Cantera Tutorial

Adiabatic flame temperature

11 septembre 2017

1 Introduction - Objectives

The aim of this tutorial is to walk non-experienced CANTERA users through several equilibrium computations of different methane-air combustion mechanisms, with the Python interface. Mainly, HP equilibrium calculations will be performed to obtain the adiabatic flame temperature.

The first script presented here will deal with a detailed methane-air scheme, at stoechiometry and under atmospheric conditions ; before a more thorough script is presented, that will allow the user to compute the adiabatic flame temperature for a wide range of operating conditions.
Two methane-air mechanisms will be used : the GRIMECH 3.0 to start with, and the global CERFACS methane-air 2S_CH4_BFER scheme to conclude on potential differences and to illustrate the fact that adaptation to other fuels is straightforward.

2 HP equilibrium calculations

2.1 Load the CANTERA module

A pre-compiled version of CANTERA will be used for this tutorial. This will enable you to run your CANTERA scripts very easily, no matter the type of environment you use. Open a terminal and simply type :

```
module load cantera/2.1.1
```

2.2 Get the script and mechanism

The script can be downloaded here : http://www.cerfacs.fr/cantera/docs/scripts/equilibrium/equil_simple.py. It should be placed in a CANTERA sub-directory of your choice.

You should also get the mechanism's file for this script, which in cantera is formatted in a **.cti** text file. As said previously, methane 'CH4' is the fuel species, and we will use the GRIMECH 3.0 to start with. Get it here : http://www.cerfacs.fr/cantera/docs/mechanisms/methane-air/DETAILED/CANTERA/gri30.cti, and place it in the same CANTERA sub-directory as your python script. Other detailed mechanisms than the GRIMech 3.0 could have been used, such as those presented on the website : http://www.cerfacs.fr/cantera/mechanisms/meth.php.

2.3 Launch your computation

Now, on the terminal where you loaded your module, go into your CANTERA sub-directory and type :

```
python equil_simple.py
```

By default, the code will run a simulation at stoechiometry, 300 K and 1 atm. As said previously, the process is held at constant pressure and constant enthalpy (HP). The execution of the script should return the state of your gas on your screen :

```
  gas:

       temperature         2224.34  K
          pressure          100000  Pa
           density        0.148294  kg/m^3
  mean mol. weight         27.4259  amu

                          1 kg           1 kmol
                       -----------     ------------
          enthalpy         -254589      -6.982e+06     J
    internal energy         -928924      -2.548e+07     J
           entropy         9880.64        2.71e+05     J/K
     Gibbs function   -2.22325e+07      -6.097e+08     J
  heat capacity c_p         1514.21       4.153e+04     J/K
  heat capacity c_v         1211.05       3.321e+04     J/K

                          X               Y          Chem. Pot. / RT
                      -------------    ------------    ------------
                 H2     0.00361596      0.000265783        -25.4583
                  H     0.000390984     1.43692e-05        -12.7292
                  O     0.000214479     0.000125121        -17.2744
                 O2     0.00457875        0.0053422        -34.5488
                 OH     0.00315576       0.00195695        -30.0035
                H2O       0.183298          0.120403        -42.7327
                HO2     4.91642e-07     5.91686e-07        -47.2779
               H2O2     4.50331e-08     5.58518e-08        -60.0071
                  C     2.19632e-17     9.61864e-18        -21.4963
                 CH     3.32419e-18     1.57798e-18        -34.2254
                CH2     9.72866e-18      4.9757e-18        -46.9546
             CH2(S)     5.89031e-19     3.01258e-19        -46.9546
                CH3     6.17384e-17     3.38449e-17        -59.6837
                CH4     3.01268e-17     1.76227e-17        -72.4129
                 CO      0.00901455      0.00920667        -38.7707
                CO2       0.0853291         0.136926         -56.045
                HCO     7.89919e-10     8.35785e-10        -51.4998
               CH2O     1.31029e-11     1.43453e-11         -64.229
              CH2OH     4.58276e-17      5.1857e-17        -76.9581
               CH3O     1.99881e-18     2.26179e-18        -76.9581
              CH3OH      3.4436e-18     4.02322e-18        -89.6873
                C2H      3.6251e-24     3.30841e-24        -55.7217
               C2H2     9.07502e-22     8.61574e-22        -68.4509
               C2H3     6.82051e-27       6.726e-27          -81.18
               C2H4      7.1398e-27     7.30326e-27        -93.9092
               C2H5     7.69301e-32     8.15186e-32        -106.638
               C2H6     5.59237e-33     6.13145e-33        -119.367
               HCCO     5.29904e-20     7.92741e-20        -72.9961
              CH2CO     7.02693e-20     1.07706e-19        -85.7253
              HCCOH     6.86468e-23     1.05219e-22        -85.7253
                  N     1.39843e-08     7.14195e-09        -13.8266
                 NH     2.32495e-09     1.27283e-09        -26.5557
```

NH2	1.07952e-09	6.30675e-10	-39.2849
NH3	2.61372e-09	1.62303e-09	-52.014
NNH	7.46413e-10	7.89836e-10	-40.3823
NO	0.00187457	0.00205093	-31.101
NO2	3.40406e-07	5.71014e-07	-48.3753
N2O	9.87967e-08	1.58548e-07	-44.9276
HNO	4.2012e-08	4.75085e-08	-43.8301
CN	4.65338e-14	4.41446e-14	-35.3229
HCN	1.53293e-11	1.51056e-11	-48.052
H2CN	5.40842e-18	5.52827e-18	-60.7812
HCNN	1.2896e-21	1.9294e-21	-61.8786
HCNO	1.00297e-16	1.57344e-16	-65.3264
HOCN	1.11817e-12	1.75416e-12	-65.3264
HNCO	2.27174e-10	3.56385e-10	-65.3264
NCO	1.52709e-11	2.33953e-11	-52.5973
N2	0.708527	0.723707	-27.6532
AR	0	0	
C3H7	3.43221e-47	5.39232e-47	-153.593
C3H8	2.47623e-48	3.98139e-48	-166.322
CH2CHO	3.3274e-25	5.2224e-25	-98.4544
CH3CHO	5.75244e-26	9.23993e-26	-111.184

2.4 Saving files

Via specific Python interface commands, it is also possible to generate **.csv** files with additional computed datas (mole fractions of the species present in the mixture, temperature, etc.). See http://www.cantera.org/docs/sphinx/html/search.html?q=&check_keywords=yes&area=default and look for the names of the quantities you want to extract. The **.csv** output files can be imported into Excel for example.

2.5 A second script

You can download a second script here : http://www.cerfacs.fr/cantera/docs/scripts/equilibrium/adiab_flame_T.py. Again, place it in a CANTERA sub-directory of your choice. On the terminal where you loaded the CANTERA module, go into this sub-directory and type :

```
python adiab_flame_T.py
```

You should see a collection of adiabatic flame temperature printed on your screen, for different equivalence ratios. We are still using the GRIMech 3.0 mechanism :

```
*******************************************************
    Computing Equilibirum for phi from 0.3 to 3.5, T = 300.0 K, P = 101325.0 Pa
    Equilibrate holding HP  constants
*******************************************************

At phi =      0.3000  Tad =   1066.2520
At phi =      0.4000  Tad =   1280.7660
At phi =      0.5000  Tad =   1480.1501
At phi =      0.6000  Tad =   1665.8199
At phi =      0.7000  Tad =   1838.2960
At phi =      0.8000  Tad =   1996.2860
At phi =      0.9000  Tad =   2133.4202
At phi =      1.0000  Tad =   2224.6418
At phi =      1.1000  Tad =   2209.5921
At phi =      1.2000  Tad =   2136.1440
At phi =      1.3000  Tad =   2057.1323
At phi =      1.4000  Tad =   1979.7040
At phi =      1.5000  Tad =   1904.7618
At phi =      1.6000  Tad =   1832.3245
At phi =      1.7000  Tad =   1762.2593
At phi =      1.8000  Tad =   1694.4196
At phi =      1.9000  Tad =   1628.6710
At phi =      2.0000  Tad =   1564.8930
At phi =      2.1000  Tad =   1502.9774
At phi =      2.2000  Tad =   1442.8255
At phi =      2.3000  Tad =   1384.3471
At phi =      2.4000  Tad =   1327.4601
At phi =      2.5000  Tad =   1272.0932
At phi =      2.6000  Tad =   1218.2015
At phi =      2.7000  Tad =   1165.8343
At phi =      2.8000  Tad =   1115.4233
At phi =      2.9000  Tad =   1068.7916
At phi =      3.0000  Tad =   1030.4203
At phi =      3.1000  Tad =   1003.4358
At phi =      3.2000  Tad =    985.3013
At phi =      3.3000  Tad =    972.4176
At phi =      3.4000  Tad =    962.6084
At phi =      3.5000  Tad =    954.7193
Output written to adiabatic.csv
```

At the end, a **.csv** file should have been created that contains the adiabatic flame temperature as well as the molar fractions of the species in the mixture for every equivalence ratio.

3 Several modifications

3.1 General parameters

To modify the parameters linked to the mechanism used (initial temperaure and pressure, range of equivalence ratios, ...), you just have to open the script you want to modify with your favorite text editor (vi, emacs, gedit...). Next, modify anything you want in the first section. For example, open adiab_flame_T.py :

```
###############################################################################
# Edit these parameters to change the initial temperature, the pressure, and
# the phases in the mixture.

# phases
gas = ct.Solution('gri30.cti')

# equivalence ratio range
phi_min = 0.3
phi_max = 3.5
npoints = 33

# Set the gas composition :
T = 300.0
P = 101325.0

# find fuel, nitrogen, and oxygen indices
fuel_species = 'CH4'

...
```

3.2 Try the 2S_CH4_BFER mechanism

To change the mechanism, you will need to copy the appropriate **.cti** file in your working directory. The 2S_CH4_BFER mechanism can be found here : http://cerfacs.fr/cantera/docs/mechanisms/methane-air/GLOB/CANTERA/2S_CH4_BFER.cti. Two steps are needed to run it properly :

1. **Place the file in the right folder :**

   The **.cti** file needs to be in the CANTERA sub-directory where your script is. Do not change the file name (**2S_CH4_BFER.cti** in this case).

2. **Modify your script :**

   Open your script in a text editor and import the good .cti name in the first section by replacing the following lines :

   ```
   # phases
   gas = ct.Solution('gri30.cti')
   ```

   by :

   ```
   # phases
   gas  = ct.Solution('2S_CH4_BFER.cti')
   ```

#! NOTE : If you change your fuel entirely, you will also need to replace the line :!
#

```
fuel_species = 'CH4'.
```

by :

```
fuel_species = 'whatever your species is'.
```

For exemple, if the fuel species you are using is Propane, the line sould read :

```
fuel_species = 'C3H8'
```

You should notice that the computation time with this reduced mechanism is significantly faster. But you should also notice non negligible discrepancies, especially when the composition is rich :

```
*********************************************************
    Computing Equilibirum for phi from 0.3 to 3.5, T = 300.0 K, P = 101325.0 Pa
    Equilibrate holding HP constants
*********************************************************
At phi =      0.3000  Tad =   1066.5973
At phi =      0.4000  Tad =   1282.0446
At phi =      0.5000  Tad =   1483.0925
At phi =      0.6000  Tad =   1671.6091
At phi =      0.7000  Tad =   1848.6458
At phi =      0.8000  Tad =   2013.2492
At phi =      0.9000  Tad =   2158.6389
At phi =      1.0000  Tad =   2258.7335
At phi =      1.1000  Tad =   2225.1789
At phi =      1.2000  Tad =   2129.3678
At phi =      1.3000  Tad =   2033.9179
At phi =      1.4000  Tad =   1985.0318
At phi =      1.5000  Tad =   1959.1590
At phi =      1.6000  Tad =   1934.2796
At phi =      1.7000  Tad =   1910.3129
At phi =      1.8000  Tad =   1887.2022
At phi =      1.9000  Tad =   1864.8982
At phi =      2.0000  Tad =   1843.3559
At phi =      2.1000  Tad =   1822.5343
At phi =      2.2000  Tad =   1802.3950
At phi =      2.3000  Tad =   1782.9030
At phi =      2.4000  Tad =   1764.0252
At phi =      2.5000  Tad =   1745.7311
At phi =      2.6000  Tad =   1727.9922
At phi =      2.7000  Tad =   1710.7819
At phi =      2.8000  Tad =   1694.0753
At phi =      2.9000  Tad =   1677.8490
At phi =      3.0000  Tad =   1662.0812
At phi =      3.1000  Tad =   1646.7514
At phi =      3.2000  Tad =   1631.8403
At phi =      3.3000  Tad =   1617.3299
At phi =      3.4000  Tad =   1603.2030
At phi =      3.5000  Tad =   1589.4437
Output written to adiabatic.csv
```

3.3 Add a plot !

You could also draw plots of different properties against the stoichiometric ratio, if Matplotlib is installed on your computer. You simply need to open your script adiab_flame_T.py in your favorite editor and to uncomment the last liens.