

Equilibrium Code

Introduction

Processes spontaneously occur in the direction that creates a positive S_{gen} . For chemically-reacting systems that are held to a specific T and P as the composition evolves, this implies both work and heat interactions with the surroundings. This means that, given enough time, the composition will evolve until the entropy of the system plus the entropy of the surroundings ($=Q/T$) reaches a maximum. From this point, any further spontaneous change is impossible since such a change would involve a negative S_{gen} . In the lecture and the accompanying derivation, we showed that this condition coincides with a minimization of the Gibbs energy for the system.

The equilibrium code searches out the composition that generates this minimum Gibbs energy via the procedure that is detailed in the derivation handout. All it requires is the input composition, and the desired equilibrium temperature and pressure. In spite of the apparent complexity of the equations in the handout, it is not difficult to implement the code in an Excel spreadsheet, and in fact we do this in the graduate combustion class. (As noted below, other options than fixed T and P are available in the code.)

Code Operation

Three versions of the code are available for download from the course website under the "Handouts" link. All are compressed files that will need to be unzipped. The Mac version will run on System 9 and earlier only, but those with System X can run it under the classic mode. The two Windows versions differ in two respects:

1. The older version has a much "clunkier" IO interface.
2. But, the older version allows you to enter fuel and air streams with different temperatures, while the new version constrains you to a single reactants temperature. We will find the two-temperature capability to be very useful in gas turbine calculations where the natural gas is usually at room temperature while the air is hotter since it comes off the compressor. The newer version of the code requires you to do a hand calculation in which you mix the fuel and air adiabatically (without reaction) to get an equivalent mixed temperature. This is then used as the single allowed reactants temperature for both streams.

The Mac version is the older code. My suggestion for Windows users is to use the older version of the code. The code runs very fast on any modern PC.

The new code is also loaded on the ME department computers under the name "UWEQL" as an icon on the menu bar at the bottom of the screen. The following description is for the old code. The new code follows the same general idea, but uses a screen menu to set the parameters.

Upon opening the code the following text appears:

```
0 *** CURRENT DATA STATUS ***
```

```
PROBLEM TYPE: HP, assigned enthalpy and pressure
```

```
Fuel temperature = 298.00 Kelvin
```

```
Air temperature = 750.00 Kelvin
```

```
Pressure = 5.0000E+00 atmospheres
```

```
A-F Ratio = 8.5000E-01 fuel/air equivalence ratio
```

SELECT AN OPTION:

- 0- Run with this data
- Change or review...
- 1- Fuels composition
- 2- Air composition
- 3- Fuel or reactants temperature
- 4- Air or reactants temperature
- 5- Product gas temperature
- 6- Temperature units
- 7- Pressure or pressure units
- 8- Air-fuel ratio or ratio units
- 9- Problem type

Option number? (0-9)

You enter an option number to go to each of these options.

Composition

You will want to change the fuel composition to reflect that needed by your current problem. You will also want to alter the oxidant composition if you are starting with something other than air. Composition information is provided by entering the relative number of kmoles of each component in each stream. *These numbers only set the internal composition of the fuel and air streams, and have nothing to do with the amount of fuel vs. the amount of air. The air/fuel stoichiometry is set later.* Thus, for CH₄/air combustion, the relative number of kmoles of CH₄ could be 1 or 10, both giving the same answer when the code is run. The relative number of kmoles of the O₂ and N₂ could be set to any of the following pairs:

O ₂	1.	.21	21
N ₂	3.76	.79	79

Next, you set the stoichiometry (Option 8). The code gives you the following options:

- Fuel/air equivalence ratio: This is $1/\phi_{AF}$ where ϕ_{AF} is the stoichiometry parameter we defined in class.
- Moles air/mole fuel: For stoichiometric methane/air combustion, this would be:
 $2(4.76)/1 = 9.52$
- Kg air/kg fuel.

You select which option you want to work with and enter the number that represents your desired stoichiometry. As an operational note, the code has a bug in the fuel/air equivalence ratio calculation when oxygen is present in the fuel (*e.g.*, alcohols). For these cases, it is much better to use the second or third options, which unambiguously set the stoichiometry.

Problem Type (Option 9)

The code solves four basic problem types, which are:

- HP: This problem holds the enthalpy and pressure constant as the composition evolves. It is used to calculate adiabatic equilibrium compositions where the final pressure is known. An example would be the final composition and temperature from an adiabatic Bunsen burner. The code will need your desired pressure (Option 7), and the initial temperature

of the reactants (from which it will find the h that is to be held constant, this is set in Options 3 and 4). The output will be the final T and the composition.

- TP: This problem holds the temperature and pressure constant as the composition evolves. This is used when you know the product temperature and pressure. An example would be the composition in an auto tailpipe, where you have measured the temperature. You will need to enter the output temperature (Option 5) and pressure (Option 7). The output will be the composition.
- UV: This problem holds the internal energy and volume constant as the composition evolves. This is used for situations where the system is insulated and has a fixed volume, such as a bomb calorimeter or the combustion process in an Otto cycle. You will need to enter the initial temperature (Options 3 and 4 and pressure (Option 7), from which the code will calculate both U and V . These will be held constant. The output will be the composition, and the final temperature and pressure.
- TV: This problem holds the volume constant, but forces the temperature to a fixed value as the composition evolves (via heat transfer with the surroundings). You would use this option to find the composition where you know the volume is being held constant, but you have a measurement for the final temperature. You will need to supply an initial pressure (Option 7, to find V) and a final temperature (Option 5), and the output will be the composition and the final pressure.

In each case, the code will also provide u , h and s (in /kg mixture units) for use in availability calculations.

Two following two additional "post-processing" options are available in the new code (they must be done manually in the older codes, and this will be illustrated in class). These allow you to take the products of one of the calculations above and expand them isentropically. These are:

- SP: The products are expanded at constant s to a new, lower P . As an example, you model a gas turbine combustor (adiabatic, constant $P=1$ MPa) using the HP option, with the known stoichiometry, pressure, and inlet temperature as inputs. The code gives you the conditions at the outlet of the combustor (T , h , s , composition). You then exercise the SP option, which takes this composition through the turbine to 100 kPa (you will be prompted for the desired outlet P) with s held constant. The code then gives you the new composition, T , and h that is characteristic of the turbine outlet. The Δh could then be used to calculate turbine work.
- SV: This is a similar option, but here the expansion volume ratio rather than the pressure ratio is known. The application would be an Otto cycle, where you would model the combustion as a UV problem, and then expand the products over a known volume ratio ($=1/\text{compression ratio}$), to get u , h , T , P , and composition at bottom dead center.

Output

The output comes in tabular form. The second column is output concentration in mole fraction. Be careful to take your output from the correct column, as this is a common error.

What if the inlet fuel and oxidizer are at different temperatures?

The following procedure is needed only if you use the new code. As mentioned above, in this code only one inlet reactants temperature is allowed.

In gas turbine combustors, the inlet fuel is often at room temperature, while the inlet air has been preheated as a result of passing through the compressor. To correctly do an adiabatic combustor calculation, you need to account for the h of each stream. Unfortunately, the new version of the code only allows the specification of a single inlet temperature. We get around the problem by pretending that we are premixing the fuel and oxidizer in an adiabatic, non-reacting mixer before the equilibrium reactor. This is accomplished by writing a first law for the hypothetical mixer:

$$(m_f h_f + m_a h_a)_{\text{inlet}} = (m_f h_f + m_a h_a)_{\text{outlet}}$$

With the constant specific heat assumption, this becomes:

$$m_f C_{pf}(T_2 - T_f) = m_a C_{pa}(T_a - T_2)$$

Here, T_f and T_a are respectively the inlet fuel and air temperatures, and T_2 is the outlet mixture temperature. The final issue is what temperatures to use for the C_p values. Because the mass flow of the air is typically large compared to the fuel flow, the final temperature is much closer to T_a than to T_f . For example, if the inlet fuel temperature is 300 K, and the inlet air is at 500 K, one could guess based on the mass flows that the final mix temperature would probably be near 470 K. Then, the appropriate C_p for the air would be at a temperature midway between the air endpoints (485 K), and the fuel C_p would be evaluated at the average of 470 and 300 K. The C_p 's are constant enough so it usually is not worth a second iteration based on the calculated T_2 .