Cantera Tutorial

0-D simulations of adiabatic reactors

11 septembre 2017

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 different types of reactors.

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/Reactors.py`
— And the .cti file here : `http://cerfacs.fr/cantera/docs/mechanisms/methane-air/DETAILED/CANTERA/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 Reactors.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	1000 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	1000
Time step	1ms

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. Note that the inital time is also important, and should be adapted to specific simulations.

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]
 4.010e-01   1003.912 101325.000   5.919182e+05
 4.020e-01   1003.917 101325.000   5.919182e+05
 4.030e-01   1003.922 101325.000   5.919182e+05
 4.040e-01   1003.926 101325.000   5.919182e+05
 4.050e-01   1003.931 101325.000   5.919182e+05
 4.060e-01   1003.936 101325.000   5.919182e+05
[...]
 1.043e+00   1316.091 101325.000   5.919184e+05
 1.044e+00   2525.055 101325.000   5.919182e+05
 1.045e+00   2551.152 101325.000   5.919181e+05
 1.046e+00   2550.658 101325.000   5.919181e+05
 1.047e+00   2549.675 101325.000   5.919181e+05
 1.048e+00   2548.771 101325.000   5.919181e+05
[...]
 1.399e+00   2542.375 101325.000   5.919181e+05
 1.400e+00   2542.375 101325.000   5.919181e+05
output written to Reactor_HP.csv
To view a plot of these results, run this script with the option -plot
```

2.4 Saving files

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

2.5 The plot option

As the output from the previous simulation suggests, there is also a plotting 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 Reactors.py -plot
```

This should produce a graph composed of 4 plots : Temperature, OH, H and H_2. 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
#####################################################################
```

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 = Solution('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 = Solution('2S_KERO_BFER.cti')

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

3.2 Change ploted properties

To change the properties ploted, open the file **Reactors.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.species_index('OH')])
xlabel('Time (s)');
ylabel('OH Mass Fraction');
subplot(2,2,3)
plot(tim,mfrac[:,cti.species_index('H')]);
xlabel('Time (s)');
ylabel('H Mass Fraction');
subplot(2,2,4)
plot(tim,mfrac[:,cti.species_index('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.