

Computational Methods in Combustion
Zeldovich Mechanism



Project coordinator: Dr Eng. Mateusz Żbikowski
Submission date: 03/06/2025

Contents

1	Introduction	3
2	Literature Review	3
3	Model Description	4
4	Results and Discussion	5
5	Summary	7
A	Appendix: Python Simulation Code	8

Contents

1 Introduction

The formation of nitrogen oxides (NO_x) during combustion processes is a critical issue in internal combustion engines due to their negative environmental and health impacts. NO_x emissions contribute to the formation of photochemical smog and acid rain and are subject to increasingly strict regulatory limits. The present work aims to model the generation of NO and NO_2 during the combustion process inside an engine cylinder. The simulation is conducted using the Cantera software package, employing a zero-dimensional reactor model with a dynamically changing volume to mimic the behavior of a reciprocating engine.

This study focuses on methane as the primary fuel, a representative component of natural gas, and investigates the influence of the initial temperature on the formation of NO and NO_2 . The goal is to understand the temperature-dependent trends in NO_x emissions and to visualize them in terms of mole fractions and parts per million (ppm) concentrations.

2 Literature Review

The formation of NO_x in combustion processes has been extensively studied over the decades. One of the key mechanisms for NO formation in high-temperature environments is the thermal NO mechanism, commonly known as the Zeldovich mechanism. This pathway involves the oxidation of atmospheric nitrogen in the presence of high temperatures and oxygen radicals, with significant NO formation typically occurring at temperatures above 1800 K.

Various kinetic models and computational tools, such as CHEMKIN and Cantera, have been employed to study NO_x formation in both laminar and turbulent flames, as well as in homogeneous reactors. Studies have shown that the concentration of NO increases sharply with temperature due to the exponential sensitivity of the Zeldovich mechanism to thermal conditions. Moreover, the presence of diluents such as nitrogen (N_2) and changes in the equivalence ratio also play a significant role in NO_x formation.

In engine simulation contexts, zero-dimensional (0D) reactor models are commonly used to simplify the geometry and isolate the chemical kinetics from complex flow

phenomena. These models are especially useful for preliminary design and analysis due to their computational efficiency. Additionally, several studies have implemented variable-volume reactors to simulate piston motion, allowing for more realistic modeling of combustion events.

3 Model Description

The simulation model implemented in this project is based on a zero-dimensional representation of an internal combustion engine cylinder. The Cantera 3.1.0 library was used to simulate the chemical kinetics and thermodynamic behavior of the reacting gas mixture, and the GRI-Mech 3.0 mechanism was employed to model the combustion of methane.

The engine cycle was divided into three key stages: compression, combustion, and expansion. A sinusoidal function was used to vary the volume during the compression and expansion strokes, which mimics the piston motion in a real engine. During the combustion phase, the volume remains constant to approximate the ignition and flame propagation processes that occur at (effectively) constant volume due to their speed relative to piston motion.

The model assumes a homogeneous gas mixture of $\text{CH}_4:1$, $\text{O}_2:2$, $\text{N}_2:7.52$, corresponding to stoichiometric combustion in air. The cylinder has a maximum volume of $1\text{e-}3\text{ m}^3$ and a compression ratio of 18:1, leading to a minimum volume of approximately $5.56\text{e-}5\text{ m}^3$. The piston area was assumed to be 0.01 m^2 , and engine speed was set to 1800 RPM.

To study the impact of thermal conditions on NO_x formation, three initial temperatures were considered: 1500 K, 1800 K, and 2500 K. The initial pressure was fixed at 1 atm for all cases. The simulation was performed over a single engine cycle with a total duration of approximately 0.032 seconds.

A key assumption in the model is the neglect of heat losses to the cylinder walls and any mixing or flow phenomena, which would occur in a real engine. Additionally, the reactor chemistry was frozen after the expansion phase to simulate the quenching of reactions due to cooling in the exhaust.

The NO and NO_2 concentrations were tracked over time, both in mole fraction and ppm. The model captures only the gas-phase chemical kinetics and does not include surface reactions or emissions after exhaust gas treatment. Therefore, the predicted concentrations represent the raw engine-out emissions.

Initial Temperature [K]	Peak NO [ppm]	Peak NO ₂ [ppm]
1500	11469.16	3.05
1800	14010.24	3.63
2500	17920.67	4.00

Table 1: Peak NO and NO₂ concentrations at different initial temperature

4 Results and Discussion

The simulation results reveal the dynamic formation of NO and NO₂ during the combustion cycle. The primary observation is that the amount of NO produced is strongly dependent on the initial temperature. For higher initial temperatures, the mole fraction and ppm concentration of NO rise more sharply and reach higher peak values. This behavior is consistent with the thermal NO formation mechanism.

For example, at an initial temperature of 2500 K, NO levels reach peak concentrations significantly higher than those at 1500 K. In contrast, NO₂ concentrations remain much lower than those of NO across all tested temperatures, which is expected due to its slower formation kinetics and typically lower formation rates under high-temperature conditions.

The plotted results (see Figures 1 and 2) show the temporal evolution of NO and NO₂ concentrations. The graphs display cumulative values at each time step, indicating the total concentration present in the cylinder, not the incremental change per time step.

After reaching a peak, the concentration of NO and NO₂ begins to decline. This drop is attributed to the decrease in temperature during the expansion phase, which limits the continuation of endothermic NO-forming reactions. Additionally, at lower temperatures, NO and NO₂ are subject to recombination and reverse reactions that reduce their concentrations.

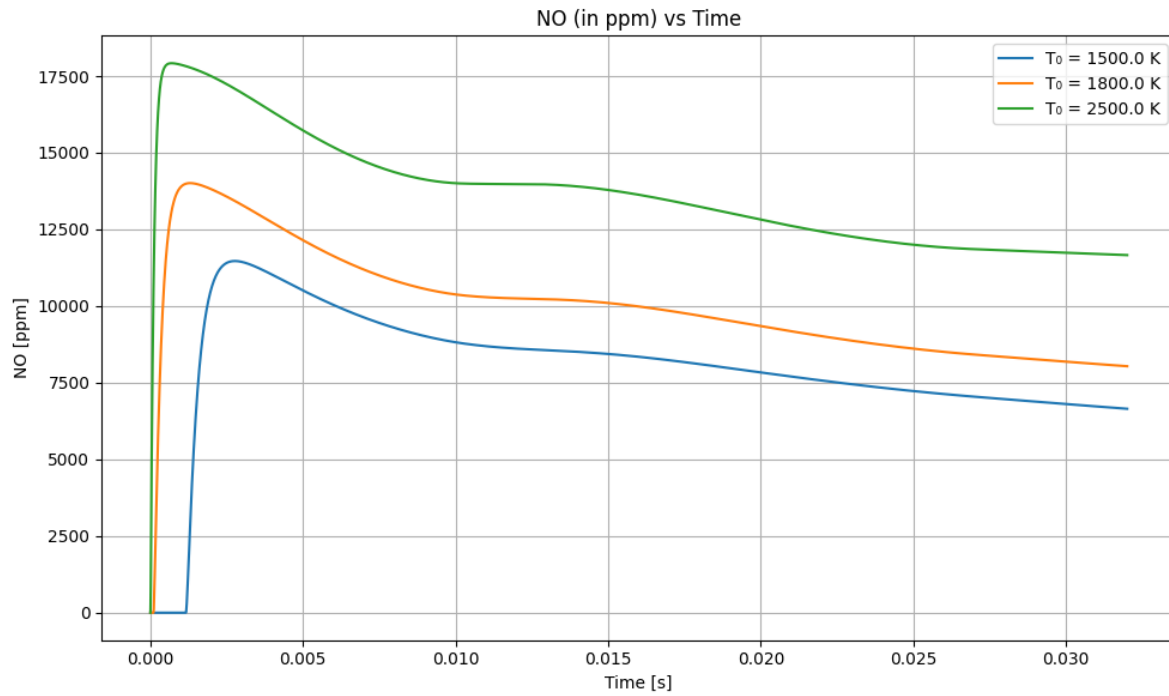


Figure 1: NO concentration in ppm over time for different initial temperatures.

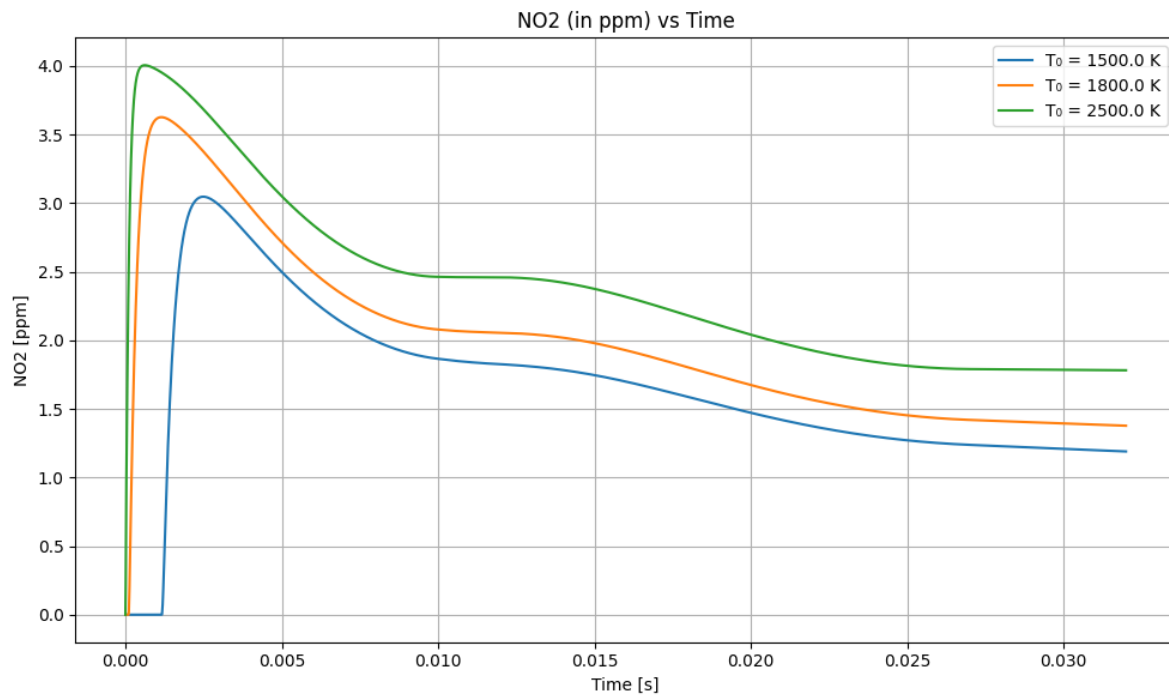


Figure 2: NO₂ concentration in ppm over time for different initial temperatures.

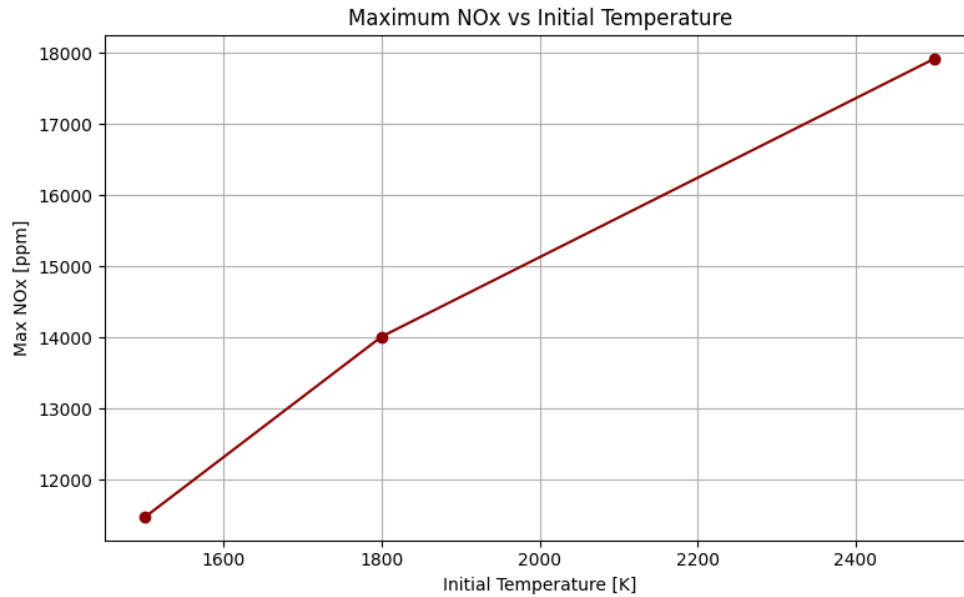


Figure 3: Maximum NO_x concentration in ppm vs. initial temperature.

5 Summary

This study successfully implemented a zero-dimensional combustion model using Cantera to simulate NO_x formation during a simplified engine cycle. Methane was used as a fuel, and different initial temperatures were evaluated. The results clearly demonstrate the sensitivity of NO formation to temperature, in line with established combustion theory.

The study shows that NO formation is dominant over NO_2 , particularly at high temperatures. These insights highlight the need for thermal management and combustion control strategies in engines to mitigate NO_x emissions.

Model Limitations

The model used in this study is subject to several simplifying assumptions:

- No heat loss to cylinder walls is included.
- Flow dynamics, turbulence, and inhomogeneities are neglected.
- The reaction mechanism does not include post-exhaust aftertreatment or particulate matter formation.

- Reactions are artificially frozen after the expansion phase.

Despite these limitations, the model provides valuable qualitative insight into temperature effects on NO_x formation.

Future work may include testing different fuels (e.g., n-heptane or n-dodecane), pressure variations, or equivalence ratios to explore broader emission characteristics.

A Appendix: Python Simulation Code

```

1  import cantera as ct
2  import numpy as np
3  import matplotlib.pyplot as plt
4
5  # ENGINE PARAMETERS
6  V_max = 1.0e-3 # Max volume [m^3]
7  compression_ratio = 18.0
8  V_min = V_max / compression_ratio
9  engine_speed = 1800 # RPM
10 cycle_duration = 60.0 / engine_speed # [s]
11 A = 0.01 # Piston area [m^2]
12
13 # COMBUSTION SETTINGS
14 mechanism = 'gri30.yaml'
15 composition = 'CH4:1,O2:2,N2:7.52'
16 P_initial = 1e5 # [Pa]
17 T_initial_values = [1500.0, 1800.0, 2500.0] # [K]
18
19 # VOLUME FUNCTIONS
20 def cylinder_volume(t, t_comp, t_burn, t_expand):
21     if t < t_comp:
22         return V_max - 0.5 * (V_max - V_min) * (1 - np.cos(np.pi * t / t_comp))
23     elif t < t_comp + t_burn:
24         return V_min
25     else:
26         t_rel = t - (t_comp + t_burn)
27         return V_min + 0.5 * (V_max - V_min) * (1 - np.cos(np.pi * t_rel / t_expand))
28
29 def volume_derivative(t, t_comp, t_burn, t_expand):
30     if t < t_comp:
31         return 0.5 * np.pi * (V_max - V_min) / t_comp * np.sin(np.pi * t / t_comp)
32     elif t < t_comp + t_burn:
33         return 0.0
34     else:
35         t_rel = t - (t_comp + t_burn)
36         return 0.5 * np.pi * (V_max - V_min) / t_expand * np.sin(np.pi * t_rel /
37                                t_expand)
38
39 # TIME SETTINGS
40 t_comp = 0.01 # 10 ms compression
41 t_burn = 0.002 # 2 ms combustion
42 t_expand = 0.015 # 15 ms expansion
43 freeze_time = t_comp + t_burn + t_expand

```



```

43 end_time = freeze_time + 0.005
44 dt = 1e-5 # Time step [s]
45
46 # STORAGE FOR RESULTS
47 results = []
48
49 for T_initial in T_initial_values:
50     gas = ct.Solution(mechanism)
51     gas.TPX = T_initial, P_initial, composition
52
53     reactor = ct.IdealGasReactor(gas, volume=V_max, energy='on')
54     env = ct.Reservoir(ct.Solution(mechanism))
55     wall = ct.Wall(reactor, env)
56     wall.area = A
57     wall.velocity = 0.0
58
59     sim = ct.ReactorNet([reactor])
60
61     times, T, P, NO, NO2, NO_ppm, NO2_ppm = [], [], [], [], [], [], []
62     t = 0.0
63
64     while t < end_time:
65         wall.velocity = volume_derivative(t, t_comp, t_burn, t_expand) / A
66
67         if t > freeze_time:
68             reactor.chemistry_enabled = False
69
70         sim.advance(t)
71
72         times.append(t)
73         T.append(reactor.T)
74         P.append(reactor.thermo.P)
75         no_x = reactor.thermo['NO'].X[0]
76         no2_x = reactor.thermo['NO2'].X[0]
77         NO.append(no_x)
78         NO2.append(no2_x)
79         NO_ppm.append(no_x * 1e6)
80         NO2_ppm.append(no2_x * 1e6)
81         t += dt
82
83     results.append({
84         'T_init': T_initial,
85         'time': times,
86         'T': T,
87         'P': P,
88         'NO': NO,
89         'NO2': NO2,
90         'NO_ppm': NO_ppm,
91         'NO2_ppm': NO2_ppm
92     })
93
94
95 # PLOTTING
96 plt.figure(figsize=(10, 6))
97 for res in results:
98     plt.plot(res['time'], res['NO_ppm'], label=f" T  $\hat{C}_{\text{NO}}$ ={res['T_init']}K")

```

```

99 plt.xlabel('Time[s]')
100 plt.ylabel('NO[ppm]')
101 plt.title('NO(in ppm) vs Time')
102 plt.legend()
103 plt.grid()
104 plt.tight_layout()
105 plt.show()
106
107 plt.figure(figsize=(10, 6))
108 for res in results:
109     plt.plot(res['time'], res['NO2_ppm'], label=f" T Ć = {res['T_init']}K")
110 plt.xlabel('Time[s]')
111 plt.ylabel('NO2[ppm]')
112 plt.title('NO2(in ppm) vs Time')
113 plt.legend()
114 plt.grid()
115 plt.tight_layout()
116 plt.show()
117
118 temps = [res['T_init'] for res in results]
119 max_nox = [max(np.array(res['NO_ppm']) + np.array(res['NO2_ppm'])) for res in results]
120
121 plt.figure(figsize=(8, 5))
122 plt.plot(temps, max_nox, 'o-', color='darkred')
123 plt.xlabel('Initial Temperature[K]')
124 plt.ylabel('Max NOx[ppm]')
125 plt.title('Maximum NOx vs Initial Temperature')
126 plt.grid()
127 plt.tight_layout()
128 plt.show()
129
130 print("Maximum NO and NO2 concentrations (in ppm) for each initial temperature:")
131 print("{:<15}{:<15}{:<15}".format("T_init[K]", "NO[ppm]", "NO2[ppm]"))
132 for res in results:
133     T_init = res['T_init']
134     max_NO = max(res['NO_ppm'])
135     max_NO2 = max(res['NO2_ppm'])
136     print("{:<15}{:<15.2f}{:<15.2f}".format(T_init, max_NO, max_NO2))

```

Listing 1: Python simulation code