Cantera: a simulation of pressure loss in a leaking tank

Andrzej Borkowski

June 13, 2019

1 Problem description

The goal of the project was to simulate the loss of pressure in a tank containing a mixture of flammable gases.
The code is written in Python, using Cantera library for simulation of chemical and physical process and Matplotlib, PIL and Tkinter for creating a visual interface.

2 Simulation model

In the simulation the tank is treated as a zero - dimensional, thermically isolated reactor, from which the gases escape to the enviroment - a reservoir containing air. The leak is simulated by a valve.
The fuel consists of hydrogen, methane, ethane and propane in mole proportion set by user and is mixed with air or oxygen according to chosen equivalence ratio. The temperature and pressure of the mixture are also set by user.
Ignition in tank may or may not happen, depending on the beginning conditions. After the parameters are set, the program prints the state of gas at the beginning of the simulation.

3 Model of leakage

The leakage is simulated as a valve. Cantera uses model in which the mass of gas flowing through a valve is described by the following equation:
Q = KvΔ P
To determine the value of Kv another equation was used:
Q = $\alpha A\sqrt{(2g\rho}\ \Delta$ P) [1]

[1] Bomelburg, H.J.,1977,"Estimation of gas leak rates through very small orifices and channels."

Following these equations, Kv depends on two variables and the equation for Kv may be shown in the following way:

$Kv = k\sqrt{(\rho/\Delta P)}$

Where the k factor depends on the size of the orifice and other factors that should be determined empyrically. In this simulations it is set constant:

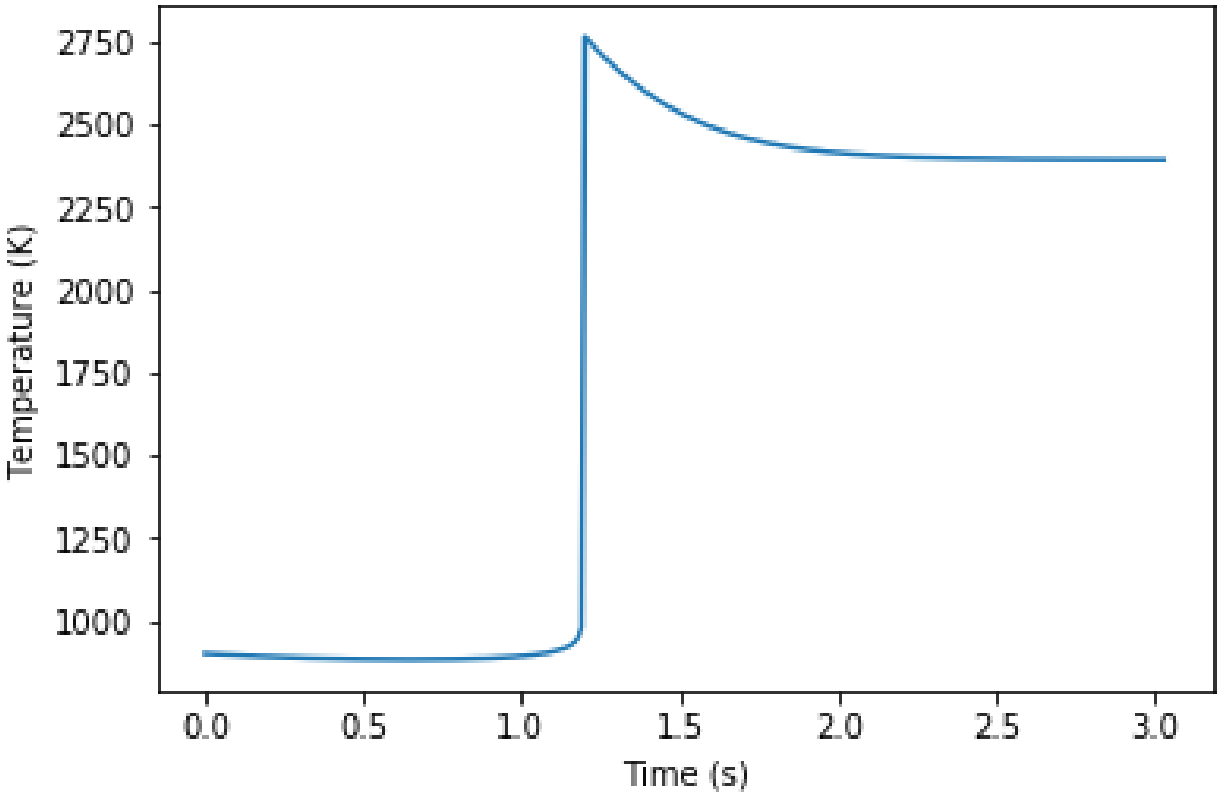
$k = 0.001$

For calculations factor ρ is read as the density of gas inside the tank and ΔP is the difference in pressure inside and outside of the tank.

4 Solution

The simulation proceeds with time steps equal 0,01 s, for each step printing and saving to file the number of step, time, temperature and pressure. The name of the output file is "tankleak.csv".

At the end of the printed results of calculations the program generates two plots for pressure and temperature change in time, based on the values saved in file.



/2/2

Figure 1: An example of a plot of change of temperature in time.

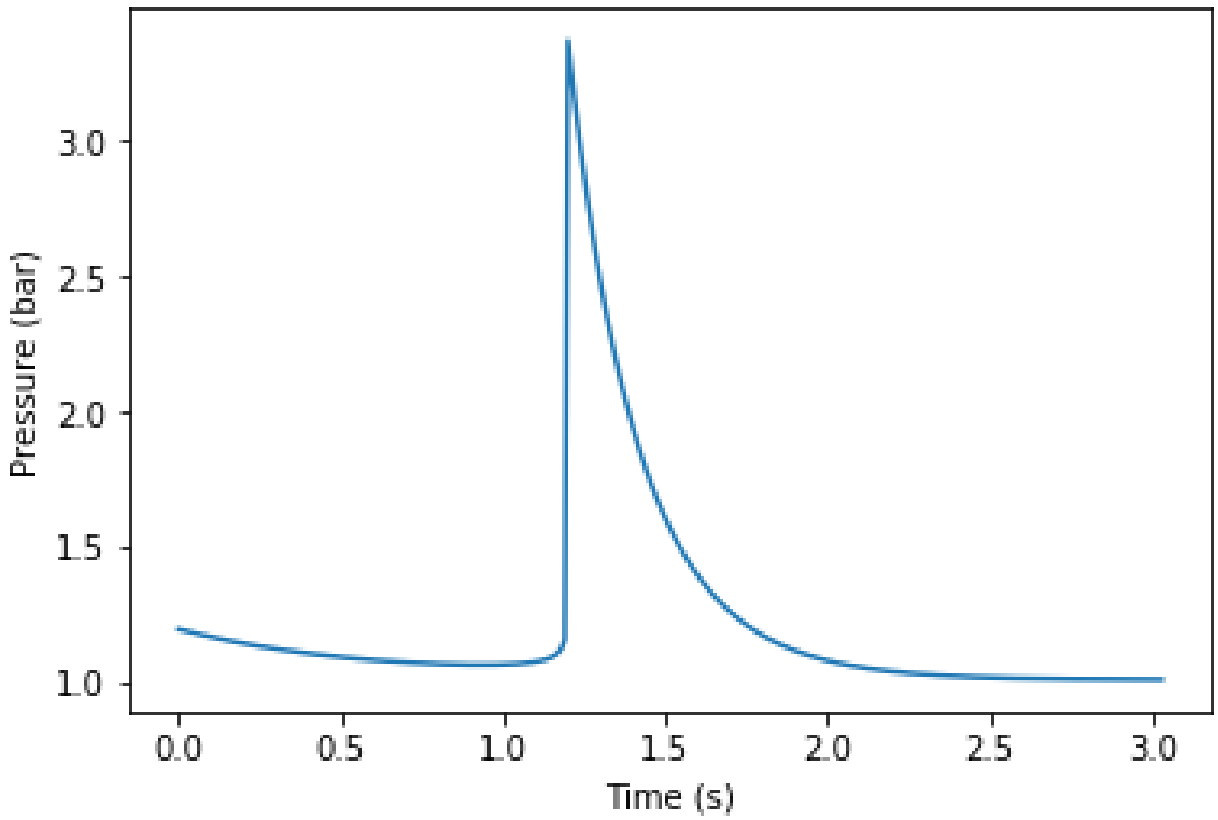


Figure 2: An example of a plot of change of pressure in time.

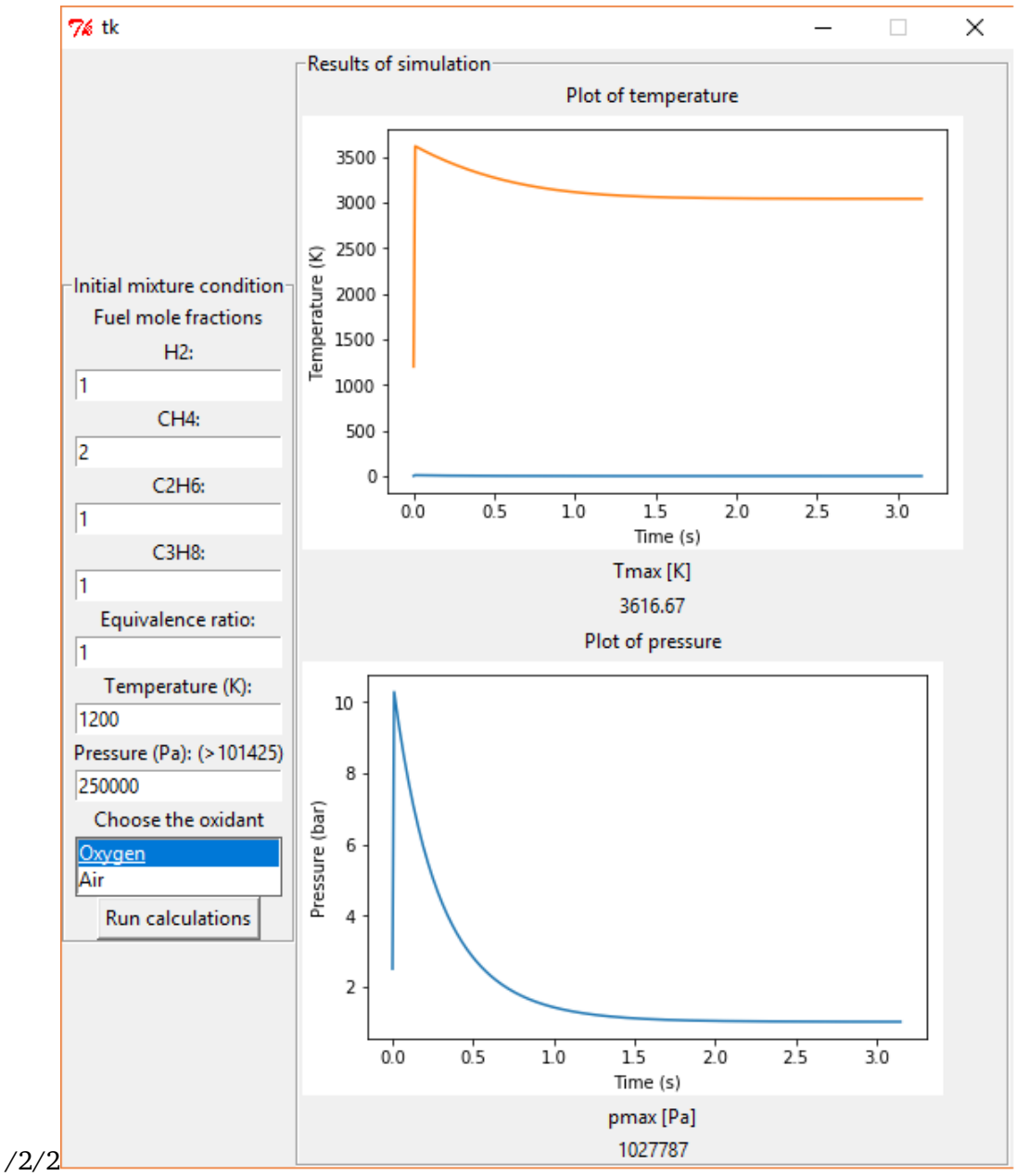
5 Graphical user interface

To make the program easier, quicker and more visually clear in use it has been updated with a graphical interface. The interface features two frames: left - for setting the initial state of the simulation and right - the output frame. After initialising the code, both frames open in a Tkinter window.

The left frame is titled "Initial mixture condition" and contains input fields for all previously mentioned initial variables. The input boxes contain default values which do not have to be changed. The choice of oxidant is realised by a list box. In this case there is no default value and the chosen oxidant becomes unchecked again in case of change of other input values. Please pay attention to this while using the interface - make choosing the oxidant the last action before starting the simulation, otherwise the simulation will not begin. To initialize the calculations with chosen parameters, press the "Run calculations" button.

The right frame is titled "Results of simulation". It contains plots of pressure (blue) and temperature (orange) as well as maximum values of these variables. These values are set as zero by default and

change to calculated value after running the simulation, which makes the plots appear as well. The code allows for series of calculations without closing the window. However, a common problem is that after first launch of the program the plots do not scale properly. If this happens, just close the window and run the code again.



/2/2

Figure 3: An example of GUI view after running the calculations.

6 Libraries

The simulation requires os, csv, cantera, matplotlib.pyplot, Tkinter, PIL and numpy libraries.

7 Sources

For modelling the leakage I have used equations provided in Bomelburg, H.J.,1977,"Estimation of gas leak rates through very small orifices and channels." As for programming solutions I have based on the Python examples provided on the Cantera homepage: "combustor.py" and "reactor2.py". Other solutions used in the code were either found on https://stackoverflow.com/, http://effbot.org/tkinterbook/ or invented for the purpose of this project.