



Faculty of Power and Aeronautical Engineering

WARSAW UNIVERSITY OF TECHNOLOGY

Effect of equivalence ratio on flame temperature and exhaust composition of ethane-air & methane-air mixtures

Computational Methods in Combustion

Authors:

Aleksandra Sadowska 320984

Aleksander Bronkowski 320923

Checking:

dr inż. Mateusz Żbikowski

Warsaw, 2024-05-30

Contents

1	Introduction	3
1.1	Description	3
1.2	Stoichiometric equations	3
1.3	Initial conditions	3
2	Method description	3
3	Results	4
3.1	Methane mixture	4
3.2	Ethane mixture	5
4	Conclusions	6
5	References	6
6	Python code	7

1 Introduction

1.1 Description

The equivalence ratio is a dimensionless parameter used in combustion analysis to compare the actual fuel-air ratio to the stoichiometric fuel-air ratio and is defined as their quotient:

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

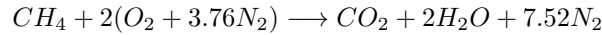
Let us note that:

- when $\phi = 1$, the mixture is stoichiometric, meaning there is exactly enough oxygen to completely combust the fuel;
- when $\phi < 1$, the mixture is lean, meaning there is excess air (or oxygen) relative to the amount of fuel;
- when $\phi > 1$ the mixture is rich, meaning there is excess fuel relative to the amount of air (or oxygen).

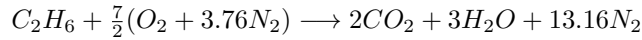
This work aims to present the effect of the equivalence ratio on the adiabatic flame temperature and mass fractions of the main exhaust components of two combustible mixtures - ethane- and methane-air mixture.

1.2 Stoichiometric equations

Stoichiometric equation for methane-air combustion is given as follows:



Similarly, for ethane-air mixture:



1.3 Initial conditions

$$p = 101325 \text{ Pa}$$

$$T = 298.15 \text{ K}$$

2 Method description

The code simulates the combustion of ethane and methane, calculating the adiabatic flame temperature and equilibrium composition for different equivalence ratios (ϕ). It uses the Cantera library for combustion modeling and NumPy and Matplotlib for computations and plotting. It loads gas mixture data from the `gri30.yaml` file, sets gas compositions, and performs adiabatic equilibrium at constant pressure. The `calculate_adibatic_flame_temp` function generates data on flame temperature and chemical composition. Then, the `plot_species_mass_fractions` and `plot_adibatic_flame_temp` functions create plots of mass fractions of selected chemical species and flame temperature as a function of phi. The final functions, `make_calc_for_ethane` and `make_calc_for_methane`, perform these calculations and create plots for ethane and methane, saving the results to graphic files and displaying them.

3 Results

3.1 Methane mixture

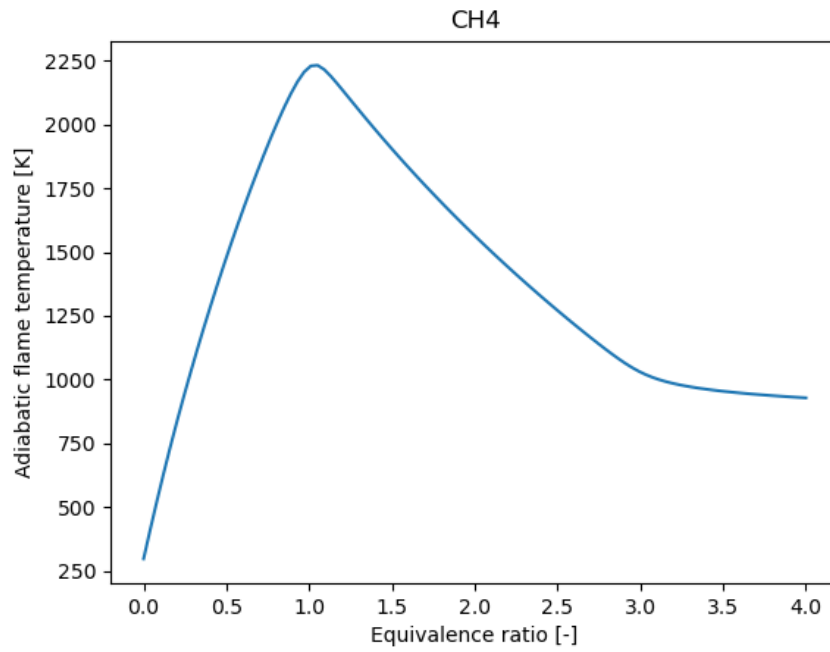


Figure 1: Methane flame temperature as a function of the equivalence ratio

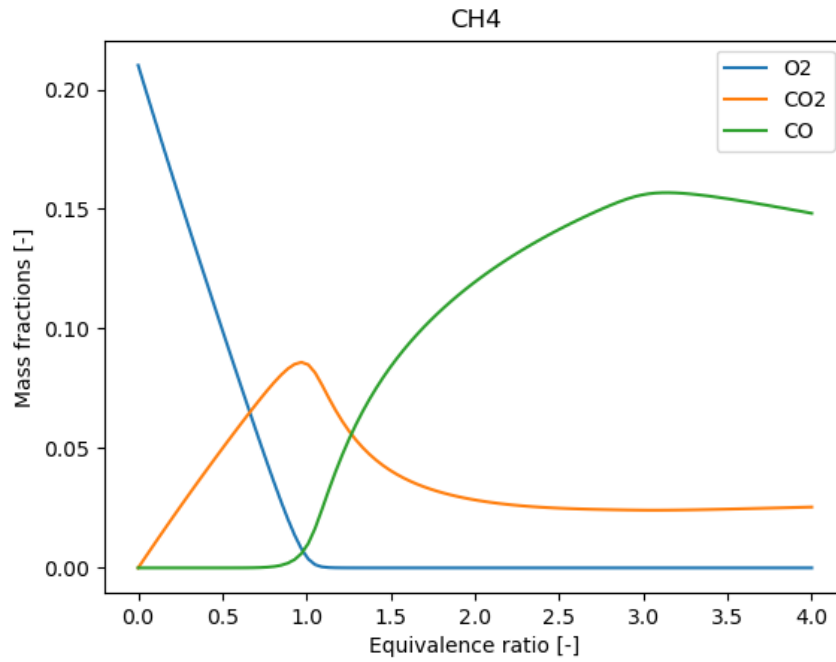


Figure 2: Methane exhaust components as a function of the equivalence ratio

3.2 Ethane mixture

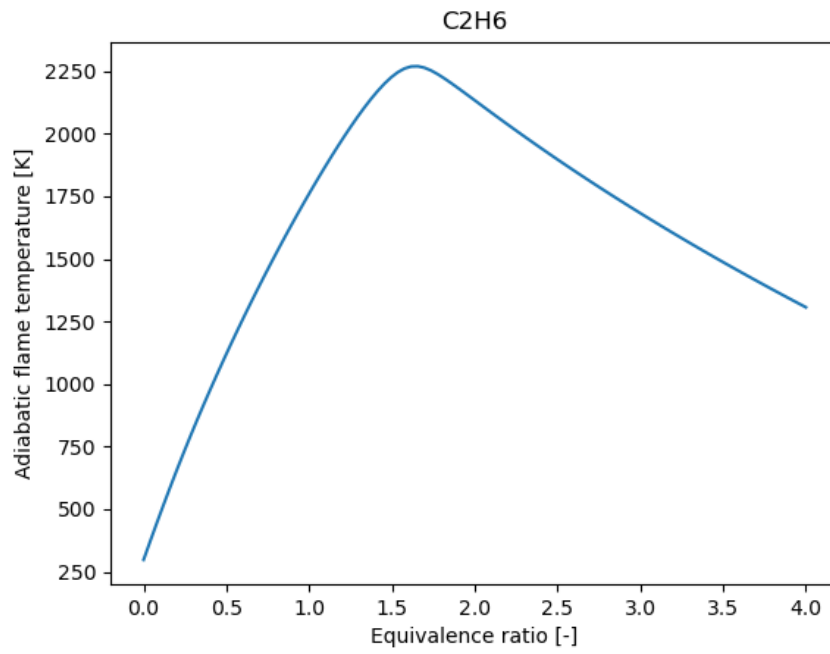


Figure 3: Ethane flame temperature as a function of the equivalence ratio

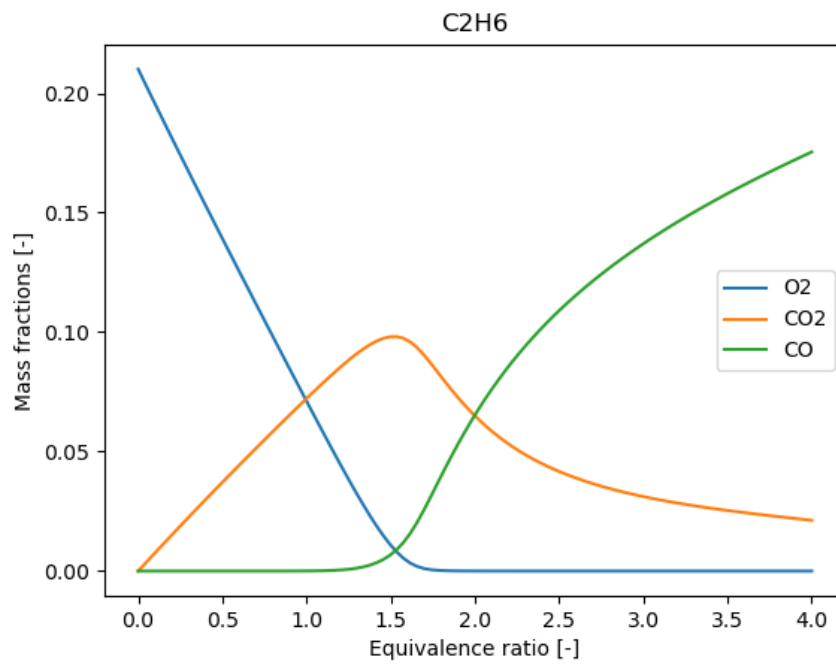


Figure 4: Ethane exhaust components as a function of the equivalence ratio

4 Conclusions

- When the equivalence ratio is small ($\phi < 1$), the percentage of O_2 in the exhaust components is very high, as not all of the fuel gets consumed in the combustion process.
- With ϕ approaching 1, the amount of O_2 drops significantly, quickly approaching 0 as ϕ exceeds the value of 1.
- On the contrary, the percentage of CO is negligible when $\phi < 1$ and rises drastically once the ratio reaches 1 for methane and approximately 1.5 for ethane. This happens because a lean mixture allows the fuel to be more completely combusted.
- The highest temperature is similar for both methane and ethane (around 2250K), however, for methane it is reached for a lower equivalence ratio - about 1 for methane and 1.5 for ethane. The highest combustion temperature typically occurs when the mixture is stoichiometric because the fuel and oxygen are perfectly balanced for complete combustion.
- The CO_2 concentration peaks at around the same equivalence ratio as the highest methane or ethane temperature respectively.
- At $\phi < 1$ the combustion temperatures are generally lower because the excess air absorbs heat, leading to cooler combustion products.
- When $\phi > 1$, combustion temperatures are also lower compared to the stoichiometric case because the excess fuel consumes heat to vaporize and partially oxidize

5 References

- Plastics to Energy, Chapter 10 (Gasification of Plastic Solid Waste and Competitive Technologies) - S.A. Salaudeen, Animesh Dutta
- Effect of Equivalence Ratio on Pollutant Formation in $CH_4O/H_2/NH_3$ Blend Combustion - Jingyun Sun, Qianqian Liu
- <https://en.wikipedia.org/wiki/Methane>
- <https://en.wikipedia.org/wiki/Ethane>

6 Python code

```
1 import cantera as ct # Cantera library for combustion simulations
2 import numpy as np   # Numerical Python library for mathematical operations
3 import matplotlib.pyplot as plt # Library for creating plots
4 # Load the gas mixture from the gri30 database
5 gas = ct.Solution("gri30.yaml")
6 def calculate_adibatic_flame_temp(fuel_species, phi_min, phi_max, npoints, T, P):
7     """
8     Calculate adiabatic flame temperature and equilibrium composition for a given fuel.
9     Parameters:
10    fuel_species (str): The chemical formula of the fuel (e.g., 'C2H6' for ethane).
11    phi_min (float): The minimum equivalence ratio.
12    phi_max (float): The maximum equivalence ratio.
13    npoints (int): The number of points to calculate between phi_min and phi_max.
14    T (float): Initial temperature of the gas mixture.
15    P (float): Initial pressure of the gas mixture.
16
17    Returns:
18    tuple: Arrays of equivalence ratios, adiabatic flame temperatures, and equilibrium compositions.
19    """
20    # Indices for the fuel, oxygen, and nitrogen species
21    ifuel = gas.species_index(fuel_species)
22    io2 = gas.species_index("O2")
23    in2 = gas.species_index("N2")
24
25    # Air composition: Nitrogen to Oxygen ratio
26    air_N2_O2_ratio = 3.76
27
28    # Calculate stoichiometric oxygen needed for the fuel
29    if fuel_species == 'C2H6':
30        stoich_O2 = 2 * gas.n_atoms(fuel_species, "C") + 0.25 * gas.n_atoms(fuel_species, "H")
31    elif fuel_species == 'CH4':
32        stoich_O2 = gas.n_atoms(fuel_species, "C") + 0.25 * gas.n_atoms(fuel_species, "H")
33
34    # Create arrays to store equivalence ratios, temperatures, and compositions
35    phi_values = np.linspace(phi_min, phi_max, npoints)
36    tad = np.zeros(npoints)
37    xeq = np.zeros((gas.n_species, npoints))
38
39    # Loop over each equivalence ratio
40    for i, phi in enumerate(phi_values):
41        # Set the composition of the gas mixture
42        X = np.zeros(gas.n_species)
43        X[ifuel] = phi # Fuel mole fraction
44        X[io2] = stoich_O2 # Oxygen mole fraction
45        X[in2] = stoich_O2 * air_N2_O2_ratio # Nitrogen mole fraction
46
47        # Set the gas state: temperature, pressure, and composition
48        gas.TPX = T, P, X
49        # Equilibrate the mixture adiabatically at constant pressure
50        gas.equilibrate("HP")
51
52        # Store the adiabatic flame temperature and equilibrium composition
53        tad[i] = gas.T
54        xeq[:, i] = gas.X
55        print(f"At phi= {phi:.4f} Tad = {gas.T:.4f}")
56
57    return phi_values, tad, xeq
58
59 def plot_species_mass_fractions(fuel_species, phi_values, xeq, species_names):
60     """
61     Plot the mass fractions of selected species against the equivalence ratio.
62     Parameters:
63    fuel_species (str): The chemical formula of the fuel.
64    phi_values (array): Array of equivalence ratios.
65    xeq (array): Array of equilibrium compositions.
66    species_names (list): List of species names.
67    """
68    # Select species to plot
69    for species in ["O2", "CO2", "CO"]:
```

```

70         if species in species_names:
71             index = species_names.index(species)
72             plt.plot(phi_values, xeq[index, :], label=species)
73
74     plt.xlabel("Equivalence ratio [-]")
75     plt.ylabel("Mass fractions [-]")
76     plt.legend(loc="best")
77     plt.title(fuel_species)
78     plt.savefig(f'mass_fractions_{fuel_species}.png')
79     plt.show()
80
81 def plot_adibatic_flame_temp(fuel_species, phi_values, tad):
82     """
83     Plot the adiabatic flame temperature against the equivalence ratio.
84     Parameters:
85     fuel_species (str): The chemical formula of the fuel.
86     phi_values (array): Array of equivalence ratios.
87     tad (array): Array of adiabatic flame temperatures.
88     """
89     plt.plot(phi_values, tad)
90     plt.xlabel("Equivalence ratio [-]")
91     plt.ylabel("Adiabatic flame temperature [K]")
92     plt.title(fuel_species)
93     plt.savefig(f'flame_temp_{fuel_species}.png')
94     plt.show()
95
96 def make_calc_for_ethane():
97     """
98     Perform calculations and plots for ethane (C2H6).
99     """
100     # Parameters for ethane
101     fuel_species = "C2H6"
102     phi_min = 0.3
103     phi_max = 4
104     npoints = 100
105     T = 298.15 # Initial temperature in Kelvin
106     P = 101325.0 # Initial pressure in Pascals
107
108     # Calculate adiabatic flame temperature and equilibrium composition
109     phi_values, tad, xeq = calculate_adibatic_flame_temp(fuel_species, phi_min, phi_max, npoints, T, P)
110     # Plot results
111     plot_species_mass_fractions(fuel_species, phi_values, xeq, gas.species_names)
112     plot_adibatic_flame_temp(fuel_species, phi_values, tad)
113
114 def make_calc_for_methane():
115     """
116     Perform calculations and plots for methane (CH4).
117     """
118     # Parameters for methane
119     fuel_species = "CH4"
120     phi_min = 0.3
121     phi_max = 4
122     npoints = 100
123     T = 298.15 # Initial temperature in Kelvin
124     P = 101325.0 # Initial pressure in Pascals
125
126     # Calculate adiabatic flame temperature and equilibrium composition
127     phi_values, tad, xeq = calculate_adibatic_flame_temp(fuel_species, phi_min, phi_max, npoints, T, P)
128     # Plot results
129     plot_species_mass_fractions(fuel_species, phi_values, xeq, gas.species_names)
130     plot_adibatic_flame_temp(fuel_species, phi_values, tad)
131
132 # Perform calculations and plots for ethane
133 make_calc_for_ethane()
134
135 # Perform calculations and plots for methane
136 make_calc_for_methane()

```