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

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:

$$CH_4 + 2(O_2 + 3.76N_2) \longrightarrow CO_2 + 2H_2O + 7.52N_2$$

Similarly, for ethane-air mixture:

$$C_2H_6 + \frac{7}{2}(O_2 + 3.76N_2) \longrightarrow 2CO_2 + 3H_2O + 13.16N_2$$

1.3 Initial conditions

$$P1 = 101325\ Pa$$
$$P2 = 202650\ Pa$$
$$P3 = 303975\ Pa$$

$$T1 = 273\ K$$
$$T2 = 298\ K$$
$$T3 = 323\ K$$

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_adiabatic_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

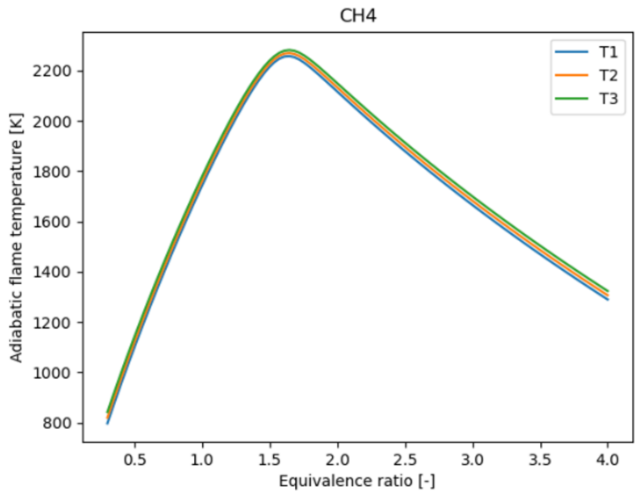

Figure 1: Methane flame temperature as function of the equivalence ratio, P1 = 1 atm

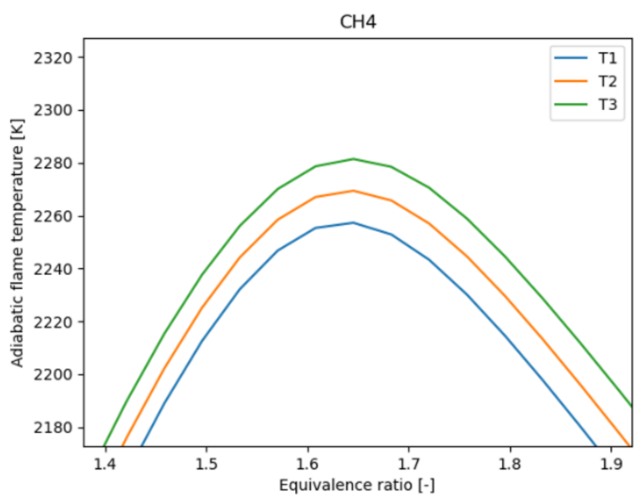

Figure 2: Methane flame temperature as function of the equivalence ratio, P1 = 1 atm

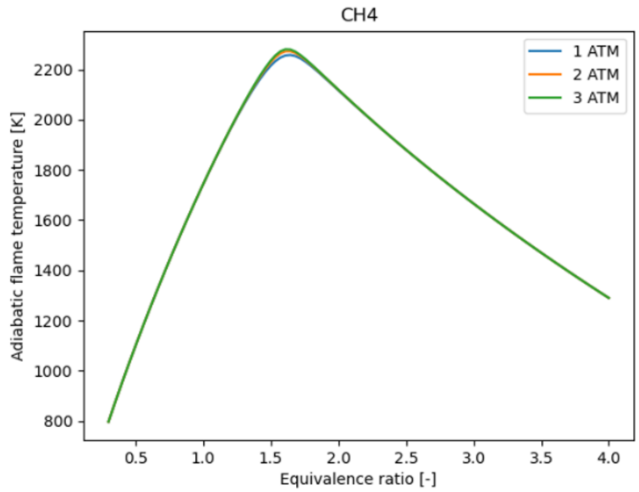

Figure 3: Methane flame temperature as function of the equivalence ratio, T1 = 273 K

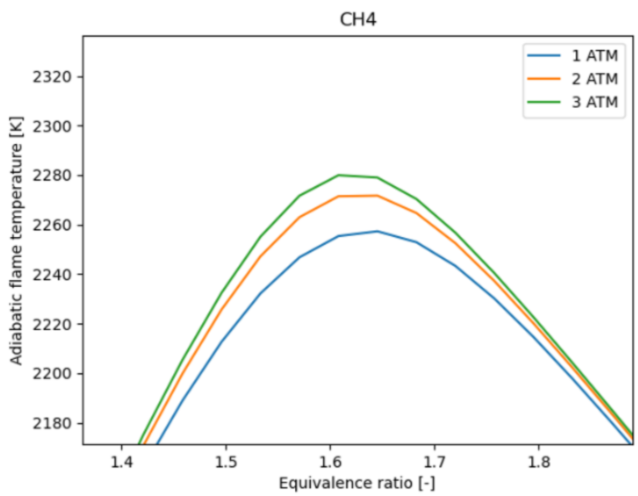

Figure 4: Methane flame temperature as function of the equivalence ratio, T1 = 273 K

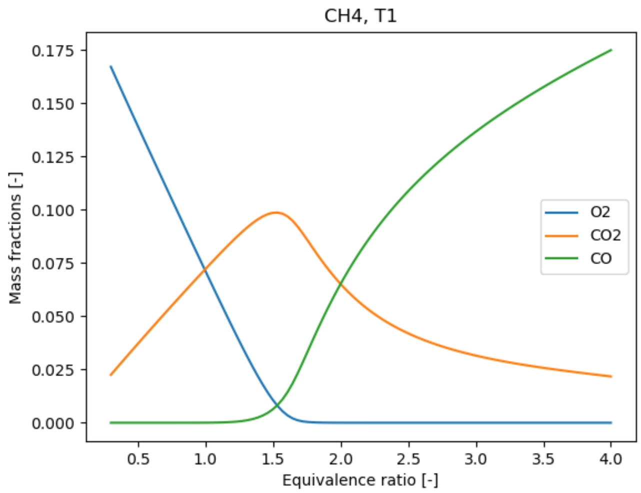

Figure 5: Methane exhaust components as a function of the equivalence ratio, P1 = 1 atm, T1 = 273 K

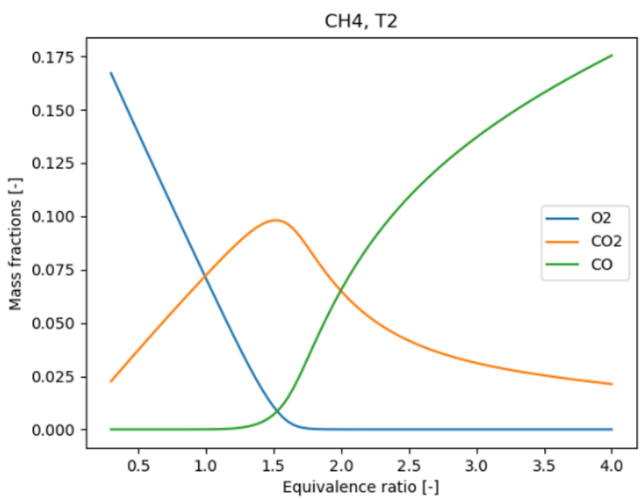

Figure 6: Methane exhaust components as a function of the equivalence ratio, P1 = 1 atm, T2 = 298 K

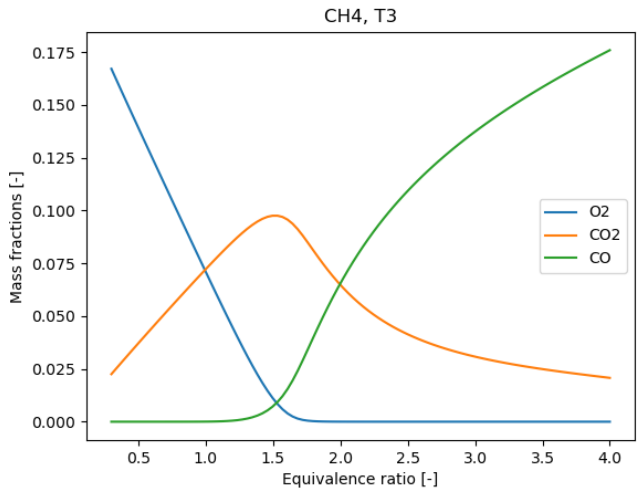

Figure 7: Methane exhaust components as a function of the equivalence ratio, P1 = 1 atm, T3 = 323 K

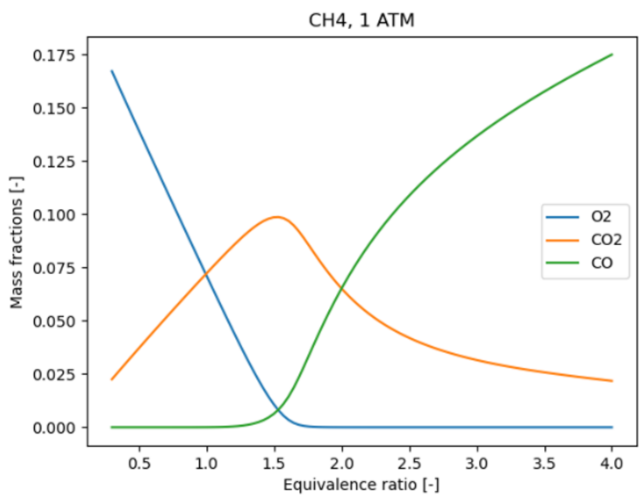

Figure 8: Methane exhaust components as a function of the equivalence ratio, T1 = 273 K, P1 = 1 atm

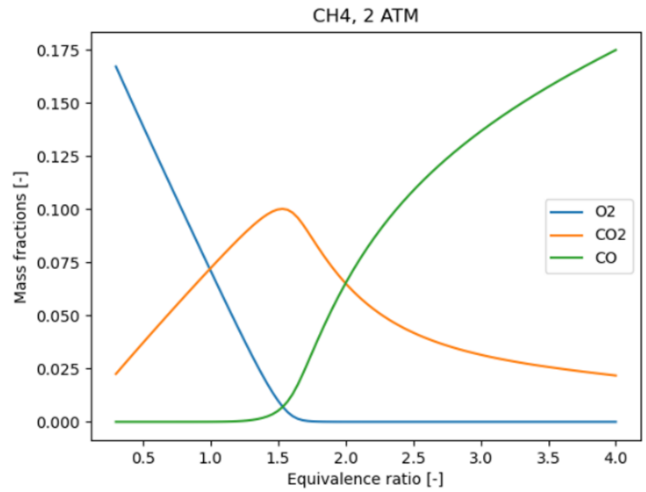

Figure 9: Methane exhaust components as a function of the equivalence ratio, T1 = 273 K, P2 = 2 atm

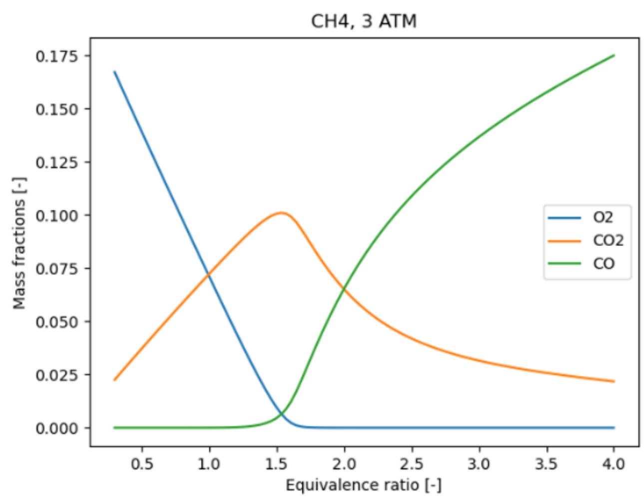

Figure 10: Methane exhaust components as a function of the equivalence ratio, T1 = 273 K, P3 = 3 atm

3.2 Ethane mixture

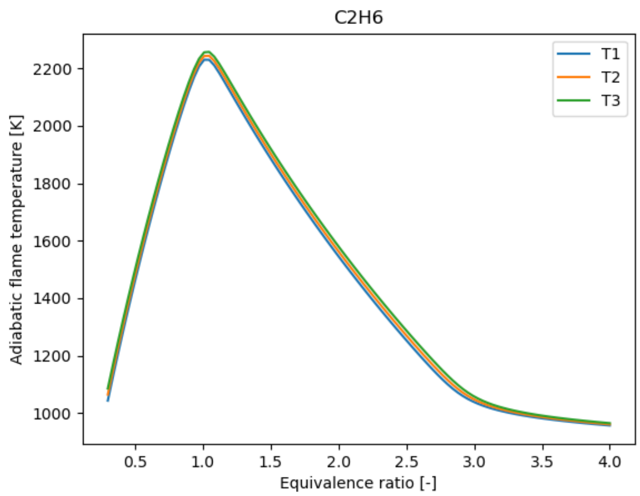

Figure 11: Ethane flame temperature as function of the equivalence ratio, P1 = 1 atm

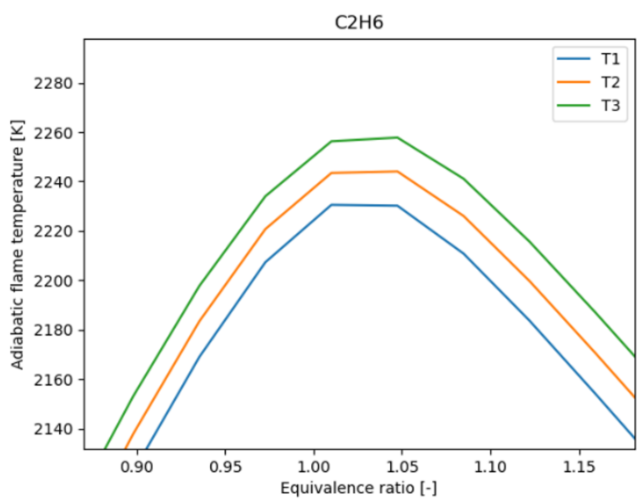

Figure 12: Ethane flame temperature as function of the equivalence ratio, P1 = 1 atm

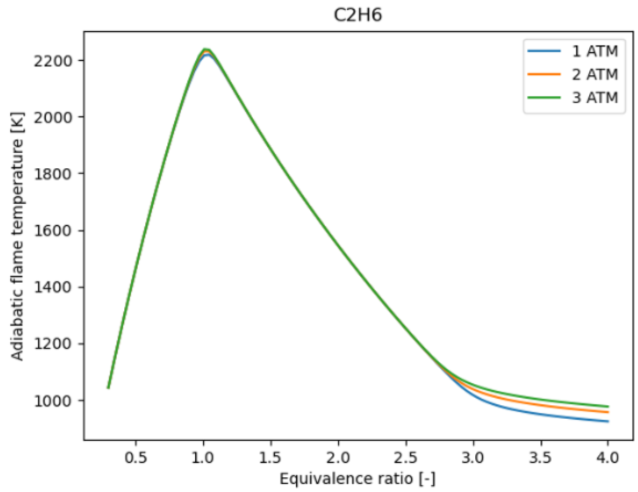

Figure 13: Ethane flame temperature as function of the equivalence ratio, T1 = 273 K

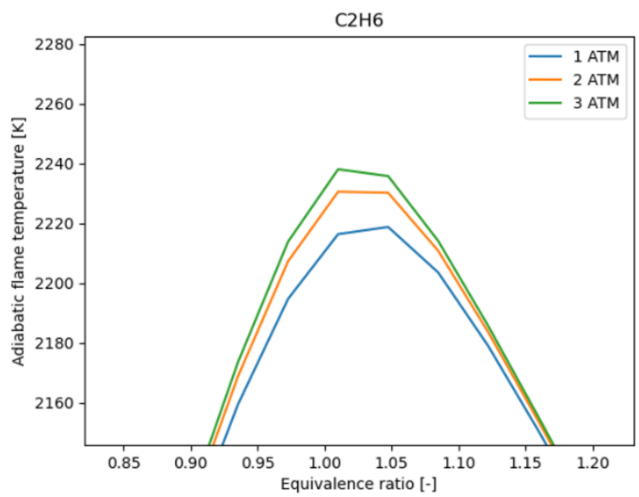

Figure 14: Ethane flame temperature as function of the equivalence ratio, T1 = 273 K

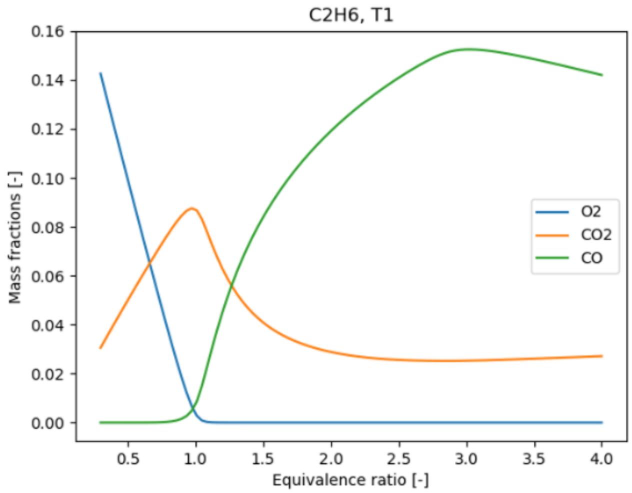

Figure 15: Ethane exhaust components as a function of the equivalence ratio, T1 = 273 K, P1 = 1 atm

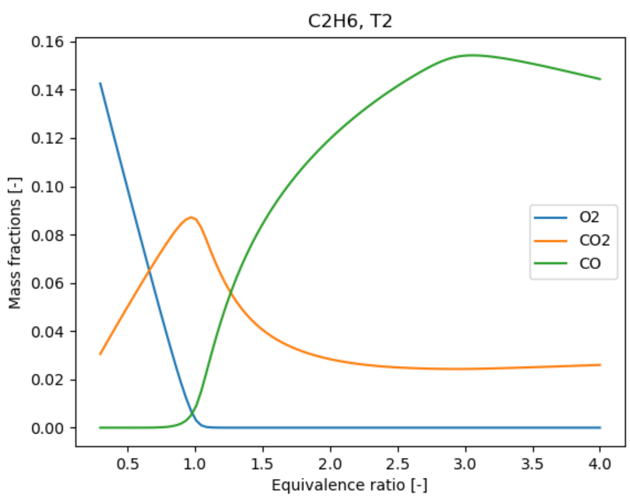

Figure 16: Ethane exhaust components as a function of the equivalence ratio, T2 = 298 K, P1 = 1 atm

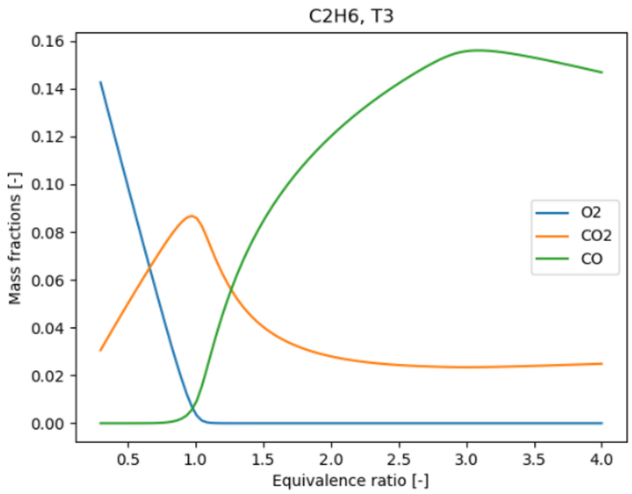

Figure 17: Ethane exhaust components as a function of the equivalence ratio, T3 = 323 K, P1 = 1 atm

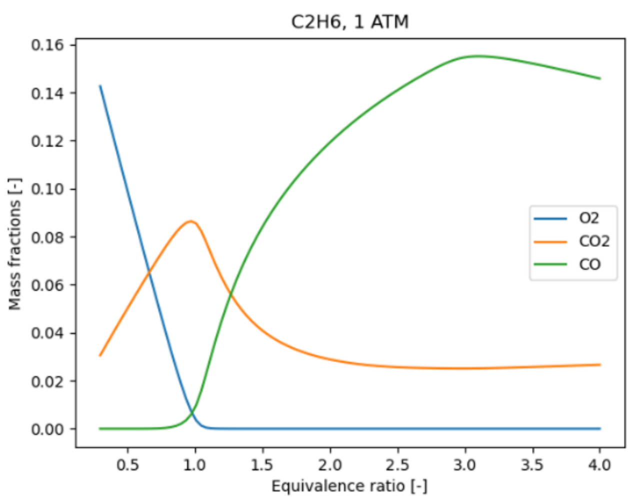

Figure 18: Ethane exhaust components as a function of the equivalence ratio, T1 = 273 K, P1 = 1 atm

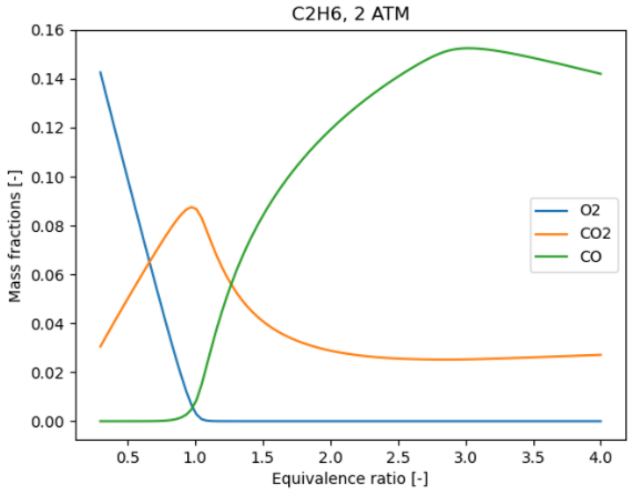

Figure 19: Ethane exhaust components as a function of the equivalence ratio, T1 = 273 K, P2 = 2 atm

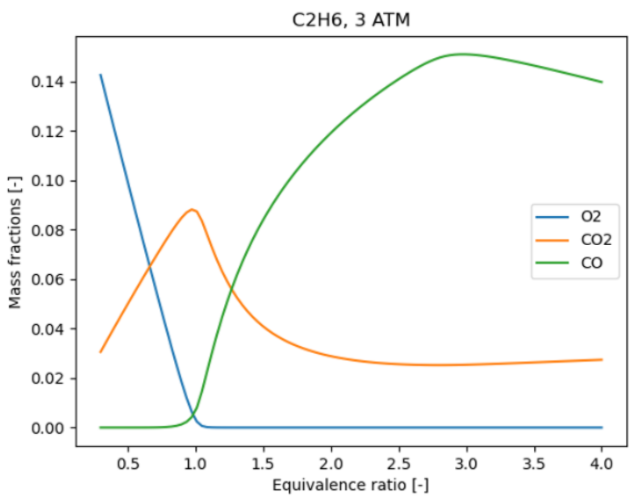

Figure 20: Ethane exhaust components as a function of the equivalence ratio, T1 = 273 K, P3 = 3 atm

4 Conclusions

- When the equivalence ratio is small (<< 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 CH4O/H2/NH3 Blend Combustion - Jingyun Sun, Qianqian Liu
- https://en.wikipedia.org/wiki/Methane
- https://en.wikipedia.org/wiki/Ethane

6 Python code

```python
import cantera as ct  # Cantera library for combustion simulations
import numpy as np  # Numerical Python library for mathematical operations
import matplotlib.pyplot as plt  # Library for creating plots

# Load the gas mixture from the gri30 database
gas = ct.Solution("gri30.yaml")

def calculate_adibatic_flame_temp(fuel_species, phi_min, phi_max, npoints, T, P):
    """
    Calculate adiabatic flame temperature and equilibrium composition for a given fuel.

    Parameters:
    fuel_species (str): The chemical formula of the fuel (e.g., 'C2H6' for ethane).
    phi_min (float): The minimum equivalence ratio.
    phi_max (float): The maximum equivalence ratio.
    npoints (int): The number of points to calculate between phi_min and phi_max.
    T (float): Initial temperature of the gas mixture.
    P (float): Initial pressure of the gas mixture.

    Returns:
    tuple: Arrays of equivalence ratios, adiabatic flame temperatures, and equilibrium compositions.
    """
    # Indices for the fuel, oxygen, and nitrogen species
    ifuel = gas.species_index(fuel_species)
    io2 = gas.species_index("O2")
    in2 = gas.species_index("N2")

    # Air composition: Nitrogen to Oxygen ratio
    air_N2_O2_ratio = 3.76

    # Calculate stoichiometric oxygen needed for the fuel
    if fuel_species == 'C2H6':
        stoich_O2 = 2 * gas.n_atoms(fuel_species, "C") + 0.25 * gas.n_atoms(fuel_species, "H")
    elif fuel_species == 'CH4':
        stoich_O2 = gas.n_atoms(fuel_species, "C") + 0.25 * gas.n_atoms(fuel_species, "H")

    # Create arrays to store equivalence ratios, temperatures, and compositions
    phi_values = np.linspace(phi_min, phi_max, npoints)
    tad = np.zeros(npoints)
    xeq = np.zeros((gas.n_species, npoints))

    # Loop over each equivalence ratio
    for i, phi in enumerate(phi_values):
        # Set the composition of the gas mixture
        X = np.zeros(gas.n_species)
        X[ifuel] = phi  # Fuel mole fraction
        X[io2] = stoich_O2  # Oxygen mole fraction
        X[in2] = stoich_O2 * air_N2_O2_ratio  # Nitrogen mole fraction

        # Set the gas state: temperature, pressure, and composition
        gas.TPX = T, P, X
        # Equilibrate the mixture adiabatically at constant pressure
        gas.equilibrate("HP")

        # Store the adiabatic flame temperature and equilibrium composition
        tad[i] = gas.T
        xeq[:, i] = gas.X
        print(f"At phi= {phi:.4f} Tad = {gas.T:.4f}")

    return phi_values, tad, xeq

def plot_species_mass_fractions(fuel_species, phi_values, xeq, species_names):
    """
    Plot the mass fractions of selected species against the equivalence ratio.

    Parameters:
    fuel_species (str): The chemical formula of the fuel.
    phi_values (array): Array of equivalence ratios.
    xeq (array): Array of equilibrium compositions.
```

```python
    species_names (list): List of species names.
    """
    # Select species to plot
    for species in ["O2", "CO2", "CO"]:
        if species in species_names:
            index = species_names.index(species)
            plt.plot(phi_values, xeq[index, :], label=species)

    plt.xlabel("Equivalence ratio [-]")
    plt.ylabel("Mass fractions [-]")
    plt.legend(loc="best")
    plt.title(fuel_species)
    plt.savefig(f'mass_fractions_{fuel_species}.png')
    plt.show()

def plot_adiabatic_flame_temp(fuel_species, phi_values, tad):
    """
    Plot the adiabatic flame temperature against the equivalence ratio.

    Parameters:
    fuel_species (str): The chemical formula of the fuel.
    phi_values (array): Array of equivalence ratios.
    tad (array): Array of adiabatic flame temperatures.
    """
    plt.plot(phi_values, tad)
    plt.xlabel("Equivalence ratio [-]")
    plt.ylabel("Adiabatic flame temperature [K]")
    plt.title(fuel_species)
    plt.savefig(f'flame_temp_{fuel_species}.png')
    plt.show()

def make_calc_for_ethane(T, P):
    """
    Perform calculations and plots for ethane (C2H6).
    """
    # Parameters for ethane
    fuel_species = "C2H6"
    phi_min = 0.3
    phi_max = 4
    npoints = 100

    # Calculate adiabatic flame temperature and equilibrium composition
    phi_values, tad, xeq = calculate_adibatic_flame_temp(fuel_species, phi_min, phi_max, npoints, T, P)
    return phi_values, tad, xeq

def make_calc_for_methane(T, P):
    """
    Perform calculations and plots for methane (CH4).
    """
    # Parameters for methane
    fuel_species = "CH4"
    phi_min = 0.3
    phi_max = 4
    npoints = 100

    # Calculate adiabatic flame temperature and equilibrium composition
    phi_values, tad, xeq = calculate_adibatic_flame_temp(fuel_species, phi_min, phi_max, npoints, T, P)
    return phi_values, tad, xeq

def make_adiabatic_plot_for_few_cases(phi_values, tad, fuel_species):
    for i in range(len(phi_values)):
        plt.plot(phi_values[i], tad[i])
    plt.xlabel("Equivalence ratio [-]")
    plt.ylabel("Adiabatic flame temperature [K]")
    plt.title(fuel_species)
    plt.legend(['T1', 'T2', "T3"])
    plt.savefig(f'flame_temp_{fuel_species}_P1.png')
    plt.show()

def make_species_plot_for_few_cases(phi_values, tad, fuel_species, species_names):
    for i in range(len(phi_values)):
```

```
        plt.figure()
        for species in ["O2", "CO2", "CO"]:
            if species in species_names:
                index = species_names.index(species)
                plt.plot(phi_values[i], xeq[i][index, :], label=species)
        plt.xlabel("Equivalence ratio [-]")
        plt.ylabel("Mass fractions [-]")
        plt.legend(loc="best")
        plt.title(f'{fuel_species}, T{i+1}')
        plt.savefig(f'mass_fractions_{fuel_species}_T{i}_P1.png')
        plt.show()

T_list = [273, 298, 323]
P_list = [101325, 202650, 303975]

phi_values = [0,0,0]
tad = [0,0,0]
xeq = [0,0,0]

# Perform calculations and plots for ethane
for i in range(len(P_list)):
    phi_values[i], tad[i], xeq[i] = make_calc_for_ethane(T_list[i],P_list[0])

make_adiabatic_plot_for_few_cases(phi_values, tad, "CH4")
make_species_plot_for_few_cases(phi_values, tad, "CH4", gas.species_names)

# Perform calculations and plots for methane
# for i in range(len(P_list)):
#     phi_values[i], tad[i], xeq[i] = make_calc_for_methane(T_list[i],P_list[1])

# make_adiabatic_plot_for_few_cases(phi_values, tad, "C2H6")
# make_species_plot_for_few_cases(phi_values, tad, "C2H6", gas.species_names)
```