = \xi_2 + \xi_4 = 9.08 \text{ mol} \quad (19)$$

$$n_{CH_4} = \xi_4 = 9.52 \text{ mol} \quad (20)$$

Calculation Details Using MATLAB

The solution of this problem required two tasks that could be performed numerically: finding a complete mathematically independent sub-set of the reactions taking place and solving equations (12) through (14) for the values of ξ_1 , ξ_2 and ξ_4 . A MATLAB script file named IndEqns.m is provided with Supplemental Unit S1. That script can be used to check an equation/reaction matrix to determine whether the corresponding equations/reactions are mathematically independent and to identify a complete mathematically independent subset, if they are not. To do so, the script file must be located in the current MATLAB working directory or in the MATLAB search path.

Before executing IndEqns, a reaction/equation matrix named C must be created from the set of reactions/equations to be tested. Then, upon execution, the script will go through the process of determining rank and eliminating reactions/equations that are not independent automatically. Upon completion, the output will consist of a variable named n_ind that is equal to the number of mathematically independent reactions/equations in the original set and a matrix named IndEqns that is the reaction/equation matrix for one mathematically independent subset containing n_ind of the original reactions/equations.

Thus, the reaction matrix, equation (7), was entered at the MATLAB command prompt with the variable name C. The script was then executed by typing "IndEqns," and the resulting output shown in Listing 1 shows that there are three mathematically independent reactions. Comparing the rows of the independent reaction matrix to the original reaction matrix shows that the independent reactions are reactions (1), (2) and (4).

```

>> C = [-1 -2 1 0 0 0
0 -3 1 -1 1 0
-1 1 0 1 -1 0
-1 -3 0 0 1 1
0 -4 0 -1 2 1
0 1 1 0 -1 -1]

C =

    -1    -2     1     0     0     0
     0    -3     1    -1     1     0
    -1     1     0     1    -1     0
    -1    -3     0     0     1     1
     0    -4     0    -1     2     1
     0     1     1     0    -1    -1

IndEqns

n_ind =
     3

IndEqs =

    -1    -2     1     0     0     0
     0    -3     1    -1     1     0
    -1    -3     0     0     1     1

```

Listing 1. Data entry and resulting output when the MATLAB template file IndEqns.m is used to determining the number of mathematically independent equations.

The second task that could be performed numerically is solving equations (12) through (14). 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_1_4.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 n_{CO}^0 , $n_{H_2}^0$, f_{CO} , $R_{H_2/CO}$ and S_{CH_3OH/CO_2} . Listing 2 shows Example_1_4.m after these modifications have been made.

```

function z = Example_1_4
% Known quantities and constants (in consistent units)
n0_CO = 34.;
n0_H2 = 66.;
fCO = 0.4;
R = 1.5;
S = 8.3;

```

Listing 2. 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. Listing 3 shows how this was done in Example_1_4.m and how code to evaluate the functions, f_i , was entered so that the vector named f contains the value of f_1 as its first element, the value of f_2 as its second element and so on.

```
% Function that evaluates the equations
function f = evalEqns(z)
    x1 = z(1);
    x2 = z(2);
    x4 = z(3);
    f = [
        fC0*n0_CO - x1 - x4;
        R*(n0_CO - x1 - x4) - (n0_H2 - 2*x1 - 3*x2 - 3*x4);
        x1 + x2 + 5*x4;
    ];
end % of internal function evalEqns
```

Listing 3. Portion of the modified template file SolvNonDif.m showing the entry of the code to evaluate the functions being solved.

The third required modification is where guesses for the unknowns are provided. The guesses are entered in the array named z_guess . They must be entered in order with z_1 first and z_n last. If you follow my recommendation in the previous modification, the statements where variables with more meaningful names were defined will serve as a key to remind you which variable is z_1 , which is z_2 , and so on. Listing 4 shows how the guesses are entered in this required modification.

```
% guesses for the solution
z_guess = [
    0.1
    0.1
    0.1
];
```

Listing 4. 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 5 shows how this was done in the present problem, where we need to calculate the final moles of each species from the extents of the reactions according to equations (12) through (20). 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_1_4.m is executed. (An alternative approach is to eliminate this part of Example_1_4.m and calculate these quantities at the MATLAB command prompt after Example_1_4.m has been executed since Example_1_4.m returns the values of the unknowns in the vector z when it is finished. I prefer including them in Example_1_4.m because doing so creates a permanent documentation of how the calculations were performed.)

```
% Calculate the final composition
x1 = z(1);
x2 = z(2);
x4 = z(3);

nC0 = n0_C0 - x1 - x4
nH2 = n0_H2 - 2*x1 - 3*x2 - 3*x4
nCH3OH = x1 + x2
nC02 = -x2
nH2O = x2 + x4
nCH4 = x4
```

Listing 5. 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_1_4.m can be executed by typing Example_1_4 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_1_4.m is shown in Listing 6. After a message regarding whether a solution was found, the final values of the functions, f_i , are printed; these should be very close to zero. That is followed by the values of the additional quantities calculated as a result of the final modification to the template file. Finally, the values of the unknowns are listed.


```
>> Example_1_4

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>

f =
    1.0e-07 *
    -0.0336
     0.3879
     0.0154

nC0 =
    20.4000

nH2 =
    30.6000

nCH3OH =
     3.6439

nC02 =
     0.4390

nH2O =
     9.0780

nCH4 =
     9.5171

ans =
     4.0829
    -0.4390
     9.5171
```

Listing 6. Output from the modified template file SolvNonDif.m used to solve this problem.