Cantera Tutorial

0-D simulations of adiabatic reactors

18 juillet 2014

1 Introduction - Objectives

The aim of this tutorial is to walk non-experienced CANTERA users through the computation of 0-D adiabatic reactors simulations. It will provide the temporal evolution of the properties of the mixture inside the reactor.

Like in the other tutorials, we will work with the detailed Methane-Air GRIMECH 3.0 mechanism, and we will perform both constant pressure (HP) and constant volume simulations (UV).

2 Steps

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
```

2.2 Get the script and mechanism

- The script can be downloaded here : http://www.cerfacs.fr/cantera/docs/scripts/reactor/reactor_HP_UV1.py
- And the .cti file here : http://www.cerfacs.fr/cantera/docs/mechanisms/methane-air/Gri30/gri30.cti
- They should be placed in a CANTERA sub-directory of your choice.

As for the other tutorials, Methane 'CH4' is the fuel species. Other detailed mechanism 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 reactor_HP_UV.py
```

This script is automated, so most of the modifications of the general parameters (temperature, pressure, ...) can be done directly in the terminal. You will have to type the values of properties asked on screen. For this exemple, the calculation was performed with the following settings :

Thermodynamic Properties :

Stoichiometric ratio	1
Initial Temperature	1500 K
Initial Pressure	1 bar
Condition	HP

The constant enthalpy and pressure reactor is chosen by typing **HP** in the terminal.

Next, set the computational properties :

Computational Properties :

Number of time steps	50
Time step	5ms

It is important to run the simulation for a time that is long enough for the mixture to ignite. Also, you want your time steps to be small enough for the continuation to run properly, and to catch the ignition time more accurately.

If all is properly set up, the program will return the evolution of some of the mixture's thermodynamic properties with time :

```
time [s] ,    T [K] ,    p [Pa] ,    u [J/kg]
 5.000e-05   1504.029 101325.000   8.448990e+05
 1.000e-04   1504.012 101325.000   8.449017e+05
 1.500e-04   1504.036 101325.000   8.448908e+05
 2.000e-04   1504.126 101325.000   8.448591e+05
 2.500e-04   1504.309 101325.000   8.447992e+05
[...]
 1.100e-03   1727.958 101325.004   7.676147e+05
 1.150e-03   2439.340 101325.002   5.102525e+05
 1.200e-03   2547.896 101325.001   4.823086e+05
 1.250e-03   2616.853 101325.001   4.648514e+05
 1.300e-03   2662.295 101325.000   4.534731e+05
 1.350e-03   2692.284 101325.000   4.460252e+05
[...]
 2.400e-03   2742.218 101325.000   4.342545e+05
 2.450e-03   2742.013 101325.000   4.343269e+05
 2.500e-03   2741.812 101325.000   4.343976e+05
```

2.4   Saving files

Eventually, the datas are written in a **.csv** output file that can be imported into Excel.

2.5 The plot option

There is also a ploting option in this script : if Matplotlib is properly installed on your operating system (help to install it on MacOS can be found here : http://avbpedia.cerfacs.fr/uploads/Other/Tutorial_MacOS.pdf), you can plot the results by running the script with the option **-plot** as follows :

```
python reactor_HP_UV.py -plot
```

Of course, you can modify the properties you whish to plot by opening the script in your favorite text editor (vi, gedit, emacs, ...) and modifying the last part, under :

```
####################################################################
# Plot your results
####################################################################
```

See the next section.

3 Several modifications

3.1 Try another mechanism

If you would like to change the mechanism used, you will need to get the **.cti** file and copy it in your working CANTERA sub-directory. Then you will need to alter your script :

In a text editor, modify the two following lines of the file **reactor_HP_UV.py** :

```
#Mechanism used for the process
cti = importPhase('gri30.cti')

#Gaseous fuel species
fuel_species = 'CH4'
```

For example, if you decided to use the 2S_KERO_BFER.cti mechanism, you would need to write :

```
#Mechanism used for the process
cti = importPhase('2S_KERO_BFER.cti')

#Gaseous fuel species
fuel_species = 'KERO'
```

3.2 Change ploted properties

To change the properties ploted, open the file **reactor_HP_UV.py** in a text editor and modify this :

```
from matplotlib.pylab import *
clf
subplot(2,2,1)
plot(tim,temp[:])
xlabel('Time (s)');
ylabel('Temperature (K)');
subplot(2,2,2)
plot(tim,mfrac[:,cti.speciesIndex('OH')])
xlabel('Time (s)');
ylabel('OH Mass Fraction');
subplot(2,2,3)
plot(tim,mfrac[:,cti.speciesIndex('H')]);
xlabel('Time (s)');
ylabel('H Mass Fraction');
subplot(2,2,4)
plot(tim,mfrac[:,cti.speciesIndex('H2')]);
xlabel('Time (s)');
ylabel('H2 Mass Fraction');
show()
```

To plot the pressure against time, for example, you can switch **temp[:]** by **press[:]**... or you can add a new plot altogether!

To see another species' mass fraction, you just have to change the species name in magenta above by the name of another one, present in the **.cti file** you use.

Don't forget to change the titles, so that they fit your plots.