Bartosz Hyży 313742, Michał Kret 313778

**Warsaw University of Technology**

**Faculty of Power and Aeronautical Engineering**

Computer Methods in Combustion

# Validation of Cantera based Propellant Analysis Tool by comparing results obtained with it for Methane-Oxygen and Methane-Nitrous dioxide propellants with NASA CEA results

Warsaw 06.06.2023

**Abstract**

In this article comparison between results obtained from NASA CEA and results from PAT - in-house developed Cantera based program in Matlab™environment were discussed. Two different propellants' characteristics were compared for mixtures with Methane as fuel and molecular oxygen or Nitrous dioxide as oxidizers for a range of oxidizer to fuel mass ratios. A detailed description of PAT software and difficulties connected with solid propellant combustion were discussed as well.

# Contents

# 1 Introduction

Students' Space Association has been developing Rocket Propulsion Analysis Tool (RPAT in short) for the last three years. Written in Matlab™environment it is applied to simulate process of grain regression for solid and hybrid rocket propulsion systems. It allows to accurately determine ballistic properties of rocket engine, however requires additional information regarding thermodynamic and kinetic properties of combustion products. Until now, they have been determined by the propellant producer and were supplied only for certain compositions. The goal of this project was to develop a tool allowing members of Students' Space Association to analyse their own propellant compositions and characteristics of engines without use of external software or other sources. Program was written with Cantera library in Matlab™environment to integrate it with RPAT.

However creating model for accurate simulation of combustion of commonly used HTPB - AP solid propellants has turned out to be much more time and resource consuming than it was expected. For now software only has capabilities to determine properties of liquid propellant mixtures. In this project it has been applied to compare methane based propellants with two oxidizers, molecular oxygen and Nitrous dioxide.

# 2 State of the Art

The new millennium is ushering in a renewed interest in methane as a liquid rocket propellant. There has been sporadic interest in this propellant since the early 1960s in the United States, however, no serious development activity or flight vehicle has used this relatively inexpensive and easily handled cryogenic fuel [1]. The main advantage of methane based propellants is it low toxicity both before and after combustion and wide range of compatible oxidizers, which are also non toxic and safe for storage. In this article LOX and NO2 were considered, as they are easy to obtain and store, while being sufficient for sounding rocket engines.

# 3 Program description

## 3.1 App interface summary

To ease the use of software for future users, an user interface app was developed in Matlab App Designer. [2] Below an image of the app window was included with description of its functions.

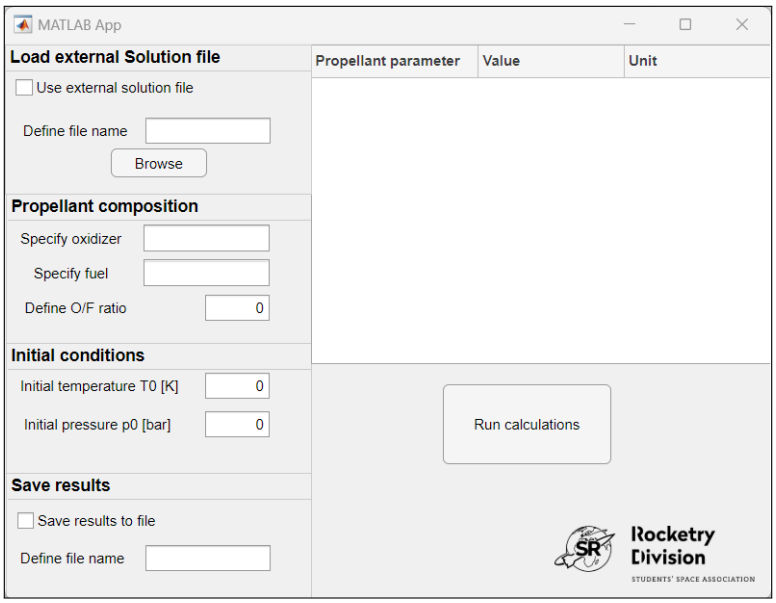


Figure 3.1: App main window

In "Propellant composition" panel oxidizer and fuel are defined by entering chemical formulas (for example CH4 or O2) and O/F ratio. In "Initial conditions" panel initial pressure and temperature values are defined, respectively in bar and in K. To save results to a file user must check the box and then define the name of the file, .txt file type is automatically chosen. "Load external Solution file" panel is for future uses with our own solution models, it hasn't been fully implemented yet. After pressing "Run calculations" app runs the script and presents results in table window, as well as writing them to file.

## 3.2 Summary of code

Program consists of two functions. PropellantComposition() defines composition of analyzed propellant and initial mixture conditions. EquilibratePropellant() finds state of equilibrium using equilibrate and transport function [3] and returns values of chosen parameters, valid from point of view of hybrid and solid propellants. Only parameters missing from ones required by RPAT are those that require empirical data such as regression rate curve coefficients and erosive burn coefficients. Code of both functions is presented below:

```
function propellant=PropellantComposition(oxid,fuel,of_ratio,T0,p0)
%PropellantComposition() - function used to define composition of
%propellant and initial parameters
%Michal Kret MKWS 2023
%Initializing data file and gas solution from GRI30 library
    propellant=Solution('gri30.yaml');
    n_sp=nSpecies(propellant); %Number of all the species in .yaml file
    m_mole=molecularWeights(propellant); %Array of molar weights [kg/
       kmol]
    x=zeros(n_sp,1); %Array of molar fractions
    i_fuel=speciesIndex(propellant,fuel); %Finding indices of fuel
       species
    i_oxid=speciesIndex(propellant,oxid); %Finding indices of oxidizer
       species
    m_mole_fuel=m_mole(i_fuel); %Assigning molar weight to fuel
    m_mole_oxid=m_mole(i_oxid); %Assigning molar weight to oxidizer
    x(i_fuel)=1; %Setting mole fraction to 1 for fuel
%Setting mole fraction of oxidizer so it is equal to mass of_ratio
    x(i_oxid)=of_ratio*m_mole_fuel/m_mole_oxid;
%Applying mass fractions and initial conditions to solution object
    set(propellant,'P',p0,'Temperature',T0,'MoleFractions',x);
end
```

```
function Propellant_parameters = EquilibratePropellant(propellant)
%Equilibrate propellant finds the state of propellant equilibrium with
%frozen enthalpy and  pressure, and returns specified parameters
   necessary
%Bartosz Hyzy 2023
    Propellant_parameters.names=zeros(9,1);
    Propellant_parameters.values=zeros(9,1);
    Propellant_parameters.units=zeros(9,1);
    Propellant_parameters.values(1)=density(propellant); %Returns
       density of propellant before combustion [kg/m3]
    transport=Transport(propellant,'default');
    equilibrate(propellant,'HP'); %Find state of equilibrium of
       propellant
    Propellant_parameters.values(2)=temperature(propellant); %Return
       reached temperature [K]
    Propellant_parameters.values(3)=cp_mass(propellant)/cv_mass(
       propellant); %Returns adiabatic constant
    Propellant_parameters.values(4)=cp_mass(propellant)-cv_mass(
       propellant); %Returns gas constant
    Propellant_parameters.values(5)=meanMolecularWeight(propellant); %
       Returns molar weight of combustion products
    Propellant_parameters.values(6)=soundspeed(propellant); %Returns
       sound speed in combustion products
    Propellant_parameters.values(7)=viscosity(transport); %Dynamic
       viscosity of combustion products
    Propellant_parameters.values(8)=thermalConductivity(transport); %
       Thermal conductivity coefficient of exhaust gases
    Propellant_parameters.values(9)=cp_mass(propellant)*
       Propellant_parameters.values(7)/Propellant_parameters.values(8);
      %Prandtl number of combustion products
    Propellant_parameters.names={'rho';'T';'kappa';'R';'m_mol';'a';'mu';
       'alpha';'Pr'}; %Defining names for parameters in table
    Propellant_parameters.units={'kg/m3';'K';'';'J/kg/K';'kg/kmol';'m/s'
       ;'Pa*s';'W/m/K';''}; %Defining units for parameters in table
    Propellant_parameters=struct2table(Propellant_parameters); %Transfer
        from struct to table
end
```

# 4 Program validation

In order to validate calculations performed by PAT, the program output was then compared with values obtained from NASA Chemical Equilibrium with Applications program (NASA CEA), which was commonly used in Students' Space Association to estimate thermodynamic characteristics of combustion products. Similarly to problem definition in PAT, here the HP solver will be also used. As needed parameters strongly depends on oxidizer- fuel ratio, subsequent simulations will be carried out for successive values of this parameter. As previously mentioned, the simulations will be performed

using methane as the fuel, $N_2O$ and $O_2$ will be used as oxidizers. Both cases will be calculated for the following O/F ratios: 2, 2.5, 3, 3.5 and 4. For the comparison between outputs, typical parameters needed to calculate inner ballistics of rocket engine, namely: temperature, kappa and molar mass obtained from each case will be presented below. It is important to mention that outputs may differ due to different input values for used substances defined in both CEA and Cantera libraries. Input files for CEA calculations was defined as following:

```
prob
o/f=2,2.5,3,3.5,4 hp p,bar=1 t,k=2000
react
  fuel=CH4 wt=1.0 t,k=300
  oxid=O2 wt=1.0 t,k=300
output
     siunits
    plot t mw gam
end
```

```
prob
o/f=2,2.5,2.875,3,3.5 hp p,bar=1 t,k=2000
react
  fuel=CH4 wt=1.0 t,k=300
  oxid=NO2 wt=1.0 t,k=300
output
     siunits
    plot t mw gam
end
```

Below, mentioned comparisons are presented.

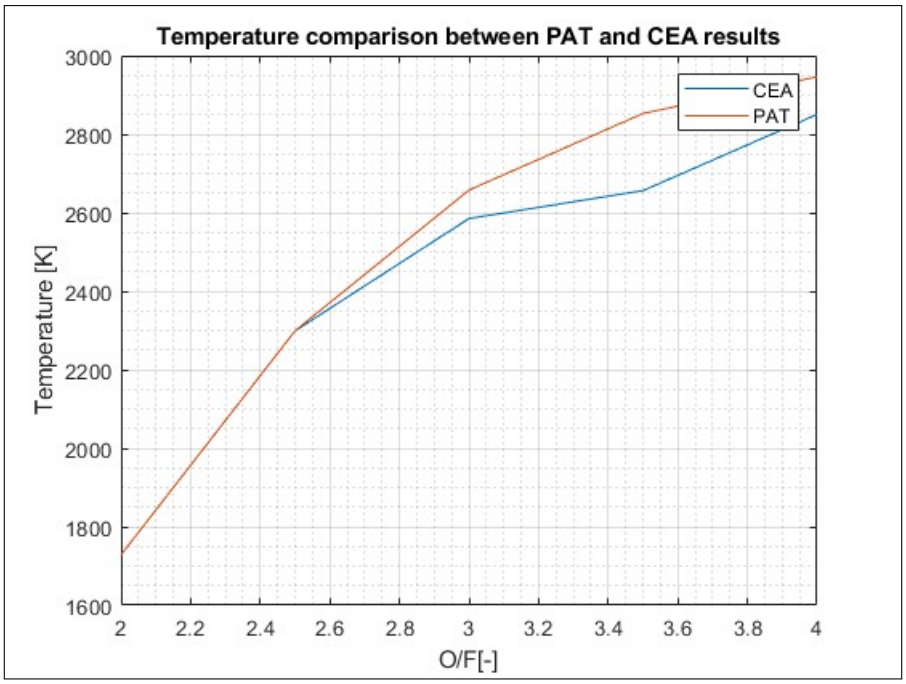


Figure 4.1: Comparison of CH4-NO2 temperature

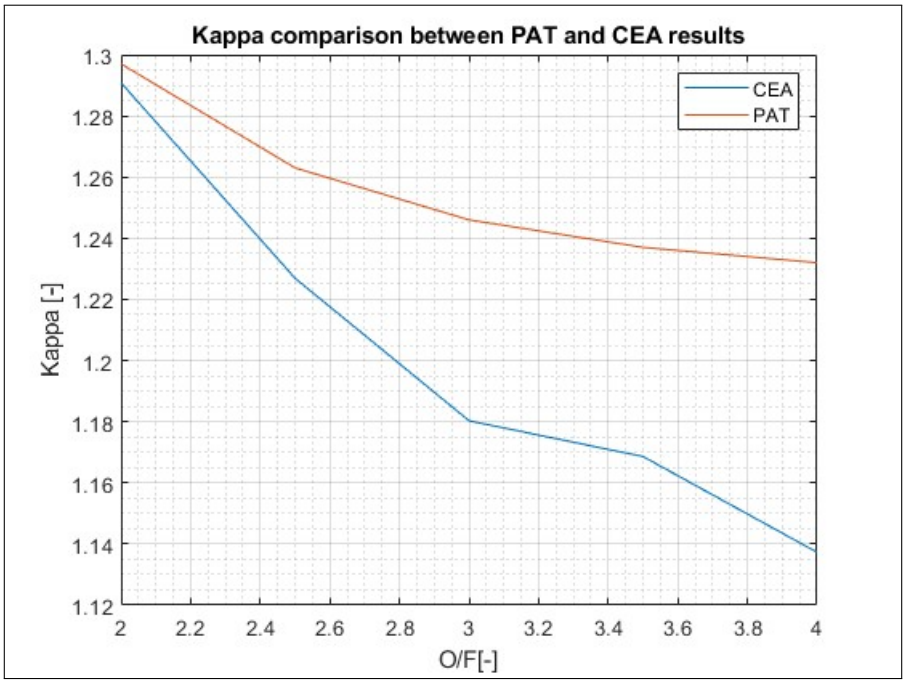


Figure 4.2: Comparison of CH4-NO2 kappa

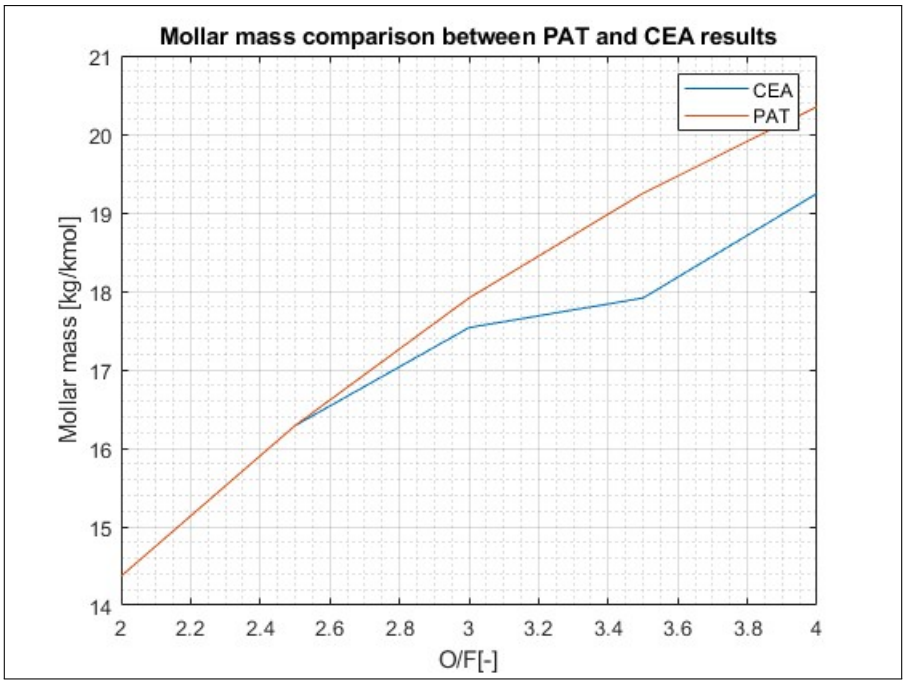


Figure 4.3: Comparison of CH4-NO2 mollar mass

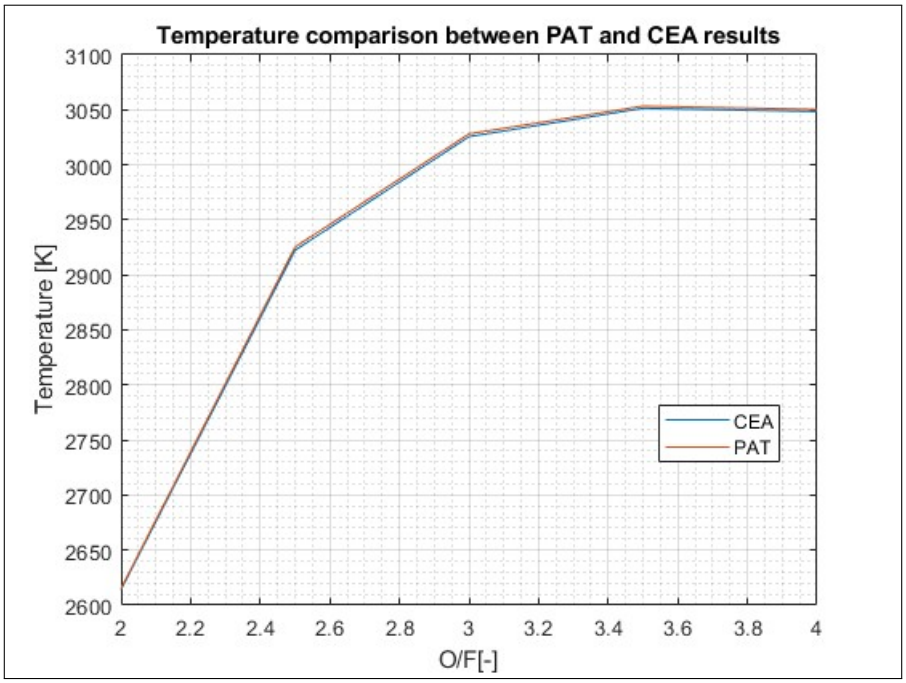


Figure 4.4: Comparison of CH4-O2 temperature

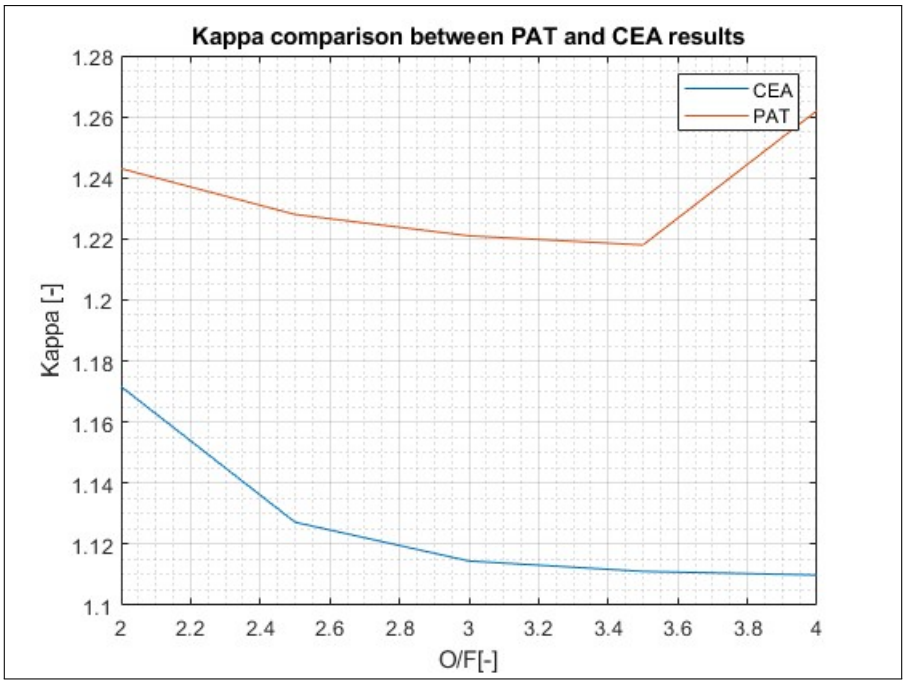


Figure 4.5: Comparison of CH4-O2 kappa

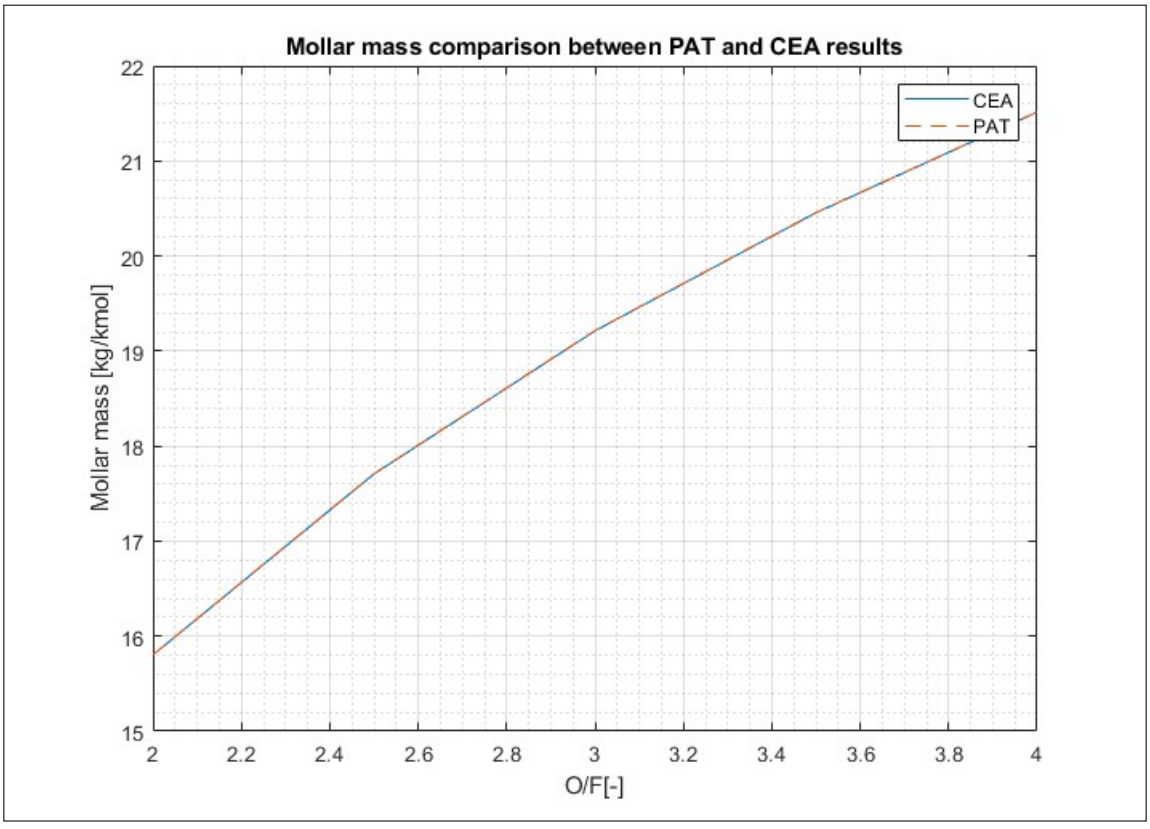


Figure 4.6: Comparison of CH4-O2 molar mass

The results of the program vary noticeably for some o/f ratios when simulated with NO2. Some discontinuity in the molar mass and temperature dependence of the ratio values can be observed here. This is probably due to the fact that additional effects are modelled by the more extensive

NASA CEA library, although the results obtained from the PAT program cannot be completely discarded here, as the differences are negligible and localised, and the nature of the curve appears to be even more natural than that obtained from NASA CEA. As for the results of temperature and molar mass calculations for the case of using O2 as an oxidant, the solutions are identical. It can therefore be concluded that the definitions of the reaction mechanism with oxygen coincide in both libraries in the case of simpler O2, while in the case of NO2, with which more complex compounds are obtained and a large number of reactions can occur, the definitions are no longer so compatible. In both cases considered, however, a discrepancy in the results of the adiabatic exponent can be observed. This difference is due to the way the specific heat of reaction is defined at constant pressure. Here, the value of this parameter is determined assuming a constant, 'frozen' mixture[4]. This assumption results in the neglect of the variable component of the Cp value, which in turn results in an error in the determination of the gamma value. This error is well known and there are some rather elaborate solutions that will serve to further develop the program.

# 5 Conclusions

Validation shows that PAT can accurately determine characteristics of propellants, consisting of simple chemical compounds. For more complex chemical reactions differences occur between NASA CEA and Cantera, caused by different models and input values. The biggest differences are obtained for adiabatic coefficient calculation.

Analyzing presented charts it can be concluded that temperature reached during combustion of CH4-LOX is higher than temperature of CH4-NO2 mixtures for the same O/F ratios. Since exit velocity is directly proportional to temperature it can be deducted that it will be higher for CH4-LOX mixtures, resulting in higher thrust and specific impulse. However temperatures this high cause problems connected with mechanical design of engine components.

PAT has proven to be an accurate tool to determine characteristics of propellants however only limited for liquid ones for now. In the future a study will be carried out to model processes occurring during solid and hybrid propellants combustion to broaden program capabilities. Other functions may be added as well, such as multiple simulations at once, interpolation of experimental data and further integration with RPAT.

# References


[1] Todd Neill et al. "Practical uses of liquid methane in rocket engine applications". In: *Acta Astronautica* 65.5 (2009), pp. 696–705. ISSN: 0094-5765. DOI: https://doi.org/10.1016/j.actaastro.2009.01.052. URL: https://www.sciencedirect.com/science/article/pii/S0094576509000630.

[2] *Matlab documentation*. https://www.mathworks.com/help/matlab. Accessed: 2023-06-02.

[3] *Cantera documentation*. https://cantera.org/documentation/index.html. Accessed: 2023-06-02.

[4] *Implementing CEA calculations using Cantera*. https://kyleniemeyer.github.io/rocket-propulsion/thermochemistry/cea_cantera.html. Accessed: 2023-06-05.