Warsaw University of Technology
Faculty of Power and Aeronautical Engineering

Computational Methods in Combustion

Effect of the equivalence coefficient on methane flame temperature and exhaust composition

Author:
Justyna Oleszycka *(313851)*

Checking:
dr inż. Mateusz Żbikowski

15/06/2023

Contents

1 Introduction

The objective of this project is to calculate the adiabatic flame temperature in constant pressure and visualize the mass fractions of selected exhaust components as a function of equivalence ratio of methane burned in air.
In order to complete the calculations Cantera was used.

1.1 Stoichiometric equasion

Stoichiometric equation representing methane-air combustion is given as followed:

$$CH_4 + 2(O_2 + 3,76N_2) = 2H_2O + CO_2 + 2 \cdot 3,76N_2$$

1.2 Equivalence ratio

Also known as "fuel ratio" :

$$\phi = \frac{F/A}{(F/A)_s}$$

$$\phi = \frac{1}{\lambda}$$

Where:

- F - mole number of fuel
- A - mole number of air
- $(F/A)_s$ - stoichiometric ratio
- λ - excess air coefficient

1.3 Initial parameters

$$p = 101325 \quad Pa$$

$$T = 300 \quad K$$

$$\phi = 0,3 \div 3,5$$

2 Method description

Using Cantera the basic reactor and gas models were prepared. All the simplifications and assumptions of the model are in line with those adopted in Cantera (e.g. the gas model is ideal, the combustion chamber is determined by the program conditions etc.).

Arrays are initialized to store the equivalence ratios, adiabatic flame temperatures, and species mole fractions at each data point.

The equilibrium analysis is performed using a loop that iterates over the specified number of data points. At each iteration, the current equivalence ratio is calculated, and the mole fractions of fuel, oxygen, and nitrogen are set accordingly.

3 Results

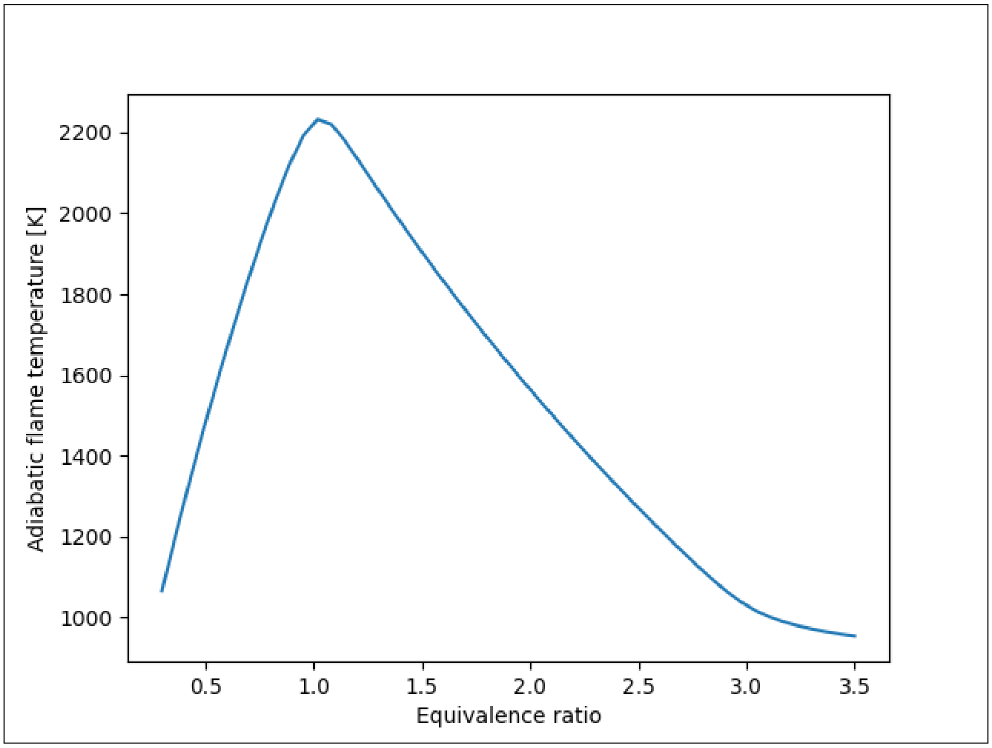

Figure 1: Flame temperature in equivalence ratio function

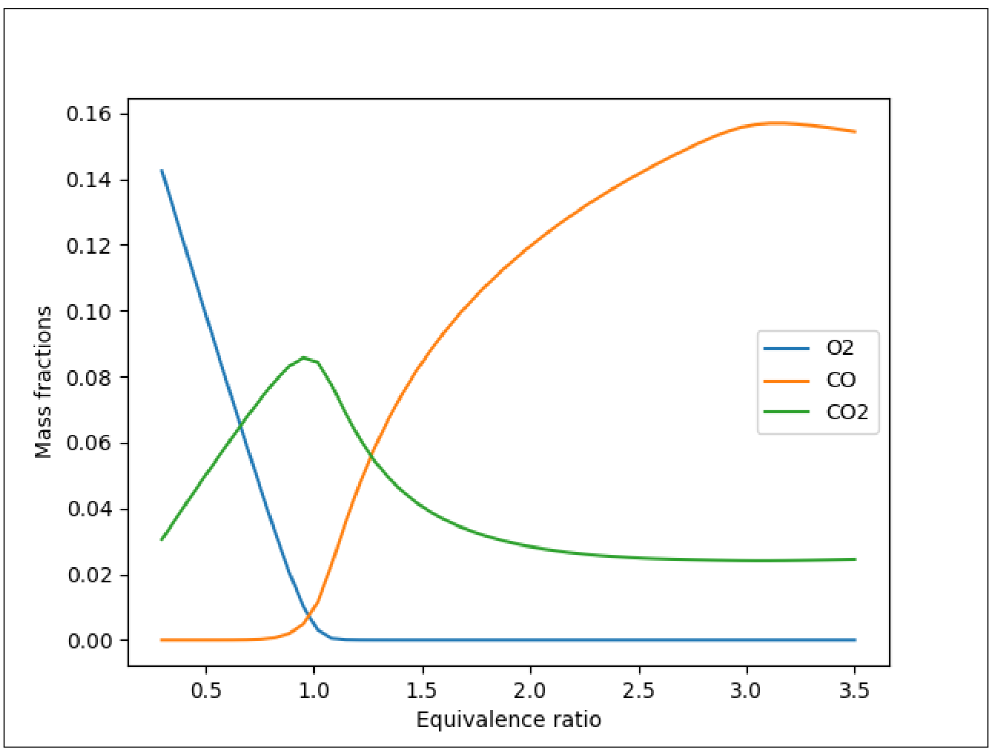

Figure 2: Eshaust components percentage in equivalence ratio function

4 Conclusions

- The adiabatic flame temperature represents the maximum temperature achieved during the combustion process under adiabatic conditions. As presented in "Results" section, maximum temperature is reached for equivalence ratio equal 1, that means stoichiometric ratio.
- If the stoichiometric ratio is lower than 1, in exhaust components O_2 is detected, that means that we have excess of air.
- If the stoichiometric ratio is higher than 1, in exhaust components CO is being detected, that means that we have deficiency of air.
- The plotted curves of the mass fractions of selected exhaust components provide insights into the composition of the combustion products at different equivalence ratios.
- The code utilizes the stoichiometry of the fuel to determine the proper ratios of oxygen and nitrogen in the air composition. The stoichiometric oxygen requirement is calculated based on the fuel's carbon and hydrogen content. This information is important for achieving complete combustion and understanding the fuel-air mixture requirements.

5 Bibliography

1. Materials provided by the instructor
2. Marian Gieras, Spalanie - wybrane zagadnienia w zadaniach, 2011
3. https://en.wikipedia.org/wiki/Methane
4. https://en.wikipedia.org/wiki/Air

6 Python code

```
import cantera as ct
import numpy as np
import matplotlib.pyplot as plt


# import the gas
gas = ct.Solution("gri30.yaml")
# equivalence ratio range
phi_min = 0.3
phi_max = 3.5
npoints = 50
# set the gas composition
T = 300.0
P = 101325.0
# find fuel, nitrogen and oxygen indices
fuel_species = "CH4"
ifuel = gas.species_index(fuel_species)
io2 = gas.species_index("O2")
in2 = gas.species_index("N2")
# air composition
air_N2_O2_ratio=3.76
stoich_O2 = gas.n_atoms(fuel_species, "C")+0.25*gas.n_atoms(fuel_species, "H")
# some arrays to hold the data
phi = np.zeros(npoints)
tad = np.zeros(npoints)
xeq = np.zeros((gas.n_species, npoints))
solution
for i in range(npoints):
phi[i] = phi_min+(phi_max-phi_min)*i/(npoints-1)
X = np.zeros(gas.n_species)
X[ifuel] = phi[i]
X[io2] = stoich_O2
X[in2] = stoich_O2*air_N2_O2_ratio
# set the gas state
gas.TPX = T, P, X
# equilibrate the mixture adiabatically at constant P
gas.equilibrate("HP")
tad[i] = gas.T
xeq[:,i ]= gas.X
print("At phi= ", "%10.4f"%(phi[i])+" Tad = ", "%10.4f"%(tad[i]))
# results
# mass fractions of selected species
for i, cas in enumerate(gas.species_names):
if cas in ["O2","CO2","CO"]:
plt.plot(phi,xeq[i,:],label=cas)
plt.xlabel("Equivalence ratio")
plt.ylabel("Mass fractions")
plt.legend(loc="best")
plt.show()
# adiabatic flame temperature
```