COMPLETE DOCUMENTATION

ODEKO

ONE DIMENSIONAL EQUILIBRIUM, KINETICS AND OPTIMIZATIONS

RELEASE v1.0

Author:
Nicolás de Jong Cantariño

November 12, 2023

Acknowledgements

Firstly, the author would like to thank Juan Manuel Tizón Pulido for his role as bachelor thesis supervisor during the initial development of ODEKO and his continued help after the end of the bachelor thesis.

The author, as well as Juan Manuel Tizón Pulido would like to thank:

- MEC & UPM for the propulsion department collaboration scholarship which I benefited from and in which I worked on aspects of ODEKO.
- INTA & Jesús Sánchez, director of Millistic and Optoelectronic department for their material and non-material support to the department of propulsion of ETSIAE.

Furthermore, publicly available libraries and tools used internally by ODEKO should also be noted:

- Qt - Qt is a cross-platform C++ framework for GUI design. It simplifies development with a widget toolkit and signals and slots for communication. It's open source with both open-source and commercial licensing options, making it a versatile choice for efficient and polished application development across different platforms. The ODEKO GUI was created using Qt.
- QCustomPlot - QCustomPlot is a C++ library for Qt that enables easy creation of customizable and interactive plots and charts in Qt applications. It provides a simple interface for developers to generate high-quality plots with various options for customization, making it a valuable tool for visualizing data in Qt-based projects. All plots present in the ODEKO GUI employ this library.
- CVodes - CVodes is a numerical integration library for solving ordinary differential equations (ODEs) and differential-algebraic equations (DAEs). It is part of the SUNDIALS (Suite of Nonlinear and Differential/Algebraic Equation Solvers) package and is widely used in scientific and engineering applications for simulating dynamic systems. CVodes is designed to be efficient and scalable, making it suitable for a variety of numerical simulations. ODEKO employs CVodes for the integration of the differential equations required to compute the ODK model.
- Eigen - Eigen is a C++ template library for linear algebra that provides high-performance, versatile, and easy-to-use matrix and vector operations. It is header-only, meaning there is no need for separate compilation, making it convenient to integrate into projects. Eigen is widely used in scientific computing, machine learning, and computer graphics for its efficiency and expressive syntax. ODEKO employs Eigen for the resolution of linear equation systems, required to compute chemical equilibrium.

Abstract

With an ever more prevalent interest in the space industry, the design and production of rocket engines has become as important as it ever was. Therefore, simulations that accurately represent the behaviour of the rocket are of the uttermost importance. ODEKO presents an open-access free simulation software that allows the user to quickly obtain data on 1D simulations of the internal flow of a rocket engine nozzle.

The aim of this document is not only to provide a users guide to the self-developed application ODEKO; but also present the theoretical background of the application, which allows the user to have an improved understanding of the simulations performed and, consequently, make better use of the tool. Moreover, it also shows a series of examples and results of the application, in order to help the user better understand its output and the conclusions that may be extracted from it.

ODEKO allows the user to define their own propellant, specifying the species and reactions that will be considered in the calculations. The nozzle is also defined in-application and, optionally, a grain geometry may also be specified if solid propellants are going to be studied. The application offers three main calculation models: chemical equilibrium (infinite reaction speed), frozen flow (non-reacting) or kinetics (where the reaction rates of the species are considered). Moreover, the program allows the user to optimize the nozzle for a given grain geometry and propellant.

The program is aimed to researchers that want to conduct early approximations, teachers that want to show their students simple cases that they can compare with analytical models and to amateur rocket enthusiasts that are starting to develop their own engines and require a means to predict their capabilities.

The software and future updates of it, as well as its documentation can be found at the ODEKO website and the author can be contacted at odeko@nickdejongc.com.

Contents

List of Figures

List of Tables

List of Symbols

A	Area of the nozzle sections $[m^2]$
$\mathcal{A}$	Auxiliary variable for ODK system.
A_b	Combustion area for the grain $[m^2]$.
A_c	Section area of the cylindrical combustion chamber $[m^2]$.
A_i	Arrhenius A constant for reaction i $[Combination\ of : mol, cm, s, K]$.
A_{min}	Minimum value for the area in the distribution calculated by ODK.
A_t	Throat area $[m^2]$.
a	Speed of sound (frozen) $[\frac{m}{s}]$.
a_e	Speed of sound (equilibrium) $[\frac{m}{s}]$.
a_{ij}	Number of kg-atoms of element i per kg-mol of specie j.
a_v	Constant a in Vielle's law $[\frac{m}{s}]$.
$\mathcal{B}$	Auxiliary variable for ODK system.
b_i	Number of kg-atoms of element i per kg of mixture.
b_i^o	Number of kg-atoms of element i per kg of propellant.
$\mathcal{C}$	Auxiliary variable for ODK system.
C_E	Thrust coefficient $[-]$.
C_p^o	Molar standard-state heat capacity at constant pressure $[\frac{J}{kg-molK}]$.
$C_{p,e}$	Equilibrium specific heat capacity at constant pressure $[\frac{J}{kg-molK}]$.
$C_{p,f}$	Frozen heat capacity at constant pressure $[\frac{J}{kg-molK}]$.
$C_{p,j}$	Molar heat capacity at constant pressure for specie j $[\frac{J}{kg-molK}]$.
$C_{p,r}$	Reacting specific heat capacity of mixture at constant pressure $[\frac{J}{kg-molK}]$.
$C_{v,e}$	Equilibrium specific heat capacity of mixture at constant volume $[\frac{J}{kg-molK}]$.

$C_{v,f}$	Frozen specific heat capacity of mixture at constant volume $[\frac{J}{kg-molK}]$.
$c*$	Characteristic velocity of propellant $[\frac{m}{s}]$.
c	Constant in Bartz equation (4.8) $[-]$.
D	Diameter at a given nozzle point $[m]$.
D_t	Diameter at the throat of the nozzle $[m]$.
E	Thrust $[N]$.
E_i	Activation energy for reaction i $[\frac{cal}{molK}]$.
g	Gibbs free energy $[\frac{J}{kg}]$.
g_j^o	Standard state gibbs free energy of specie j $[\frac{J}{kg-mol}]$.
H_j^o	Molar standard-state enthalpy of specie j $[\frac{J}{kg-mol}]$.
H_j	Molar enthalpy of specie j $[\frac{J}{kg-mol}]$.
h	Specific enthalpy of mixture $[\frac{J}{kg}]$.
h_0	Specific enthalpy of reactants $[\frac{J}{kg}]$.
ΔH_i^o	Net molar standard-state enthalpy production in reaction i $[\frac{J}{kg-mol}]$.
h_g	Convective heat transfer coefficient $[\frac{J}{m^2 \cdot s \cdot K}]$.
$\mathcal{I}$	Parameter to differentiate if the independent variable is the pressure or the area in the ODK problem.
I_{sp}	Specific impulse $[\frac{m}{s}]$.
$K_{c,i}$	Equilibrium constant of reaction i (concentration).
$K_{f,i}$	Forward rate constant of reaction i.
$K_{p,i}$	Equilibrium constant of reaction i (pressure).
$K_{r,i}$	Reverse rate constant of reaction i.
L_c	Length of the propellant charge $[m]$.
$\mathcal{M}$	Molecular mass of the mixture $[\frac{kg}{kg-mol}]$.
$\mathcal{M}_j$	Molecular mass of specie j $[\frac{kg}{kg-mol}]$.
$[M]$	Molar concentration of the mixture $[\frac{mol}{cm^3}]$.
M	Mach number $[-]$.

$\dot{m}$	Mass flow $[\frac{kg}{s}]$.
m	Exponent for Reynolds number in Bartz equation (4.8) $[-]$.
Nu	Nusselt number.
n	kg-mol of mixture per kg of mixture $[\frac{kg-mol}{kg}]$.
n	Exponent for Prandtl number in Bartz equation (4.8) $[-]$.
n_c	Number of charges of solid propellant.
n_j	kg-mol of specie j per kg of mixture $[\frac{kg-mol}{kg}]$.
n_v	Constant n in Vielle's law $[-]$.
Pr	Prandtl number.
P	Pressure of the mixture $[Pa]$.
P_{inj}	Pressure at the injection point of the combustion chamber $[Pa]$.
p_s	Number of points in star of solid propellant charge.
q_i	Rate of reaction of reaction i.
Q_1	Mass source term $[\frac{mol-kg}{m^3 s}]$.
Q_2	Momentum source term $[\frac{mol-kg}{m^3 s}]$.
Q_3	Energy source term $[\frac{mol-kg}{m^3 s}]$.
Q_j	Specie mass source term $[\frac{mol-kg}{m^3 s}]$.
q_1	Reduced mass source term $[\frac{mol-kg}{m^3 s}]$.
q_2	Reduced momentum source term $[\frac{mol-kg}{m^3 s}]$.
q_3	Reduced energy source term $[\frac{mol-kg}{m^3 s}]$.
q_j	Reduced specie mass source term $[\frac{mol-kg}{m^3 s}]$.
Re	Reynolds number.
R	Universal gas constant 8314.51 $[\frac{J}{kg-mol K}]$.
R_c	Universal gas constant 1.9872 $[\frac{cal}{mol K}]$.
R_g	Gas constant of mixture $[\frac{J}{kg K}]$.
R_j	Gas constant of specie j $[\frac{J}{kg K}]$.
R_m	Universal gas constant 8.31451 $[\frac{J}{mol K}]$.

$R_{e,c}$	Initial exterior radius of the propellant charge $[m]$.
$R_{i,c}$	Initial interior radius of the propellant charge $[m]$.
$R_{p,c}$	Initial radius at the star points of the propellant charge $[m]$.
S_j^o	Molar standard-state entropy of specie j $[\frac{J}{kg-molK}]$.
S_j	Molar entropy of specie j $[\frac{J}{kg-molK}]$.
s	Specific entropy of mixture $[\frac{J}{kgK}]$.
s_0	Specific entropy of reactants $[\frac{J}{kgK}]$.
ΔS_i^o	Net molar standard-state entropy production in reaction i $[\frac{J}{kg-molK}]$.
T	Temperature of the mixture $[K]$.
t_b	Burn time. $[s]$
u	Fluid velocity $[m/s]$.
v	Velocity of the fluid $[\frac{m}{s}]$.
$[X_k]$	Molar concentration of species k $[\frac{mol}{cm^3}]$.
x_j	Molar fraction of specie j $[-]$
x	Longitudinal coordinate of nozzle, x=0 represents the throat $[m]$.
$\mathcal{Y}(x)$	Known function respect to the variable x.
y	Combustion advance parameter $[m]$.
y_f	Value of the advance parameter at which the combustion ends $[m]$.
z	Compressibility factor in equation of state $[-]$.

Greek letters

α_{ki}	efficiency of the k body third body in reaction i.
β_i	Arrhenius β constant for reaction i $[-]$.
γ	Frozen value for gamma $[-]$.
γ_s	Equilibrium value for gamma $[-]$.
ε	Correction factor for chemical equilibrium convergence.
κ	Curvature radius at a given nozzle point $[m]$.
λ	Conductivity of mixture $[\frac{\mu W}{cmK}]$.

λ_j	Conductivity of specie j $[\frac{\mu W}{cmK}]$.
λ_i	Lagrange multiplier.
μ	Viscosity of mixture $[\mu Poisson]$.
μ_j	Viscosity of specie j.
$\mu_i j$	Viscosity interaction parameter between species i and j.
ξ_j^o	Molar standard-state chemical potential of specie j $[\frac{J}{kg-mol}]$.
ξ_j	Molar chemical potential of specie j $[\frac{J}{kg-mol}]$.
π_i	Reduced lagrangian multiplier.
ρ	Density of the mixture $[\frac{kg}{m^3}]$.
ρ_p	Density of the solid propellant $[\frac{kg}{m^3}]$.
σ	Criteria for near-throat pressure convergence.
υ''_{ki}	Products stoichiometric coefficient of specie k in reaction i.
υ'_{ki}	Reactants stoichiometric coefficient of specie k in reaction i.
υ_{ki}	Net stoichiometric coefficient of specie k in reaction i.
ϕ_{ij}	Viscosity interaction coefficient.
χ_k	Chemical symbol of specie.
ψ_{ij}	Conductivity interaction coefficient.
$\dot{\omega_k}$	Reaction rate of specie k $[\frac{mol-kg}{m^3 s}]$
ω_μ	Temperature exponent for viscosity in boundary layer.
ω_λ	Temperature exponent for conductivity in boundary layer.
ω_c	Temperature exponent for C_p in boundary layer.
ω_κ	Exponent for curvature correction term.

Sub-indexes

am	Intermediate variable between the static value and the value at the nozzle wall.
c	Variable at the exit of the combustion chamber, entrance of the nozzle.
e	Equilibrium variable.
f	Frozen variable.

r	Reacting variable.
t	Variable at the throat of the nozzle.
w	Value at the wall
	Super-indexes
(k)	Value at the k^{th} iteration.
o	Standard state (1bar).

List of Abbreviations

AP	Ammonium perchlorate.
FAC	Finite Area Combustor.
GUI	Graphical User Interface.
HTPB	Hydroxyl-terminated polybutadiene.
IAC	Infinite Area Combustor.
LOX	Liquid Oxygen.
NE	Number of chemical elements present.
NC	Nitrocellulose.
NG	Number of gaseous species in the mixture.
NG	Nitroglycerin.
NK	Number of chemical reactions considered during calculations.
NR	Number of reactants in propellant.
NS	Number of species in the mixture.
NT	Number of species in the mixture with transport properties in the database.
ODE	One Dimensional Equilibrium. It refers to a flow which is constantly in chemical equilibrium, its reaction time is considered as zero.
ODK	One Dimensional Kinetics. It refers to a flow which calculates its composition considering the time required for reactions to occur.

1

Introduction

Contents

1.1 Objective

The ODEKO project aims to provide an user-friendly graphical interface for one dimensional simulations of the internal flow of a rocket engine's nozzle. This will allow users to easily simulate the desired configurations without the need of extensive knowledge on the topic. Moreover, it will provide tools to compare the effect of various nozzles, propellants or computation methods, allowing for a more precise study for the more experienced user. Furthermore, it also allows the user to perform calculations employing solid grains, as well as optimizing nozzles for them.

Even though it intends to be easy to use, it also offers a wide variety of simulation options. Moreover, by giving the user the freedom to modify almost all parameters but setting appropriate default values for most of them, ODEKO aims to fit a wide variety of users with different levels of expertise and requirements.

1.2 Motivation

Although ODEKO is not the first software to compute nozzle properties through chemical equilibrium, it aims to offer a novel approach to this type of software. The intention was to create a software with a much larger scope.

Firstly, due to my own work in developing solid rocket engines (for amateur high-power rocketry), I wanted the software to have capabilities of simulating the time evolution of the properties of solid rocket motors for a given grain geometry. Furthermore, to help in the design of the nozzle for said engines I also envisioned an optimization tool embedded in the software.

Furthermore, the ODEKO software was started as by bachelor's thesis project, with one of the objectives of my supervisor (Juan Manuel Tizón Pulido) being the simulation of the internal flow through chemical kinetics. Consequently, another objective was to have the capability to simulate through chemical equilibrium, frozen flow and/or chemical kinetics, as well as compare the three approaches.

Moreover, existing software felt limited in many aspects. In consequence, my objective was to develop a versatile application, which would allow the advances user to modify any parameter and have a very detailed control on all aspects of the software. Nevertheless, the intention was to simultaneously develop and accessible GUI and set of pre-defined parameters that would also allow the inexperienced user to take advantage of ODEKO's capabilities.

1.3 Development

The ODEKO software was entirely developed by Nicolás de Jong. The project started as a bachelor's thesis under the supervision of Juan Manuel Tizón Pulido for the degree of Aerospace Engineering at Madrid's Technic University (UPM - ETSIAE). After the end of the bachelor's thesis, the development was continued. The intention is to keep improving the software and bringing new updates with bug fixes and new features.

Nevertheless, as the ODEKO software is a side-project and not my main occupation updates will not be very frequent, although it will be attempted to fix the more critical bugs that might appear as quickly as possible.

New versions of the software and the documentation, as well as other updates and information can be found at the ODEKO website and the author can be contacted at odeko@nickdejongc.com.

1.4 Contents

The program consists of 4 main tabs:

- **Propellant Tab:** Allows the user to fully define a propellant and save it to the database. Moreover it also offers the posibility to select which species/reactions are considered and it shows the evolution of c* with pressure.
- **Grain geometry Tab:** Allows the user to define a combustion geometry. It also provides a graph of combustion area relative to advance parameter. Finally, it offers some degree of optimization for certain geometries.
- **Nozzle Tab:** Allows the user to define different nozzle geometries. It also allows to optimize the geometry of the nozzle for a given propellant and combustion geometry (requires Vielle's parameters).
- **Calculations Tab:** It is in charge of performing and displaying the various types of available calculations, such as chemical equilibrium (ODE) or chemical kinetics (ODK).

1.5 User target

ODEKO has three main user targets:

- **Researchers:** In this case the program would be used in an early design stage. It offers very precise results given its execution time and it serves as a means to quickly discard a bulk of configurations, leaving only a few for a more intricate (and time consuming) simulation. Moreover, it allows the user to configure every last bit of the program, which is extremely helpful in an early stage of the design process. The possibility to hand-select which species are involved allows to easily compare their effects on the final properties and it serves of great use for research purposes.
- **Students:** A simplified version of the program can be toggled at any time, which hides many of its more complicated aspects. This allows students to easily navigate the GUI. Reactions with limited amount of species can be used to compare ODEKO's result with what the students achieve analytically. Generally, this tool will offer as a great introduction to the effects of the different components and parameters of a rocket engine.
- **Amateur rocket engine design:** One of the main struggles of amateur rocket motor development is the nozzle design. ODEKO will give more than enough means to optimize a nozzle for a given propellant and the desired characteristics. It will also allow to study the effect of different grain geometries and the possibility to reuse the same nozzle for different configurations. Moreover, as ODEKO gives the expected chamber pressure for a given grain geometry, propellant and nozzle, the combustion chambers can be designed accordingly.

2

Chemical Equilibrium modelling

Contents

A set of assumptions were made in the following theoretical model:

- Perfect gas, ideal-gas law.
- Complete adiabatic combustion of reactants, isentropic expansion in the nozzle and homogeneous mixing.
- Condensed phase particles sufficiently small to have no velocity or temperature lags in respect to the fluid.

2.1 Thermodynamic variables

Firstly, the set of variables to be used will be defined. The variable n_j represents the number of kg-mol of a specie divided by the kilograms of mixture. Therefore, its sum corresponds to the inverse of the molecular mass of the mixture, as seen in Equation 2.1, where NG represents the number of gaseous species.

$$\frac{1}{\mathcal{M}} = n = \sum_{j=1}^{NG} n_j \tag{2.1}$$

In consequence, the gas constant for the mixture corresponds to:

$$R_g = R \cdot n \tag{2.2}$$

The C_p of a species in its standard state (1bar) is given by a set of least-squares coefficients, which have the format shown in Equation 2.3, which is used to define the c_p of a specie in the software (see section 5.1.2).

$$\frac{C_p^o}{R} = \sum_{i=1}^{N_{terms}} a_i \cdot T^{e_i} \tag{2.3}$$

The values for coefficients 'a' and exponents 'e' are stored in the database. Generally several intervals are defined, in order to better fit the coefficients to the temperature range. Note that, if the current temperature is out of the range of definition, C_p is considered constant an equal to either its value for the maximum temperature in the range or the minimum, and a warning flag is raised for the user. All other computations are performed in the same manner.

$$\begin{aligned} \frac{H^o}{RT} &= \frac{\int C_p^o dT}{RT} \\ \frac{S^o}{RT} &= \int \frac{C_p^o}{RT} dT \end{aligned} \tag{2.4}$$

Equation 2.4 shows how to calculate enthalpy and entropy of the specie. Said integral is done analytically, since the expression for C_p will always be a polynomial. Using Equation 2.5 the chemical potential of the specie may also be calculated. Where g^o represents gibbs free energy at the standard state.

$$\frac{\xi^o}{RT} = \frac{1}{RT}\frac{\delta g^o}{\delta n} = \frac{H^o}{RT} - \frac{S^o}{R} \tag{2.5}$$

Due to the perfect gas hypothesis, the enthalpy has no pressure dependence, being only a function of temperature. For gaseous species, entropy and chemical potential are dependent on pressure and molar fraction when part of a mixture, as seen in Equation 2.6.

$$\begin{aligned}
\xi_j &= \begin{cases} \xi_j^o + RT\ln\frac{n_j}{n} + RT\ln\frac{P}{1bar} & (j = 1, ..., NG) \\ \xi_j^o \quad (j = NG+1, ..., NS) \end{cases} \\
S_j &= \begin{cases} S_j^o - R\ln\frac{n_j}{n} - R\ln\frac{P}{1bar} & (j = 1, ..., NG) \\ \chi_j^o \quad (j = NG+1, ..., NS) \end{cases} \\
H_j &= H_j^o
\end{aligned} \tag{2.6}$$

The specific thermodynamic variables of the mixture can be written as seen in Equation 2.7. Note the change in units (relative to kg instead of kg-mol).

$$\begin{aligned}
h &= \sum_{j=1}^{NS} n_j H_j^o \\
s &= \sum_{j=1}^{NS} n_j S_j
\end{aligned} \tag{2.7}$$

2.2 Equilibrium calculations

The condition for chemical equilibrium is the minimization of free energy. In this case, Gibbs energy will be minimized (note that the minimization of Helmholtz energy may also be used, although it is not meant for constant pressure problems, characteristic of a rocket engine). Gibbs energy by kg of mixture is given by

$$g = \sum_{j=1}^{NS} \xi_j n_j \tag{2.8}$$

The process followed in order to obtain the iteration equations is the one given in [3]. In order to perform the minimization, a set of mass-balance constrains are added:

$$\begin{aligned} b_i - b_i^o &= 0 \qquad (i = 1, ..., NE) \\ b_i &= \sum_{j=1}^{NS} a_{ij} n_j \qquad (i = 1, ..., NE) \end{aligned} \tag{2.9}$$

b_i represents the kg-atoms of an element per kg of mixture (of propellant in the case of b_i^o), a_{ij} represent the kg-atoms of element i per kg-mol of specie j. A term 'G' is defined as:

$$G = g + \sum_{i=1}^{NE} \lambda_i (b_i - b_i^o) \tag{2.10}$$

Where λ_i are the Lagrangian multipliers. Therefore, the condition for equilibrium becomes:

$$\delta G = \sum_{j=1} NS \left(\xi_j + \sum_{i=1}^{NE} \lambda_i a_{ij} \right) \delta n_j + \sum_{i=1}^{NE} (b_i - b_i^o) \delta \lambda_i = 0 \tag{2.11}$$

As seen in [3] the iteration equations become:

$$\Delta \ln n_j = \sum_{i=1}^{NE} a_{ij} \pi_i + \Delta \ln n + \frac{H_j^o}{RT} \Delta \ln T - \frac{\xi_j}{RT} \qquad (j = 1, ..., NG) \tag{2.12}$$

$$- \sum_{i=1}^{NE} a_{ij} \pi_i - \frac{H_j^o}{RT} \Delta \ln T = - \frac{\xi}{RT} \qquad (j = NG + 1, ..., NS) \tag{2.13}$$

$$\sum_{j=1}^{NG} a_{ij} n_j \Delta \ln n_j + \sum_{j=NG+1}^{NS} a_{ij} \Delta n_j = b_i^o - b_i \qquad (i = 1, ..., NE) \tag{2.14}$$

$$\sum_{j=1}^{NG} n_j \Delta \ln n_j - n \Delta \ln n = n - \sum_{j=1}^{NG} n_j \tag{2.15}$$

$$\sum_{j=1}^{NG} \frac{n_j H_j^o}{RT} \Delta \ln n_j + \sum_{j=NG+1}^{NS} \frac{H_j^o}{RT} \Delta n_j + \left(\sum_{j=1}^{NS} \frac{n_j C_{p,j}^o}{R} \right) \Delta \ln T = \frac{h_0 - h}{RT} \tag{2.16}$$

$$\sum_{j=1}^{NG} \frac{n_j S_j}{R} \Delta n_j + \sum_{j=NG+1}^{NS} \frac{S_j^o}{R} \Delta n_j + \left(\sum_{j=1}^{NS} \frac{n_j C_{p,j}^o}{R} \right) \Delta \ln T = \frac{s_0 - s}{R} + n - \sum_{j=1}^{NG} n_j \tag{2.17}$$

Where $\pi_i = -\lambda_i / RT$. The system of equations written in a matrix, (eliminating $\Delta \ln n_j$ via the substitution of Equation 2.12) is included in Appendix A.

For chemical equilibrium in the combustion chamber, enthalpy is constant (since a combustion is taking place), therefore Equation 2.12 through Equation 2.16 are used. Throughout the nozzle entropy is conserved, therefore Equation 2.12 through Equation 2.15 and Equation 2.17 are used. Entropy is conserved in the nozzle because it is considered adiabatic (no heat flow), viscosity has been disregarded and reaction terms cancel out due to the equilibrium condition. This can be proven (as seen in [6]) by writing the entropy differential:

$$TdS = d\left[\sum_{j=1}^{NS} n_j H_j^o\right] - \frac{1}{\rho}dP - \sum_{j=1}^{NS} \xi_j dn_j \tag{2.18}$$

If the equations of energy and momentum are substituted in Equation 2.18, and considering the definition for the chemical potential given in Equation 2.5, the following expression is obtained:

$$TdS = -\sum_{j=1}^{NS} \xi_j dn_j = -dg \tag{2.19}$$

Which is identically zero, since the equilibrium criteria is the minimization of gibbs free energy ($dg = 0$).

2.2.1 Convergence criteria

The value for a new iteration (k+1) is obtained in the following manner:

$$\begin{aligned} \ln {n_j}^{(k+1)} &= \ln {n_j}^{(k)} + \varepsilon^{(k)}(\Delta \ln n_j)^{(k)} \qquad (j = 1, ..., NG) \\ n_j^{(k+1)} &= n_j^{(k)} + \varepsilon^{(k)}(\Delta n_j)^{(k)} \qquad (j = NG+1, ..., NS) \\ \ln n^{(k+1)} &= \ln{(k)} + \varepsilon^{(k)}(\Delta \ln n)^{(k)} \\ \ln T^{(k+1)} &= \ln T^{(k)} + \varepsilon^{(k)}(\Delta \ln T)^{(k)} \end{aligned} \tag{2.20}$$

The values used in the first iteration are preset estimates, which can be changed in the preferences of the application at any time. The default values are: $T^{(0)} = 3800, n^{(0)} = 0.1, n_j^{(0)} = 0.1/NG$. If the value of n for a given specie drops below a certain threshold (by default 10^{-8}) it is automatically set to 0 and the inverse of its logarithm is not computed in order to save computing power.

The value for $\varepsilon^{(k)}$ is a correction value meant to avoid drastic changes in values which may affect convergence capabilities. Said value is computed at every iteration following Equation 2.21. The value for ε_2 is only computed for species that have a value of n_j below a certain threshold (also specified in preferences).

$$\begin{aligned} \varepsilon_1 &= \frac{2}{max(5|\Delta \ln T|, 5|\Delta \ln n|, |\Delta \ln n_j|)} \\ \varepsilon_2 &= min\left|\frac{-\ln \frac{n_j}{n} - 9.2103404}{\Delta \ln n_j - \Delta \ln n}\right| \\ \varepsilon^{(k)} &= min(1, \varepsilon_1, \varepsilon_2) \end{aligned} \tag{2.21}$$

The convergence criteria are written in Equation 2.22. Note that the values written for convergence represent the default values, but they may be changed by the user at any time.

$$\begin{aligned} \frac{n_j|\Delta \ln n_j|}{sum_{j=1}^{NS} n_j} &\leq 0.5 \cdot 10^{-5} \qquad (j = 1, ..., NG) \\ \frac{|\Delta n_j|}{\sum_{j=1}^{NS}} &\leq 0.5 \cdot 10^{-5} \qquad (j = NG + 1, ..., NS) \\ \left| b_i^0 - b_i \right| &\leq (b_i^0)_m ax \cdot 10^{-6} \qquad (i = 1, ..., NE) \\ |\Delta \ln T| \leq 10^{-4} \left| \frac{s_0 - s}{R} \right| &\leq 0.5 \cdot 10^{-4} \end{aligned} \tag{2.22}$$

The expressions for convergence criteria and correction values are the ones used in reference [3]. Note that if the user allows the program to account for species with $n_j < 10^{-8}$ and additional test has to be conducted:

$$\left| \frac{\pi_i^{(k)} - \pi_i^{(k+1)}}{\pi_i^{(k+1)}} \right| \leq 0.001 \qquad (i = 1, ..., NE) \tag{2.23}$$

2.2.2 Ions

If ions are present in the mixture several details must be kept in mind. Firstly, if during the calculation all ions were to disappear from the iteration (drop below the minimum value and their n set to 0). The equilibrium equation for electrons becomes null and may prove problematic when trying to solve the linear system. Therefore, this equation must be removed. Moreover if, after convergence, there are still ions present in the mix an additional convergence test must be performed:

$$|\Delta \pi_e| = \left| \frac{-\sum_{j=1}^{NG} a_{ej} n_j}{\sum_{j=1}^{NG} a_{ej}^2 n_j} \right| \leq 0.0001 \tag{2.24}$$

If the test is not passed, the values of the ionic species must be altered as indicated by equation 2.25 until the test is successfully passed.

$$(\ln n_j)^{(k+1)} = (\ln n_j)^{(k)} + a_{ej} \Delta \pi_e \qquad (j = 1, ..., NG) \tag{2.25}$$

Note that, if only positive or only negative ions are present, the ion balance can never be achieved and, therefore, ions must be removed from the mixture. Furthermore, if a new equilibrium is calculated using a previous one as seed for the iterations, if ions were removed it would be recommended to re-include the ion equation in the system, since ions might be present for the new mixture.

2.2.3 Condensed phases

Initially, only gaseous species and ions are considered in the mix. After convergence, a test is performed in order to check if any condensed phases should be added.

$$\frac{\delta G}{\delta n_j} = \left(\frac{\xi_j^o}{RT}\right)_c - \sum_{i=1}^{NE} \pi_i a_{ij} < 0 \tag{2.26}$$

If more than one condensed species passes the test in Equation 2.26 the one that decreases Gibbs energy the most (has the most negative value in Equation 2.26) is added. It is recommended to divide the quantity computed in Equation 2.26 by the molecular mass in order to achieve better results whenever more than one species passes the test and their values have to be compared.

Once the program converges again, the test is re-conducted. This process continues until no more condensed phases are added or removed. After every convergence, if the value of n_j for a condensed species is negative it is removed from the mixture.

Furthermore, only species whose validity temperature range contains the temperature achieved after convergence are checked. After every convergence it is also checked if any condensed specie is no longer in its valid temperature range and has to be removed from the mixture. Note that only one specie can be removed due to this condition, afterwards a new equilibrium will be reached and the condition shown in Equation 2.26 will be re-chequed. If no condensed species pass the test the temperature range condition will be re-chequed. This process is continued until no more species should be added or removed.

Whenever a condensed specie with $n_j \neq 0$ is removed, it is recommended to reset all mixture components, since it is possible that said condensed specie was the only one to have a specific atom in its composition. Therefore, after its removal it may be difficult for the program to reintroduce said atom in the mixture, requiring lots of iterations or maybe not even being capable of it. Nonetheless, if a specific atom is unintentionally removed from the mixture, soft resets are set in place in the program in order to reintroduce it.

Alternative methods for condensed phases are present in the literature. The advantage of the method here presented is that it does not require the user nor program to establish a relation in between condensed phases and gas phases, focusing exclusively on the minimization of Gibbs free energy. It is important to note that this method generally requires many iterations, since many convergences need to be reached in order to check for possible condensed phases.

Gas constant for the mixture

If no condensed phases are present, the gas constant is defined as seen in Equation 2.2. On the other hand, if condensed phases are present, the perfect gas equation can only be applied to the gaseous subset. Nevertheless, it is useful to define a new gas constant in order to include the effects of the presence of condensed species. Therefore, the gas constant used when condensed phases are present is:

$$R_g' = R\left(\sum_{j=1}^{NG} n_j\right)\left(1 - \sum_{j=NG+1}^{NS} \frac{n_j}{\mathcal{M}}\right) \tag{2.27}$$

In order for this new gas constant to be valid, the flow must be strongly interacting with the condensed phase, meaning that their temperatures and velocities must match at all times (which is explained in detail in [6]).

2.2.4 Phase changes between condensed phases throughout the nozzle

If a condensed phase (generally a liquid) is present in the combustion chamber or throughout the nozzle a phase change between condensed species might arise. Since the temperature is decreasing along the nozzle, it may reach the value at which the liquid phase transitions to its solid state. Said transition can not be modelled as instant. Therefore, during a given nozzle region, both species coexist. In this region in which the phase change takes place, the flow is isotherm.

Once the melting temperature is reached it is maintained constant until the original specie has completely disappeared. At the melting point, the chemical potentials for both condensed species coincide (the least-squares coefficients are fitted to ensure this fact). As will be seen in subsection 2.3.1 a notable discontinuity will be present for many variables, the most critical being the speed of sound.

The system of equations does not need to be modified and can be solved normally. Nonetheless, it is important to make sure that both species are allowed to coexist. This is easily achieved by slightly increasing the temperature range in which they are defined, so they slightly overlap. Moreover, it will be convenient to include a check to test if the phase transition is taking place, as some changes will need to be applied to calculate the equilibrium variables (as shown in subsection 2.3.1).

2.2.5 Matrix singularities

The main matrix singularity that may occur is the presence of a condensed phase whose elemental composition is a linear combination of two other condensed species. In this case the matrix cannot be solved and one of the species must be removed. The program raises a flag making the user aware of this occurrence, since it may lead to a wrong or partially wrong result. Generally, this constitutes a rare occurrence, thanks to the criteria for including condensed species, which only allows for one inclusion per convergence. Moreover, the final mixture for rocket engines do not tend to have multiple condensed species present.

Another possible singularity was already mentioned in section subsection 2.2.2 and occurs when all ions have $n_j = 0$. In this case, the row and column of the matrix corresponding to the electron are composed of only 0. In order to avoid this the program just needs to remove the electron equation.

Finally, if all species containing one of the elements are temporarily eliminated or two rows of the matrix are identical the matrix cannot be solved. In this case all species currently set to zero are reset to $n_j = 10^{-6}$. This is considered a soft reset and may not always solve the singularity. If it fails repeatedly either a stronger reset needs to be applied or the calculation process needs to be aborted.

2.3 Frozen and equilibrium thermodynamic properties

Throughout this section values for 'frozen' and 'equilibrium' properties will be calculated. Typically, only frozen values will be used; although, as will be seen in section 2.5 at times it is required to use the 'equilibrium' values, specially when applying equations that are usually meant for non-reacting flows (such as the M=1 condition at the throat of the nozzle). The 'equilibrium' values do not necessarily carry a physical meaning, in contrast to their 'frozen' counterparts. Nonetheless, they introduce the effect of chemical equilibrium into formulas that will be used in section 2.5. For example, the condition of $M = 1$ at the throat is applied for the equilibrium speed of sound (therefore, the frozen $M = 1$ is not achieved at the throat).

As will be seen in chapter 5, when the flow is neither non-reacting nor in equilibrium certain equations or conditions cannot be used due to the lack of a proper 'reacting' definition for certain parameters, such as the speed of sound.

The first value to calculate is $C_{p,e}$, since it will be used for other calculations. $C_{p,e} = C_{p,f} + C_{p,r}$ where the frozen and reacting values correspond to:

$$C_{p,f} = \sum_{j=1}^{NS} C^o_{p,j} n_j \tag{2.28}$$

$$\begin{aligned} \frac{C_{p,r}}{R} = \sum_{i=1}^{NE} \left(\sum_{j=1}^{NG} \frac{a_{ij} n_j H^o_j}{RT} \right) \left(\frac{\delta \pi_i}{\delta \ln T} \right)_P + \sum_{j=NG+1}^{NS} \frac{H^o_j}{RT} \left(\frac{\delta n_j}{\delta \ln T} \right)_P + \\ + \left(\sum_{j=1}^{NG} \frac{n_j H^o_j}{RT} \right) \left(\frac{\delta \ln n}{\delta \ln T} \right)_P + \sum_{j=1}^{NG} \frac{n_j (H^o_j)^2}{R^2 T^2} \end{aligned} \tag{2.29}$$

Where the derivatives with respect to $\ln T$, as indicated in [3] can be obtained from the following set of equations, which is also included in Appendix A in matrix form.

$$
\begin{aligned}
\sum_{i=1}^{NE}\sum_{j=1}^{NG} a_{kj}a_{ij}n_j\left(\frac{\delta\pi_i}{\delta\ln T}\right)_P + \sum_{j=NG+1}^{NS} a_{kj}\left(\frac{\delta n_j}{\delta\ln T}\right)_P + \sum_{j=1}^{NG} a_{kj}n_j\left(\frac{\delta\ln n}{\delta\ln T}\right)_P = \\
= -\sum_{j=1}^{NG}\frac{a_{kj}n_jH_j^o}{RT} \qquad (k=1,....,NE) \\
\sum_{i=1}^{NE} a_{ij}\left(\frac{\delta\pi_i}{\delta\ln T}\right)_P = -\frac{H_j^o}{RT} \qquad (j=NG+1,...,NS) \\
\sum_{i=1}^{NE}\sum_{j=1}^{NG} a_{ij}n_j\left(\frac{\delta\pi_i}{\delta\ln T}\right)_P = -\sum_{j=1}^{NG}\frac{n_jH_j^o}{RT}
\end{aligned}
\tag{2.30}
$$

Note that the coefficients for said system coincide with part of the coefficients used in the equilibrium matrix. The values for the frozen values of C_v, γ and the speed of sound can be calculated as indicated in Equation 2.31.

$$
\begin{aligned}
C_{v,f} &= C_{p,f} - R_g \\
\gamma &= \frac{C_p}{C_v} \\
a &= \sqrt{R_gT\gamma}
\end{aligned}
\tag{2.31}
$$

On the other hand, its equilibrium counterparts which account for the equilibrium state of the mixture, are calculated with Equation 2.32.

$$
\begin{aligned}
C_{v,e} &= C_{p,e} + \frac{R_g\left(\frac{\delta\ln V}{\delta\ln T}\right)_P^2}{\left(\frac{\delta\ln V}{\delta\ln P}\right)_T} \\
\gamma_s &= -\frac{\gamma}{\left(\frac{\delta\ln V}{\delta\ln P}\right)_T} \\
a_e &= \sqrt{R_gT\gamma_s}
\end{aligned}
\tag{2.32}
$$

Note that the value for R_g is different if condensed phases are present, as was explained in subsection 2.2.3. The thermodynamic derivatives present in the equation can be obtained in the following manner:

$$
\begin{aligned}
\left(\frac{\delta\ln V}{\delta\ln T}\right)_P &= 1 + \left(\frac{\delta\ln n}{\delta\ln T}\right)_P \\
\left(\frac{\delta\ln V}{\delta\ln P}\right)_T &= -1 + \left(\frac{\delta\ln n}{\delta\ln P}\right)_T
\end{aligned}
\tag{2.33}
$$

The value for $\left(\frac{\delta \ln n}{\delta \ln T}\right)_P$ is obtained from Equation 2.30, whilst $\left(\frac{\delta \ln n}{\delta \ln P}\right)_T$ is obtained from the following system of equations, which is also included in Appendix A:

$$\begin{aligned}
&\sum_{i=1}^{NE}\sum_{j=1}^{NG} a_{kj}a_{ij}n_j \left(\frac{\delta \pi_i}{\delta \ln P}\right)_T + \sum_{j=NG+1}^{NS} a_{kj}\left(\frac{\delta n_j}{\delta \ln P}\right)_T + \\
&+\sum_{j=1}^{NG} a_{kj}n_j \left(\frac{\delta \ln n}{\delta \ln P}\right)_T = -\sum_{j=1}^{NG} a_{kj}n_j \qquad (k = 1, ..., NE) \\
&\sum_{i=1}^{NE} a_{ij}\left(\frac{\delta \pi_i}{\delta \ln P}\right)_T = 0 \qquad (j = NG+1, ..., NS) \\
&\sum_{i=1}^{NE}\sum_{j=1}^{NG} a_{ij}n_j \left(\frac{\delta \pi_i}{\delta \ln P}\right)_T = -\sum_{j=1}^{NG} n_j
\end{aligned} \tag{2.34}$$

It is clear from Equation 2.33 and Equation 2.32 that when the mixture's equilibrium composition is close to constant for a local pressure range (and therefore can be approximated with frozen composition), the values of the thermodynamic variables tend to their frozen definitions. This obviously was a requirement for the equilibrium model to be coherent with the frozen values of the variables.

2.3.1 Equilibrium properties during a phase change between condensed species

As was explained by Gordon in [5], during the transition between condensed phases, at which temperature stays constant, derivatives with respect to temperature are not properly defined. Therefore, expressions such as Equation 2.32 cannot be used. Gordon presents an alternative definition for the isentropic isothermal case:

$$\begin{aligned}
\gamma_{S,T} &= -\frac{1}{(\ln V/\ln P)|_T} \\
a_e &= \sqrt{R_g T \gamma_{S,T}}
\end{aligned} \tag{2.35}$$

It should be pointed out that, during this phase transition, two rows of Equation 2.34 are equal to each-other. Consequently, the solution is undefined. Luckily, the variable $\left(\frac{\delta \ln n}{\delta \ln P}\right)_T$ can still be solved for, which is the only value required to compute the equilibrium variables as shown in Equation 2.35. This can be achieved given that for the variables $\frac{\delta n_j}{\delta \ln P}$ corresponding to the transitioning phases only the value of their sum is relevant, consequently, one column can be removed (as well as the duplicate row) and a solvable system is recovered.

Note that this definition change will cause a notable discontinuity in the values of gamma and the velocity of sound. It is also important to point out that, as was explained in subsection 2.2.3, the value for R_g needs to account for the presence of condensed species.

In this isothermal region, the equilibrium values for C_p and C_v are not defined, therefore only their frozen counterparts can be used and the application only offers frozen variables output for this region. Note as well that even if equilibrium variable present a discontinuity at the start and end of the isothermal region, frozen variables are always continuous (although some show a discontinuity on their first derivative).

2.4 Transport properties

Transport properties for a given specie are approximated by a least-squares polynomial, as shown in Equation 2.36.

$$\left.\begin{matrix}\ln \mu_j \\ \ln \lambda_j\end{matrix}\right\} = A \ln T + \frac{B}{T} + \frac{C}{T^2} + D \tag{2.36}$$

Where μ_j, λ_j represent the viscosity and conductivity of specie j respectively and A,B,C,D are the polynomial coefficients. In order to calculate the mixture values, two interaction coefficients must be computed first:

$$\phi_{ij} = \frac{1}{4}\left[1 + \left(\frac{\mu_i}{\mu_j}\right)^{1/2}\left(\frac{\mathcal{M}_j}{\mathcal{M}_i}\right)^{1/4}\right]^2 \left(\frac{2\mathcal{M}_j}{\mathcal{M}_i + \mathcal{M}_j}\right)^{1/2} \tag{2.37}$$

$$\psi_{ij} = \phi_{ij}\left[1 + \frac{2.41(\mathcal{M}_i - \mathcal{M}_j)(\mathcal{M}_i - 0.142\mathcal{M}_j)}{(\mathcal{M}_i + \mathcal{M}_j)^2}\right] \tag{2.38}$$

Where $\mathcal{M}$ is the molecular mass of the specie. An alternative form to calculate ϕ_{ij} is given in Equation 2.39. It requires the interaction parameter for species i and j to be known.

$$\phi_{ij} = \frac{\mu_i}{\mu_{ij}} \frac{2\mathcal{M}_j}{\mathcal{M}_i + \mathcal{M}_j} \tag{2.39}$$

The expressions for the mixture viscosity and thermal conductivity are:

$$\mu = \sum_{i=1}^{NT} \frac{x_i \mu_i}{x_i + \sum_{\substack{j=1 \\ j \neq i}}^{NT} x_j \phi_{ij}} \tag{2.40}$$

$$\lambda_f = \sum_{i=1}^{NT} \frac{x_i \lambda_i}{x_i + \sum_{\substack{j=1 \\ j \neq i}}^{NT} x_j \psi_{ij}} \tag{2.41}$$

Where x is the molar fraction of the specie. Note that, the conductivity given in expression 2.41 is its frozen value. Therefore, its equilibrium counterpart needs to be calculated as was done for the thermodynamic variables in section 2.3. The viscosity of the mixture is the same no matter if the mixture is in chemical equilibrium or non-reacting. The equilibrium conductivity can be expressed as $\lambda_e = \lambda_f + \lambda_r$, where λ_r corresponds to:

$$\lambda_r = R \sum_{i=1}^{NK} \frac{\Delta H_i^o}{RT} \lambda_{r,i} \tag{2.42}$$

Where ΔH_i^o corresponds to the net enthalpy variation of reaction i. The values for $\lambda_{r,i}$ are obtained from the system in Equation 2.43.

$$\sum_{j=1}^{NK} \tau_{ij} \lambda_{r,j} = \frac{\Delta H_i^o}{RT} \qquad (i = 1, ..., NK) \tag{2.43}$$

Where τ_{ij} is given by:

$$\tau_{ij} = \sum_{k=1}^{NT} \sum_{l=k+1}^{NT} \left(\frac{RT}{PD_{kl}} x_k x_l \right) \left(\frac{\upsilon_{ik}}{x_k} - \frac{\upsilon_{il}}{x_l} \right) \left(\frac{\upsilon_{jk}}{x_k} - \frac{\upsilon_{jl}}{x_l} \right) \tag{2.44}$$

The value of υ_{ik} corresponds to the net coefficients of specie k in reaction i. And $\frac{RT}{PD_{kl}}$ equates to:

$$\frac{RT}{PD_{kl}} = \frac{5\mathcal{M}_k \mathcal{M}_l}{3A_{kl}^* \mu_{kl} (\mathcal{M}_k + \mathcal{M}_l)} \tag{2.45}$$

The factor A_{kl}^* is the cross-section ratio, which is substituted by a value of 1.1, additionally, μ_{kl} can be obtained from Equation 2.39.

2.5 One Dimensional Chemical Equilibirum (ODE)

The first step to solve the flow through the nozzle is to calculate the equilibrium composition at the combustion chamber (constant enthalpy). The enthalpy of the reactants is easily calculated since their starting temperature is given by the user or the database. The chamber pressure (or alternatively, the grain geometry, which would require to iterate the ODE calculations) is also given. Therefore, the composition at the combustion chamber is obtained without any complications.

Subsequently, the composition and thermodynamic properties of the nozzle's throat are to be calculated. A first estimation for the throat's pressure is conducted:

$$\frac{P_{inf}}{P_t} = \left(\frac{\gamma_s + 1}{2}\right)^{\gamma_s/(\gamma_s - 1)} \tag{2.46}$$

The convergence criteria is:

$$\left|\frac{v_t^2 - a_{e,t}^2}{u_t^2}\right| \leq 0.4 \cdot 10^{-4} \tag{2.47}$$

And the correction performed to the pressure until the convergence is satisfied is:

$$P_t^{(k+1)} = P_t^{(k)} \frac{1 + \gamma_s M^2}{1 + \gamma_s} \tag{2.48}$$

The velocity of the fluid is calculated with Equation 2.49. Where h_{inf} corresponds to the enthalpy of combustion.

$$u = \sqrt{2(h_{inf} - h)} \tag{2.49}$$

Once the throat has been calculated, the rest of the points of the nozzle are calculated in the same manner. Firstly, an initial estimate for the pressure is obtained (Equation 2.50 for subsonic nozzle and Equation 2.51 for supersonic nozzle).

$$\begin{gathered}
\ln \frac{P_{inf}}{P} = \frac{\ln \frac{P_{inf}}{P_t}}{\frac{A}{A_t} + 10.587 \left(\ln \frac{A}{A_t}\right)^3 + 9.454 \ln \frac{A}{A_t}} \qquad \left(\frac{A}{A_t} \geq 1.09\right) \\
\ln \frac{P_{inf}}{P} = \frac{\ln 0.9 \frac{P_{inf}}{P_t}}{\frac{A}{A_t} + 10.587 \left(\ln \frac{A}{A_t}\right)^3 + 9.454 \ln \frac{A}{A_t}} \qquad \left(1.0001 < \frac{A}{A_t} < 1.09\right)
\end{gathered} \tag{2.50}$$

$$\begin{gathered}
\ln \frac{P_{inf}}{P} = \ln \frac{P_{inf}}{P_t} + \sqrt{3.294 \left(\ln \frac{A}{A_t}\right)^2 + 1.535 \ln A A_t} \qquad \left(1.0001 < \frac{A}{A_t} < 2\right) \\
\ln \frac{P_{inf}}{P} = \gamma_s + 1.4 \ln \frac{A}{A_t} \qquad \left(\frac{A}{A_t} \geq 2\right)
\end{gathered} \tag{2.51}$$

Afterwards, the value for the pressure is iterated according to the following formula:

$$\left(\ln \frac{P_{inf}}{P}\right)^{(k+1)} = \left(\ln \frac{P_{inf}}{P}\right)^{(k)} + \left(\frac{\delta \ln \frac{P_{inf}}{P}}{\delta \ln \frac{A}{A_t}}\right)_s \cdot \left[\ln \frac{A}{A_t} - \ln \frac{\rho_t u_t}{\rho u}\right] \tag{2.52}$$

Where the partial derivative can be expressed as:

$$\left(\frac{\delta \ln \frac{P_{inf}}{P}}{\delta \ln \frac{A}{A_t}}\right)_s = \left(\frac{\gamma_s u^2}{u^2 - a^2}\right) \tag{2.53}$$

After all points are calculated (equilibrium composition along with thermodynamic and transport variables if requested) the rocket parameters may be calculated as well, employing the following equations:

$$\begin{aligned} E &= \dot{m} v_{exit} + (P_{exit} - P_a) A_{exit} \\ I_{sp} &= \frac{E}{\dot{m}} \\ c^* &= \frac{P_{inf} A_t}{\dot{m}} \\ C_E &= \frac{I_{sp}}{C*} \end{aligned} \tag{2.54}$$

The mass flow is calculated at the throat. Even though it should be constant along the nozzle, very small variations are present due to the nature of the numerical method. Since all points are calculated establishing that their mass flow should equal that of the throat, it is regarded as the most representative value.

2.5.1 Frozen flow

If the flow is set to 'freeze' (stop reacting) after the throat, the procedure is simple. After the frozen point composition is conserved. The iteration procedure remains the same. Since the equilibrium equations can no longer be used, the temperature of the mixture must also be iterated (for each pressure iteration there must be a temperature convergence). The iteration is performed using the constant entropy condition (non-viscous, non-reacting flow with no heat transfer), as shown in Equation 2.55.

$$(\ln T)^{(k+1)} = (\ln T)^{(k)} + \frac{s_{inf} - s}{C_{p,f}} \tag{2.55}$$

The convergence condition is:

$$\left|\frac{s_{inf} - s}{C_{p,f}}\right| \leq 0.5 \cdot 10^{-4} \tag{2.56}$$

On the other hand, it the frozen point is located before the throat an additional iteration procedure for the throat is required. In order to calculate the parameters of the frozen point (specially its pressure, required to obtain its composition) the throat must be computed first. The only option is to calculate the throat in its equilibrium condition. Once the frozen point is calculated the throat can be re-calculated with the frozen composition. Note that this will fundamentally change the parameters of the nozzle and; therefore, the frozen point will no longer be correct (it will not be coherent with the new throat values). This process is iterated until the frozen point and throat are compatible.

Note that this process usually requires many iterations, although it is relatively fast since only two points are being computed back and forth. Generally frozen flow is much faster to compute in spite of the additional iteration procedure required.

Note that, once the flow freezes, all parameters used must be also 'frozen' and equilibrium variables can no longer be used for any iteration or convergence criteria, since the flow no longer is in chemical equilibrium.

The option to 'freeze' the flow can also be included in optimizations, IAC and FAC models or computations with grain geometry.

2.5.2 Convergence criteria during isothermal regions caused by transitions between condensed phases

If the throat of the nozzle is contained at the start of the isothermal region of a condensed phase transition, the velocity of sound is discontinuous at that given point, as described by Equation 2.35. Therefore, the convergence criteria applied in Equation 2.48 can no longer be used. In its place, as described by Gordon in [5], the following criteria can be used:

$$\ln {P_t}^{k+1} = \ln {P_t}^{k} + \left(\frac{\delta \ln P}{\delta \ln T}\right)_S (\ln T_m - \ln T) \tag{2.57}$$

Where T_m is the melting temperature between both phases and T is the temperature of the mixture. The derivative value can be expressed as:

$$\left(\frac{\delta \ln P}{\delta \ln T}\right)_S = \frac{c_P}{R_g \left(\frac{\delta \ln V}{\delta \ln T}\right)_P} \tag{2.58}$$

2.5.3 Finite Area Combustor (FAC)

Whilst the IAC model considers that the combustion conditions are that of a fictitious section of the nozzle with infinite area and null velocity; the FAC model sets a difference between this fictitious point and the combustion chamber with a given section area.

The FAC model assumes that the velocity at the injector is negligible and that the velocity at the combustion chamber exit (entrance of the nozzle) is only due to a differential in pressure in the combustion chamber. It also considers a cylindrical combustion chamber (constant section area). Therefore, the momentum equation can be written as:

$$P_{inj} = (P + \rho u^2)_c = \left(P + \frac{\dot{m}u}{A}\right)_c \tag{2.59}$$

In order to solve said system for a given P_{inj} an iteration procedure is needed. Firstly the composition for the injector is calculated, since its pressure is known by the user. Afterwards, 3 points need to be iterated: the end of the combustor (beginning of the nozzle), the throat and an imaginary point of infinite area which represents an equivalent nozzle of infinite starting area with null velocity at the start (which only expands isentropically). An initial estimate for P_{inf} is achieved through the following formula:

$$P_{inf} = P_{inj}\left(\frac{1.0257 - 1.2318\frac{A_c}{A_t}}{1 - 1.26505\frac{A_c}{A_t}}\right) \tag{2.60}$$

Where A_c represents the Area of the cylindrical combustor (which should coincide with the area at the start of the nozzle, although this is left up to the user). Afterwards, the throat is calculated, which in term allows to calculate the first section of the nozzle. Once the pressure and velocity for the first section of the nozzle are obtained, using Equation 2.59 a value for P_{inj} is obtained. Of course, it this value does not coincide with the user-defined value the estimation for P_{inf} was erroneous and an iteration procedure needs to be conducted. The convergence criteria is:

$$\frac{P_{inj} - (P + \rho u^2)_c}{P_{inj}} \leq 2 \cdot 10^{-5} \tag{2.61}$$

If the criteria is not met, the correction applied to P_{inf} is given by:

$$P_{inf}^{(k+1)} = P_{inf}^{(k)} \frac{P_{inj}}{(P + \rho u^2)_c} \tag{2.62}$$

It is important to note that if the nozzle is frozen before the throat, the necessary iteration in order to obtain the throat's parameters (explained in subsection 2.5.1) must be performed for each value of P_{inf}.

It should also be pointed out that FAC calculations are not necessarily better than IAC (which is the reason why both options are offered). In the case of liquid propellants, if the injector fulfills the condition of approximately null velocity for the flow (or if said non-zero velocity is known) the FAC model can be a better approximation than IAC. On the contrary, in the case of solid or hybrid propellants it can be a worse model. Firstly, the grain geometry is combusting and therefore modifying the area of the combustion chamber, although this may be solved by changing the cross section area of the chamber for each step. Additionally, the pressure throughout the chamber is assumed to not be constant in the FAC model, therefore different parts of the grain would burn at different speeds due to

Vielle's law. Moreover, the combustion of the grain expels the gaseous products at an unknown speed, which further invalidates the FAC model. Therefore, the FAC model is not appropriate for these cases and even though the IAC offers no solution to said problems it is probably best used for solid and hybrid propellants.

Note that, if the velocity of the flow at the injector point was known by the user, Equation 2.59 could be modified in order to include that initial velocity and improve the quality of this model.

2.5.4 Grain combustion

The process for grain combustion calculation is reduced to an iteration of the parameter P_{inf}, which is assumed to be the pressure of combustion for the grain. Firstly, the values at the throat are calculated using ODE and an initial estimate for P_{inf} (given by the user). Note that, if the flow is frozen before the throat, an additional iteration needs to be conducted to obtain the throat parameters, as explained in subsection 2.5.1. Generally, Vielle's Law would be used, along with the condition that the mass flow through the throat and that being created by the combustion should be equal, equation [9].

$$P_c = \left(\frac{\rho_t A_t u_t}{\rho_p A_b a}\right)^{1/n_v} \tag{2.63}$$

Where A_b is the combustion area, ρ_p is the density of the solid propellant and a_v, n_v are Vielle's parameters. P_c represents the combustion pressure necessary for this equilibrium to take place. Nevertheless, this procedure, although perfectly viable for a simplified rocket model, requires the use of the value of c^*. On a simplified model, the mass flow at the throat would be substituted by $\frac{P_c A_t}{c^*}$, whilst in ODE the mass flow is computed using the conditions at the throat. On Figure 2.1 a simplified qualitative example for the mass flow dependence with pressure for the nozzle throat is given, along with the dependence with pressure of the combustion mass flow for two sets of Vielle's parameters. The iteration consists on calculating the mass flow for the throat and then obtaining the pressure that gives that same mass flow for Vielle's law, which allows to calculate a new mass flow for the throat. Iterating this procedure the result would be achieved. Examining Figure 2.1 it is evident that for $n > 1$ this procedure converges, but for $n < 1$ it does not (which is the more common scenario).

An the ideal model, this divergence is avoided by the substitution of c^*, whilst in the ODE model a different approach was taken. For a given pressure, the throat area that makes both mass flows be equal is calculated using Equation 2.64. The value for the chamber pressure is corrected using Equation 2.65 until the value for the throat area converges to the real one for the given tolerance. It is important to note that correction in Equation 2.65 is different depending on the value of Vielle parameter 'n', due to the behaviour difference visible in Figure 2.1.

$$A_t = \frac{A_b \rho_p a_V (P_c)^{n_V}}{\rho_t u_t} \tag{2.64}$$

$$\begin{aligned} P_c^{(k+1)} &= P_c^{(k)} \frac{(A_t)_{calculated}}{(A_t)_{real}} \qquad n_V < 1 \\ P_c^{(k+1)} &= P_c^{(k)} \frac{(A_t)_{real}}{(A_t)_{calculated}} \qquad n_V > 1 \end{aligned} \tag{2.65}$$

Once the value of the throat converges, it can be stated that the chamber pressure used is correct. This procedure needs to be performed for every point describing the combustion area curve.

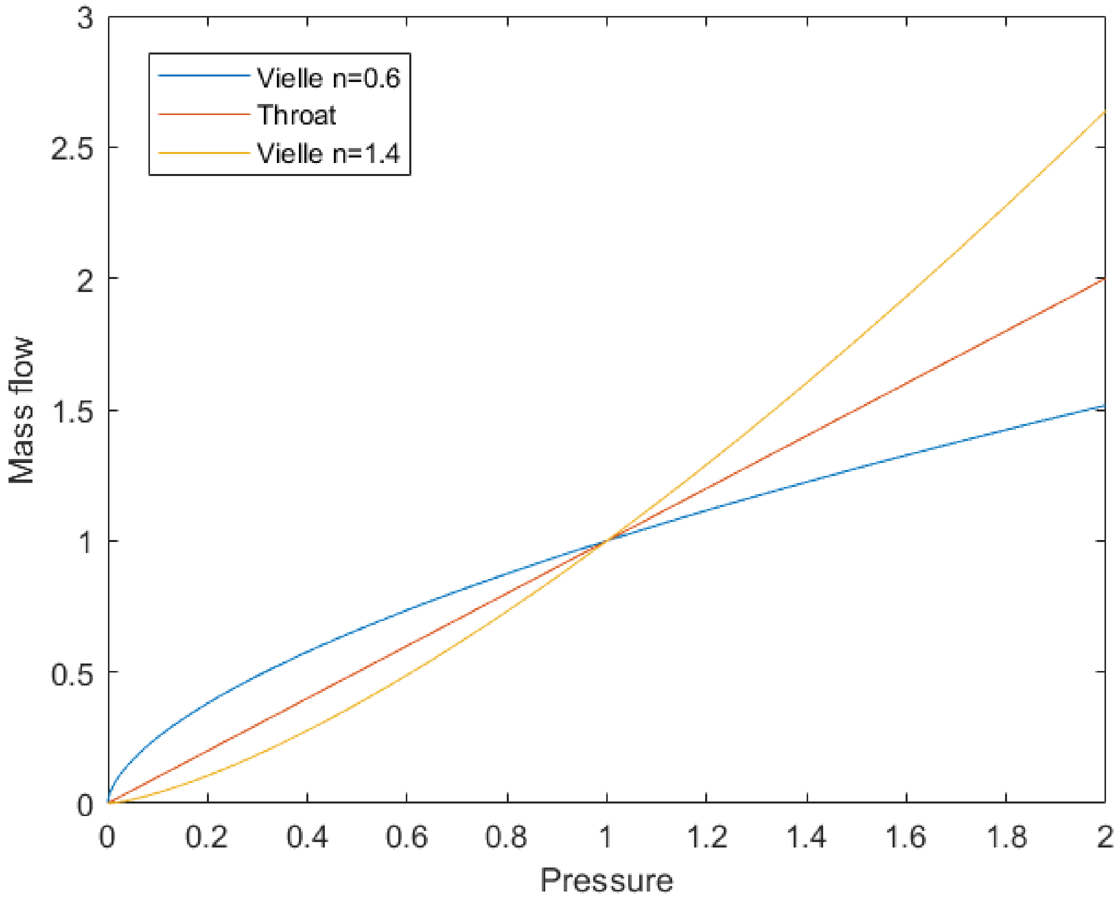


Figure 2.1: Qualitative example for mass flow pressure dependence

Finally, the combustion time can be calculated as:

$$t_b = \int_0^{y_f} \frac{1}{a_v (P_{inf})^{n_v}} dy \tag{2.66}$$

Where the variable y represents the advance parameter of the combustion, where y_f is the advance parameter for which the combustion ends. Subsequently, it is possible to obtain a relation between time and advance parameter using Equation 2.66. This allows to plot the rocket parameters relative to the combustion time and obtain a thrust curve relative to time, which is extremely useful in order to compare the model with real combustion of solid propellants.

The program offers the time curves along with a mean value for the parameters, obtained by use of Equation 2.67 where thrust has been used as an example:

$$\bar{E} = \frac{1}{t_b} \int_0^{t_b} E \cdot dt \tag{2.67}$$

Note that this value is equivalent to the thrust of a constant pressure combustion with the same total impulse.

It is important to clarify that transitory effects in the variation of the combustion surface area were not accounted for, the process was considered stationary. For most rocket engines the transitory effects of the combustion surface are negligible, since the density of the propellant is much higher than the density of the fluid (which is explained in detail in [9]).

Units for Vielle's law

It is important to note that, if the pressure unit is changed in Vielle's law, the value for the constant 'a' also needs to be modified appropriately. As an example, in Equation 2.68 an example unit change from MPa to Pa is shown. In order to maintain the same results for Vielle's law, the unit change needs to be incorporated in the 'a' constant.

$$\begin{aligned}
(\dot{r}\ [m/s]) &= (a\ [m/s])\,(P_c\ [MPa])^n \\
(\dot{r}\ [m/s]) &= \left(a\ [m/s]\ (10^{-6n})\right)(P_c\ [Pa])^n \\
(\dot{r}\ [m/s]) &= \left(a^{'}\ [m/s]\right)(P_c\ [Pa])^n
\end{aligned} \tag{2.68}$$

3

Chemical Kinetics modelling

Contents

3.1 Reaction rates

In order to model the chemical kinetics of the mixture it is vital to first obtain the reaction rates of each specie. The following equations are written as described in [7]. Given a set of reactions of the form:

$$\sum_{k=1}^{NS} \upsilon'_{ki}\chi_k \rightleftharpoons \sum_{k=1}^{NS} \upsilon''_{ki}\chi_k \tag{3.1}$$

Where υ'_{ki} and υ'_{ki} are the stoichiometric coefficient of specie k in reaction i as reactant and product respectively. χ_k is the chemical symbol of specie k. Reaction rates can be written as:

$$\dot{\omega}_k = \sum_{i=1}^{NK} \upsilon_{ki} q_i \tag{3.2}$$

Where index 'k' refers to a specie and index 'i' refers to a reaction. Additionally, $\upsilon_{ki} = \upsilon''_{ki} - \upsilon ki'$. The value for the rate of progress (q_i) can be calculated with the following expression:

$$q_i = K_{f,i} \prod_{k=1}^{NS} [X_k]^{\upsilon'_{ki}} - K_{r,i} \prod_{k=1}^{NS} [X_k]^{\upsilon''_{ki}} \tag{3.3}$$

Where $[X_k]$ is the molar concentration of specie k. The forward rate constant is calculated making use of the Arrhenius equation:

$$K_{f,i} = A_i T^{\beta_i} exp\left(\frac{-E_i}{R_c T}\right) \tag{3.4}$$

Where E_i is the activation energy of reaction i, A_i and β_i are constants related to the reaction. The gas constant (R_c) needs to have compatible units with the activation energy (typically calories). The reverse rate constant ($K_{r,i}$) has to be coherent with the forward rate constant and equilibrium constant ($K_{c,i}$), therefore:

$$\begin{aligned} K_{r,i} &= \frac{K_{f,i}}{K_{c,i}} \\ K_{c,i} &= K_{pi} \left(\frac{1bar}{R_m T}\right)^{\sum_{k=1}^{NS} \upsilon_{k,i}} \\ K_{p,i} &= exp\left(\frac{\Delta S_i^o}{R} - \frac{\Delta H_i^o}{RT}\right) \end{aligned} \tag{3.5}$$

Note that in Equation 3.5, the composition equilibrium constant ($K_{c,i}$) is calculated based on the pressure equilibrium constant ($K_{p,i}$). The expression uses a pressure of 1bar, this is because it is assumed to be the standard pressure, although older data sets may have a standard pressure of 1atm.

If efficiencies for third bodies have been specified, Equation 3.3 must be switched for Equation 3.6. Note that if α_{ki} has not been specified for a given specie its value defaults to 1. Therefore, by default the first factor would be the total concentration of the mixture.

Note as well, that this constant is computed through the enthalpy and entropy of the reaction (ΔH_i^o and ΔS_i^o respectively), which must be computed using the same reaction coefficients (ν) employed previously and subsequently at the standard pressure (1bar).

$$q_i = \left(\sum_{k=1}^{NS} \alpha_{ki} [X_k]\right) \left(K_{fi} \prod_{k=1}^{NS} [X_k]^{\upsilon'_{ki}} - K_{ri} \prod_{k=1}^{NS} [X_k]^{\upsilon''_{ki}}\right) \tag{3.6}$$

Note that, in Equation 3.6, third bodies can be molecules involved in that very reaction, molecules involved only in other reactions or inert molecules that do not react. Therefore, the presence of a non-reacting molecule can have an effect on the mixture's reaction rates.

It is important to highlight that, if a reaction involves electrons, it must have its reverse rates explicitly defined or only react in its forward direction.

An alternative formulation for the forward reaction constants is Landau-Teller, although it was not used in this application, mainly since most reactions in chemkin datafiles only include Arrhenius constants. Pressure dependence was not included in this formulation.

It should be pointed out that, if a change of unit is desired (with respect to the units in which the Arrhenius constants are given), it is best to perform it after the rates are computed. Both the rate of reaction and the reaction rate of a specie have set units ($[\frac{mol-kg}{m^3 s}]$), but the intermediate values such as the forward and backward constants do not, as their units depend on the specific reaction under consideration. Consequently, on Equation 3.5, R_m is used as the gas constant, in order to use the same units as the Arrhenius constants and delay the unit conversion.

3.2 Equations

From the Navier-Stokes equations, disregarding viscous and non-stationary terms the following system of equations can be obtained:

$$\frac{d}{dx}(\rho v A) = Q_1 \tag{3.7}$$

$$\rho v \frac{dn_j}{dx} = \dot{\omega}_j + Q_j \tag{3.8}$$

$$\rho v \frac{dv}{dx} + \frac{dP}{dx} = Q_2 \tag{3.9}$$

$$\frac{dh}{dx} + v\frac{dv}{dx} = Q_3 \tag{3.10}$$

Where the terms Q_1, Q_2, Q_3, Q_j are generic terms that may be used to account for effects not explicitly present in the equations. And n_j represents the number of kg-mol of a specie per kg of mixture. The state equation is:

$$P\rho = zR_gT \tag{3.11}$$

Where the compressibility factor (z) has been included. Given the definition of h given in section 2.1, Equation 2.7 we can express its longitudinal derivative as:

$$\frac{dh}{dx} = \sum_{j=1}^{NS} H_j^o \frac{dn_j}{dx} + \left(\sum_{j=1}^{NS} n_j \frac{dH_j^o}{dT}\right)\frac{dT}{dx} = \sum_{j=1}^{NS} h_j \frac{dn_j}{dx} + C_{p,f}\frac{dT}{dx} \tag{3.12}$$

Where $C_{p,f}$ follows the definition of Equation 2.28. Substituting Equation 3.8 and Equation 3.12 into Equation 3.10 the following expression is obtained:

$$\sum_{j=1}^{NS} H_j^o \left(\frac{\dot{\omega}_j + Q_j}{\rho v}\right) + C_{p,f}\frac{dT}{dx} + v\frac{dv}{dx} = Q_3 \tag{3.13}$$

Differentiating the equation of state and substituting it into Equation 3.8 the following equation is obtained:

$$\frac{1}{P}\frac{dP}{dx} = \frac{1}{\rho}\frac{d\rho}{dx} + \frac{1}{z}\frac{dz}{dx} + \frac{1}{T}\frac{dT}{dx} + \frac{R}{R_g}\sum_{j=1}^{NS}\left(\frac{\dot{\omega}_j + Q_j}{\rho v}\right) \tag{3.14}$$

Substituting into it Equation 3.8 the following equation is obtained:

$$\frac{1}{P}\frac{dP}{dx} = \frac{1}{\rho}\frac{d\rho}{dx} + \frac{1}{z}\frac{dz}{dx} + \frac{1}{T}\frac{dT}{dx} + \frac{R}{R_g}\sum_{j=1}^{NS}\left(\frac{\dot{\omega}_j + Q_j}{\rho v}\right) \tag{3.15}$$

Substituting Equation 3.9 into Equation 3.13 and dividing it by R_gT, Equation 3.16 is obtained.

$$\frac{1}{R_gT}\sum_{j=1}^{NS} H_j^o \left(\frac{\dot{\omega}_j + Q_j}{\rho v}\right) + \frac{C_{p,f}}{R_gT}\frac{dT}{dx} + \frac{Q_2 - \frac{dP}{dx}}{\rho R_g T} = Q_3 \tag{3.16}$$

Reordering Equation 3.7, Equation 3.8, Equation 3.9, Equation 3.16 and Equation 3.15; the following system of equations can be obtained:

$$\frac{1}{v}\frac{dv}{dx} + \frac{1}{\rho}\frac{d\rho}{dx} + \frac{1}{A}\frac{dA}{dx} = \frac{Q_1}{\rho v A} \tag{3.17}$$

$$\frac{dn_j}{dx} = \frac{\dot{\omega}_j + Q_j}{\rho v} \tag{3.18}$$

$$\frac{1}{v}\frac{dv}{dx} + \frac{zR_gT}{Pv^2}\frac{dP}{dx} = \frac{Q_2}{\rho v^2} \tag{3.19}$$

$$\frac{1}{T}\frac{dT}{dx} - \frac{R_g}{C_{p,f}}\frac{z}{P}\frac{dP}{dx} = \frac{R_g}{C_{p,f}}Q_3 - \frac{Q_2}{\rho T C_{p,f}} - \frac{1}{C_{p,f}T}\sum_{j=1}^{NS} H_j^o\left(\frac{\dot{\omega}_j + Q_j}{\rho v}\right) \tag{3.20}$$

$$\frac{1}{\rho}\frac{d\rho}{dx} + \frac{1}{z}\frac{dz}{dx} + \frac{1}{T}\frac{dT}{dx} - \frac{1}{P}\frac{dP}{dx} = -\frac{R}{R_g}\sum_{j=1}^{NS}\left(\frac{\dot{\omega}_j + Q_j}{\rho v}\right) \tag{3.21}$$

Which can be written in matrix form as:

$$
\begin{bmatrix}
0 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & \frac{zR_gT}{v^2} & 0 & 0 & 0 \\
0 & 0 & 0 & -z\frac{R_g}{C_{p,f}} & 0 & 1 & 0 \\
1 & 0 & 0 & -1 & 1 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
\frac{1}{z}\frac{dz}{dx} \\
\frac{dn_j}{dx} \\
\frac{1}{v}\frac{dv}{dx} \\
\frac{1}{P}\frac{dP}{dx} \\
\frac{1}{\rho}\frac{d\rho}{dx} \\
\frac{1}{T}\frac{dT}{dx} \\
\frac{1}{A}\frac{dA}{dx}
\end{bmatrix}
=
\begin{bmatrix}
q_1 \\
\left(\frac{\dot{\omega}_j+Q_j}{\rho v}\right) \\
q_2 \\
\frac{R_g}{C_{p,f}}(q_3 - q_2\frac{v^2}{R_gT}) - \mathcal{A} \\
-\mathcal{B}
\end{bmatrix}
\tag{3.22}
$$

Where a set of new variables have been introduced:

$$
\begin{aligned}
q_1 &= \frac{Q_1}{\rho v A} \\
q_2 &= \frac{Q_2}{\rho v^2} \\
q_3 &= \frac{Q_3}{R_gT} \\
\mathcal{A} &= \frac{R}{C_{p,f}}\sum_{j=1}^{NS}\frac{H_j^o}{RT}\left(\frac{\dot{\omega}_j+Q_j}{\rho v}\right) \\
\mathcal{B} &= \sum_{j=1}^{NS}\left(\frac{\dot{\omega}_j+Q_j}{\rho v}\right)
\end{aligned}
\tag{3.23}
$$

In order to use the pressure as the independent variable a set of operations must be performed on the system presented in Equation 3.22. Therefore, (I) will be replaced by (I)-(III)+(IV)-(V) and (V) will be replaced by (V) - (IV). The resulting system of equations is:

$$
\begin{bmatrix}
-1 & 0 & 0 & -\frac{zR_gT}{v^2} - z\frac{R_g}{C_{p,f}} + 1 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & \frac{zR_gT}{v^2} & 0 & 0 & 0 \\
0 & 0 & 0 & -z\frac{R_g}{C_{p,f}} & 0 & 1 & 0 \\
1 & 0 & 0 & -1 + z\frac{R_g}{C_{p,f}} & 1 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\frac{1}{z}\frac{dz}{dx} \\
\frac{dn_j}{dx} \\
\frac{1}{v}\frac{dv}{dx} \\
\frac{1}{P}\frac{dP}{dx} \\
\frac{1}{\rho}\frac{d\rho}{dx} \\
\frac{1}{T}\frac{dT}{dx} \\
\frac{1}{A}\frac{dA}{dx}
\end{bmatrix}
=
\begin{bmatrix}
q_1 - q_2 + \frac{R_g}{C_{p,f}}(q_3 - q_2\frac{v^2}{R_gT}) - \mathcal{A} + \mathcal{B} \\
\left(\frac{\dot{\omega}_j+Q_j}{\rho v}\right) \\
q_2 \\
\frac{R_g}{C_{p,f}}(q_3 - q_2\frac{v^2}{R_gT}) - \mathcal{A} \\
-\mathcal{B} - \frac{R_g}{C_{p,f}}(q_3 - q_2\frac{v^2}{R_gT}) + \mathcal{A}
\end{bmatrix}
\tag{3.24}
$$

Generally it would be desired to use the Area as a function of the longitudinal coordinate of the nozzle, x, as an independent variable. Even though this variable is known beforehand (since we are computing the evolution through a given nozzle), it cannot be used. If the first equation of the matrix is used to compute the pressure value near the throat as a function of the area a problem arises. Near the throat, the term $1/\left(1 - \frac{zR_gT}{v^2} - z\frac{R_g}{C_{p,f}}\right)$ becomes singular (even if it is not apparent). If we use frozen variables and assume z=1, that term could be substituted by $\frac{\gamma u^2}{u^2-a^2}$. This term is clearly singular at the throat. As was mentioned beforehand, this conversion cannot be used and the speed at the throat is not necessarily the frozen speed of sound. Nonetheless, this serves as a way to show the singularity of this term near the throat. Therefore, at least near the throat, the independent variable has to be the pressure. As will be explained in section 3.3 a pressure distribution will need to be assumed and corrected in various iterations.

$$\begin{aligned}
&\frac{dn_j}{dx} = \left(\frac{\dot{\omega}_j + Q_j}{\rho v}\right)\\
&\frac{1}{v}\frac{dv}{dx} = q_2 - \frac{zR_gT}{v^2}\mathcal{C}\\
&\frac{1}{T}\frac{dT}{dx} = \frac{R_g}{C_{p,f}}(q_3 - q_2\frac{v^2}{R_gT}) - \mathcal{A} + z\frac{R_g}{C_{p,f}}\mathcal{C}\\
&\frac{1}{\rho}\frac{d\rho}{dx} = -\mathcal{B} - \frac{R_g}{C_{p,f}}(q_3 - q_2\frac{v^2}{R_gT}) + \mathcal{A} - \frac{1}{z}\frac{dz}{dx} - \left(z\frac{R_g}{C_{p,f}} - 1\right)\mathcal{C}\\
&\frac{1}{A}\frac{dA}{dx} = \left(q_A + \frac{1}{z}\frac{dz}{dx} + \left(\frac{zR_gT}{v^2} + z\frac{R_g}{C_{p,f}} - 1\right)\mathcal{Y}\right)\mathcal{I} + (1-\mathcal{I})\mathcal{Y}\\
&\frac{1}{P}\frac{dP}{dx} = \mathcal{C}
\end{aligned} \tag{3.25}$$

The system in Equation 3.24 will be rewritten introducing the variable $\mathcal{Y}(x)$, which represents a known function of the longitudinal coordinate (x), let it be pressure ($\frac{1}{P}\frac{dP}{dx}$) or area ($\frac{1}{A}\frac{dA}{dx}$) depending on which part of the nozzle the calculation is taking place in. In order to differentiate both cases, a parameter $\mathcal{I}$ will be introduced, which can take two values: either 1 for pressure or 0 for area. The final system of equations is given in Equation 3.25. Where two new variables have been introduced:

$$\begin{aligned}
&\mathcal{C} = \left(q_A + \frac{1}{z}\frac{dz}{dx} - \mathcal{Y}\right)\left(1 - \frac{zR_gT}{v^2} - z\frac{R_g}{C_{p,f}}\right)^{-1}(1-\mathcal{I}) + \mathcal{I}\mathcal{Y}\\
&q_A = q_1 - q_2 + \frac{R_g}{C_{p,f}}(q_3 - q_2\frac{v^2}{R_gT}) - \mathcal{A} + \mathcal{B}
\end{aligned} \tag{3.26}$$

3.3 One Dimensional Chemical Kinetics (ODK)

In order to solve the system of differential equations presented in Equation 3.25 an iteration must be performed. Moreover, a set of assumptions must be made. Firstly, the only user-fixed parameters are the chamber pressure, nozzle geometry and propellant. Note that, if the user were to set a grain geometry, the whole process soon to be described would need to be iterated in order to obtain the pressure at the chamber. The combustion chamber and first section of the nozzle are assumed to be in chemical equilibrium (calculated as explained in section 2.5). Furthermore, a pressure for the throat region must be assumed since, as was explained in section 3.2 in this region the expression for given area becomes singular. Said initial estimation for the pressure distribution near the throat will be the one calculated by ODE.

There are several targets that need to be satisfied in order to confidently say that the solution is correct. Firstly, all area values for the near-throat points calculated with the pressure distribution need to coincide with the real values. Moreover, the minimum area calculated must match the throat's area both in value and position. In order to be able to match those values a set of tests and iterations will be performed. As is to be expected, if this process is performed successfully, no discontinuities should be present at the switches between given area and pressure.

Given that the description of the nozzle is discrete (and therefore the pressure evolution as well), for the integration of the ODK equations in between two points the value of area/pressure and its derivative must be interpolated from nearby points.

The first point of the nozzle is calculated using ODE. Therefore, the initial conditions for the differential equation system are obtained through an incorrect value of the throat's parameters. In consequence, the initial conditions must be changed to match the values at the throat calculated by ODK. Said conditions are all determined by the mass flow at the throat (which is used to determine all other parameters in ODE calculations). For the initial estimate, the minimum area given by ODK will either be smaller or bigger than the one calculated by ODE; consequently, the mass flow for the initial section ODE calculation will be changed in order to more closely match the throat area, following Equation 3.27.

$$\dot{m}^{(k+1)} = \dot{m}^{(k)} \cdot exp\left(\left(\ln \frac{\rho|_{A_{min}} u|_{A_{min}} A_t}{\dot{m}^{(k)}}\right) \cdot 150 \frac{A_t - A_{min}}{A_t}\right) \tag{3.27}$$

The convergence criteria is:

$$\left|\frac{A_t - A_{min}}{A_t}\right| \leq 4 \cdot 10^{-4} \tag{3.28}$$

Given that the ODK equations are dependant on the history up to that point, all points up to the throat must be computed in order to obtain the values at the point of minimum area. Moreover, this point could be further downstream from the geometrical throat. Furthermore, due to the necessity to interpolate the pressure, more points will need to be computed past the change of independent variable back to the area. This way the interpolation for the pressure will be valid for the current iteration. Once the magnitude of the minimum area corresponds to the geometrical throat, a correction on the pressure estimation near the throat is also conducted:

$$\begin{aligned}
P(x)^{(k+1)} &= \frac{P_{inf}}{exp\left(\left(\ln\frac{P_{inf}}{P(x)^{(k)}}\right)+\sigma\right)} && if(x \neq x|_{A_t}) \\
P(x)^{(k+1)} = P^{(k)}exp&\left(\left(\frac{Interpolation(P)}{P^{(k)}}\right)\cdot 0.1\cdot min\left(\frac{x|_{A_{min}}}{\Delta x}, 1.0\right)\right) && if(x = x|_{A_t})
\end{aligned} \tag{3.29}$$

Where σ is defined as a convergence criteria for all points except the throat itself:

$$\begin{aligned}
|\sigma| = \left|\frac{\ln\frac{A(x)}{A(x)|_{ODK}}}{min((1-\frac{zR_gT}{v^2}-z\frac{R_g}{C_{p,f}}), -10^{-2})}\right| &\leq 4\cdot 10^{-4} && if(x < x|_{A_t}) \\
\left|\frac{A_t|_{ODK}-A_t}{A_t}\right| &\leq 4\cdot 10^{-4} && if(x = x|_{A_t}) \\
|\sigma| = \left|\frac{\ln\frac{A(x)}{A(x)|_{ODK}}}{max((1-\frac{zR_gT}{v^2}-z\frac{R_g}{C_{p,f}}), 10^{-2})}\right| &\leq 4\cdot 10^{-4} && if(x > x|_{A_t})
\end{aligned} \tag{3.30}$$

If the convergence test is not passed the correction is applied and the calculations are performed again. Given that the pressure value near and at the throat has been modified, the mass flow will also experience some variation. This implies that the first section will need to be recalculated using ODE and all the nozzle up to the second switch of independent variable will be recalculated using ODK. This will change the computed pressure and area distributions (in their respective domains). In order to come back to this step, the first convergence criteria must be passed again.

It is important to highlight that the values for A_{min} and $A_t|_{ODK}$ are not necessarily the same. The first one represents the minimum value of the area calculated by ODK, this value can be located anywhere and not coincide with any of the points employed in the calculation. On the other hand, the second value corresponds to the area value calculated by ODK at the point corresponding to the geometrical throat. Since this two values need to be equal to eachother, the convergence criteria is the same for both (although one is mainly affected by the mass flow value whilst the other by the pressure at the throat and points prior to it). Furthermore, it should also be chequed that the point at which the minimum area occurs concides with (or at least is very close to) the geometrical throat. Consequently, when all the previous convergence criteria are passed an additional one is set in place:

$$\left|\frac{x|_{A_{min}}}{\Delta x}\right| \leq 0.1 \tag{3.31}$$

This ensures that the minimum area computed with ODK corresponds not only in value but also in position to the geometrical throat area. If this test is passed as well, the solution for the ODK problem is deemed correct. Otherwise, another correction is performed and the process is restarted:

$$\dot{m}^{(k+1)} = \dot{m}^{(k)} \frac{A_t}{A_{min}} \tag{3.32}$$

Once the process terminates, all the variables of interest for the various points are calculated, as well as the region of the nozzle past the second switch in independent variable, as it was not necessary to compute up until this point.

3.3.1 Convergence & number of nozzle points

The solver for the differential equations has the need to perform very small steps, due to the stiff nature of the problem. Since it would not be feasible to have such a small spacing within nozzle points, the program interpolates the values based on the surrounding nozzle points. Nonetheless, for a given precision in the tolerances selected for the calculation, a minimum amount of points describing the nozzle is needed.

One very clear indication of this is the conservation of the mass flow. Since the condition set for said conservation is not strict (instead of assigning it a value it is limited to a null derivative) the mass flow may experiment changes. It the value of Δx is very large and the tolerance for the solver is very wide, those changes become more prominent. As seen in Equation 3.27 the iteration is very susceptible to changes in mass flow. Moreover, the convergence for mass flow is the first and most important, since it determines the behaviour of the entire problem.

Moreover, an excess amounts of points would not only require the program to achieve convergence in a greater quantity of points, but it would also complicate convergence for the throat position, since the value for Δx would be excessively small.

In spite of this occurrences, for amounts of points that the average user would regularly use, convergence is achieved. Moreover, varying the number of points has no effect (inside a not extreme range of values). The set tolerances play a more important role in convergence. A more detailed explanation along with examples can be found in **??**.

4

Other models used in ODEKO

Contents

4.1 Models for nozzle curves

4.1.1 Conical nozzle

Currently, the program offers two types of nozzle geometries. Firstly, a conical nozzle, with the option to add fillets. If a fillet radius is specified, the cone will be tangent to the fillet circumference. Fillets can be added at the entrance, exit and throat, as seen in Figure 4.1. If the user specifies an angle for the convergent or divergent region, the length of said region is calculated as seen in Equation 4.1, where L is the length of the nozzle region (convergent or divergent), R_{fe}, R_{ft} are the fillet radii for the entrance (or exit) and throat respectively and α is the slope of the cone. On the contrary, if the user specifies the length, the slope of the cone is obtained by solving Equation 4.1.

$$L = (R_{fe} + R_{ft})\sin\alpha + [(R_e - R_t) - (R_{fe} + R_{ft})(1 - \cos\alpha)]\frac{\cos\alpha}{\sin\alpha} \tag{4.1}$$

Moreover, a set of checks are set in place in order to ensure that the nozzle can be constructed. The checks for set angle are given in Equation 4.2 and for set length in Equation 4.3, where R_e can symbolize either exit radius or entrance radius (depending on the region in question).

$$\begin{gathered}(R_{fe} + R_{ft})(1 - \cos\alpha) < R_e - R_t \\ 0 < \alpha < \frac{\pi}{2}\end{gathered} \tag{4.2}$$

$$\begin{gathered}L^2 + (R_e - R_t - R_{fe} - R_{ft})^2 > (R_{fe} + R_{ft})^2 \\ L\sqrt{L^2 + (R_e - R_t - R_{fe} - R_{ft})^2 - (R_{fe} + R_{ft})^2} > (R_{fe} + R_{ft})(R_e - R_t - R_{fe} - R_{ft})\end{gathered} \tag{4.3}$$

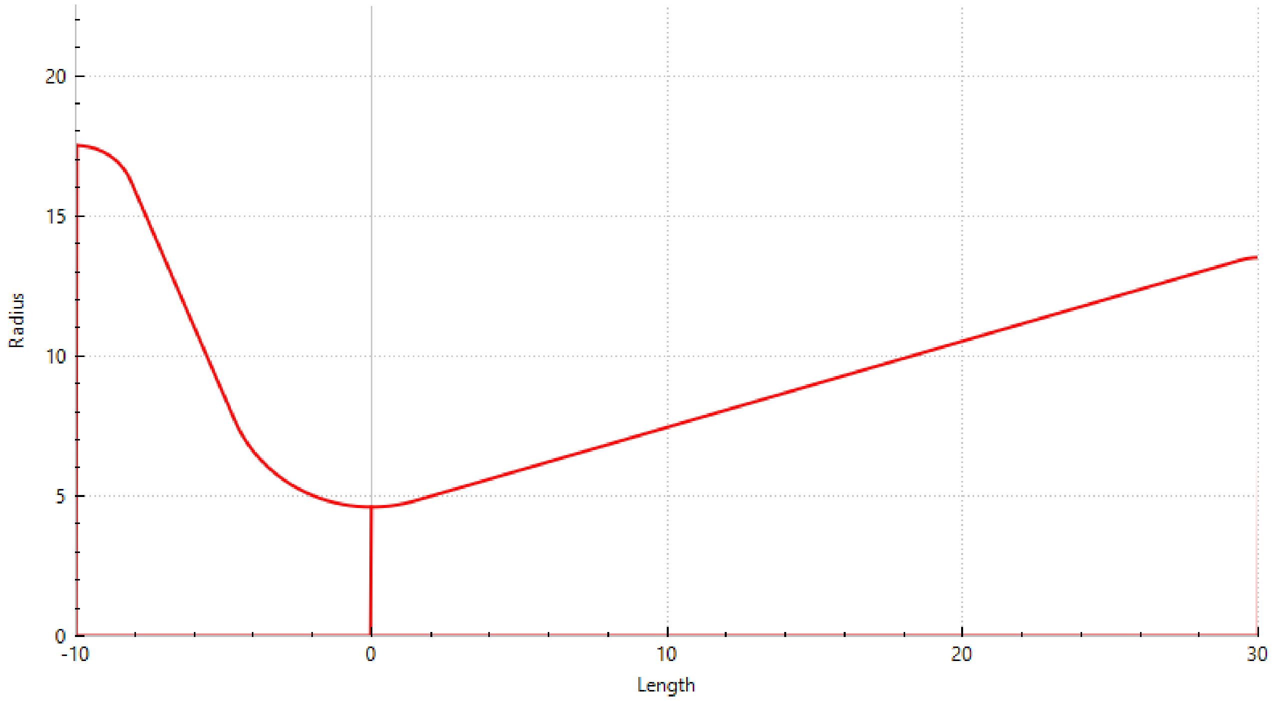


Figure 4.1: Example of conical nozzle

4.1.2 Bell nozzle

Bezier curves (also known as bell nozzles or parabolic nozzles) are also available. The convergent region is modelled as a conical nozzle, but offering the user the possibility to specify different fillet radius for the convergent and divergent regions of the throat. The divergent region consists of a parabola tangent to the fillet circumference at the throat. The user may specify its initial and final angle. An example for a Bezier curve is present in Figure 6.15.

The equation for the Bezier curve is given in Equation 4.4, where N and E are the first and last points of the parabola and Q is the intersection point between the tangent lines at N and E. A set of checks are set in place in order to ensure that the angles specified by the user are valid, described in Equation 4.5, where R_{ftd} is the fillet radius at the divergent region of the throat, α_0 is the initial angle of the parabola and α_f is the final angle of the parabola. This checks ensure that the set of parameters given define a valid nozzle geometry.

$$\begin{aligned} x(t) &= (1-t^2)N_x + 2(1-t)t \cdot Q_x + t^2 E_x \\ y(t) &= (1-t^2)N_y + 2(1-t)t \cdot Q_y + t^2 E_y \end{aligned} \tag{4.4}$$

$$\begin{aligned} L_{div} > R_{ftd} \sin \alpha_0 &= N_x \\ R_t < Q_y &< R_{exit} \\ 0 < Q_x &< L_{div} \\ \alpha_0 &> \alpha_f \end{aligned} \tag{4.5}$$

4.2 Combustion area for different grain geometries

4.2.1 Cylindrical geometry

The program allows the user to choose between frontal, cylindrical and star geometry. For the later two, the user may also specify if the charges burn radially, at the bases or both. In the case of the cylindrical charges with 3 combustion fronts (bases + radial), the expression used is the following:

$$A_b = n_c 2\pi \left(R_{e,c}^2 - (R_{i,c} + y)^2 + (R_{i,c} + y)(L_c - 2y) \right) \qquad 0 < y < y_f \tag{4.6}$$

In which y_f is the smallest of $R_{e,c} - R_{i,c}$ & $\frac{L_c}{2}$. This expression can also be used to calculate for which charge length there is a smaller deviation on the combustion area values (for which the combustion is closer to neutral). This optimization process can be performed in the application. Since computing hundreds of values is extremely fast, the combustion evolution for a set of lengths are computed and the one with the least deviation is selected as the optimum.

4.3 Convective heat coefficient

The convective heat transport coefficient (h_g) can be written in terms of the Nusselt number as:

$$h_g = Nu \cdot \frac{\lambda_f}{D} \tag{4.7}$$

As shown in [2] the Nusselt number in pipelines can be expressed as:

$$Nu = c \cdot Re^m Pr^n \tag{4.8}$$

In order to achieve a representative value for the convective heat transport coefficient variables should use intermediate values between their value at the nozzle wall and their static value. In this case, the average of both values was taken (marked with sub-index 'am'). Average values for the Reynolds, Prandtl and Nusselt numbers can be written as:

$$Re_{am} = \frac{\rho_{am} \cdot u \cdot D}{\mu_{am}} = \rho u \frac{T}{T_{am}} \frac{D}{\mu_{am}} = \frac{u_t \rho_t D_t}{\mu} \left(\frac{A_t}{A}\right)^{\frac{1}{2}} \left(\frac{T}{T_{am}}\right)^{1+\omega_\mu} \tag{4.9}$$

$$Pr_{am} = \frac{\mu_{am} C_{p,am}}{\lambda_{am}} = \frac{\mu C_p}{\lambda} \left(\frac{T_{am}}{T}\right)^{\omega_\mu - \omega_\lambda + \omega_c} \tag{4.10}$$

$$Nu_{am} = \frac{h_{g,am} D}{\lambda_{am}} = \frac{h_{g,am} D_t}{\lambda} \left(\frac{T}{T_{am}}\right)^{\omega_\lambda} \left(\frac{A}{A_t}\right)^{\frac{1}{2}} \tag{4.11}$$

Pressure has been assumed constant throughout the boundary layer and transport properties have been considered exclusively dependant on temperature: $\mu \propto T^{\omega_\mu}$; $\lambda \propto T^{\omega_\lambda}$; $C_p \propto T^{\omega_c}$.

The convective heat transport coefficient can be written using Equation 4.9 to Equation 4.11. In Equation 4.12 the expression was written with an additional term, in order to take into account the curvature radius at the throat (κ).

$$h_{g,am} = c \frac{\lambda}{D_t} \left(\frac{u_t \rho_t D_t}{\mu}\right)^m \left(\frac{\mu C_p}{\lambda}\right)^n \left(\frac{2T}{T + T_w}\right)^{(1+\omega_\mu)m - \omega_\lambda - n(\omega_\mu - \omega_\lambda + \omega_c)} \left(\frac{A_t}{A}\right)^{\frac{m+1}{2}} \left(\frac{D_g}{\kappa}\right)^{\omega_\kappa} \tag{4.12}$$

$$c = 0.026 \qquad m = 0.8 \qquad n = 0.4 \qquad \omega_\mu = 0.6 \qquad \omega_\lambda = 0.6 \qquad \omega_c = 0 \qquad \omega_\kappa = 0.1 \tag{4.13}$$

The default values for the constants are shown in Equation 4.13. The user is able to edit those values in the preferences window. A temperature for the wall of the nozzle must be specified by the user. An example plot for the value of the convective heat transport coefficient was also included in Figure 4.2. The calculation was performed for SpaceX's Raptor engine (described in subsection 6.3.1). It is notable that, as it is widely know, refrigeration is of the uttermost importance in the convergent nozzle and, specially, at the throat, whilst being not as critical in the divergent region.

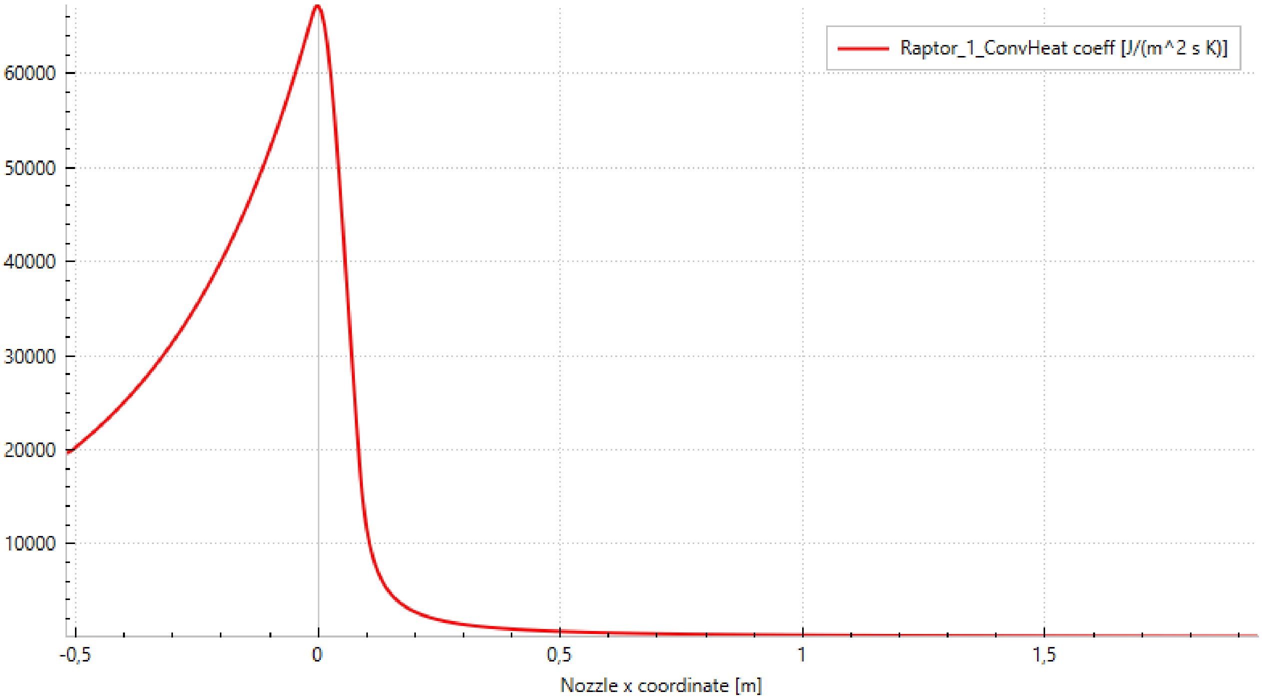


Figure 4.2: Convective heat transport coefficient along the nozzle

4.4 Optimization process

4.4.1 Nozzle optimization

The main optimization is performed in the Nozzle tab, where the user can obtain the optimum values for the throat radius and the area ratio at the exit point. Several targets can be set for optimization: chamber pressure, combustion time, thrust, specific impulse and total impulse. The first two take a target value that the user wants to achieve. The later three search for the maximum value of the target, whilst complying with the restrictions set by the user: minimum and maximum chamber pressure, minimum throat area and minimum area ratio at the exit. As well as restrictions set by the nozzle: throat area must be strictly smaller than the entrance area and exit area.

Fixed chamber pressure

Although it is grouped as a optimization, it does not require any optimization procedure, as the value can be directly calculated. Nevertheless, it is used as part of the other optimization processes. Since the chamber pressure is given by the user, the pressure at the throat can be directly calculated as seen in chapter 2. Moreover, employing Vielle's equation, it is also posible to calculate the mass flow created by the combustion process. Since said mass flow needs to equal the flow through the nozzle, the area of the throat can easily be calculated as:

$$A_t = \frac{G}{u_t \rho_t} \tag{4.14}$$

Where the mass flow was given by the combustion and the throat parameters can be easily calculated with the ODE model knowing the chamber pressure (without the need to specify a throat area). Note that the combustion area used needs to be a fixed value (with a fixed throat, the pressure must vary if so does the combustion area). The user may choose between the initial, final, maximum, minimum or average combustion area.

Fixed combustion time

The initial estimate for the chamber pressure for a given combustion time is performed assuming a constant area of combustion (equal to the average combustion area of the grain geometry). The recession rate can be calculated as: $\dot{r} = \frac{y_f}{t_b}$. Employing Vielle's law, a value for the chamber pressure can be obtained. Afterwards, the optimum throat area for said value is computed (as explained in the previous subsection). Finally, a calculation for the grain combustion for the given throat is performed as explained in subsection 2.5.4. Correction in Equation 4.15 is applied to the throat area until the computed burn time deviates less than 0.1% from the target value, where A_0 is the area at the entrance of the nozzle. Note that the correction performs small changes in the area value even for relatively big disparities between the computed and target time, this is because the combustion time is highly susceptible to the value of the throat area.

$$\begin{aligned} A_t^{k+1} &= A_t^k \cdot exp\left[\frac{A_t^{(k)}}{A_0} \ln\left(\frac{(t_b)_{target}}{(t_b)_{computed}}\right)\right] \qquad n_V < 1 \\ A_t^{k+1} &= A_t^k \cdot exp\left[\frac{A_t^{(k)}}{A_0} \ln\left(\frac{(t_b)_{computed}}{(t_b)_{target}}\right)\right] \qquad n_V > 1 \end{aligned} \tag{4.15}$$

Maximum thrust, specific impulse or total impulse

In order to locate the throat and exit area values for the maximum value of the target, a custom simple particle swarm method was employed. Firstly, a set of 10 equally spaced points in the user-defined allowable pressure range are selected (they must include both boundary points). Note that 10 is an arbitrary number and can be modified by the user. For said points using a prior optimization method, the optimal throat is calculated (for the given representative area of combustion). Afterwards, the exit of the nozzle is computed setting the pressure of the mixture equal to the ambient pressure (adapted flow); therefore the usual ODE iteration is not needed. Finally, the target value is computed. In the case of the total impulse, all area points are computed with the new nozzle in order to obtain the time evolution and be able to integrate the thrust values.

The two points with the highest target values are selected as the new boundaries of the range. The process is repeated until the range size is smaller than the set tolerance, at that point, the highest target value is selected as the optimum. Note that only points that comply with all the restrictions (such as minimum throat area) are eligible as new boundary points.

4.4.2 Reactant ratio optimization

Additionally, a simpler optimization can also be performed in the propellant tab. If only two reactants have been selected, the user is offered the possibility to optimize the ratio in between the two reactants for maximum c^*. The optimization process is trivial, a set of different values for the reactant relation are computed and the one with the highest c^* is selected. This method is feasible since calculations for hundreds of points barely take a couple seconds to compute.

Note that this optimization is performed for a given combustion pressure and reactant temperature that the user may specify in the preferences menu. Moreover, it is also important to clarify that the reactant ratio for maximum c^* may not always be the better propellant. This is due to the effect of the reactant ratio on the propellant weight, among other factors (such as nozzle refrigeration and tank pressurization). For example, H-LOX rockets generally use higher oxidizer ratio than the optimum value for c^*.

Moreover, in the propellant tab the user may also plot the evolution of the propellant c^* with chamber pressure, in order to study its effect. Generally, most propellants increase their c^* with pressure and a logarithmic-like curve.

5

User Manual

Contents

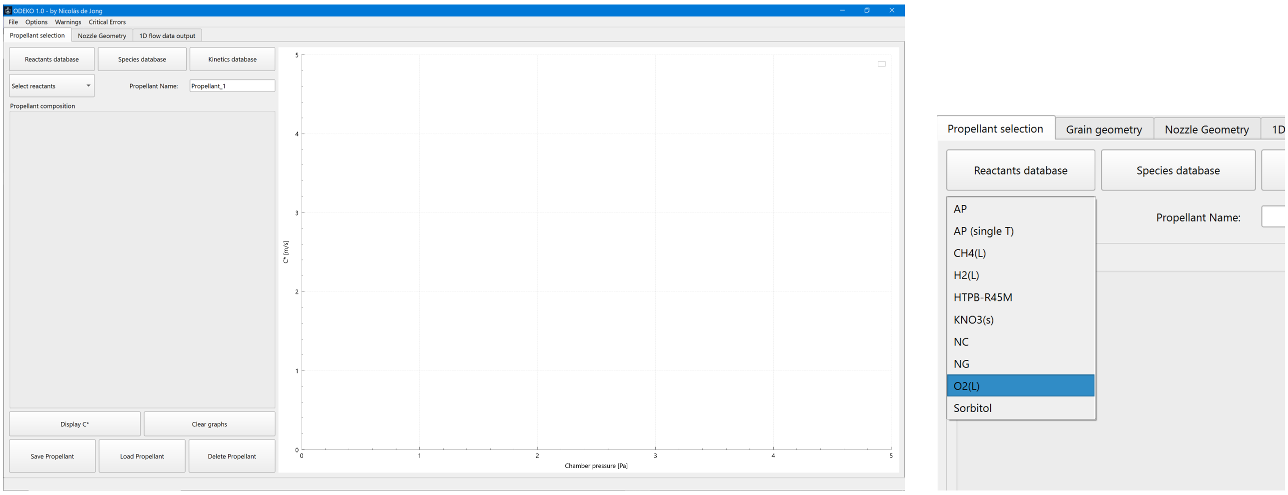
5.1 Propellant definition tab

This tab is in charge of the propellant definition. It allows the user to define their own propellants or modify already existing ones. It also offers the possibility to add information about Vielle's parameters or the species and chemical reactions to be considered during calculations (as well as modifying their respective databases). The propellants defined in this tab will be used in the calculations tab (and optionally in the nozzle tab in order to perform nozzle optimizations for solid propellants).

5.1.1 Definition of a new propellant

Right after installation, ODEKO comes packaged with a basic database that includes the most common species and reactions and example reactants and propellants. As will be subsequently explained in subsection 5.1.2, the user may alter the databases or add new elements to them. For the propellant definition, it will be assumed that the user has already added all necessary components to the databases or is using the default settings.

In Figure 5.1b the user has clicked on 'select reactant' opening a dropdown list with reactants present in the database. The user can select any set of reactants, which will appear on the space below, as seen in Figure 5.2. The program will not allow the user to add twice the same reactant. No other restrictions are applied. As shown in Figure 5.2, each reactant has a 'delete' button, allowing the user to remove it from the propellant. Each reactant also comes with an empty line box which allows the user to specify its relative mass in the propellant. The sum of all reactants relative mass does not need to add up to any specific number, each reactant will be assigned the ratio between its number and the total sum (calculations are performed per unit mass of total propellant, only the ratio is relevant).



Propellant Name: Propellant_1

(a) Empty propellant tab

(b) Selection of reactants

Figure 5.1: Selection of reactants in propellant tab

Once all reactants have been selected and their relative masses specified, the user may save the propellant. Note that if the definition is not properly done, the application will give out an error message and not save the propellant (this includes specifying a name for the propellant). If the propellant name is already taken, the application will let the user know that this data will be overwritten and allow them to cancel the operation. Furthermore, the software will also not allow the propellant to be saved if the selected reactions contain a specie that has not been selected itself (see section 5.1.2 and section 5.1.2).

It is important to remark that, when a propellant is saved it does not only include the information about its reactants (and optionally Vielle's parameters as explained in subsection 5.1.4) but it also includes information about the species and reactions to be considered during calculations involving said propellant. This information can be specified in the species and kinetics databases as explained in section 5.1.2 and section 5.1.2 respectively. As explained in subsection 5.1.5, this information is also loaded with the propellant both on this tab and on the calculations tab (see section 5.4).

Once the propellant has been saved it is immediately available on the calculations tab (see section 5.4) and on the optimizations section of the nozzle tab (see subsection 5.3.3).

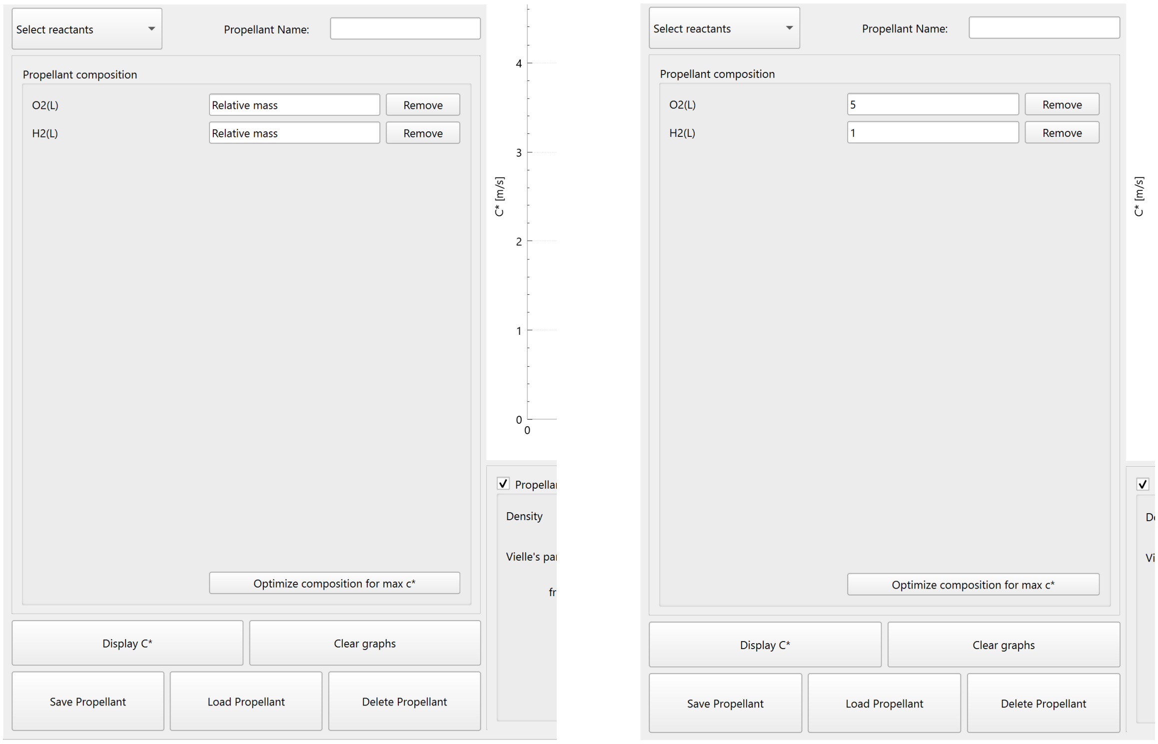


Figure 5.2: Two selected reactants in the composition list

5.1.2 Data Bases

The propellant tab has three buttons that allow the user to access the databases. It is recommended to the inexperienced user to not access this databases nor modify their content. On the other hand, the more experienced user is more than welcome to add new items or modify which species/kinetics are to be considered in the calculations. Note that ODEKO comes with default content for the databases which allows the user to simulate the most common types of propellants. Nonetheless, the user is able to add manually or through external files new items to the databases. The user is responsible for the correctness of the data that is input into the databases, since the simulation results are directly dependant on said information. ODEKO performs several checks when inputting data into the database (invalid values, coherence...) but the ultimate responsibility lies on the user's side.

Specie Database

This database contains the information about all the chemical species that could be considered by ODEKO. The user may edit them or add new ones. In order to add new species the user may either add them manually or import a thermodynamic data file, which follows NASA's CEA format (as explained in section 5.8.2). The user may also import transport data files to complement this information in already defined species (also in NASA's CEA format), but never as a means to define a new one (format explained in section 5.8.2). More input formats are expected to be made available in future releases.

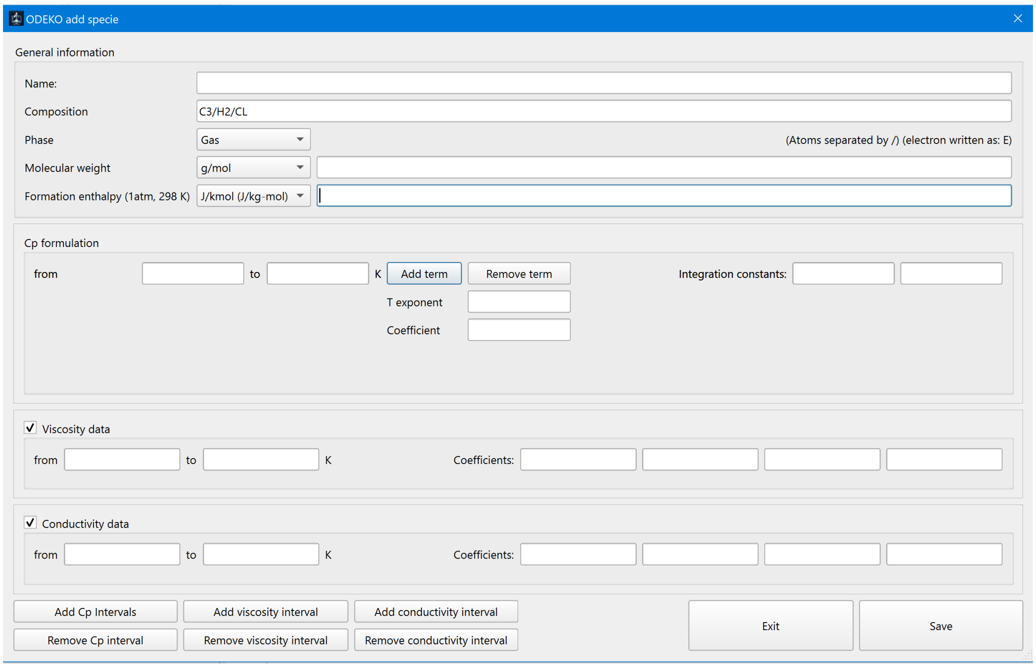
If the user edits a specie (or manually adds one) a window with all the information about the specie will be opened, where the user can edit it. This window is shown in Figure 5.3. Note that transport properties are not required to define the specie, but if the user wants to obtain accurate transport properties for the internal flow it is crucial to have transport properties defined for as many species as possible. An in depth-explanation of the required data for the transport properties (least-squares fitting) can be found at the full documentation of ODEKO.

Relevant parameters in specie definition (see Figure 5.3):

- **Name:** It is just used for the user to identify the specie, it can be any character string.
- **Composition:** It should be written as atom identifier (e.g. H for Hydrogen), followed by number of said atoms (it can be relative amount to the other atoms), if more atoms are present an inclined bar should be added to separate them (/). Example: *C3/H2/Cl* . For ions, the electron can be written as 'E' which should have a positive index for negative ions an negative index for positive ions. Note that ODEKO performs checks on this syntax and will let the user know if there are any errors present in the composition.
- **Molecular Weight:** Note the units, $\frac{g}{mol} = \frac{kg}{kg-mol}$ & $\frac{kg}{mol} = 10^{-3}\frac{kg}{kg-mol}$
- **Formation enthalpy:** Currently this value is not used in calculations (it is used for reactant

calculations, as explained in section 5.1.2).

- **Cp:** The user may define the Cp of the specie in as many intervals as he wishes, which any number of terms for the minimum squares fitting in each interval. Note the presence of two integration constants (regarding enthalpy and entropy calculations). For more detailed information consult the theory manual of the application.


ODEKO add specie

General information

Name:
Composition: C3/H2/CL
Phase: Gas
(Atoms separated by /) (electron written as: E)
Molecular weight: g/mol
Formation enthalpy (1atm, 298 K): J/kmol (J/kg-mol)

Cp formulation

from … to … K
Integration constants:
T exponent
Coefficient

☑ Viscosity data

from … to … K
Coefficients:

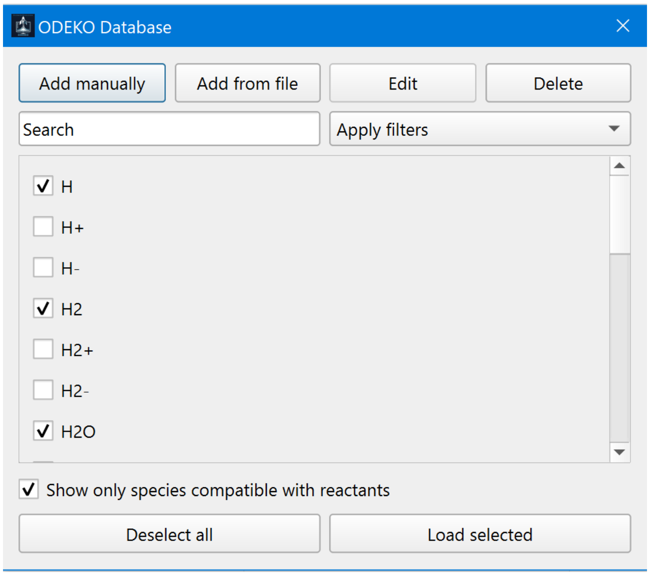
☑ Conductivity data

from … to … K
Coefficients:

Figure 5.3: Manual edit/addition of specie

Note that, by default, only the species compatible with the currently selected reactants are shown in the database. 'Compatible' means that said species are made exclusively out of atoms present in the reactants. The user may uncheck the box at the bottom of the window (see Figure 5.4) or modify the preferences (see section 5.6) in order to show all species present in the database instead. By default, incompatible species will be shown as 'disabled' which will not allow the user to unmark them or edit them, this behaviour can also be modified in the preferences window. Note that if the preferences are changed so that all species appear as enabled, even though incompatible species can be checked or unchecked that will not have any effect on the current propellant (although those selection will appear when a future propellant is compatible with said species).

Note that in the database window there are two separate behaviours on exit. If the user saves the loaded species, the check-state of the species will be saved, else it will be discarded and returned to its previous state. The checkboxes indicate which species are to be considered during calculations (out of the compatible ones). This information will be saved along with the propellant (and also loaded along a propellant). Considering all species will offer the most precise results, but will also be more computationally expensive. Therefore, the experienced user may unselect species which are known to not be present in the mixture at any point, in order to speed up the process. Note that not considering a specie present in the mixture will fundamentally alter the obtained results.



☑ H
☐ H+
☐ H-
☑ H2
☐ H2+
☐ H2-
☑ H2O

☑ Show only species compatible with reactants

Figure 5.4: Specie database

As seen in Figure 5.4, the user is able to perform a search in the database, or apply a set of filters. The search option will only show species whose name contains the specified character string. The filters will allow the user to choose which types of species are to be shown (all, gaseous, ions, solid, liquid, with transport properties). Note that if a search option or filter is applied, the buttons 'deselect all' and 'select all' (which after 'deselect all' has been pressed appears in its place) only affect the shown species (after the filter).

Reactant Database

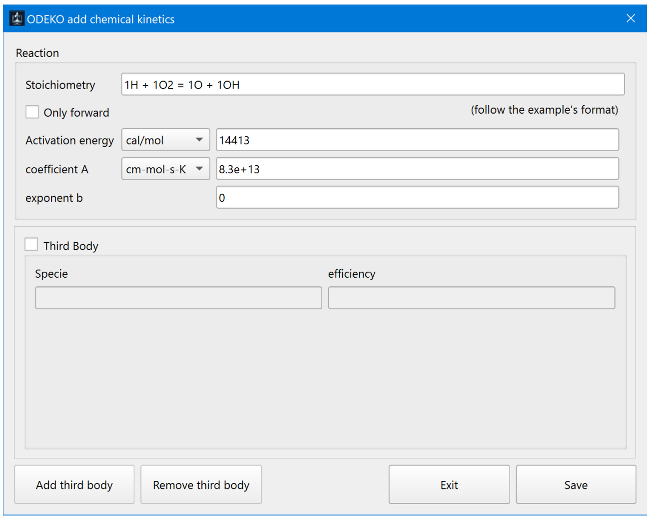
The database for reactants is quite similar to that for species, with an important distinction. Reactants do not require the user to specify a function for its C_p, it is possible to exclusively define their enthalpy of formation (along with the temperature of formation). The reasons behind this are further explained in the theory manual. Nonetheless, if the user wants to study the impact of initial propellant temperature on the rocket behaviour it is necessary to have C_p defined for the reactants, or else the same results will be given for all initial temperatures (the user will be given a warning message letting them know that the reactants are defined for a single temperature).

Moreover, the reactant database has no check/uncheck options, since reactants for the propellant are chosen as explained in subsection 5.1.1. Nonetheless, reactants may be selected to be edited. Therefore, it only has one behaviour on exit, since it saves no information about selected items. Furthermore, the reactant database also has search and filter options (all, gas, solid, liquid). Adding reactants manually and from a file function in the same fashion as species, except for the already mentioned differences.

Kinetics Database

ODK calculations (One Dimensional Kinetics) require a set of possible reactions between the species in the mixture to be specified, so does the calculation of equilibrium conductivity of the mixture as well as its Prandlt number (both properties require a set of reactions even wqhen computing through ODE, for details refer to the full documentation). The reactions of ODEKO's database can be accessed here. The behaviour of this database is quite similar to that of the species database. The user may check or uncheck reactions in order to indicate if they are going to be considered in the saved propellant or not. Note that said selections will also be loaded along with the propellant. The inclusion of reactions severely impacts the speed of computations of ODK. Therefore, it is recommended to find in the literature a suitable kinetic scheme that offers precise results for the given propellant without requiring a great amount of reactions to be considered.

Reactions may be added manually or through an import file which follows ANSYS CHEMKIN format, detailed in section 5.8.2. Adding (or editing) a reaction manually will open a window were all the relevant data can be specified (shown in Figure 5.5). Note that the definition of 'third bodies' is optional. It is important to keep in mind that, when writing the stoichiometry in a manually added reaction, only species that are defined in the specie database should be used. Same principle applies to the species considered as 'third bodies'.


ODEKO add chemical kinetics

Reaction

Stoichiometry: 1H + 1O2 = 1O + 1OH

☐ Only forward

(follow the example's format)

Activation energy: cal/mol 14413

coefficient A: cm-mol-s-K 8.3e+13

exponent b: 0

☐ Third Body

Figure 5.5: Manual addition of a chemical reaction

It is important to highlight that, as also happened in the species database, by default only compatible reactions are shown (reactions whose involved species themselves are also compatible). This behaviour can be changed in preferences (refer to section 5.6). Note that, if a reaction were to be selected but one of its species was not, a problem would arise. Therefore, ODEKO gives out a warning informing of this and does not allow the propellant to be saved until this problem is fixed (either deselecting the reaction or selecting the appropriate species). On the other hand, species present as 'third bodies' are not required to be included in the mixture. When using the propellant a warning will show up indicating that some considered third bodies are not present in the mixture. A the inclusion of third bodies not present in the mixture does not change the result of the calculations (but offers more versatility, so the reaction can be used in several different mixtures).

5.1.3 c^* graphs

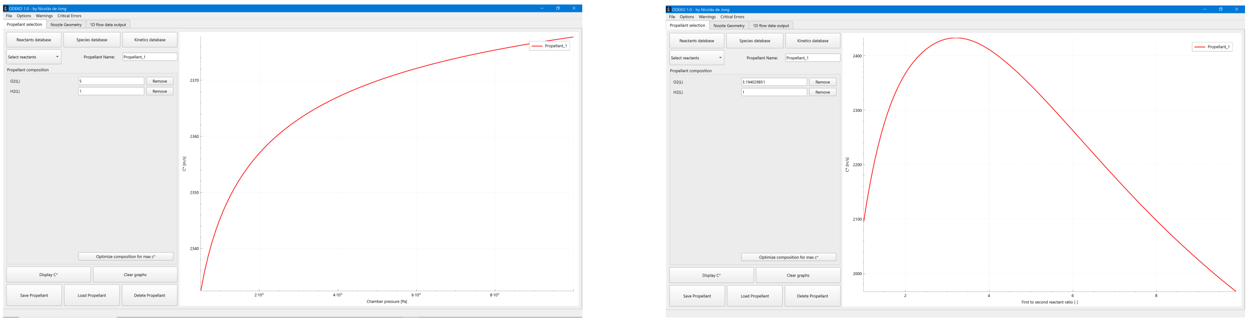
The user has the option to graph the value of the c^* of the given propellant with respect to pressure or mixture ratio. Note that this is performed on the same tab and does not require a grain or nozzle definition (it only accounts for combustion chamber values).

c^* with respect to chamber pressure

At any time, if at least one reactant is present the user may plot the c^* of the propellant with respect to pressure, by clicking on the button on the lower left of the screen (Figure 5.6a). As many plots as desired can be represented in the same graph (the user may change the composition and re-plot the graph). Note that the name for each graph present in the legend is the name that was given to the propellant. If there are two or more plots with the same name, a numeric index will be appended in order to differentiate them. If a different type of curve were to be plotted (c^* with respect to mixture ratio) all current plots of the graph would be cleared (since the x axis is changing). The upper and lower limits of the x axis may be modified in the preferences menu (refer to 5.6).

c^* with respect to mixture ratio

This type of curves may only be plotted if exactly two reactants are present. When this condition is satisfied, a button will appear giving the user this option (Figure 5.6b). Note that the mixture ratio in the x axis is the ratio between the first reactant added and the second. Therefore, it would be wise to have the oxidizer as the first reactant for most common propellant combinations. Nonetheless, the user may change the lower and upper limits of the x axis at any time in the preferences menu (refer to section 5.6). The plot is also performed for a constant chamber pressure, which also may be modified in the preferences menu. This plot also performs an "optimization" where the highest value of the propellant's c^* is found in the given mixture ratio range; afterwards, said ratio is set as the reactants relative masses. As was mentioned in section 5.1.3, the user may plot as many curves as they want as long as they are of the same type (mixture ratio in this case), plotting c^* over pressure would change the x axis and in consequence clear the previous plots.



Propellant Name: Propellant_1
O2(L) 5
H2(L) 1

Propellant Name: Propellant_1
O2(L) 3.194029851
H2(L) 1

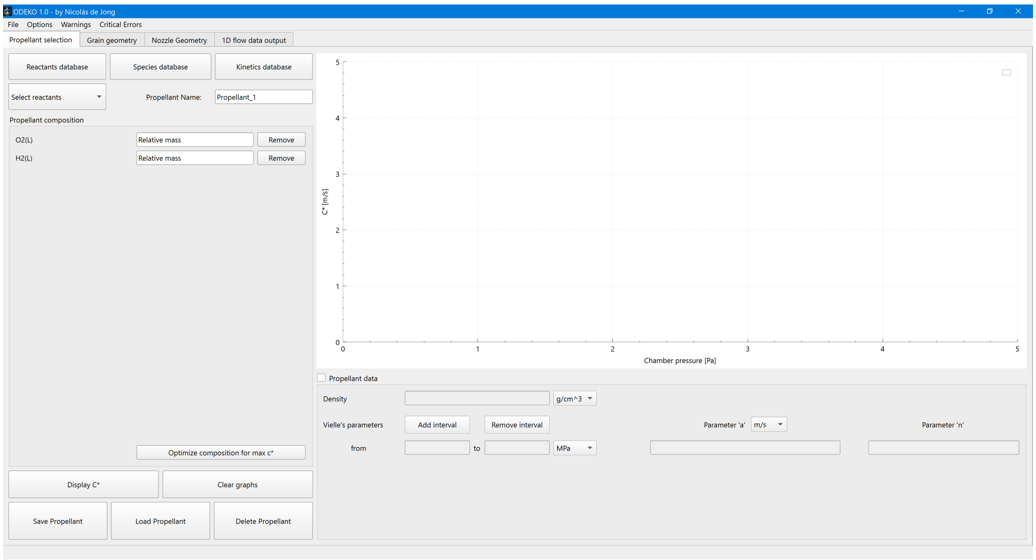
(a) c^* with respect to pressure

(b) c^* with respect to mixture ratio

Figure 5.6: c^* plots in ODEKO for HLOX propellant

5.1.4 Vielle's parameters

The user also has the option (although it is not required) to specify the propellant's Vielle's parameters in the lower right section of the propellant tab. Note that this region is hidden by default (as all options related to grain geometries and solid motor simulation are hidden in a default installatioin). Nevertheless, the user can toggle through the options menu (section 5.5.2), as well as set the solid grain options enabled by default in preferences (section 5.5.2). On Figure 5.1a the propellant tab with hidden solid motor information was shown, whilst on Figure 5.7 the user can see the same tab when said option was untoggled in the options menu.



Propellant Name: Propellant_1
O2(L) Relative mass
H2(L) Relative mass
☐ Propellant data
Density g/cm^3
Vielle's parameters
Parameter 'a' m/s
Parameter 'n'
from to MPa

Figure 5.7: Propellant tab with shown solid motor information

Specifying Vielle's parameters as well as propellant density is required if the user wants to simulate a grain combustion with this propellant (although it is not necessary for simulations with constant chamber pressure). Vielle's parameters can be defined for different pressure intervals of the propellant (as many as the user is willing to add). This definition also requires the definition of the propellant density (to be used in grain combustion calculations).

It is important for the user to be aware of the units employed during the definition. The user may select out of a variety of units, both for speed and pressure. Note that, when the a and n constants are given, they must be coherent with the units of pressure and speed selected. The value for n does not depend on the units. On the other hand, the value of a depends not only on the units used for speed but also on the pressure units used during the linear fitting, changing said units will also change the value of a. It is recommended to the inexperienced user to not try and perform a unit conversion and write the values in the units in which they were found in the literature. The user may also choose if an upper pressure limit is to be specified or not in the preferences menu (refer to section 5.6). The only difference is that, if an upper limit is specified a warning will be set if a pressure above this limit is used in calculations (but the parameters for the last interval will be used anyway).

5.1.5 Advanced use of this tab

The fact that the considered species and reactions during calculations are saved along with the propellant offers many possibilities to the user. Firstly, it allows for quick comparison of different kinetic schemes (or species considerations) for the same propellant. The user may just save two copies of the propellant each with a different scheme and perform calculations with both in the calculations tab.

Moreover, since the kinetic scheme and considered species are dependant on the propellant, it is very helpful to have them tied to each other. Therefore, if a propellant is to be used, the user does not need to stop and think about the kinetic scheme or try to find in their notes which scheme to use (as long as it was adequately chosen when defining the propellant). Therefore, a proper definition allows the user to quickly change propellants during calculations without worrying about kinetic schemes.

Still, even though it is useful to only worry about the schemes during the propellant definition, it may seem tedious to some to have to define the scheme for each propellant (specially if they are similar in nature and the same scheme can be applied to all of them). ODEKO easily allows to avoid this hassle. When new reactants are added, the default scheme is that which was used for the last propellant. Moreover, if a propellant is loaded the default scheme will change to the one of that propellant. Consequently, if the user wants to define a new propellant which uses the same scheme as a previously defined one, the user may just load it to get its scheme and then change the composition in order to obtain the new propellant. Note that, the only species and reactions whose consideration is changed when loading a propellant are those who are compatible with said propellant. Therefore, if a O-C-H-N propellant is loaded, the inclusion of species with Cl will not change, it will remain as it was (corresponding to the selection of the last propellant that included Cl). On the contrary, species containing exclusively O-C-H-N will be modified accordingly to the new propellant (either including them or excluding them). Therefore, if a propellant is being used as a scheme base for a new one, it is important that they have the same elements present (or to be aware that it is needed to manually modify or attend to the elements in which they differ).

5.1.6 Export

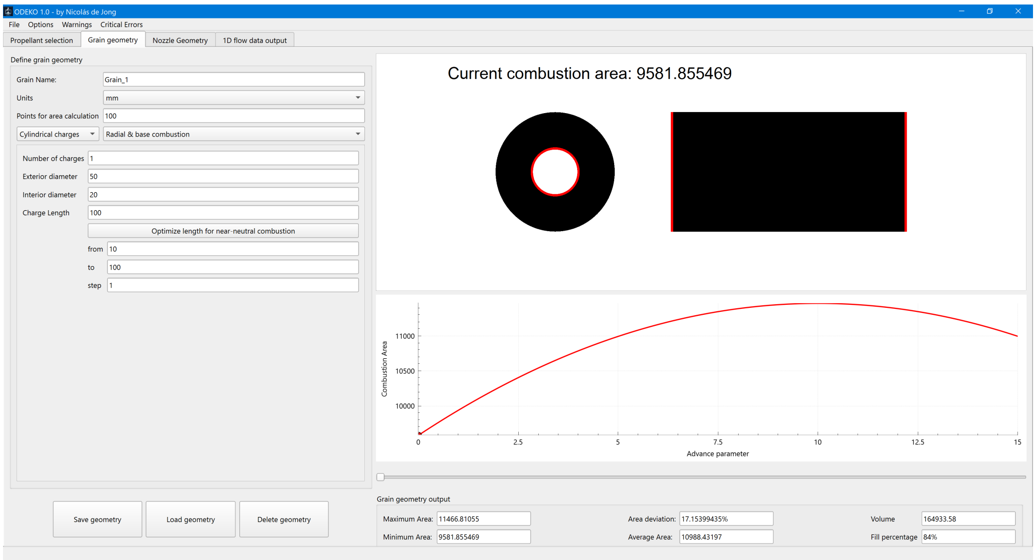
As explained in section 5.5.1, in every tab the user has the ability to export the present data, via a button in the file menu. Note that only one .csv file will be created, which includes all selected information. In the propellant tab, the user is able to choose which of the following sets of data are to be exported:

- Propellant composition: Exports to a .csv file information about which reactants form the propellant and their relative masses.
- Propellant data: Exports to a .cvs file additional information about the propellant: density and Vielle's parameters definition.
- Considered species: Exports to a .csv file a full list of all the species that are considered in calculations performed with this propellant.
- Considered reactions: Exports to a .csv file a full list of all the reactions that are considered in calculations performed with this propellant.
- Cstar curve: Exports as a .png file the current c^* graph present in the propellant tab (including the legend).

5.2 Grain geometry definition tab

By default, this tab is completely hidden from a new user. Nevertheless, it can be easily toggled on in the options menu (section 5.5.2), by deselecting the option "hide solid grain simulations". The user can disable this option by default in preferences (section 5.5.2), in order to have this tab visible every time the software is launched. Enabling this option also enables aspects related to solid grain combustion on the other three tabs, as explained in subsection 5.1.4, subsection 5.3.3 and section 5.4.2.

This tab is in charge of the definition of grain geometries for solid propellant combustion. Note that, in contrary to propellant and nozzle definition, grain definition is not required to perform calculations. Grain geometry is only used for nozzle optimizations or to study the time evolution of solid propellant rocket engines. The default grain geometry tab can be seen in Figure 5.8.



Grain Name: Grain_1
Units: mm
Points for area calculation: 100
Cylindrical charges
Radial & base combustion
Number of charges: 1
Exterior diameter: 50
Interior diameter: 20
Charge Length: 100
from: 10
to: 100
step: 1

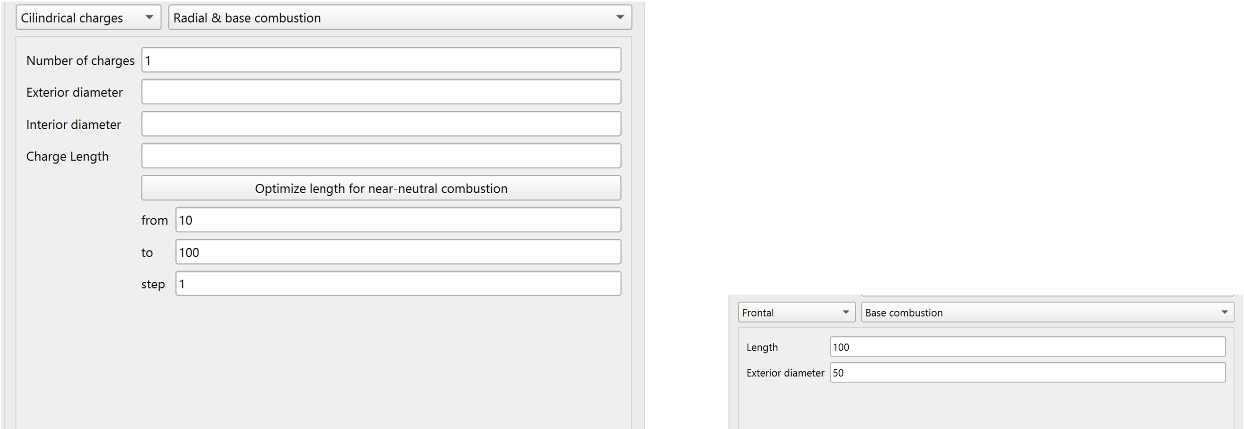
Current combustion area: 9581.855469

Maximum Area: 11466.81055
Minimum Area: 9581.855469
Area deviation: 17.15399435%
Average Area: 10988.43197
Volume: 164933.58
Fill percentage: 84%

Figure 5.8: Base grain geometry definition tab

5.2.1 Grain definition

In order to define the grain geometry the user must specify a name, the number of area points to be used during computations, the units in which the values will be inputted and the type of geometry and the geometry parameters. In the subsequent subsections each type of geometry definition will be explained. The type of combustion must also be specified, either radial, through the bases or both types simultaneously. The type of combustion affects both the shape and values of the combustion area curve. Note that the number of charges does not change the shape of the curve (they all have the same combustion) but it does multiply its value.



Cilindrical charges
Radial & base combustion
Number of charges: 1
Exterior diameter:
Interior diameter:
Charge Length:

from: 10
to: 100
step: 1

Frontal
Base combustion
Length: 100
Exterior diameter: 50

(a) Definition of Cylindrical charge **(b)** Definition of frontal combustion charge

Figure 5.9: Two grain geometrz options currently available

Cylindrical charges

In Figure 5.9a the parameters used in cylindrical charge definition are shown. Note that, for cylindrical charges, as seen in the figure, an optimization of the charge length is possible. Said optimization searches for the charge length that results in the least difference between maximum and minimum areas of combustion, standardized with the average area. Therefore, the closest to neutral combustion is obtained for said length. The optimization allows the user to set the minimum and maximum lengths between which the search is performed and the step to be used in the search.

Frontal combustion

The parameters from frontal combustion are: number of charges, length and exterior diameter. This type of geometry only allows for frontal combustion (it makes no sense to have a radial combustion without an interior hollow space). Furthermore, frontal combustion assumes that combustion only takes place in one of the bases of the charge. On the other hand, cylindrical charges with base combustion have both bases as active combustion surfaces.

Star charges

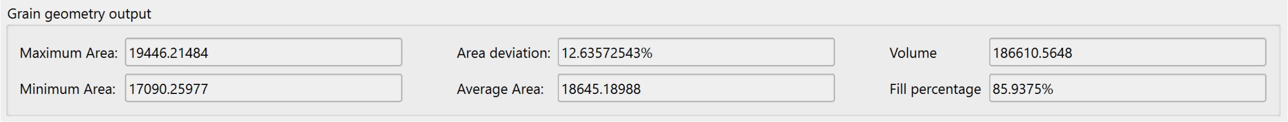
Currently star charges are not supported, but they will be implemented in version 1.1.

Custom charges

Currently custom charges are not supported, but it will be implemented in the future.

5.2.2 Output

The grain geometry tab has three output components. Firstly, on the bottom right of the screen a set a numerical values are displayed (Figure 5.10) for the following parameters: maximum area, minimum area, area deviation, mean area, volume occupied by propellant and percentage of chamber volume occupied by propellant. The area deviation is computed as the difference between maximum and minimum combustion area divided by the average combustion area.


Grain geometry output

Maximum Area: 19446.21484
Minimum Area: 17090.25977
Area deviation: 12.63572543%
Average Area: 18645.18988
Volume 186610.5648
Fill percentage 85.9375%

Figure 5.10: Grain geometry output parameters

Additionally, two connected visual aids are also given, as seen in Figure 5.8. The plot represent the evolution of the area relative to the advance parameter from the start of the combustion until the entirety of the grain has been combusted. The image on top represents the shape of the grain, as seen from the top and from the side. There is a scrollbar which the user may move, this will change the "current" advance parameter, which will be marked in the graph with a dot. Moreover, the grain representation will change in order to show its state for the current advance parameter (the combustion surfaces, marked red, will advance). The value of the combustion area for said advance parameter is also displayed.

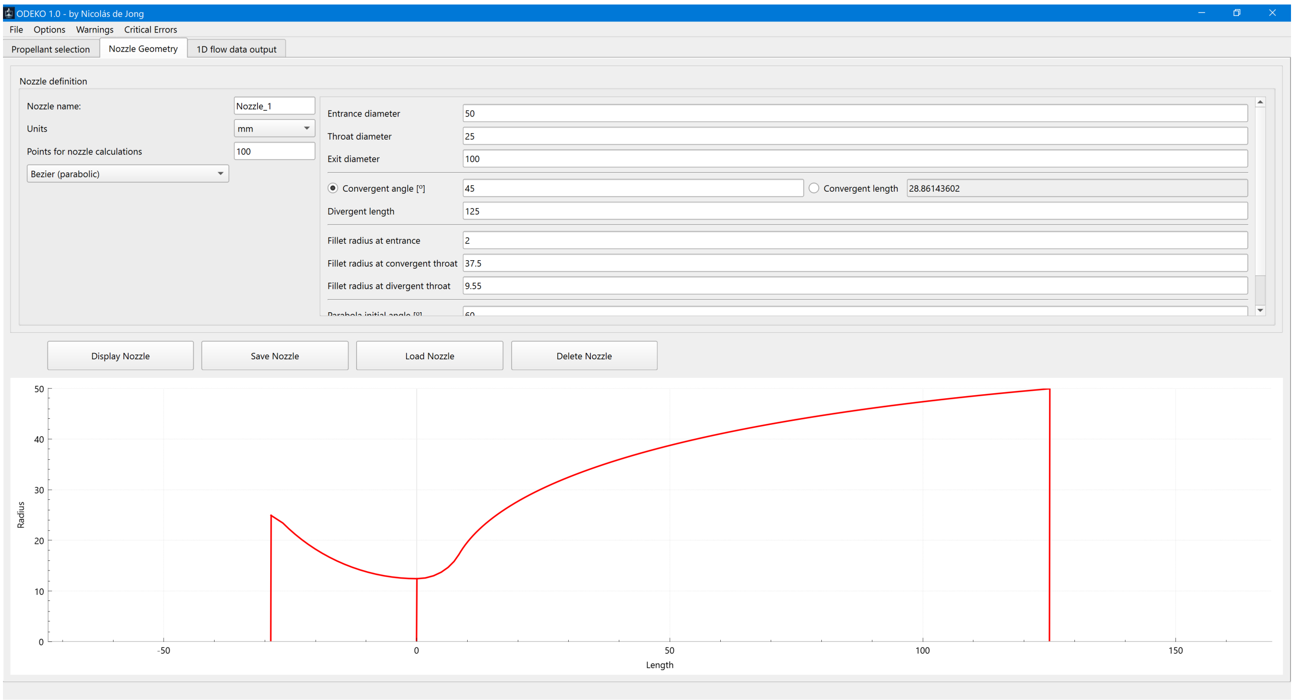
5.2.3 Export

As explained in section 5.5.1, in every tab the user has the ability to export the present data, via a button in the file menu. Note that only one .csv file will be created, which includes all selected information. In the grain geometry tab, the user is able to choose which of the following sets of data are to be exported:

- Grain parameters: Exports to a .csv file all parameters related to the definition of the grain geometry.
- Grain image: Exports to a .png file the current state of the grain image of the tab (upper right).
- Combustion area curve: Exports to a .png file the current state of the combustion area plot of the tab (lower right).
- Combustion area values: Exports to a .csv file a table containing the combustion area value for each value of the advance parameter (for the number of total points specified in the grain definition).

5.3 Nozzle definition tab

This tab is in charge of the nozzle geometry definition. Along with the propellant definition, the nozzle definition is required to perform any calculations. One launching the application a default nozzle will be shown in the window, but the user may make any modifications to it. The user might choose between two nozzle types and input/modify their parameters. Moreover, if the menu option "hide solid motor simulation" is toggled off (see section 5.5.2), the optimization section might be accessed, in which the user can select a grain geometry, as well as an engine characteristic in order to optimize the nozzle shape.



Nozzle name: Nozzle_1

Units: mm

Points for nozzle calculations: 100

Bezier (parabolic)

Entrance diameter: 50

Throat diameter: 25

Exit diameter: 100

● Convergent angle [º]: 45 ○ Convergent length: 28.86143602

Divergent length: 125

Fillet radius at entrance: 2

Fillet radius at convergent throat: 37.5

Fillet radius at divergent throat: 9.55

Parabola initial angle [º]: 60

Figure 5.11: Empty nozzle geometry definition tab

5.3.1 Nozzle definition

In the upper left corner of the screen (as seen in Figure 5.11) the user must define the name of the nozzle, the units to be used for the parameters input, the number of points to be considered during calculations and the type of nozzle to be defined. The user might select between two nozzle types: conical or parabolic. The specific parameters that need to be inputted for each are shown on the upper right section of the screen and will be explained next.

Conical Nozzle

The parameters for the conical nozzle are shown in Figure 5.12. Note that the red lines are generated by ODEKO (nozzle shape) whilst the green and blue lines have been manually added to aid the user understand the various parameters. Fillets are not required to define the conical nozzle but can be optionally specified. Length and angle (for convergent and divergent regions respectively) are mutually exclusive. The user may define either length or angle and ODEKO will compute the other and display its value to the user.

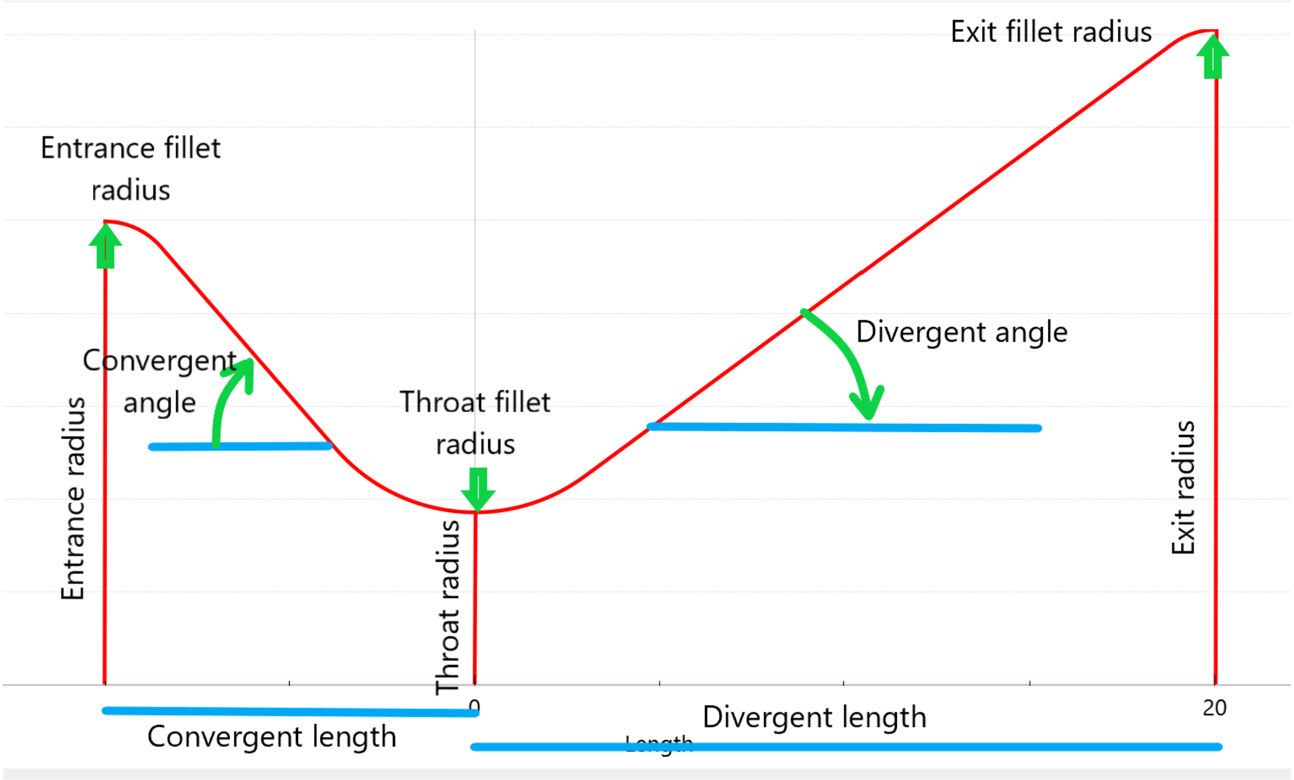


Figure 5.12: Example for conical nozzle

Parabolic Nozzle

The parameters for the parabolic nozzle are shown in Figure 5.13. Where the red lines are generated by ODEKO (nozzle shape) and the green and blue lines have been manually added to indicate the meaning of all parameters that need to be specified for the geometry definition. Note that convergent length and convergent angle are mutually exclusive, only one of them can be set by the user (the value for the other one will be computed by the application and displayed for the user).

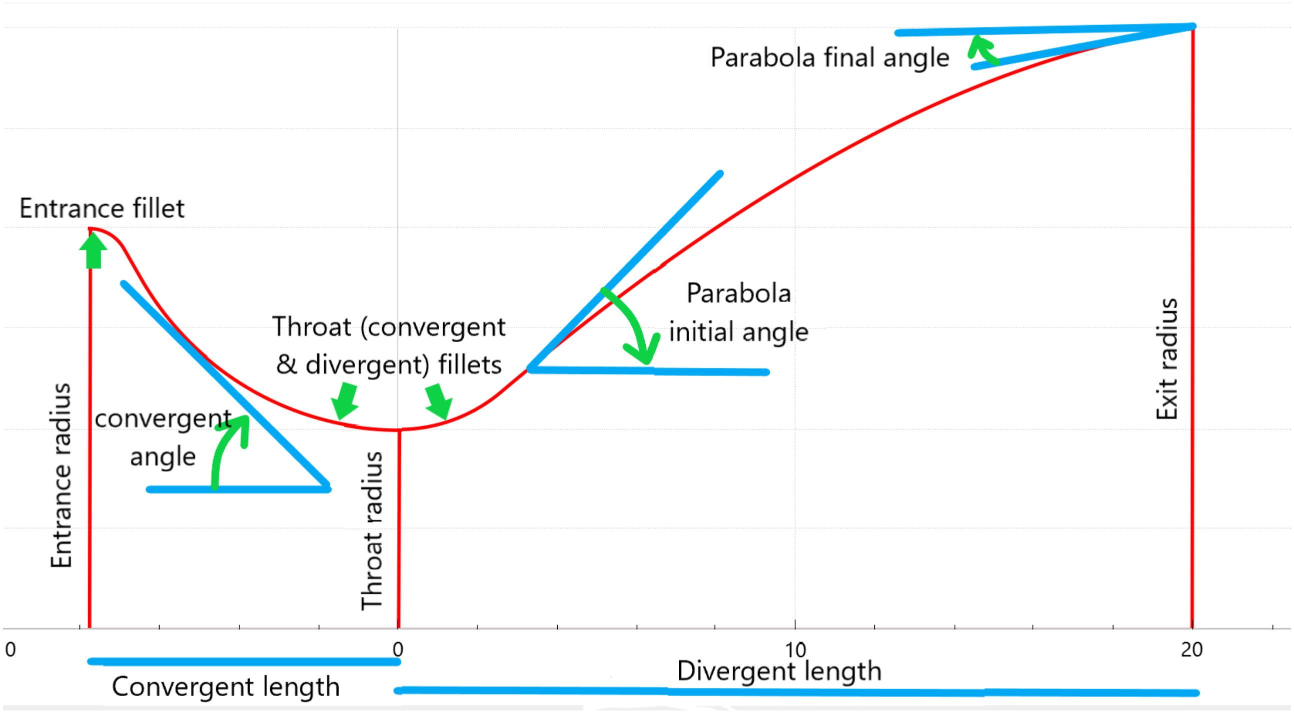


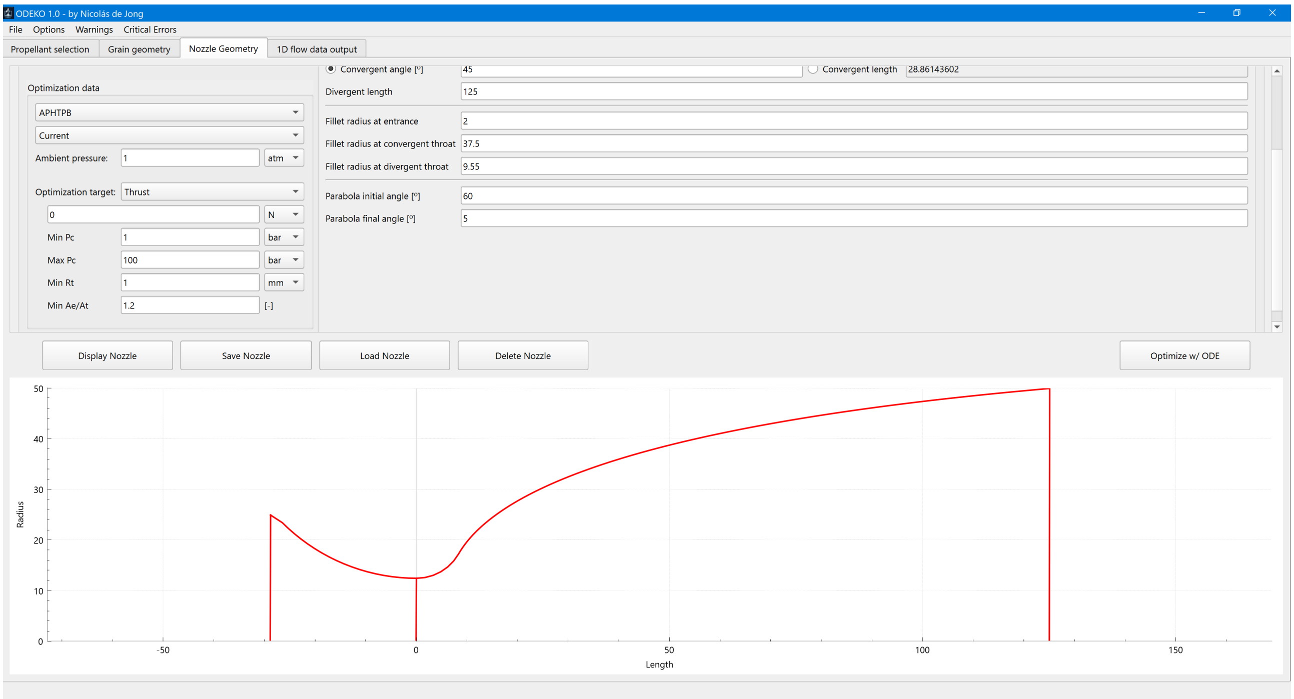
Figure 5.13: Example for bezier (parabolic) nozzle

5.3.2 Display options

In the preferences menu (see section 5.6) the user may select how the nozzle is represented. If only the curve is plotted, the throat will be tangent to the x axis. If the axis is also represented, the nozzle will be seen as shown in Figure 5.12 and Figure 5.13. Note that, in these figures, vertical lines in the entrance, throat and exit are present, this behaviour can be toggled off in the preferences menu as well. The user can also choose to represent the entire nozzle, which will show the section of the nozzle instead of just a quarter section. Finally, the user may also specify if they want the nozzle to be colorfilled or not. Note that all these options are merely aesthetic and have no impact on calculations.

5.3.3 Optimization

If the user is employing a solid propellant with known Vielle's parameters, combustion geometry and propellant density, the nozzle can be optimized for this configuration attending to one out of five target parameters. Note that this optimization required toggling on the solid motor simulation in the options menu (section 5.5.2). On Figure 5.11 the nozzle tab was shown with the solid motor options hidden, on Figure 5.14 this option has been untoggled, so the user can access the optimization menu.



Optimization data

APHTPB

Current

Ambient pressure: 1 atm

Optimization target: Thrust

0 N

Min Pc 1 bar

Max Pc 100 bar

Min Rt 1 mm

Min Ae/At 1.2 [-]

● Convergent angle [º] 45 ○ Convergent length 28.86143602

Divergent length 125

Fillet radius at entrance 2

Fillet radius at convergent throat 37.5

Fillet radius at divergent throat 9.55

Parabola initial angle [º] 60

Parabola final angle [º] 5

Figure 5.14: Nozzle tab with solid motor options shown

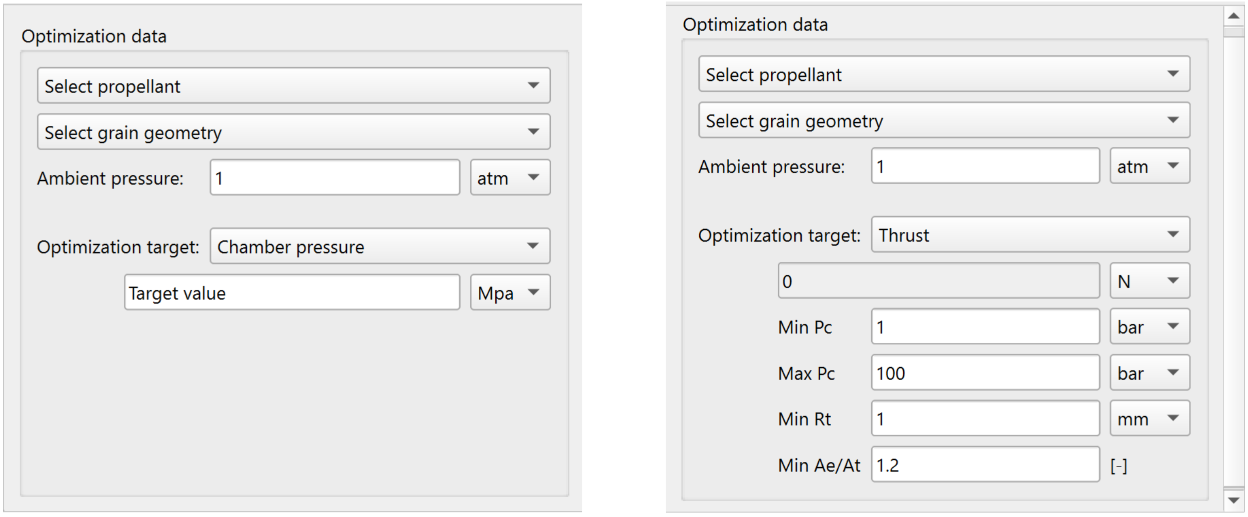
Chamber pressure & Combustion time

The user may specify a given chamber pressure or combustion time for the rocket engine. The application will compute the throat area that achieves said target value. The user might also choose if the nozzle is to be adapted or not. If not, the exit area is not modified since its not relevant for these two parameters and the value of the atmospheric pressure will have no effect in these optimizations. On the other hand, if the nozzle is adapted the ambient pressure will be used to dictate the exit area required (once the throat has been computed).

Note that the optimization for a given chamber pressure does not require any iteration, resulting in a very fast process. On the other hand, optimization for a given combustion time does require some iteration, but it still is a relatively fast procedure.

Thrust, specific impulse & total impulse

The user can optimize the nozzle so that it maximizes the thrust, specific impulse or total impulse value. ODEKO will compute the values for the throat area and exit area that achieve said maximum values whilst complying with the restrictions set by the user (minimum and maximum chamber pressure, minimum throat radius and minimum area relation). Note that it is necessary to specify the atmospheric pressure for which the optimization is taking place. It should be noted that all these optimization require many iterations and can be quite slow (specially if optimizing for total impulse).



Optimization data

Select propellant

Select grain geometry

Ambient pressure: 1 atm

Optimization target: Chamber pressure

Target value Mpa

Optimization data

Select propellant

Select grain geometry

Ambient pressure: 1 atm

Optimization target: Thrust

0 N

Min Pc 1 bar

Max Pc 100 bar

Min Rt 1 mm

Min Ae/At 1.2 [-]

(a) Parameters for thrust optimization

(b) Parameters for chamber pressure optimization

Figure 5.15: Nozzle optimization parameters

Abort optimization

After the optimization is started the "Optimize" button will be disabled. After a given time, if the optimization has not yet finished, the button will be substituted by an "Abort optimization" button. If the user presses the button the optimization will be halted and all its progress will be lost. The time until the abort button appears can be selected in the preferences menu (see section 5.5.2) and defaults to 15s.

The use of the abort button might be specially useful if the optimization is stuck (or specially slow) and the maximum number of allowable iterations was set to a large number (which can also be modified from the preferences menu). Nevertheless, for expensive optimizations such as thrust, specific impulse or specially total impulse, it is not strange to have to wait even a few minutes for the optimization to succeed. The abort method is not implemented for pressure optimizations, as they require no iterations and are almost instant. Note that the abort signal is only checked for at the start of every iteration, so it might take a few seconds from the push of the button until the process is fully stopped. If the process is not stopped even after pressing the abort button the user should report this as a bug.

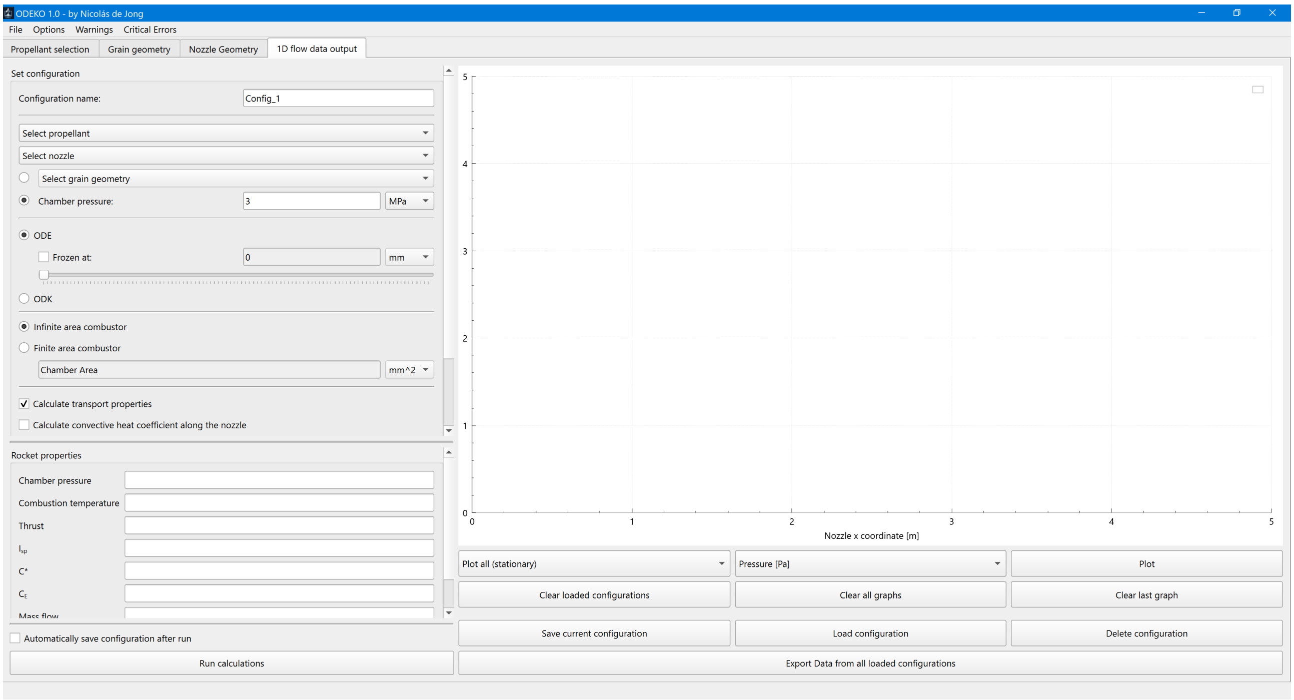
5.3.4 Export

As explained in section 5.5.1, in every tab the user has the ability to export the present data, via a button in the file menu. Note that only one .csv file will be created, which includes all selected information. In the nozzle geometry tab, the user is able to choose which of the following sets of data are to be exported:

- Nozzle parameters: Exports to a .csv file all parameters related to the definition of the nozzle geometry.
- Nozzle points coordinates: Exports to a .csv file a table with the axial and radial coordinate of all points in the nozzle.
- Nozzle points areas: Exports to a .csv file a table with the area values for all points in the nozzle.
- Nozzle image: Exports to a .png file an image of the currently defined nozzle.

5.4 Calculations tab

The calculations tab allows the user to simulate the internal flow of a rocket engine nozzle, employing the propellant and nozzle (and optionally grain geometry) that were defined in the previous tabs. The optimization has multiple parameters that the user is able to modify, in order to best understand them it is recommended to refer to the full documentation. The main two simulation processes are chemical equilibrium and chemical kinetics.



Configuration name: Config_1

● Chamber pressure: 3 MPa

● ODE

☐ Frozen at: 0 mm

● Infinite area combustor

☑ Calculate transport properties

Figure 5.16: Empty calculations tab

5.4.1 Input information

A short explanation of the different input parameters of the rocket engine simulation is given in this section. Nonetheless, it may prove beneficial to consult the full documentation in order to properly understand their implications in the simulation process. All these parameters are inputted in the left section of the data tab, as can be seen in Figure 5.16. Note that the option to select a grain geometry will be hidden by default, as the options menu "hide solid motor simulation" is toggled on startup (see section 5.5.2).

- Propellant: Propellant to be used, as defined in the Propellant tab. Note that to use a defined propellant it needs to be saved first in its respective tab. Loading the propellant will also load the selected set of species and chemical reactions to be used in the calculations.
- Nozzle: Nozzle to be used, it can either be a previously saved nozzle or the nozzle currently

displayed in the nozzle tab (this means the last nozzle that was plotted without errors).

- Chamber pressure / grain geometry: Here the user may select a constant chamber pressure for which to show results along the nozzle's axial coordinate or a grain geometry in order to compute the time evolution of the rocket engine's properties. The grain geometry to be selected can either be a previously saved one or the one currently displayed in the grain geometry tab (this means the last grain that was computed without errors). This is further explained in section 5.4.2 and section 5.4.2 respectively.
- ODE/ODK: Model to be used, either chemical equilibrium (which also allows to freeze the flow at a given point of the nozzle) or chemical kinetics. Note that ODK is much slower than ODE, although more precise (as long as the chosen kinetic scheme is precise itself). For more in-depth information about the differences and inner-workings of both computation methods please consult the full documentation.
- IAC/FAC: Indicates how to model the combustor, either with infinite area or finite area. If the user is not aware of these concepts it is recommended to always use IAC (FAC works best for liquid rocket engines with a well known injection velocity which needs to be inputted through the preferences menu).
- Calculate transport properties: Whether transport properties for the flow should be computed. Not performing this computation will result in a slightly faster simulation. Note that, in order to obtain valid transport properties, most of the mixture species need to have their transport properties defined in the database (refer to section 5.1.2).
- Calculate convective heat transfer coefficient: Whether the convective heat coefficient should be computed. Not performing this computation will affect very little the overall simulation time. In the preferences menu (section 5.6) many parameters for this computation can be edited by the user. Note as well that computing the convective heat transfer coefficient requires the computation of transport properties.
- Ambient pressure: The ambient pressure for which the various rocket parameters (such as thrust) are computed.
- Set uniform temperature for reactants: The temperature at which the propellant is before the combustion. This value will affect slightly the overall performance. Note that, in order for this parameter to have an effect on the simulation, the reactants must have their C_p defined (refer to section 5.1.2).

5.4.2 Computation

As mentioned in the prior section, the user may either choose a constant chamber pressure at which to run the simulation or a grain geometry. If a constant chamber pressure is chosen the simulation is considered ’stationary’ and axial variables are shown. If a grain geometry is used the simulation is considered ’time dependant’ and the time evolution of variables is presented.

Stationary

When a given chamber pressure is selected, the output is ’stationary’. This means that the chamber pressure is assumed constant and output is given for that specific scenario. The evolution along the nozzle axis of different variables is presented. The variables that can be plotted are: pressure, temperature, density, enthalpy, entropy, molecular weight, gamma, C_p, C_V, velocity of sound, fluid velocity, mach number, mass flow and the molar fraction of the various present species. Moreover, if transport properties are computed, the user may also plot conductivity, viscosity and Prandtl number. The user has also the option to calculate the convective heat transfer coefficient. Values in the combustion chamber are not present in the plots but are shown if the table display is selected (see section 5.4.3).

Time-dependant

If a grain geometry is specified (along with a solid propellant and its corresponding Vielle’s parameters), the problem no longer is stationary. Due to the grain geometry information, an evolution of the chamber pressure relative to time can be computed. Therefore, the plots represent the evolution of key variables with time. Those variables are: Chamber pressure, combustion temperature, thrust, specific impulse, c^*, C_E and mass flow. In this case no variables along the nozzle axis are presented. Nevertheless, the user always has the option to select a given point in time (for which the chamber pressure is given by this calculation) and simulate that specific point to obtain the spatial evolution in the nozzle of the various variables. Note that in this case the values present in the rocket paremeters section (section 5.4.3) represent the average values over time.

Abort Calculations

After the calculation is started the "Run calculations" button will be disabled. After a given time (which can be selected in preferences menu and defaults to 15s), if the optimization has not yet finished, the button will be substituted by an "Abort calculations" button. If the user presses the button the calculation will be halted and all its progress will be lost. This might be specially useful if the computation is stuck (or specially slow) and the maximum number of allowable iterations was set to a large number (which can also be modified from the preferences menu).

The abort procedure is currently only implemented for ODK calculations, as complex kinetics schemes can be very computationally expensive. On the other hand, stationary ODE simulations are almost instant, so the abort button has not been implemented for them. For time-dependant simulations the abort behaviour is also present, as depending on the grain geometry they can be quite computationally taxing. Note that the abort signal is only checked for at the start of every iteration, so it might take a few seconds from the push of the button until the process is fully stopped.

5.4.3 Output

The output for the calculations tab can either be represented in a table or in a graph (the user may change this in the preferences window, refer to section 5.6). Moreover, the main characteristics of the engine are displayed in the lower left section of the screen.

Graph display

The graph is the option set by default and it allows the user to plot any of the given variables along the nozzle axis or over time (depending on the type of calculation performed). The user can plot as many variables as desired and either plot different configurations (different calculations made that session). This allows the user to compare the same variable for different calculation methods (such as between ODE and ODK) or compare how a change in the propellant or nozzle affects the results. A legend is automatically added, using the name of the configuration and the name of the plotted variable. The user can plot simultaneously different variable types, but their different scales might make the graph unusable. Nevertheless, the preferences menu (section 5.6) has an option to normalize all variables, making mixed plots more readable. On the preferences menu the user can also find an option to always shown the $y = 0$ point of the axis, instead of just the region where the variable is located.

When a calculation is finished, the currently selected variables is automatically plotted (either along the length of the nozzle or respect to time, depending on the calculation type). If a plot is already present, the new one will be added to the graph. Moreover, this calculation will be added to the list of current configurations, so it can be plotted at any time without the need to perform the calculation again (saving valuable computing time).

In order to plot a variable from an already computed configuration, said configuration needs to be selected, as shown in Figure 5.17. Note that the possibility to choose 'all stationary' or 'all time dependant' is also present, which will perform the plotting for all configurations that fit said category. Afterwards, the user will select the variable to be plotted at the drop-down list at the right of the previous one. Finally, if the plot button is pressed, the variable will be represented for all the selected configurations and the corresponding items will be added to the legend. This can be done any number of times and the new plots will be added to the existing ones.

The user may clear the last plot or all plots in the graph at any point in time. This clear will not eliminate the calculations made, allowing the data to be used again for plots (without needing to redo calculations that might be computationally expensive). The user can clear the session's calculations at any time as well, but in this case it will not be possible to plot them unless computed again. Clearing calculations is not required but might be useful if too many calculations are made and the list is becoming too large.

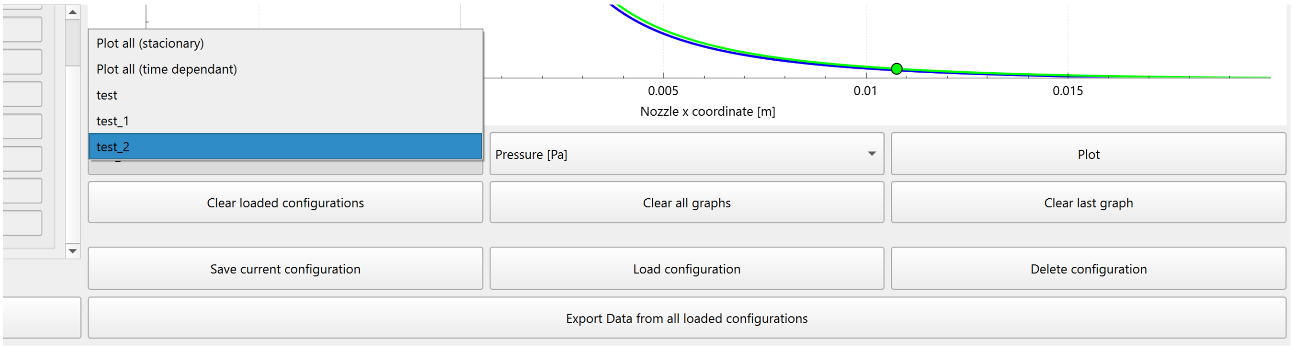


Figure 5.17: Selection of an already computed configuration

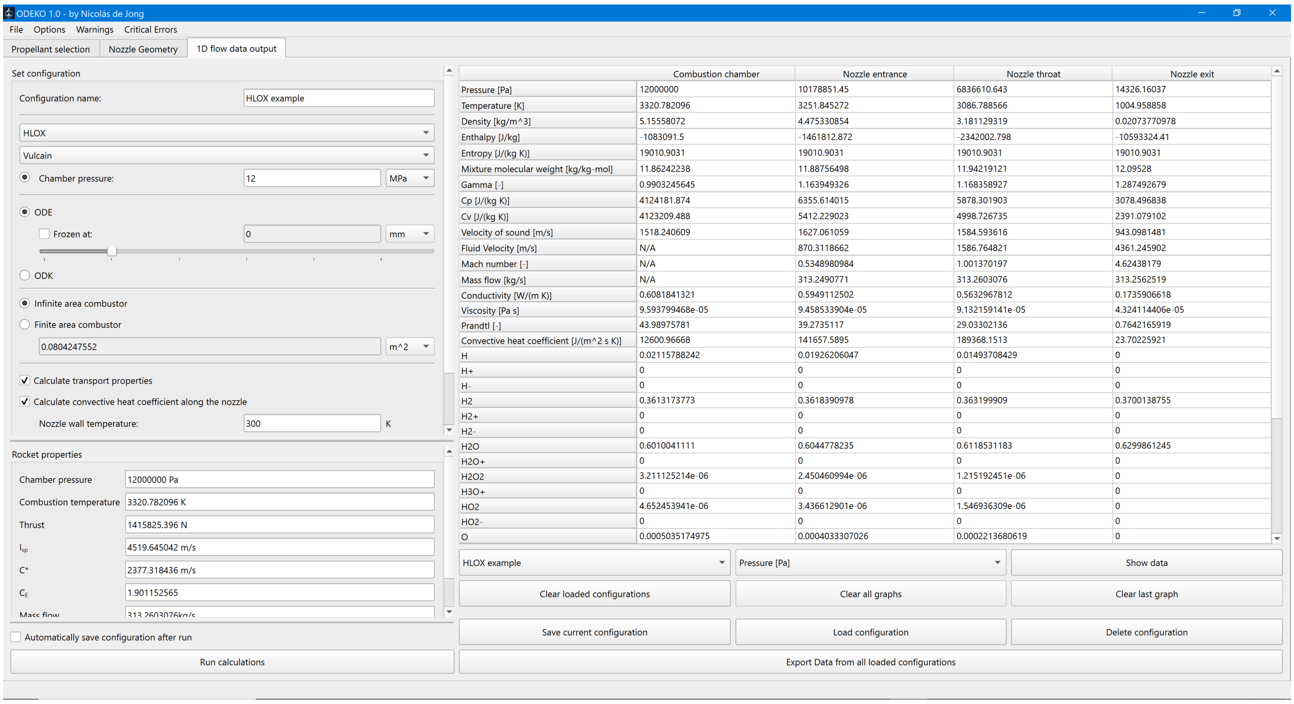
Table display

By default, the display tab is the aforementioned graph display. Nevertheless, the user may change at any time the display type to a table through the preferences menu (see section 5.6). This will change the look of the calculations tab from the one shown in Figure 5.16 to the one presented in Figure 5.18.

When using the tab display, all variables are shown simultaneously; therefore, the variable selector used for the graph display is disabled. As can be seen in the image, the table display also includes information about the combustion chamber, which was not present in the graph display. The table display only allows the data of one configuration to be shown, only the graph display can show multiple configurations at once. Nevertheless, in the table display the user can quickly select a different configuration and show its data on the table.

Note that, for variables that have not been computed (for example transport variables if the option is not selected) their respective cells will show N/A. This also applies to the velocity, mach number and mass flow in the combustion chamber, as these variables are not computed there for the model chosen in the image example. For more details on the computation method refer to the full documentation.

By default, only the variables at the combustion chamber, nozzle entrance, nozzle throat and nozzle exit are shown in the table. If the flow is frozen at a given point, the variables at said point will also be shown in the table. For time dependant computations the points at $t = 0, t = t_b/2, t = t_b$ will be shown, with t_b being the combustion time. If the user wishes to have the data at all computed points this can be achieved by exporting the current configuration, as explained in subsection 5.4.5.



Configuration name: HLOX example

HLOX

Vulcain

● Chamber pressure: 12 MPa

● ODE

☐ Frozen at: 0 mm

○ ODK

● Infinite area combustor

○ Finite area combustor

0.0804247552 m^2

☑ Calculate transport properties

☑ Calculate convective heat coefficient along the nozzle

Nozzle wall temperature: 300 K

Chamber pressure	12000000 Pa
Combustion temperature	3320.782096 K
Thrust	1415825.396 N
I_{sp}	4519.645042 m/s
C*	2377.318436 m/s
C_E	1.901152565
Mass flow	313.2603076kg/s

☐ Automatically save configuration after run

HLOX example

Pressure [Pa]

	Combustion chamber	Nozzle entrance	Nozzle throat	Nozzle exit
Pressure [Pa]	12000000	10178851.45	6836610.643	14326.16037
Temperature [K]	3320.782096	3251.845272	3086.788566	1004.958858
Density [kg/m^3]	5.15558072	4.475330854	3.181129319	0.02073770978
Enthalpy [J/kg]	-1083091.5	-1461812.872	-2342002.798	-10593324.41
Entropy [J/(kg K)]	19010.9031	19010.9031	19010.9031	19010.9031
Mixture molecular weight [kg/kg-mol]	11.86242238	11.88756498	11.94219121	12.09528
Gamma [-]	0.9903245645	1.163949326	1.168358927	1.287492679
Cp [J/(kg K)]	4124181.874	6355.614015	5878.301903	3078.496838
Cv [J/(kg K)]	4123209.488	5412.229023	4998.726735	2391.079102
Velocity of sound [m/s]	1518.240609	1627.061059	1584.593616	943.0981481
Fluid Velocity [m/s]	N/A	870.3118662	1586.764821	4361.245902
Mach number [-]	N/A	0.5348980984	1.001370197	4.62438179
Mass flow [kg/s]	N/A	313.2490771	313.2603076	313.2562519
Conductivity [W/(m K)]	0.6081841321	0.5949112502	0.5632967812	0.1735906618
Viscosity [Pa s]	9.593799468e-05	9.458533904e-05	9.132159141e-05	4.324114406e-05
Prandtl [-]	43.98975781	39.2735117	29.03302136	0.7642165919
Convective heat coefficient [J/(m^2 s K)]	12600.96668	141657.5895	189368.1513	23.70225921
H	0.02115788242	0.01926206047	0.01493708429	0
H+	0	0	0	0
H-	0	0	0	0
H2	0.3613173773	0.3618390978	0.363199909	0.3700138755
H2+	0	0	0	0
H2-	0	0	0	0
H2O	0.6010041111	0.6044778235	0.6118531183	0.6299861245
H2O+	0	0	0	0
H2O2	3.211125214e-06	2.450460994e-06	1.215192451e-06	0
H3O+	0	0	0	0
HO2	4.652453941e-06	3.436612901e-06	1.546936309e-06	0
HO2-	0	0	0	0
O	0.0005035174975	0.0004033307026	0.0002213680619	0

Figure 5.18: Table display in Calculations tab

Rocket parameters

The rocket parameters, shown in the lower left of the screen (Figure 5.16), are: chamber pressure, combustion temperature, thrust, specific impulse, c^*, C_E, mass flow and combustion time. For the stationary case, the combustion time is not given since there is no way to compute this value. For the time-dependant case, the combustion time can be computed using the specified grain geometry. Moreover, in this case, the average values for the rocket parameters considering the entire combustion are given.

The rocket parameters shown by the application correspond to the latest calculation performed or the latest selected configuration (whichever happened last). This means that the user can quickly change between the values of different configurations just be selecting them as shown in Figure 5.17.

5.4.4 Configuration save & load

If the current configuration is saved, only the input parameters will be saved (not the output nor results of the calculation). In the same fashion, if a configuration is loaded, it needs to be calculated afterwards in order to plot or use its results. The intention behind saving or loading a configuration is to be able to have a 'benchmark' with which to compare results that can be quickly loaded (so the user does not have to go through every input parameter). This is also useful if the user wants to perform small modifications to an already existing configuration.

5.4.5 Export

As explained in section 5.5.1, in every tab the user has the ability to export the present data, via a button in the file menu. For this tab, an export button is also present on the lower right corner. Note that only one .csv file will be created, which includes all selected information. In the calculations tab, the user is able to choose which of the following sets of data are to be exported:

- Parameters used for calculations: Exports to a .csv file all the parameters used in the calculation (this also includes propellant, nozzle and optionally grain geometry names).
- Configurations to be exported: The user must choose if they want only the currently selected configuration data to be exported or the data of all configurations. This will affect all other options present in the export.
- Graphs: The user may choose to export to a .png file the currently plotted graph or multiple ones. In the second case, the user can specify which types of graphs they want to export (thermodynamic, transport, composition, time dependant). Note that, if multiple configurations are being considered, only one graph of each type will be exported and it will contain the plots for all the considered configurations.
- Tables: The user may choose to export the output data in table format to a .csv file. The user may choose to export thermodynamic, transport and/or composition data. By default this only exports data for the key points of the nozzle (combustion chamber, entrance, throat, exit and frozen point if present), but the user can specify that the data should be exported for all points of the nozzle instead. For time dependant calculations the user only needs to specify that all points should be exported, the other options will be ignored.
- Rocket parameters: Exports to a .csv file the rocket parameters of the current/all configuration/s (as selected). If the configuration is time dependant, the export takes the mean values.

5.5 Menu

The menu can be accessed at the top of the screen, no matter in which tab the user is at, since it is always present on top of the tap content. There are four main sections of the menu each with their own subsections, which are explained in the subsequent sections of the manual.

5.5.1 File

File changes

In the file menu, the user may change which files are used to store the data of the different components of the application (known as save-files, as explained in subsection 5.8.1). The user may select any of the save file types and change its associated file, which will from there on be used to retrieve and save its corresponding data. This is of special use when exchanging component information with a colleague or if the user wishes to have two differentiated sets of component data. It is also very useful since the user can give their entire configuration data along with a bug report, to ensure replicability.

Note that, keeping the configurations file but changing other may result in inconsistencies, where a configuration is not able to be loaded due to missing information (said components where located in the default save files but not in the new selected ones).

The preferences file might also be changed in the same fashion. This way the user will be able to share specific set of preferences or quickly change between two configurations.

Export

The Export button, inside the File Menu, allows the user to export data from the current tab to an output file. The data that the user may choose to export is different for each tab and it is explained in each corresponding section: subsection 5.1.6 subsection 5.2.3 subsection 5.3.4 subsection 5.4.5. Specific information for the output files can be found on those sections, whilst a more general overview is given in subsection 5.8.3.

Input information

There are three buttons in the File menu, regarding the formats for the three input data files (thermo, transport & kinetics). These formats are further explained in section subsection 5.8.2.

About ODEKO

If the user clicks on it, a window with a short description of the application will be shown. This information is very similar to the introduction of this manual.

Open Documentation

There are two buttons on the file menu which open the User Manual and the Full documentation respectively. The User Manual is this one, whilst the full documentation contains the entire User Manual as well as the Theory documentation and some other relevant information about ODEKO.

5.5.2 Options

Hide solid motor options

As mentioned in section 5.2, by default all options related to solid motor simulations and grain geometry are hidden. The user might untoggle this checkbox in the options menu in order to enable these additional features. Furthermore, the user might change the default behaviour in the preferences menu (section 5.5.2) in order to have this option toggled off by default. The hidden features related to solid motor simulation include:

- The entire grain geometry tab, which allows the user to define and save grain geometries.
- The propellant properties section in the propellant tab, which allows the user to specify vielle's parameters and the propellant density (subsection 5.1.4).
- The optimizations in NozzleTab, which allow the user to optimize the nozzle using a given grain geometry for the desired pressure or combustion time or to maximize thrust or impulse (subsection 5.3.3).
- The time-dependant simulations in the Calculations tab, which show the time evolution of the rocket parameters for a given grain geometry (section 5.4.2).

the entire grain geometry tab, the option to specify vielle's parameters and

Baby mode

Referred to in-application as 'Baby mode'. It is a simplified mode for the application that can be toggled through the options menu. It severely limits the capabilities of the program, hiding many of its functions (such as ODK calculations). Its main objective is to provide a simpler and less confusing interface for students using this program. In preferences the user may select if 'baby mode' is active by default and if the preferences are accessible in this mode. Moreover, the user may also lock the application in 'baby mode' which would disable the ability to toggle it off. This may be useful if a teacher is installing the program in a students computer and does not want this mode to be toggled off. The only way to undo the lock is to manually edit the preferences file and set it to false. A comparison between normal mode and baby mode for the propellant tab is shown in Figure 5.19.

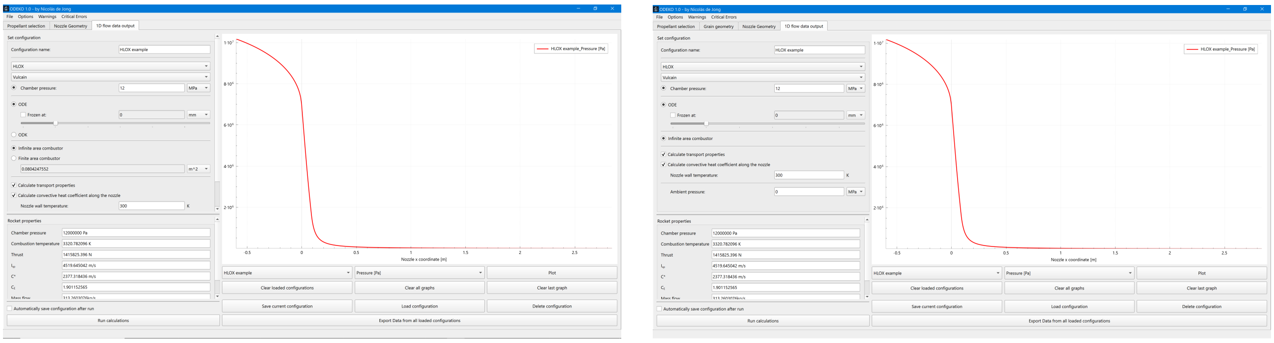


Figure 5.19: Comparison between normal mode and baby mode for the calculations tab

Visual changes

There are two visual changes available in the options menu. Firstly, the user may toggle on or off the visibility of the status bar. The status bar provides tips about the functionality of certain buttons or parts of the application when the user hovers over them. Therefore, for a new user it is useful to have the status bar visible, whilst for a more experienced one it might become redundant.

On the other hand, the user may also turn on or off a ’night mode’ for the application. Said mode changes the color palette to a darker tone, in place of the default light one. This change is only aesthetic and has no other impact, is up to user preference. A comparison of both palettes can be seen in image 5.20.

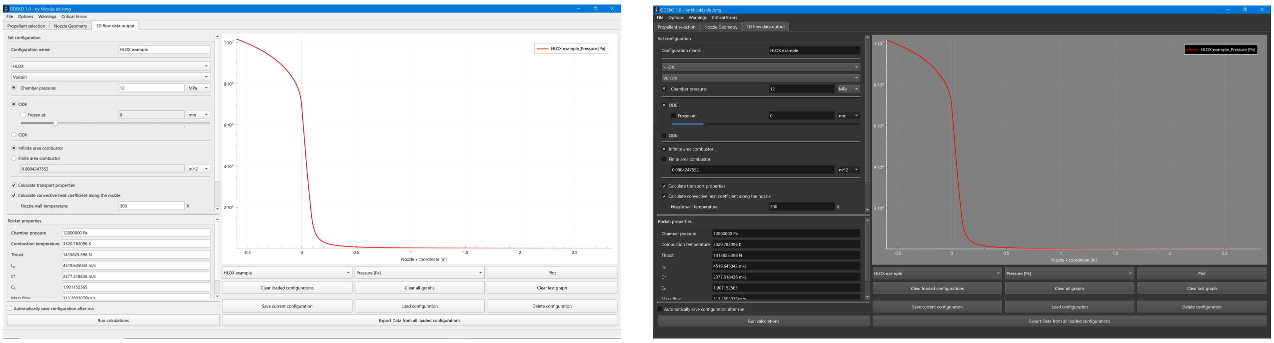


Figure 5.20: Comparison between light mode and dark mode for the calculations tab

Clear Warnings & Errors

In the options menu the user may clear the current warnings or the current critical errors. Note that this is an option that comes with a certain risk. Clearing the critical errors will allow the user to perform certain actions or calculations that would regularly be forbidden due to the errors occurred; therefore, it is only advised to clear them if the user knows for a fact that said errors will not result in any malfunction of the program. It is best to try to fix the errors rather than clear them. In the case of warnings, clearing them does not have much of an impact, and it is done automatically in between calculations.

Update Warnings & Errors

Generally speaking, warnings and errors are updated automatically after events that might trigger them. Nevertheless, the user has the option to trigger a manual update of warnings and errors through the options menu. This might be of use if the user suspects that an error/warning happen which was not displayed.

Preferences

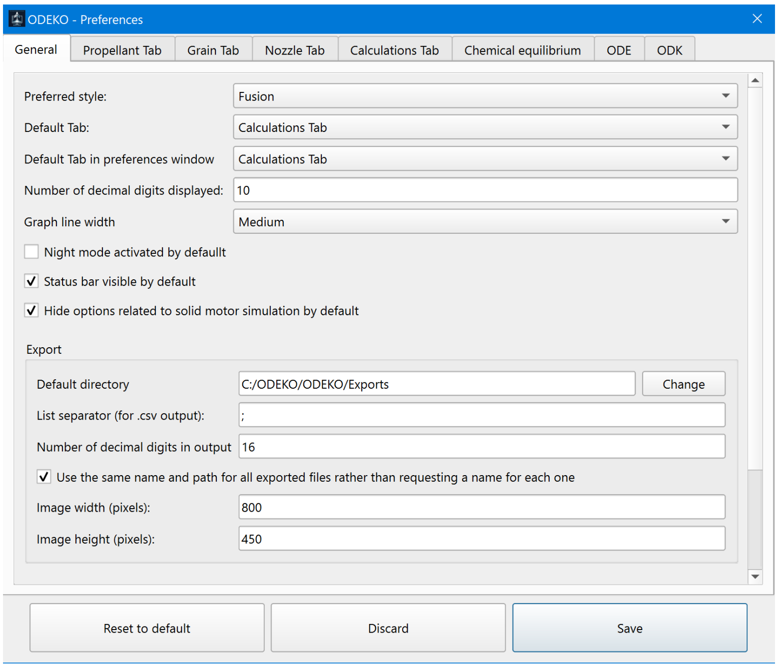
The preferences window can be accessed through this button, further information can be found in section 5.6.

5.5.3 Warning & critical erros

This two sections of the menu show the current warnings and critical errors respectively (those that occurred in the last calculation or action). Clicking on them will show a complete list of all warnings/errors along with longer descriptions, this list can also be seen in section 5.7.

5.6 Preferences

The preferences window may be accessed from the Options Menu (section 5.5). Here the user may modify different aspects of the application. Each tab of the preferences window groups items related to the same aspect of the application. The user may change any of the settings, changes will not be applied until the 'save' button is clicked. The user always has the option to discard the changes just made or to restore all preferences to their default value (as seen in Figure 5.21). Closing the window through the 'x' button will result in all changes being discarded.



Preferred style: Fusion

Default Tab: Calculations Tab

Default Tab in preferences window Calculations Tab

Number of decimal digits displayed: 10

Graph line width Medium

☐ Night mode activated by defaullt

☑ Status bar visible by default

☑ Hide options related to solid motor simulation by default

Export

Default directory C:/ODEKO/ODEKO/Exports Change

List separator (for .csv output): ;

Number of decimal digits in output 16

☑ Use the same name and path for all exported files rather than requesting a name for each one

Image width (pixels): 800

Image height (pixels): 450

Figure 5.21: Preferences window

The default settings is what the creator of this application believes to be best for the average user. Therefore, it is advised to not change them unless the user has proper knowledge of its implications. Some of the preferences relate to visual aspects of the application, which can be modified without much regard. On the other hand, some preferences affect key aspects of the calculation process, and changing them may result in catastrophic consequences. In the following subsections, preferences marked with '*' are best not changed unless the user has a deep understanding of the theoretical aspects behind the calculations performed. It is recommended to thoroughly read the theory manual of ODEKO before modifying any preferences related to the calculations of the application.

5.6.1 General

- Preferred style: The application currently offers two visual styles: 'classic windows' or 'Fusion'. This change is only visual and will not have any other implications. Note that currently, dark mode is only supported for 'Fusion' style. Defaults to "Fusion".
- Default tab: Whenever the application is opened this will be the tab that is shown by default. This may proof specially useful top users that already have all their components designed (propellant, grain & nozzle) and only wish to perform calculations. Defaults to "Calculations Tab".
- Default tab in preferences window: Whenever the preferences window is opened, this will be the tab shown by default. This may proof useful to users that change a certain preference very often. Defaults to "General".
- Number of decimal digits displayed: This will affect all numeric displays present throughout the application. It is only a visual change and does not alter the numeric precision of the calculations performed by the program. Defaults to 10.
- Graph line width: This will affect all graphs throughout the application. It is only a visual change. Defaults to "Medium".
- Night mode activated by default: Whether the application should start on night mode by default. Note that this does not forbid the user from toggling it on/off during the session. Unchecked by default.
- Status bar visible by default: Whether the status bar should be visible by default. Note that this does not forbid the user from toggling it on/off during the session. Checked by default.
- Hide options related to solid motor simulation by default: As mentioned in section 5.2, the grain geometry tab is hidden by default when starting the application. Toggling this option off will result in the grain tab being visible by default when the application is launched. This also affects the other features related to grain combustion, such as Vielle's parameters in propellant tab (subsection 5.1.4), optimizations in nozzle tab (subsection 5.3.3) and time dependant simulations in the data tab (section 5.4.2). Note that, independently of whether this preference is toggled or not, the options related to solid motors can be toggled on or off at any time from the options menu (section 5.5.2), this preference only affects the startup behaviour. Checked by default.
- Default directory: The directory to be opened by default when an export file needs to be selected. Note that this only affects the first export of the session, as after it the folder that will be opened is the folder in which the last export was saved. Restarting the software will result in this default directory being opened when making an export again. Defaults to "INSTAL_PATH/Exports".
- *List separator: As explained in 5.8.3, this character is used to separate columns in the '.csv'

output files. Said character should coincide with the one used by the application that reads said file (for example Excel). Defaults to ";".

- Number of decimal digits in output: Here the user might specify how many decimal digits are written in the output file when performing an export. Defaults to 16.
- Use the same name and path for all exported files rather than requesting a name for each one: As explained in 5.8.3, if this is chequed the program will only ask once for a output file name (even if various files are outputted). If it is unchecked, every output file will require the user to name it. Checked by default.
- Image width & height: Size (in pixels) of figures outputted by ODEKO. Note that all figures are exported in '.png' format. Defaults to 800p width and 450p height.
- Baby mode active by default: Whether 'baby mode' is active by default when the application is run. This does not forbid the user from toggling it on/off during the session. Unchecked by default.
- Block access to preferences in babymode: If checked, it will prevent the user from accessing the preferences menu if 'baby mode' is toggled. Unchecked by default.
- *Lock this application in babymode: If this item is selected the application will be locked in 'baby mode'. It will always start by default in 'baby mode' and said mode can not be untoggled from the menu nor from the preferences menu. The only way to disable 'baby mode' once this item is toggled is to manually change the preferences file. Unchecked by default.

5.6.2 Propellant tab

- Default reactant temperature: Temperature at which the reactants are before the combustion to be used by default in calculations. Note that, if a reactant is defined only for a single temperature this value will not have any effect on it (since its enthalpy is only defined for a single value). Defaults to 298.15 K (recommended, as most enthalpies of formation are given for said temperature).
- Allow limit for top pressure range in vielle's parameters: Whether the last range of Vielle's parameters should have an upper pressure limit or not. This may be useful if the definition is only valid for a given interval since, if the upper limit is set, the user will get a warning if it is surpassed during calculations (see 5.7.1). Checked by default
- Minimum & maximum pressure in c^* graph: Upper and lower limits for the x-axis in the c^* vs pressure graphs present in the Propellant tab. Defaults to 5 and 100 bar respectively.
- Number of points in c^* graph: This will impact the time it takes to compute and represent the

c^* vs pressure graphs present in the Propellant tab. Defaults to 100.

- First to second reactant minimum & maximum ratio: Upper and lower limits for the x-axis in the c^* vs mixture ratio graphs present in the Propellant tab. Defaults to 1 and 10.
- Number of mixture ratios to be tested: his will impact the time it takes to compute and represent the c^* vs mixture ratio graphs present in the Propellant tab. Defaults to 200.
- Chamber pressure for optimization: The optimal mixture ratio given is valid for a specific chamber pressure, which is here specified. Note that the optimal point does not tend to vary much with pressure (the value for the c^* will change, but the optimum mixture ratio will hardly do so generally speaking). Defaults to 10 bar.
- By default only show species/reactions that are compatible with current reactants: If it is checked the user will only see species/reactions that are compatible with the reactants when the database is opened. Note that the database has a checkbox itself to change this behaviour, this preference only affects the default state. Checked by default.
- When all species/reactions are shown, disable those which are incompatible with current reactants: If checked, incompatible species will shown as disable, this not only offers a visual difference between compatible/incompatible species but it also prevents the user from selecting incompatible ones. If the user wishes to edit any specie/reaction without worrying about the current reactants, this item is best left unchecked. Checked by default.

5.6.3 Grain tab

- Combustion front evolution of the charge: Specifies which combustion fronts are shown in the upper right image in the grain tab. This change is only visual, has no effect on computation. By default shows radial and axial perspectives.
- Maximum number of iterations: Maximum number of iterations performed in order to find the chamber pressure when the computation involves a grain geometry (and therefore a propellant with properly defined Vielle's parameters). Defaults to 50.
- *Tolerance for convergence: Tolerance applied for the convergence towards a chamber pressure compatible with the grain and nozzle geometries and propellant characteristics. Note that altering this value will impact the results. A low tolerance (»0) will give erroneous results, whilst a too fine tolerance will imply long computation times (or it might not even converge). Defaults to 0.0004.
- Pressure initial estimate: Initial estimate for the chamber pressure in grain combustion problems. Defaults to 20 bar.

5.6.4 Nozzle tab

- Nozzle display: Offers different ways to perform the nozzle display. It can be fully displayed (half section); only a quarter section; or only the curve (the throat will be contained in the x axis). This change is only visual, but this also decides how the nozzle figures are exported. Defaults to "curve + axis".
- Add vertical lines: If checked, vertical lines at the entrance, throat and exit of the nozzle will be painted. This change is only visual but it will be present in exported figures. Checked by default.
- Colorfill nozzle: Whether the nozzle interior is colorfilled or not. This change is only visual, but it will be present in exported figures. Unchecked by default.
- Optimize with: Method employed in optimization, currently only IAC (Infinite Area Combustor) is supported. Refer to the theoretical background documentation for more information about these methods. Defaults to IAC.
- Area to be used in optimizations: Employed in thrust, specific impulse and total impulse optimizations. Initial combustion area is recommended if a thrust target is required for take off. Final combustion area is recommended if a thrust target is needed at the end of propulsion. Maximum area will maximize the maximum thrust value, whilst minimum area will maximize the minimum thrust value. Average combustion area is generally the best choice, since it will offer a good optimization for the entire combustion (given a near-neutral combustion area). Defaults to "Average combustion".
- Initial temperature for reactants in optimization: Temperature at which the reactants are considered to be prior to the combustion for the optimization process. This only affects reactants that are not single defined. Defaults to 298.15 K (recommended value as most enthalpies of formation are defined for this temperature).
- Maximum number of iterations for optimization: Some optimizations may require a high number of iterations or a long computation time (specially total impulse optimization). This setting allows the user to limit it. Defaults to 200.
- *Tolerance for target value in optimization: This should be change depending on the precision with which the user wants to achieve the specified chamber pressure or combustion time. Defaults to 0.01.
- *Number of points used in particle swarm optimization: particle swarm is used for thrust, specific impulse and total impulse optimizations. A greater number will offer more reliable results, but it will increase computation time for the same number of iterations. Defaults to 10.
- Freeze flow at the throat in optimizations: Whether the optimizations should consider that the

flow is frozen at the throat of the nozzle. This setting is recommended for solid propellants, since it generally offers a better approximation than chemical equilibrium throughout the entire nozzle. Checked by default.

5.6.5 Calculations tab

- Type of display: Whether the data is displayed in a graph or in a table. This change is only visual and does not affect the export options. Note that graphs may represent more than one configurations, whilst tables cannot. On the other hand, tables clearly represent all variables simultaneously, as well as show the values in the combustion chamber. Changing this option does not remove any calculations performed, so it can be changed to view the same simulation (or set of simulations) in both ways. Defaults to Graph.
- Number of seconds before "Abort calculations" button appears: If calculations take too long to finish the "Running..." button is substituted with an "Abort calculations" button, which allows the user to stop the calculation if desired. This preference allows the user to modify how many seconds it takes the abort button to appear. This button is also present when performing optimizations in the Nozzle tab. More information on the abort behaviour can be found in section 5.3.3 and section 5.4.2. Defaults to 15s.
- Normalize values plotted in graph: If checked all plotted values will be of order one. Generally speaking values will be normalized using the maximum value of the plot or a characteristic value (such as the chamber pressure). Some values will not be affected (such as the Mach number or mass fractions) as they already are of order 1. Unchecked by default.
- Always start graph y-axis at 0: Whether the 0 value should always be visible in the y-axis. Note that this may render some graphs useless (they will appear as straight lines due to the scaling that would need to be applied). Unchecked by default.
- Show freeze point in graphs: Whether the point at which the flow was frozen should be marked in the graph or not. It is recommended to do so, since it offers more understandable graphs & output. Checked by default.
- Show points at which ODK changes from pressure to area: Whether to mark the two points in between which the throat iteration is performed for ODK problems. For more information on why these points are relevant refer to the theory documentation. Checked by default.
- Only plot frozen variables even if the flow is in equilibrium: Generally, the variables of interest for the user are the equilibrium variables. Nevertheless, this depend on which equations are these variables being used in, for some application frozen variables might be needed instead of their equilibrium counterparts. For a more in-depth explanation on the difference between both types

refer to the theory manual. Unchecked by default.

- Derivative & Interpolation precision: A higher precision increases the number of interpolation points used, each step increases the number of points by two. The 'Low' setting only uses two points. Defaults to "High".
- Integration method: Method to be employed. Further information about these methods can be found in the documentation for the theoretical background of the application. Defaults to "Trapezoidal rule".
- Flow velocity in combustion chamber: Only used to compute the convective heat transfer coefficient in the combustion chamber. Defaults to 30m/S. For more information on how the convective heat transfer coefficient is computed and the relevance of the parameters that will be presented subsequently, refer to the theory manual.
- *Coefficient for Prandtl-Reynolds product: The Nusselt number is approximated as seen in equation 5.1, as a basis to calculate the convective heat transfer coefficient. This coefficient is represented as 'c'. Defaults to 0.026.
- *Exponents for Reynolds number: Exponent 'm' in equation 5.1. Defaults to 0.8.
- *Exponent for Prandtl number: Exponent 'n' in equation 5.1. Defaults to 0.4.
- *Viscosity exponent for temperature: In the limit layer, viscosity is considered proportional to an exponent of temperature as seen in equation 5.2. Defaults to 0.6.
- *Conductivity exponent for temperature: In the limit layer, conductivity is considered proportional to an exponent of temperature as seen in equation 5.2. Defaults to 0.6.
- *Cp exponent for temperature: In the limit layer, cp is considered proportional to an exponent of temperature as seen in equation 5.2. Defaults to 0.
- *Curvature correction exponent: Exponent for the curvature correction term in Bartz equation, used to calculate the convective heat transfer coefficient. Defaults to 0.1.

$$Nu = c \cdot Re^m Pr^n \tag{5.1}$$

$$\mu \propto T^{\omega_\mu} \quad ; \quad \lambda \propto T^{\omega_\lambda} \quad ; \quad C_p \propto T^{\omega_c} \tag{5.2}$$

5.6.6 Chemical equilibrium

This section can be better understood if the user has carefully read the documentation on the theoretical background of ODEKO.

- Maximum number of iterations for equilibrium calculation: Generally convergence is achieved in very few iterations. Nonetheless, for combustion containing condensed phases, a greater number of iterations may be required. Therefore this value is advised to be kept high. Although it should not be too big in case the program gets stuck and is unable to find convergence (it would imply a great amount of time before it is forced to exit calculations). Defaults to 8000.
- Maximum iterations for ion balance convergence: After convergence is achieved, a number of iterations are performed until ions are balanced. Should take a low amount of iterations. Defaults to 100.
- Minimum composition for specie to be considered: Species with compositions lower than this value will be considered to not be present in the mixture (therefore, computation time is not wasted on them). Defaults to 1e-10.
- Minimum composition for ion to be considered: Ions with compositions lower than this value will be considered to not be present in the mixture (therefore, computation time is not wasted on them). Defaults to 1e-10.
- *Tolerance for temperature convergence: The user should not change this unless a proper understanding of the theoretical background is possessed. Defaults to 0.0001.
- *Tolerance for entropy convergence: The user should not change this unless a proper understanding of the theoretical background is possessed. Defaults to 5e-5.
- *Tolerance for molecular weight convergence: The user should not change this unless a proper understanding of the theoretical background is possessed. Defaults to 5e-6.
- *Tolerance for species convergence: The user should not change this unless a proper understanding of the theoretical background is possessed. Defaults to 1e-6.
- *Tolerance for ion convergence: The user should not change this unless a proper understanding of the theoretical background is possessed. Defaults to 0.0001.
- *Tolerance for species with extremely low composition: The user should not change this unless a proper understanding of the theoretical background is possessed. Defaults to 0.001.
- Seed temperature for combustion chemical equilibrium: Temperature used as initial estimate of combustion temperature. If calculations converge correctly, all seed temperatures should give the same result, it only affects the number of iterations it takes to achieve it. Defaults to 3800K.
- Initial estimate for mixture composition: Initial estimate for the inverse of the molecular weight of the mixture. Defaults to 0.1.
- *Initial value for introduced condensed species: When a condensed specie is introduced, it will

be considered to have the composition here specified. Defaults to 1e-5.

- *Specie composition at forced reset: When a reset is forced (due to a singularity during calculations) this is the composition to which species will be reset. Defaults to 1e-6.
- *Ion composition for ion reset: When an ion reset is forced this is the composition to which species will be reset. Defaults to 1e-5.
- *Collision cross-section ratio (A*): Used for transport properties computation, user should not need to change it. Defaults to 1.1.

5.6.7 ODE

This section can be better understood if the user has carefully read the documentation on the theoretical background of ODEKO.

- *Tolerance for throat convergence: The user should not change this unless a proper understanding of the theoretical background is possessed. Defaults to 0.004.
- *Tolerance for pressure convergence: The user should not change this unless a proper understanding of the theoretical background is possessed. Defaults to 4e-5.
- *Tolerance for temperature convergence (frozen flow): The user should not change this unless a proper understanding of the theoretical background is possessed. Defaults to 5e-5
- Maximum iterations for throat convergence: Generally should only require a small number of iterations. Defaults to 20.
- Maximum iterations for pressure convergence: Generally should only require a small number of iterations. Defaults to 10.
- Maximum iterations for temperature convergence: Generally should only require a small number of iterations. Defaults to 10.
- Maximum iterations for frozen throat convergence: it may take a greater number of iterations than the other computations. Nonetheless, it it is prematurely terminated, the precision lost can be generally disregarded. Defaults to 20.
- Flow velocity at injector plane: Specifying this value accurately will also improve the accuracy of the FAC simulations. Defaults to 0 m/S (injector velocity disregarded).
- *Pinf/Pi forced reset for FAC problem: In very rare circumstances the FAC problem may need to force a rest for the pressure relation at the throat, this value is given here. Defaults to 1.0001 (value should be kept very slightly over 1).

- *Minimum area difference to consider new initial point in FAC: If the given combustor area differs too much from the nozzle entrance's area, a new point is created for the computation. Defaults to 1e-5.
- *X percentage offset for new initial point in FAC: The offset of the new point mentioned in the previous item. Defaults to 1.05.
- *Tolerance for FAC problem convergence: This should not be changed by the user unless a deep theoretical understanding is possessed. Defaults to 2e-5.
- Maximum number of iterations for FAC Pinf calculation: Depending on the problem this may take some iterations to converge. Defaults to 20.

5.6.8 ODK

This section can be better understood if the user has carefully read the documentation on the theoretical background of ODEKO.

- *Differential equation solver relative tolerance: This value greatly impacts the maximum tolerances that the ODK calculations are able to handle. it is not recommended to be outside the range of 1e-10 to 1e-13. It does not have a significant impact on computation time. Defaults to 1e-10.
- *Differential equation solver absolute tolerance: It is not an specially important tolerance value. Nonetheless, it is not recommended to change its default value. Defaults to 1e-6.
- Maximum iterations for global convergence: This number should be big, since it encompasses all other iterations during ODK calculations. It is also very common for ODK computations to take a great amount of iterations, specially for certain propellants. Defaults to 400.
- Maximum iterations for throat convergence: This iteration is performed at the end, therefore, it should not have a high maximum amounts or it could significantly impact computation time. Defaults to 20.
- Maximum iterations for mass flow convergence: This is the most basic iteration, it is required to converge many times; therefore its maximum iterations should not be too high in case it gets stuck by an unforeseen issue. Defaults to 20.
- Maximum iterations for near-throat points convergence: The maximum iterations should be similar to those set in the previous two items. Defaults to 40.
- *Area relation for throat iteration start/end: It is recommended for the throat iteration to have a significant number of points, although not too many (it slows computation). For a nozzle of about 200-800 points, the default values are more than sufficient. Increments to those values should

not change the obtained results. If it does, an error must have occurred during computation. Defaults to 1.08 for iteration start and 1.12 for iteration end.

- *Tolerance for throat convergence: Finer tolerances than the default values are not generally possible to be achieved. If the solver tolerance is worsened, this tolerance should be changed accordingly. Defaults to 0.0001.
- *Tolerance for near-throat points convergence: Finer tolerances than the default values are not generally possible to be achieved. If the solver tolerance is worsened, this tolerance should be changed accordingly. Defaults to 0.0001.
- *Tolerance for mass flow convergence: Finer tolerances than the default values are not generally possible to be achieved. If the solver tolerance is worsened, this tolerance should be changed accordingly. Defaults to 0.0001.
- *Tolerance for throat position convergence: Finer tolerances than the default values are not generally possible to be achieved. If the solver tolerance is worsened, this tolerance should be changed accordingly. This tolerance is generally the hardest one to achieve out of the four. Defaults to 0.0001.
- *Area relevance for mass flow correction: The effect of this value can be seen in the theoretical background. This values are not fully optimized, so the intrepid user may change them to try to find better ones. Note that this may completely break convergence. Defaults to 150.
- *Maximum pressure correction for near-throat points: The effect of this value can be seen in the theoretical background. This values are not fully optimized, so the intrepid user may change them to try to find better ones. Note that this may completely break convergence. Defaults to 0.01.
- *Throat position relevance for pressure correction: The effect of this value can be seen in the theoretical background. This values are not fully optimized, so the intrepid user may change them to try to find better ones. Note that this may completely break convergence. Defaults to 0.1.

5.7 Warnings & critical errors

5.7.1 Warnings

This subsection contains a complete list of all possible warnings given by the program, along with a short explanation of what may have caused them and possible solutions. Warnings let the user now about a situation that occurred during calculations. It does not need to be a problem, although the user should check them in order to know if the results obtained may have been compromised.

- SP00: Unknown error. Its occurrence is generally due to a programming mistake, it should never happen.
- SP01: The temperature of the mixture was below the range of definition of C_p for one or more species. The value of C_p is assumed to be that of the smallest temperature in the range of definition.
- SP02: The temperature of the mixture was over the range of definition of C_p for one or more species. The value of C_p is assumed to be that of the biggest temperature in the range of definition.
- SP03: Equivalent to SP01 but related to viscosity range of definition.
- SP04: Equivalent to SP02 but related to viscosity range of definition.
- SP05: Equivalent to SP01 but related to conductivity range of definition.
- SP06: Equivalent to SP02 but related to conductivity range of definition.
- SP07: Computing of viscosity was attempted for a specie without a viscosity definition. This warning occurs when transport properties are computed and one or more species only have conductivity defined.
- SP08: Computing of conductivity was attempted for a specie without a viscosity definition. This warning occurs when transport properties are computed and one or more species only have viscosity defined.
- SP09: One of the species used was defined only for a single temperature value. The definition for said value is used for all values. The most likely outcome is that the calculations performed with this specie are invalid.
- SP10: Reached maximum number of iterations for equilibrium calculation. This may happen if the maximum iterations are limited to a very low amount or if the mixture is bad-behaved and has convergence problems (specially if a lot of condensed species are involved).
- SP11: Transport properties calculation was chequed in the configuration, but no species present

in the mixture have transport properties defined in their database. This supposes no error, but is recommended to uncheck the transport properties calculation in order to gain computation speed.

- SP12: Reached maximum number of iterations for ion balance in equilibrium calculation. This means that the mixture has converged but ion balanced could not be obtained, generally due to a very low number for the maximum of iterations.
- SP13: One or more reactions had efficiencies for a third body not present in the mixture. This supposes no problem, said species will simply not be considered since they are not present in the mixture.
- SP14: Species compositions were reset during equilibrium calculations in order to reintroduce an atom in the mixture. This may be caused by the removal of a no longer considered condensed specie. If no other warnings/errors are present, the user should disregard this event since the problem was fixed.
- SP15: Computation of equilibrium conductivity failed. Frozen value given instead. This is probably due to the selected reaction (kinetics) scheme, as it is required to compute equilibrium conductivity (even when using ODE).
- SP16: Computation of equilibrium gaseous cp failed. Frozen value of Pr given. Gaseous cp is not a direct output, but it is used to compute Pr. This is probably due to the selected reaction (kinetics) scheme, as it is required to compute equilibrium conductivity (even when using ODE).
- ODE00: Maximum number of iterations for throat calculation reached. This is extremely rare and should only occur if the maximum number of iterations permitted is extremely low.
- ODE01: Maximum number of iterations for nozzle point pressure convergence reached. Generally this is due to a extremely low limit to the maximum iterations or problems with the equilibrium calculations.
- ODE02: Maximum number of iterations for frozen temperature calculation reached. Extremely rare, probably due to bad convergence if frozen point is before the throat or to a low limit to the maximum iterations.
- ODE03: Maximum number of iterations for frozen throat calculation reached. Can be a common warning since it is a difficult to achieve convergence. Nonetheless it is not very significant if it is not achieved as long as the number of iterations performed is sufficiently large. For a sufficiently large maximum number of iterations this warning will never occur, although it may impact computing time slightly.
- ODE04: Maximum number of iterations for FAC initial conditions reached. Not common, may be solved increasing maximum number of iterations.

- ODK03: It occurs when the area relation criteria to switch from given area to given pressure calculations in ODK in the convergent nozzle is invalid. It is usually caused due to no point in the nozzle being in between the throat an the area relation given. The user should change the criteria or increase the number of points in the nozzle.
- ODK04: Equivalent to OPDK03 but for the divergent nozzle, when calculations swithc from given pressure to given area.
- ODK05: Reached maximum number of iterations for global convergence of ODK problem. Generally this limit should be set very high, since a lot of iterations are required to accomodate all the sub-iterations present in the ODK problem.
- ODK06: Reached maximum number of iterations for throat convergence of ODK problem. Generally this is caused by a very fine tolerance for the throat position, which is a very hard convergence to achieve. It is recommended to reduce this tolerance. Alternatively the iteration limit may be increased but this will probably not avoid the convergence problem.
- ODK07: Reached maximum number of iterations for near-throat points convergence in ODK. The most common cause for this warning is that the amount of points calculated with pressure is to big. It is recommended to reduce the area relations at which the switch happens.
- ODK08: Reached maximum number of iterations for mass-flow convergence in ODK. This error occurs because the tolerance set is to fine for the number of points in the nozzle. Said number of points greatly impacts how many significant figures of the mass flow are maintained constant throughout the nozzle. Therefore, if the tolerance is too fine, convergence may never be achieved.
- V00: Pressure in the combustion chamber is beneath the lower limit for Vielle's parameters definition. This will have an effect in the validity of the results obtained. It is recommended to try and re-define the intervals or to change the nozzle geometry.
- V01: Equivalent to V00 but the pressure exceeds the upper limit.
- V02: Maximum number of iterations for pressure calculations reached. Generally due to the iteration procedure being stuck near an interval change in Vielle's parameters definition. There is no way to solve this problem, but the solution offered should be very close to the correct solution.
- OPT00: Maximum number iterations for optimization reached. This can be caused either by an invalid target, too strict restrictions or a bad-behaving Vielle's intervals definition.
- OPT01: Early exit from optimization procedure. This error is caused by an unexpected behaviour during the optimization procedure, it may be caused by an unusual interval definition for Vielle's parameters. A bug report should be filed.

- F00: Species file could not be found at the location given in preferences, probably it was deleted or moved to a different location.
- F02: Reactants file could not be found at the location given in preferences, probably it was deleted or moved to a different location.
- F04: Kinetics file could not be found at the location given in preferences, probably it was deleted or moved to a different location.
- F06: Propellants file could not be found at the location given in preferences, probably it was deleted or moved to a different location.
- F08: Nozzle file could not be found at the location given in preferences, probably it was deleted or moved to a different location.
- F10: Grain file could not be found at the location given in preferences, probably it was deleted or moved to a different location.
- F12: Configurations file could not be found at the location given in preferences, probably it was deleted or moved to a different location.
- F14: Preferences file could not be found at the location given in preferences, probably it was deleted or moved to a different location.
- F16: Species file at the location given in preferences was corrupted, probably due to manually modifying the file. If the user chose not to delete the affected items on startup, the file should be corrected manually or it may cause malfunctions in the software.
- F18: Reactants file at the location given in preferences was corrupted, probably due to manually modifying the file. If the user chose not to delete the affected items on startup, the file should be corrected manually or it may cause malfunctions in the software.
- F20: Kinetics file at the location given in preferences was corrupted, probably due to manually modifying the file. If the user chose not to delete the affected items on startup, the file should be corrected manually or it may cause malfunctions in the software.
- F22: Propellants file at the location given in preferences was corrupted, probably due to manually modifying the file. If the user chose not to delete the affected items on startup, the file should be corrected manually or it may cause malfunctions in the software.
- F24: Nozzle file at the location given in preferences was corrupted, probably due to manually modifying the file. If the user chose not to delete the affected items on startup, the file should be corrected manually or it may cause malfunctions in the software.

- F26: Grain file at the location given in preferences was corrupted, probably due to manually modifying the file. If the user chose not to delete the affected items on startup, the file should be corrected manually or it may cause malfunctions in the software.
- F28: Configuirations file at the location given in preferences was corrupted, due to manually modifying the file. If the user chose not to delete the affected items on startup, the file should be corrected manually or it may cause malfunctions in the software.
- F30: Preferences file at the location given in preferences was corrupted, affected items will be returned to their default values. This may happen due to manually modifying the file or using a file from a previous version (in this case old values are unaffected and new preferences items are added to the file). All affected items were reset to its default values.

5.7.2 Critical errors

This subsection contains a complete list of all possible critical errors given by the program, along with a short explanation of what may have caused them and possible solutions. Critical errors forbid the program from continuing its calculations and may even prevent the user from starting them. They represent the occurrence of a serious problem and should be solved immediately.

- ODK00: Failure during initialization of the differential equations solver (CVodes). This prevents the solver from functioning and the ODK calculation is aborted. There is no way to solve it, a bug report must be sent to the developer.
- ODK01: Failure during calculations of the differential equations solver (CVodes). ODK calculation is aborted. There is no way to solve it, a bug report must be sent to the developer.
- ODK02: Failure during closure of the differential equations solver (CVodes). The calculations are finished, but a memory leak might have happened. There is no way to solve it, a bug report must be sent to the developer.
- U: A unit conversion failed, unit not defined in program. This error only happens if the user has tampered with the save files and manually changed a value. If it is not the case, please file a bug report to the developer.
- L00: All considered species have the same element ratio as that of the reactants. A soft-rest is set in place. This problem may arise if to little species are being considered in the mixture.
- L01: At least one condensed specie's elements are a linear combination of the others. One of the condensed species is automatically removed, but this may cause the solution to not be valid or prevent convergence.
- L02: No gaseous species were included in the calculation. Therefore, the mixture is invalid and

equilibrium calculations can not be performed.

- L03: At least one positive and one negative ion must be considered (or none at all). Having only positive or only negative ions would make impossible charge equilibrium, which is required.
- L04: One of the species present in the propellant definition was not found in the database. This error happens if the user has deleted species that a propellant was considering.
- L05: One of the reactants present in the propellant definition was not found in the database. This error happens if the user has deleted a reactant that forms part of a propellant.
- L06: One of the reactions present in the propellant definition was not found in the database. This error happens if the user has deleted a reaction that was considered by a propellant.
- L07: Propellant load failed for an undetermined reason (probably a file could not be accessed or was corrupted).
- L08: Nozzle load failed for an undetermined reason (probably a file could not be accessed or was corrupted).
- L09: Grain load failed for an undetermined reason (probably a file could not be accessed or was corrupted).
- F01: Species file could not be found at its default location, probably it was deleted or moved to a different location.
- F03: Reactants file could not be found at its default location, probably it was deleted or moved to a different location.
- F05: Kinetics file could not be found at its default location, probably it was deleted or moved to a different location.
- F07: Propellants file could not be found at its default location, probably it was deleted or moved to a different location.
- F09: Nozzle file could not be found at its default location, probably it was deleted or moved to a different location.
- F11: Grain file could not be found at its default location, probably it was deleted or moved to a different location.
- F13: Configurations file could not be found at its default location, probably it was deleted or moved to a different location.
- F15: Preferences file could not be found at its default location, probably it was deleted or moved to a different location.

- F17: Species file at its default location was corrupted, probably due to manually modifying the file. If the user chose not to delete the affected items on startup, the file should be corrected manually or it may cause malfunctions in the software.
- F19: Reactants file at its default location was corrupted, probably due to manually modifying the file. If the user chose not to delete the affected items on startup, the file should be corrected manually or it may cause malfunctions in the software.
- F21: Kinetics file at its default location was corrupted, probably due to manually modifying the file. If the user chose not to delete the affected items on startup, the file should be corrected manually or it may cause malfunctions in the software.
- F23: Propellants file at its default location was corrupted, probably due to manually modifying the file. If the user chose not to delete the affected items on startup, the file should be corrected manually or it may cause malfunctions in the software.
- F25: Nozzle file at its default location was corrupted, probably due to manually modifying the file. If the user chose not to delete the affected items on startup, the file should be corrected manually or it may cause malfunctions in the software.
- F27: Grain file at its default location was corrupted, probably due to manually modifying the file. If the user chose not to delete the affected items on startup, the file should be corrected manually or it may cause malfunctions in the software.
- F29: Configuirations file at its default location was corrupted, due to manually modifying the file. If the user chose not to delete the affected items on startup, the file should be corrected manually or it may cause malfunctions in the software.
- F31: Preferences file at its default location was corrupted, affected items will be returned to their default values. This may happen due to manually modifying the file or using a file from a previous version (in this case old values are unaffected and new preferences items are added to the file). All affected items were reset to its default values.

5.8 File structure

5.8.1 Save Files

Save files contain the information of the different components employed by the application. All save files are '.json' files. The files are both written from and read by the application in order to use the data of said components. It is highly recommended to not open nor tamper with these files in any way. The default names for the files are:

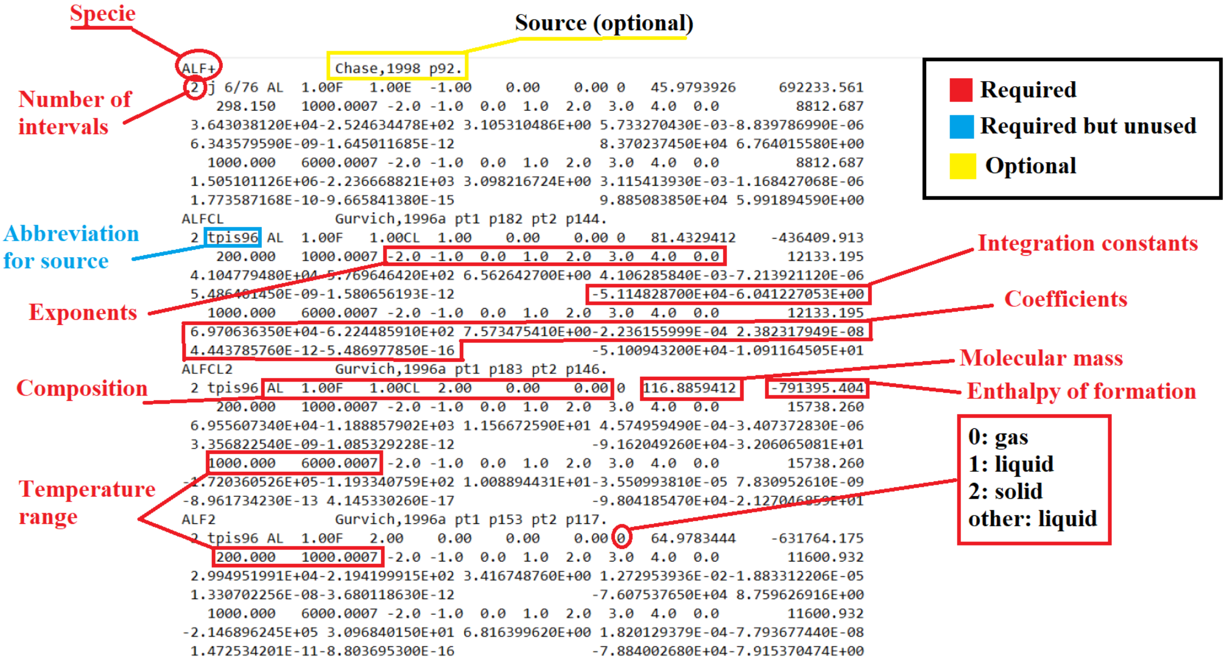
- Reactants: Contains information about the reactants of the application. This information can be modified in the Propellant Tab, clicking on the 'Reactant Database' button.
- Species: Contains information about the species of the application. This information can be modified in the Propellant Tab, clicking on the 'Specie Database' button. Note that it also contains the information about the last loaded species (further explained in subsection 5.1.5).
- Kinetics: Contains information about the chemical reactions of the application. This information can be modified in the Propellant Tab, clicking on the 'Kinetics Database' button. Note that it also contains the information about the last loaded reactions (further explained in subsection 5.1.5).
- Propellants: Contains the information necessary to describe the various propellants created by the user.
- Grains: Contains the information necessary to describe the various grain geometries created by the user.
- Nozzles: Contains the information necessary to describe the various nozzles created by the user.
- Configurations: Contains information of the different configurations for engine simulation in the 'Calculations tab' saved by the user.
- Preferences: Contains all the information regarding the users preferences. This is the only file that could be needed to be manually modified, further explained in section 5.5.2.

By default, after installation, the program comes with all these files (and a set of basic components and information). Nonetheless, the option to use different compatible files is given, as is explained in section 5.5.1. This may be useful in order to exchange save files with a colleague, in order to test the components created by them. Note that using configurations saved with a different file set may result in errors.

5.8.2 Input files

Thermodynamic data files

Currently, the only supported format for thermodynamic data input files is the one used by NASA's CEA, although other format types are planned to be implemented in future releases. Figure 5.22 shows an example of such a file, where different parts are highlighted in order to show the format to be followed. Note that all this information can also be input through the application (as seen in section 5.1.2).

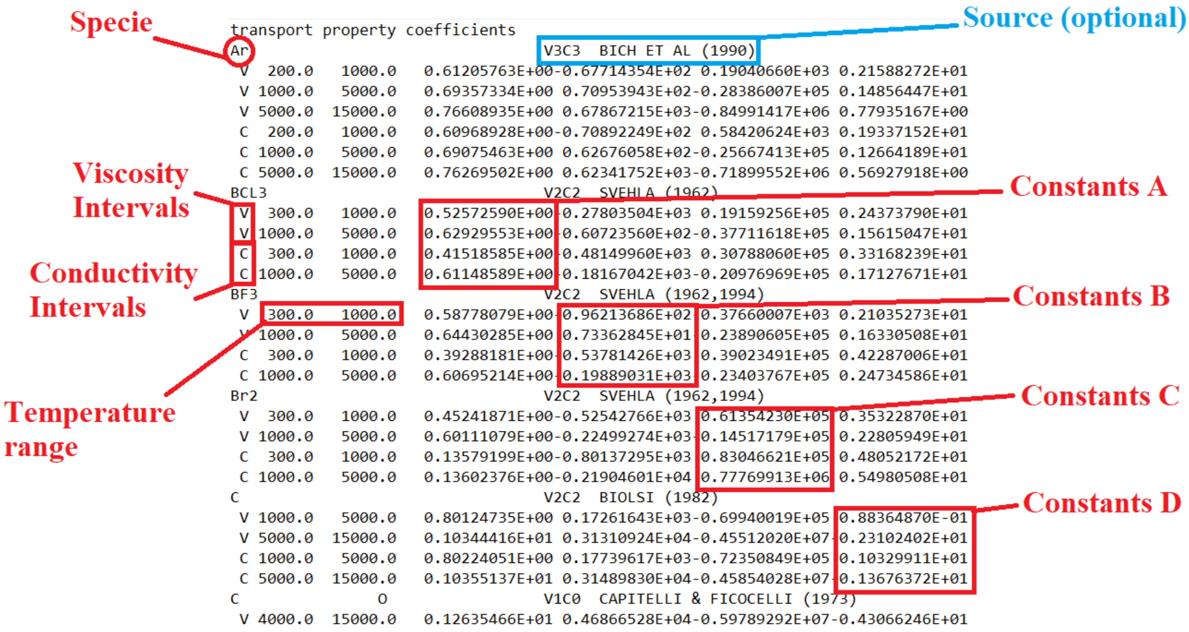


```
ALF+                  Chase,1998 p92.
 2 j 6/76 AL  1.00F   1.00E  -1.00    0.00    0.00 0   45.9793926     692233.561
    298.150   1000.0007 -2.0 -1.0  0.0  1.0  2.0  3.0  4.0  0.0           8812.687
 3.643038120E+04-2.524634478E+02 3.105310486E+00 5.733270430E-03-8.839786990E-06
 6.343579590E-09-1.645011685E-12                 8.370237450E+04 6.764015580E+00
   1000.000   6000.0007 -2.0 -1.0  0.0  1.0  2.0  3.0  4.0  0.0           8812.687
 1.505101126E+06-2.236668821E+03 3.098216724E+00 3.115413930E-03-1.168427068E-06
 1.773587168E-10-9.665841380E-15                 9.885083850E+04 5.991894590E+00
ALFCL                 Gurvich,1996a pt1 p182 pt2 p144.
 2 tpis96 AL  1.00F   1.00CL  1.00    0.00    0.00 0   81.4329412    -436409.913
    200.000   1000.0007 -2.0 -1.0  0.0  1.0  2.0  3.0  4.0  0.0          12133.195
 4.104779480E+04-5.769646420E+02 6.562642700E+00 4.106285840E-03-7.213921120E-06
 5.486401450E-09-1.580656193E-12                -5.114828700E+04-6.041227053E+00
   1000.000   6000.0007 -2.0 -1.0  0.0  1.0  2.0  3.0  4.0  0.0          12133.195
 6.970636350E+04-6.224485910E+02 7.573475410E+00-2.236155999E-04 2.382317949E-08
 4.443785760E-12-5.486977850E-16                -5.100943200E+04-1.091164505E+01
ALFCL2                Gurvich,1996a pt1 p183 pt2 p146.
 2 tpis96 AL  1.00F   1.00CL  2.00    0.00    0.00 0  116.8859412    -791395.404
    200.000   1000.0007 -2.0 -1.0  0.0  1.0  2.0  3.0  4.0  0.0          15738.260
 6.955607340E+04-1.188857902E+03 1.156672590E+01 4.574959490E-04-3.407372830E-06
 3.356822540E-09-1.085329228E-12                -9.162049260E+04-3.206065081E+01
   1000.000   6000.0007 -2.0 -1.0  0.0  1.0  2.0  3.0  4.0  0.0          15738.260
-1.720360526E+05-1.193340759E+02 1.008894431E+01-3.550993810E-05 7.830952610E-09
-8.961734230E-13 4.145330260E-17                -9.804185470E+04-2.127046859E+01
ALF2                  Gurvich,1996a pt1 p153 pt2 p117.
 2 tpis96 AL  1.00F   2.00    0.00    0.00    0.00 0   64.9783444    -631764.175
    200.000   1000.0007 -2.0 -1.0  0.0  1.0  2.0  3.0  4.0  0.0          11600.932
 2.994951991E+04-2.194199915E+02 3.416748760E+00 1.272953936E-02-1.883312206E-05
 1.330702256E-08-3.680118630E-12                -7.607537650E+04 8.759626916E+00
   1000.000   6000.0007 -2.0 -1.0  0.0  1.0  2.0  3.0  4.0  0.0          11600.932
-2.146896245E+05 3.096840150E+01 6.816399620E+00 1.820129379E-04-7.793677440E-08
 1.472534201E-11-8.803695300E-16                -7.884002680E+04-7.915370474E+00
```

Figure 5.22: Format for thermodynamic data input files

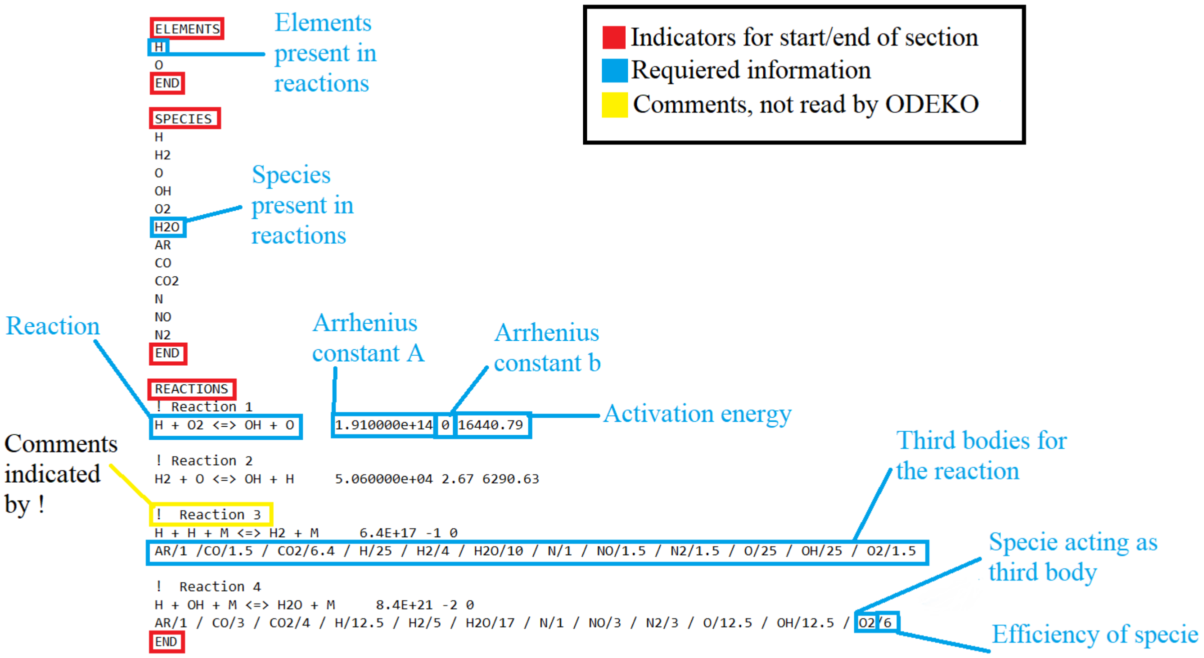
Transport properties data files

Currently, the only supported format for transport properties data input files is the one used by NASA's CEA, although other format types are planned to be implemented in future releases. Figure 5.23 shows an example of such a file, where different parts are highlighted in order to show the format to be followed. Note that all this information can also be input through the application (as seen in section 5.1.2).



```
transport property coefficients
Ar                                      V3C3  BICH ET AL (1990)
 V  200.0   1000.0   0.61205763E+00-0.67714354E+02 0.19040660E+03 0.21588272E+01
 V 1000.0   5000.0   0.69357334E+00 0.70953943E+02-0.28386007E+05 0.14856447E+01
 V 5000.0  15000.0   0.76608935E+00 0.67867215E+03-0.84991417E+06 0.77935167E+00
 C  200.0   1000.0   0.60968928E+00-0.70892249E+02 0.58420624E+03 0.19337152E+01
 C 1000.0   5000.0   0.69075463E+00 0.62676058E+02-0.25667413E+05 0.12664189E+01
 C 5000.0  15000.0   0.76269502E+00 0.62341752E+03-0.71899552E+06 0.56927918E+00
BCL3                                    V2C2  SVEHLA (1962)
 V  300.0   1000.0   0.52572590E+00 0.27803504E+03 0.19159256E+05 0.24373790E+01
 V 1000.0   5000.0   0.62929553E+00 0.60723560E+02-0.37711618E+05 0.15615047E+01
 C  300.0   1000.0   0.41518585E+00 0.48149960E+03 0.30788060E+05 0.33168239E+01
 C 1000.0   5000.0   0.61148589E+00 0.18167042E+03-0.20976969E+05 0.17127671E+01
BF3                                     V2C2  SVEHLA (1962,1994)
 V  300.0   1000.0   0.58778079E+00 0.96213686E+02 0.37660007E+03 0.21035273E+01
 V 1000.0   5000.0   0.64430285E+00 0.73362845E+01 0.23890605E+05 0.16330508E+01
 C  300.0   1000.0   0.39288181E+00 0.53781426E+03 0.39023491E+05 0.42287006E+01
 C 1000.0   5000.0   0.60695214E+00 0.19889031E+03 0.23403767E+05 0.24734586E+01
Br2                                     V2C2  SVEHLA (1962,1994)
 V  300.0   1000.0   0.45241871E+00-0.52542766E+03 0.61354230E+05 0.35322870E+01
 V 1000.0   5000.0   0.60111079E+00-0.22499274E+03 0.14517179E+05 0.22805949E+01
 C  300.0   1000.0   0.13579199E+00-0.80137295E+03 0.83046621E+05 0.48052172E+01
 C 1000.0   5000.0   0.13602376E+00-0.21904601E+04 0.77769913E+06 0.54980508E+01
C                                       V2C2  BIOLSI (1982)
 V 1000.0   5000.0   0.80124735E+00 0.17261643E+03-0.69940019E+05 0.88364870E-01
 V 5000.0  15000.0   0.10344416E+01 0.31310924E+04-0.45512020E+07 0.23102402E+01
 C 1000.0   5000.0   0.80224051E+00 0.17739617E+03-0.72350849E+05 0.10329911E+01
 C 5000.0  15000.0   0.10355137E+01 0.31489830E+04-0.45854028E+07 0.13676372E+01
C                 O                     V1C0  CAPITELLI & FICOCELLI (1973)
 V 4000.0  15000.0   0.12635466E+01 0.46866528E+04-0.59789292E+07-0.43066246E+01
```

Figure 5.23: Format for transport properties data input files



```
ELEMENTS
H
O
END

SPECIES
H
H2
O
OH
O2
H2O
AR
CO
CO2
N
NO
N2
END

REACTIONS
! Reaction 1
H + O2 <=> OH + O      1.910000e+14 0 16440.79

! Reaction 2
H2 + O <=> OH + H      5.060000e+04 2.67 6290.63

!  Reaction 3
H + H + M <=> H2 + M     6.4E+17 -1 0
AR/1 /CO/1.5 / CO2/6.4 / H/25 / H2/4 / H2O/10 / N/1 / NO/1.5 / N2/1.5 / O/25 / OH/25 / O2/1.5

!  Reaction 4
H + OH + M <=> H2O + M     8.4E+21 -2 0
AR/1 / CO/3 / CO2/4 / H/12.5 / H2/5 / H2O/17 / N/1 / NO/3 / N2/3 / O/12.5 / OH/12.5 / O2/6
END
```

Figure 5.24: Format for chemical reactions data input files

Chemical reactions data files

Currently, the only supported format for chemical reactions data input files is the one used by Ansys CHEMKIN, although other format types are planned to be implemented in future releases. Figure 5.24 shows an example of such a file, where different parts are highlighted in order to show the format to be followed. Note that all this information can also be input through the application (as seen in section 5.1.2).

5.8.3 Output files

Output files are generated by the application when a data export is conducted by the user. It generates '.csv' files, which can be read using Microsoft's Excel. Note that in order for the files to be properly read, the delimiter character used by the ODEKO must match that used by the reading application (e.g. Excel). In the preferences menu, said character may be changed (see subsection 5.6.1). Note that this character can also be changed in Excel. The name for the csv file is chosen by the user.

The export structure is different for each tab and also differs based on the selected data from that tab to be exported. Specific information can be found in the section regarding each tab (5.1.6 5.2.3 5.3.4 5.4.5).

Figures may also be exported in several tabs. The image size can be modified in the preferences menu (subsection 5.6.1). By default, the image will have the same name as the csv file. In case of multiple figures, they will have the same base name, followed by a differentiating character string. Note that this may be changed in the preferences menu (subsection 5.6.1), which would force the user to choose a name for each generated file.

5.9 Bug reports & future releases

5.9.1 Where & how to file a bug report

Bug reports can be sent to odeko@nickdejongc.com and a dedicated bug report page will soon be made available at the ODEKO website.

It is appreciated that bugs are reported with the most amount of information possible. The most important pieces of information are the error caused by the program (let it be a crash, a freeze, a malfunction...) and the setup needed to replicate the error (parameters used in the calculations, preference settings...). It is always appreciated to include the preferences and database files used by the user, so it is easier to replicate the issue.

Bug reports will be ordered by relevance, prioritizing the most critical bugs first. A new release will not be launched for each solved bug. Whenever a sufficient amount of changes or bug fixes are implemented a new release will come out. If a major bug is resolved a new release with that patch will also be put out.

5.9.2 Where to find information about new releases

Information about new releases will be posted on the ODEKO website. Here the latest version of the software as well as of the documentation will be available. Only the latest version of the software and documentation will be available, but the user will also be able to find the patch notes for all releases, which contain information on bug fixes and added features for each release with respect to the prior one.

5.9.3 Plans for version 1.1

Release 1.1 is already on the works and various changes are planned for it, the following lists contains some of the changes that are meant to be implemented. More likely than not, a decent amount of bug fixes will also be included in v1.1. Any suggestions are more than welcome, and may be sent to odeko@nickdejongc.com.

Installer upgrades:

- Improve the visuals of the installer.
- Offer the option to choose which shortcuts the user wants in the installer.
- Allow to install updates of the application instead of having to uninstall and reinstall again. This will allow the user to update the software directly from their installation instead of having to download the new version.

Cosmetic or QOL additions:

- Include an in-application bug report option.
- Add keyboard shortcuts to the application.
- Addition of progress bars to calculations and optimizations.
- Capability to delete multiple saved items simultaneously.
- More customization options!

New features:

- Improve optimization procedure (speed and accuracy).
- Allow to define star-shaped grains.
- Allow to define custom grains.
- Offer to define custom nozzles.
- Add ODK FAC options.
- Add IAC FAC with grain geometry.
- Auto-correction of corrupt save files.
- Better load checks to avoid inconsistencies between loaded files.
- Improved support for external input of thermal and transport properties as well as kinetics.

6

Examples & results

Contents

6.1 Comparison of ODE and frozen flow

The parameters of the arbitrary employed parabolic nozzle are shown in Table 6.3 and Table 6.4. The employed propellants will be referred to as LHLO (liquid hydrogen and liquid oxygen with O/F=5), NCNG (nitroglycerine to nitrocellulose ratio of 10), APHTPB (Ammonium perchlorate and HTPB with O/F=4) and RC (Potasium nitrate and sorbitol with O/F = 65/35). Chamber pressure is 3MPa for all configurations.

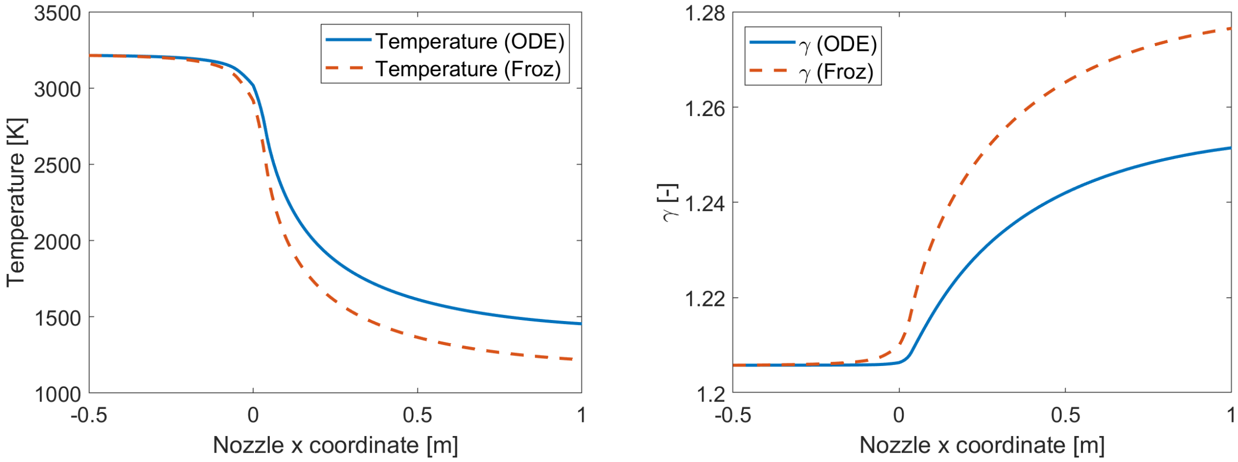
Entrance diameter	0.5m
Throat diameter	0.25m
Exit diameter	1m
Convergent length	0.5m
Divergent length	1m

Table 6.1: Main nozzle parameters

Initial parabola angle	50º
Final parabola angle	5º
Entrance fillet radius	0.1m
Convergent throat fillet radius	0.1875m
Divergent throat fillet radius	0.04775m

Table 6.2: Additional nozzle parameters

6.1.1 Propellant comparison



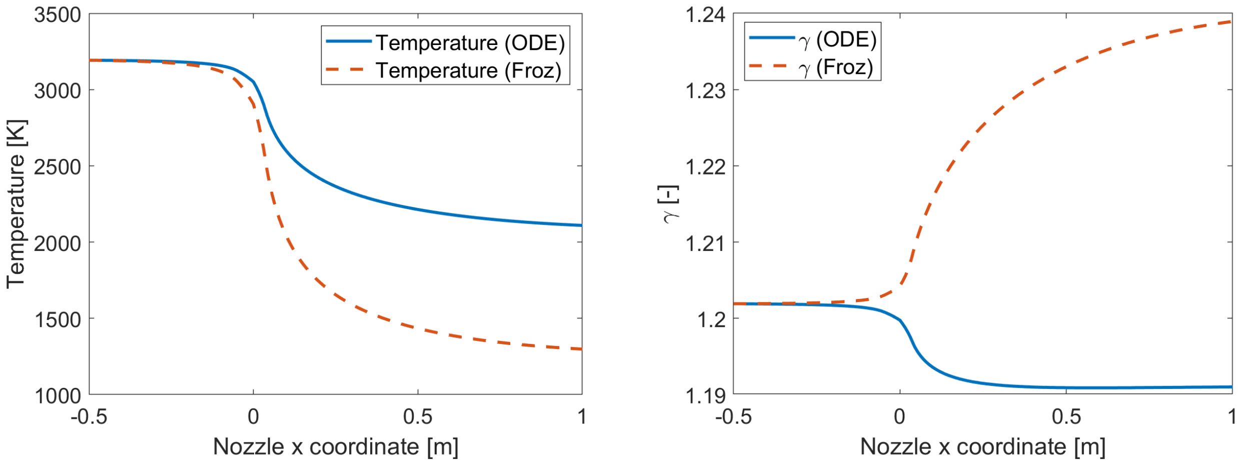
(a) Temperature evolution

(b) Gamma evolution

Figure 6.1: LHLO

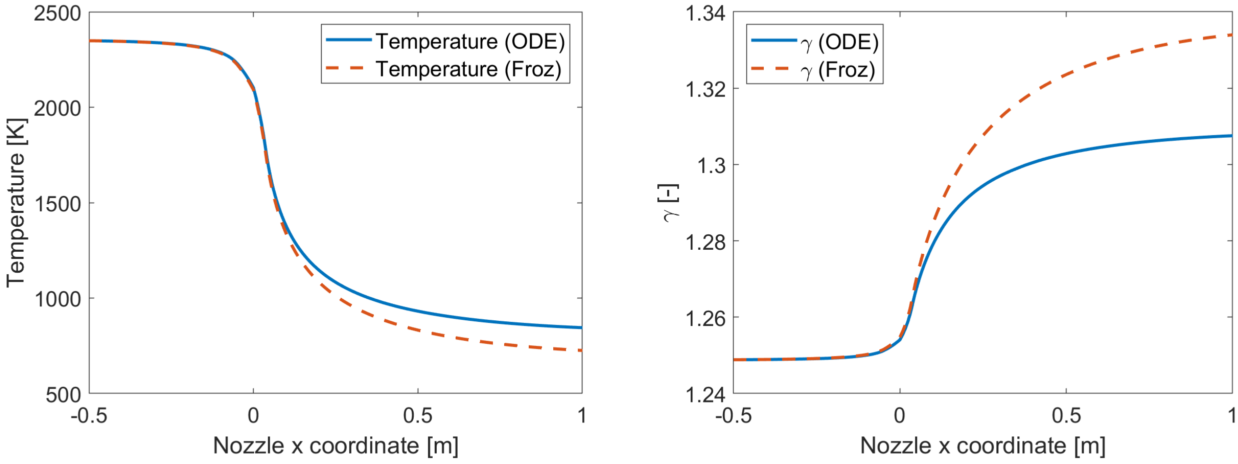
A comparison between flow through the nozzle computed with chemical equilibrium and 'freezing' said flow at the throat is given for a series of propellants, as shown in Figure 6.1, Figure 6.2 and Figure 6.3.

It is apparent that the flow temperature throughout the nozzle is higher when chemical equilibrium is conserved. On the other hand, the frozen value for gamma tends to be lower.



(a) Temperature evolution **(b)** Gamma evolution

Figure 6.2: NCNG

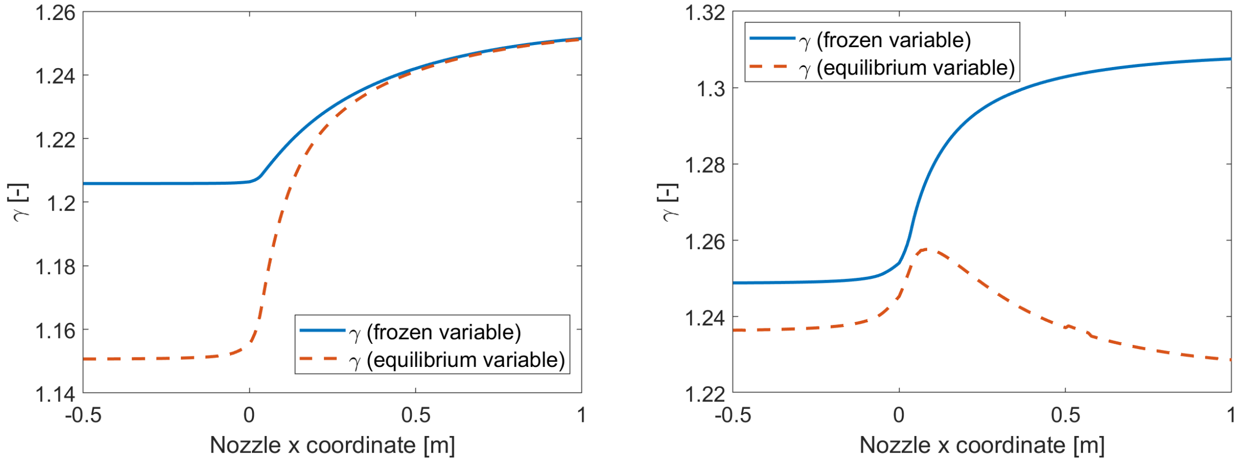


(a) Temperature evolution **(b)** Gamma evolution

Figure 6.3: APHTPB

6.1.2 Frozen & Equilibrium Variables

In this section the entire flow is computed using chemical equilibrium and the frozen and equilibrium values of gamma are compared. It is notable that, for the LHLO propellant (Figure 6.4a) the frozen and equilibrium gammas tend to the same value as the exit area increases. On the other hand, for APHTPB (Figure 6.4b) the frozen and equilibrium values differ more and more with increasing exit area.



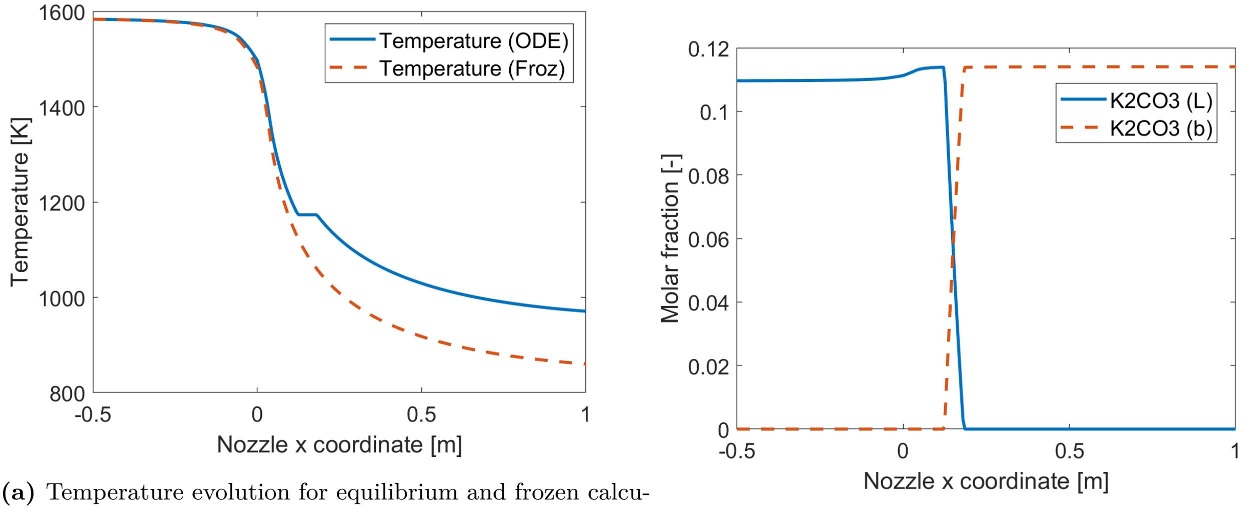
(a) Gama evolution for LHLO

(b) Gamma evolution for APHTPB

Figure 6.4: Comparison of equilibrium and frozen variables for equilibrium calculations

6.1.3 Transition Between Condensed Phases

In this section the propellant RC is used to showcase the transition between condensed phases in the nozzle. In Figure 6.5a it can be seen that the equilibrium flow has an isothermal region in the nozzle, corresponding to the phase transition. On the other hand, the flow frozen at the nozzle throat does not experience said transition and has a more typical temperature curve. On Figure 6.5b the composition of both condensed species is shown for the flow computed with chemical equilibrium.

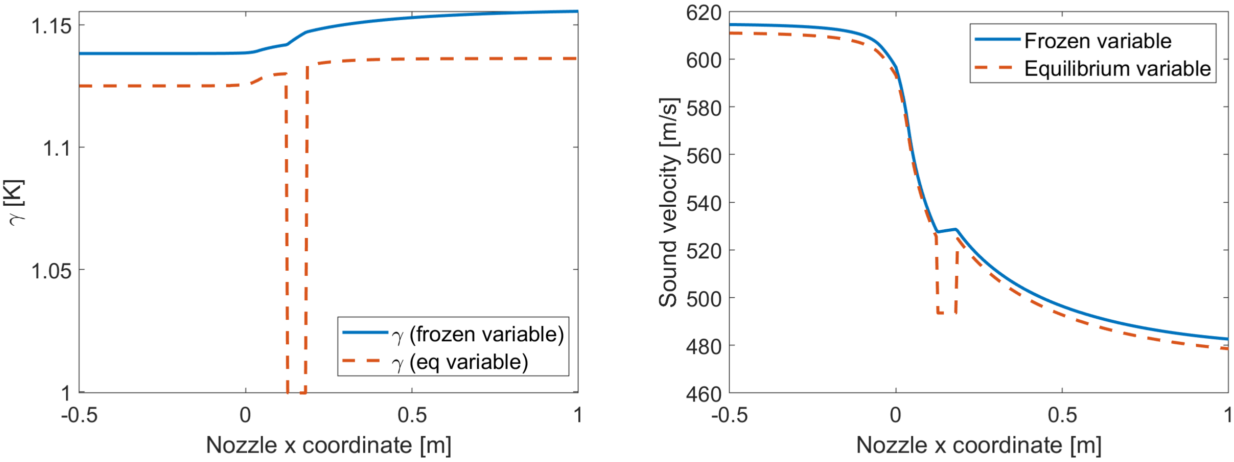


(a) Temperature evolution for equilibrium and frozen calculation

(b) Composition evolution for equilibrium calculation

Figure 6.5: Transition between condensed phases

Furthermore, for the flow computed fully with chemical equilibrium a comparison between the frozen and equilibrium variables is given in Figure 6.6. This illustrates the discontinuity present in the value of the equilibrium variables, as mentioned in prior sections. Moreover, it can be seen that the frozen value for gamma and specially that of the speed of sound have a discontinuity on their first derivative at the entrance and exit of the isothermal region. It is also notable that the equilibrium gamma is slightly less than 1 (around 0.9996) throughout the isothermal region. Moreover, the frozen speed of sound increases in value during the isothermal region, this is easily explained given that the temperature remains constant whilst the frozen gamma is increasing.



(a) Frozen & equilibrium gamma values for equilibrium calculations

(b) Frozen & equilibrium sound velocity value for equilibrium calculations

Figure 6.6: Comparison of frozen and equilibrium variables during a transition between condensed phases

6.2 Comparison between ODK, Equilibrium and Frozen Flow

To obtain the following results an arbitrary set of nozzles have been used. The parameters of the employed conical nozzle are shown in Table 6.3 and the employed parabolic nozzle shares the same parameters with the addition of those shown in Table 6.4. Unless stated otherwise, the parabolic nozzle is employed for calculations.

Entrance diameter	0.5m
Throat diameter	0.25m
Exit diameter	1m
Convergent length	0.5m
Divergent length	1m

Table 6.3: Dimensions shared by parabolic and conical nozzles employed in calculations

Initial parabola angle	50º
Final parabola angle	5º
Entrance fillet radius	0.1m
Convergent throat fillet radius	0.1875m
Divergent throat fillet radius	0.04775m

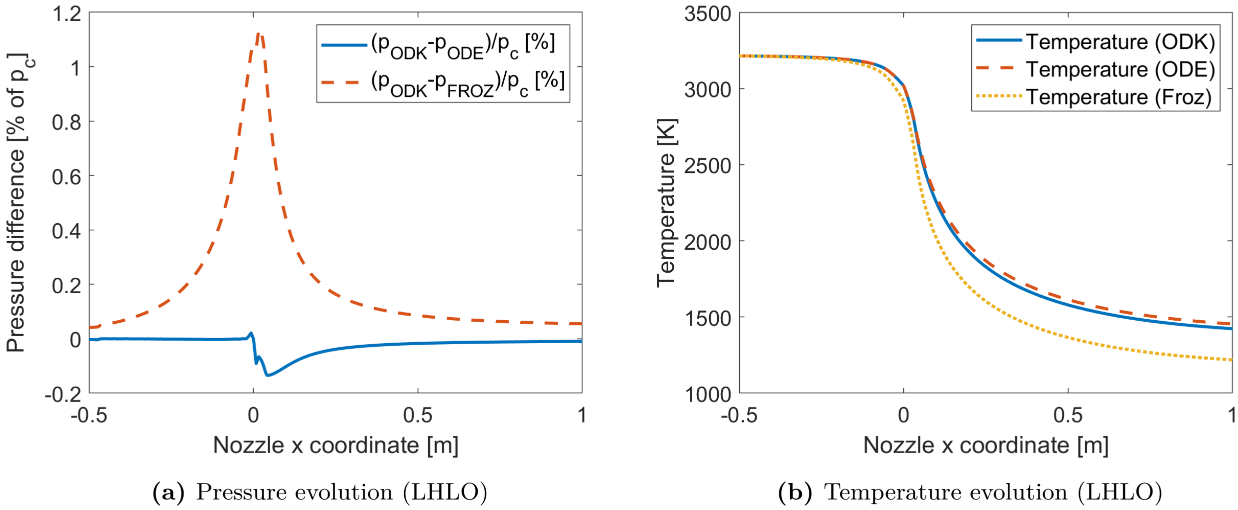
Table 6.4: Dimensions of parabolic nozzle employed in calculations

The employed propellants will be referred to as LHLO (liquid hydrogen and liquid oxygen with O/F=5), NCNG (nitroglycerine to nitrocellulose ratio of 10), APHTPB (Ammonium perchlorate and HTPB with O/F=4). Chamber pressure is 3MPa for all configurations.

6.2.1 Propellant comparison

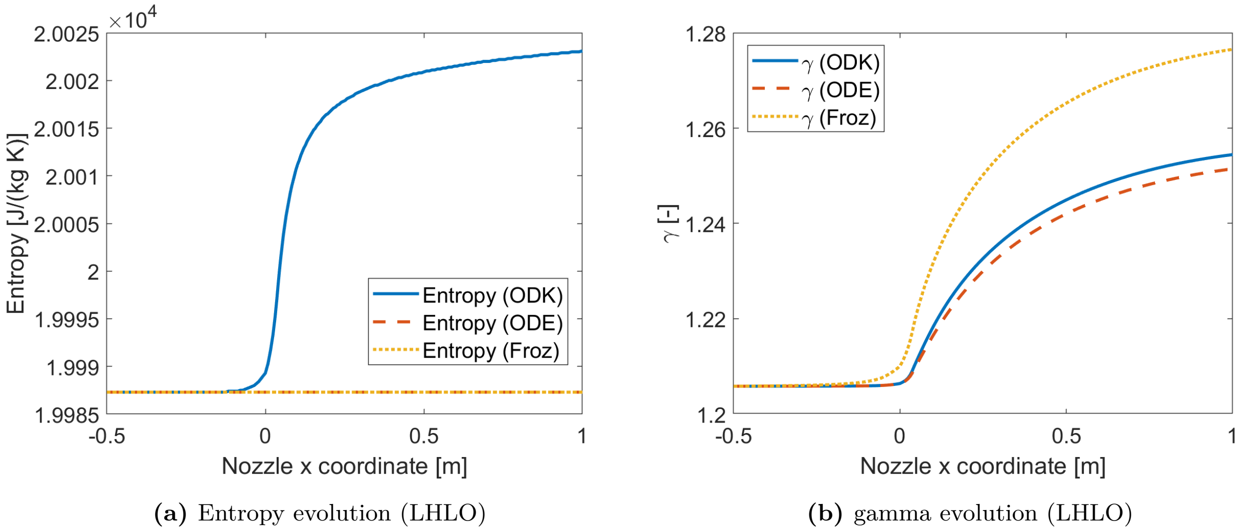
The first comparison is performed using the LHLO propellant, Figure 6.7a shows the pressure difference between the results obtained through ODK and those for ODE and frozen flow. It can be seen that the pressure difference is not very significant, specially between ODE and ODK for the LHLO propellant. On the other hand, differences in temperature, gamma and mean molecular mass as shown in Figure 6.7b, Figure 6.8b and Figure 6.9a respectively are more clearly visible. it should be pointed out that, as can be seen in the figures, the results given by ODK are very close to those given by ODE, this is due to the fast reaction rate of the LHLO mixture. Finally, it should be highlighted that, if finite, non-zero reaction rates are considered, the flow is not isotropic, as can be seen in Figure 6.8a.

The set of species and reactions employed in the calculations is explained in detail in subsection 6.2.6.



(a) Pressure evolution (LHLO) **(b)** Temperature evolution (LHLO)

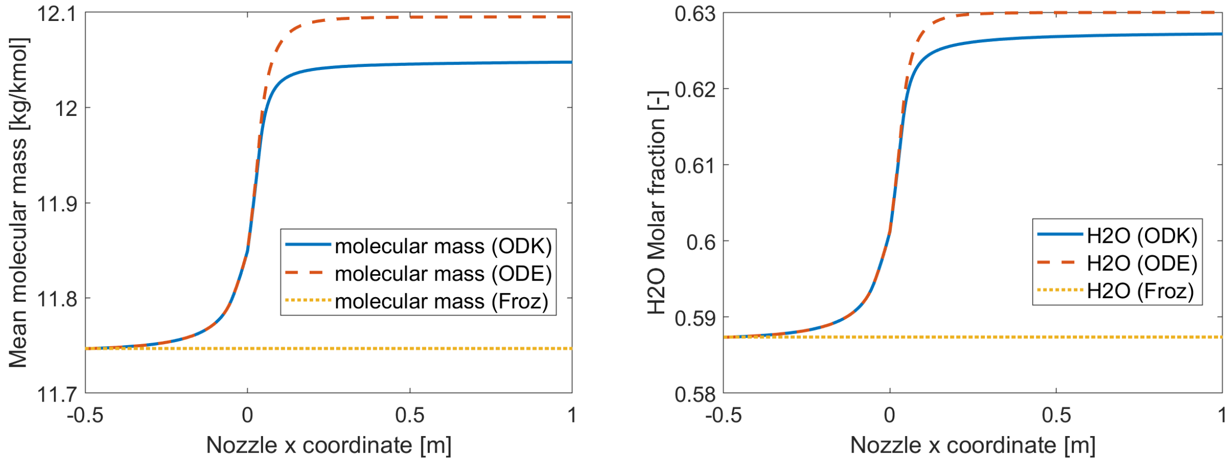
Figure 6.7: LHLO 1 - ODK



(a) Entropy evolution (LHLO) **(b)** gamma evolution (LHLO)

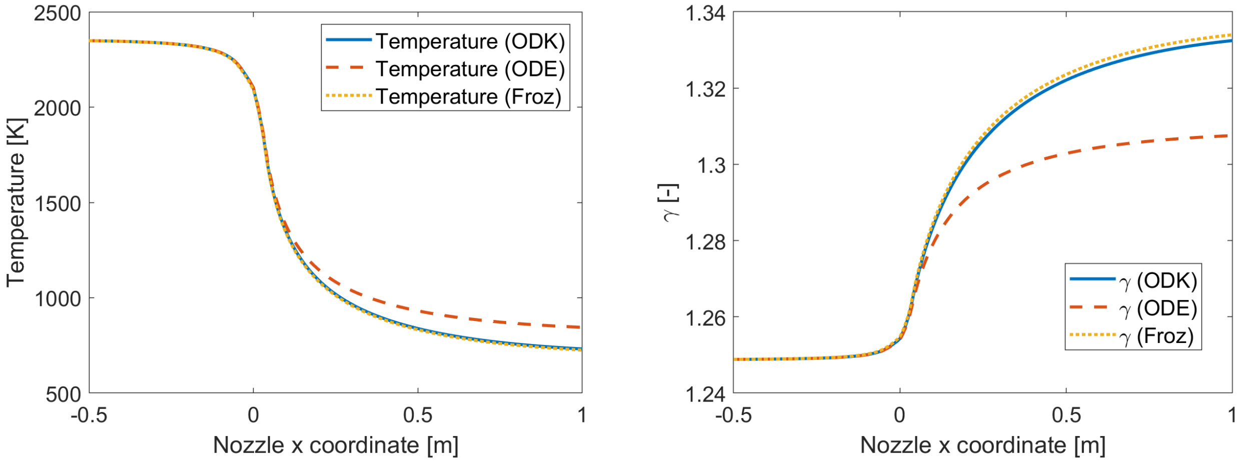
Figure 6.8: LHLO 2 - ODK

The LHLO propellant is an example of a fast reacting mixture, where the results obtained through ODK are very similar to those of ODE. Given this proximity to the equilibrium case, the entropy generation is insignificant, as can be seen in Figure 6.8a. The other extreme, where reaction rates are very slow and the results are closer to those given by frozen flow, can be shown with the APHTPB propellant. On Figure 6.10a or Figure 6.10b, this can be clearly seen. Moreover, given this proximity to frozen flow, the entropy generation is close to 0, which can be seen in Figure 6.11a.



(a) Mean molar mass evolution (LHLO) **(b)** H2O molar fraction evolution (LHLO)

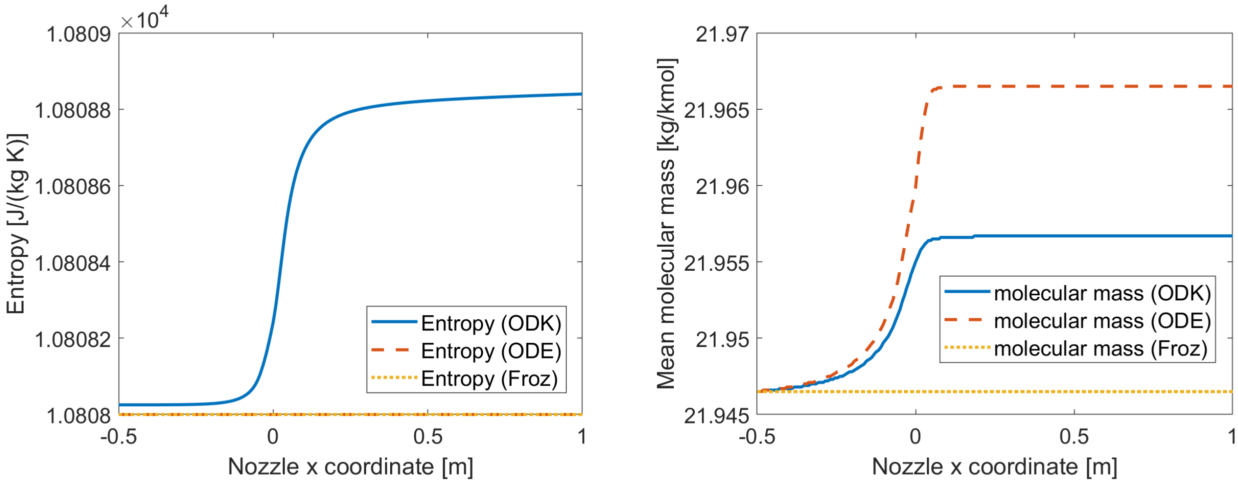
Figure 6.9: LHLO 3 - ODK



(a) Temperature evolution (HTPB) **(b)** gamma evolution (HTPB)

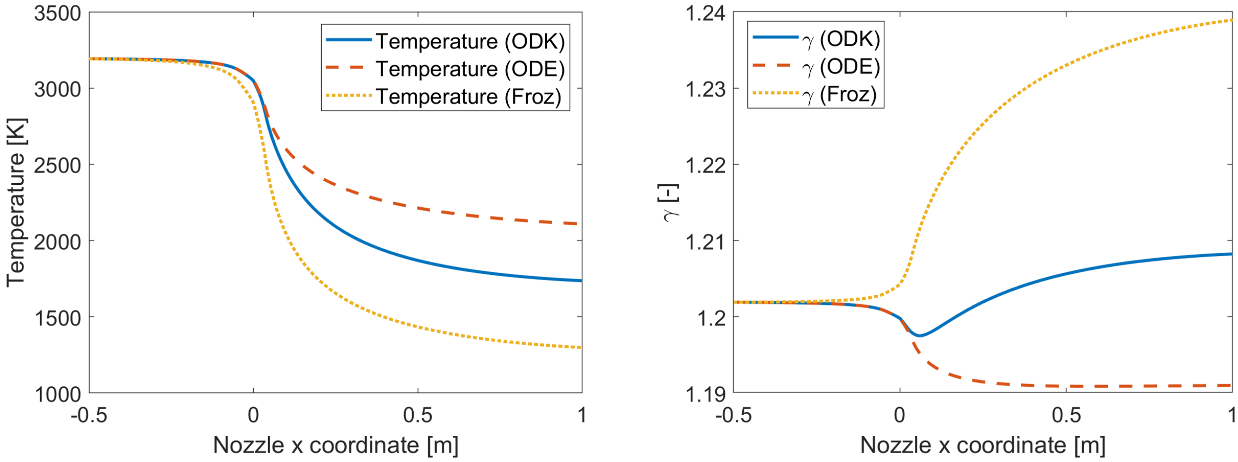
Figure 6.10: HTPB 1 - ODK

Finally, an intermediate case between the last two is shown, where the reaction rates have a moderate value, intermediate between the extremes of frozen flow and ODE. This is the case of the NCNG propellant, as shown in Figure 6.12a or Figure 6.12b. Moreover, the entropy generation (as shown in Figure 6.13a) in this case is much larger than for the APHTPB propellant or LHLO. Since in this case the mixture behaviour is not close to neither chemical equilibrium nor frozen flow, entropy generation becomes much more significant.



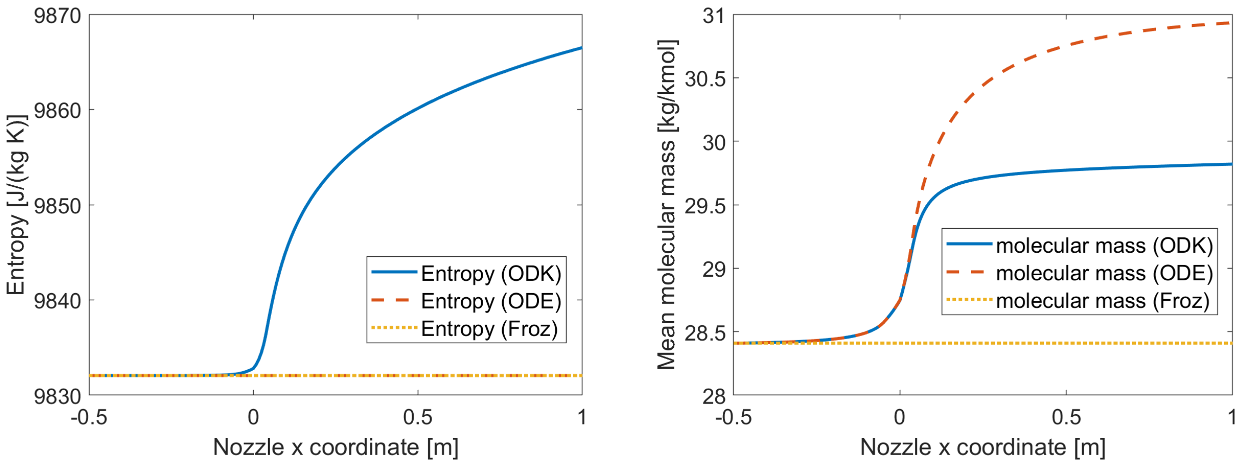
(a) Entropy evolution (HTPB) **(b)** Mean molar mass evolution (HTPB)

Figure 6.11: HTPB 2 - ODK



(a) Temperature evolution (NCNG) **(b)** Gamma evolution (NCNG)

Figure 6.12: NCNG 1 - ODK



(a) Entropy evolution (NCNG) **(b)** Mean molar mass evolution (NCNG)

Figure 6.13: NCNG 2 - ODK

6.2.2 Effects of Nozzle Geometry on ODK Calculations

If the intermediate points of the geometry are modified (for example switching from a parabolic to a conical nozzle), but the throat and exit area are left equal, calculations performed with chemical equilibrium or frozen flow will yield the same results at the exit section for both geometries. This is a result of these calculations being only dependant on the mass flow of the system (given by the throat) and the area of the section of interest (in this case the exit). On the other hand, calculations with ODK are dependant on the "history" of the flow and, consequently, the geometry up to that point. Therefore, for the previous scenario ODK will give different values for the exit section depending on the shape of the intermediate geometry. On Table 6.5 the exit pressure given by ODK is shown for the parabolic and conical nozzle as well as some variations of them. It can be seen that the different geometries have an effect on the exit pressure, although not large. As expected, the greatest difference is found between the parabolic and conical nozzle, as the present the largest differences in geometry. It is important to point out that this difference in exit pressure is not very significant and will be overshadowed by the actual difference due to divergence losses.

Nozzle	Parabolic	Conical	Conical + throat fillet R=0.05	Conical + throat fillet R=0.1
Exit pressure	19498	19563	19549	19549

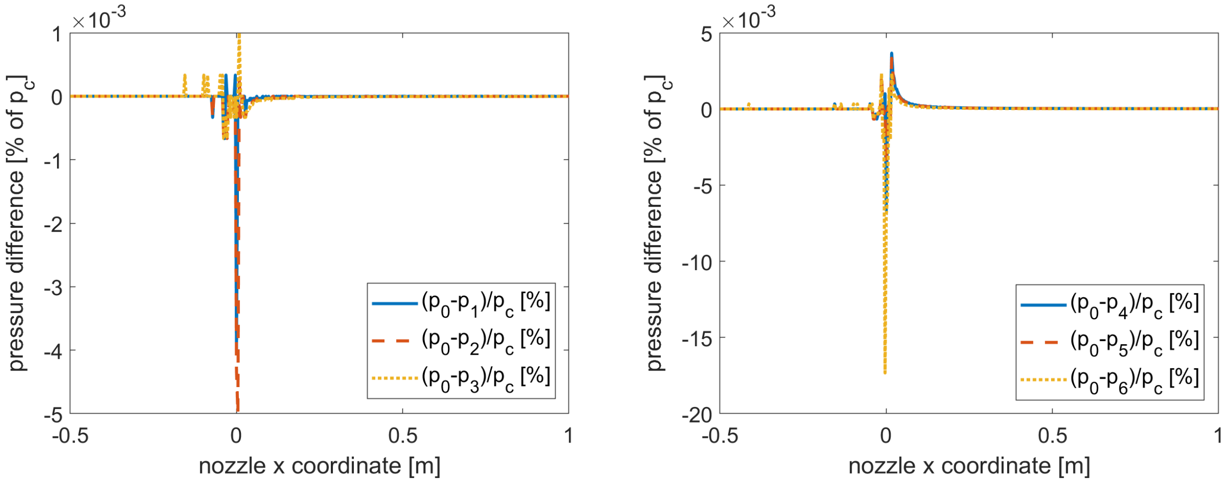
Table 6.5: Comparisons of exit pressure for different nozzle geometries

6.2.3 Effect of Selecting Different Switch Points for the Independent Variable

On Table 6.6 the pressure evolution throughout the nozzle is shown. The pairs of differently colored points represent the switch between area and pressure as independent variables and vice-versa. On Figure 6.14a and Figure 6.14b the difference in pressure between the base case (corresponding to the red points) and the other cases is shown. The base case is the one that has been used in all previous calculations. It can be seen that for cases 1 and 2, the pressure difference is very small, being always less than a hundredth of a percent of the chamber pressure. Even for case 7, where the difference is largest, it remains incredibly small. Consequently, it can be stated with confidence that the switch points have no effect on the results. It should be noted that his statement applies given that convergence has been successful for all cases, for some problems convergence could not occur for certain point pairs.

	Case 0	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Area to pressure switch	$\epsilon = 1.08$	$\epsilon = 1.06$	$\epsilon = 1.04$	$\epsilon = 1.03$	$\epsilon = 1.02$	$\epsilon = 1.01$	$\epsilon = 1.005$
Pressure to area switch	$\epsilon = 1.12$	$\epsilon = 1.08$	$\epsilon = 1.06$	$\epsilon = 1.04$	$\epsilon = 1.02$	$\epsilon = 1.01$	$\epsilon = 1.001$

Table 6.6: Area ratio for which the independent variable is switched in calculations



(a) Comparison of base case with cases 1,2,3

(b) Comparison of base case with cases 4,5,6

Figure 6.14: Effect of different switch points in ODK

6.2.4 Effect of Amount of Nozzle Points Considered

Similarly to the previous section, as long as the solution converges (or even as long as it comes very close to convergence), the number of points has no effect on the solution and the encountered differences are extremely small. Nevertheless, several things should be pointed out regarding the number of nozzle points considered in the calculations.

Firstly, if the number of points is large the computation time will obviously increase. Bringing the independent variable switch points closer to the throat will decrease said computation time, given the increased number of points there should still be sufficient points in the region of pressure as independent variable for an adequate convergence. Another solution would be to reduce the tolerances of the various convergence criteria (specially throat position or throat area), as that will greatly reduce the number of necessary iterations.

Furthermore, more point will not only increase the computation time but it will also increase the complexity of achieving a solution. The reason behind this is the increased number of points near the throat. Given the fillets present near the throat, points very close to it will have very similar area values (as well as very similar values of its derivative), this will hinder the ability of the presented convergence criteria to find the adequate solution.

Finally, an insufficient number of points may prevent convergence. Increasing the distance between the switch points may help in these cases (as there might have been to few points in the region of pressure as independent variable).

6.2.5 Effect of Modifying Tolerances of the Convergence Criteria

Generally, for the differential equation solver tolerances it is beneficial to have a strict relative tolerance, since its error is accumulated through the iteration and can have a great impact on critical variables, such as the number of decimal places of the mass flow that are maintained constant. A value around 10^{10} is recommended. The absolute tolerance is not very important in the given scenario, since it is used for variables approaching a null value. Nevertheless, a tolerance around 10^{8} is adequate. Increasing the solver tolerance does not have a significant impact on computation time.

The remaining ODK tolerances (throat points, throat area, throat position and mass flow) are very interconnected. The mass flow tolerance severely limits the capacity for the nozzle points and throat convergence, therefore it should always be at least as precise as the others. The general recommendation is a value of 10^{3} for all four. In spite of this, decreasing these tolerances will significantly speed up the calculations (so will reducing the number of points describing the nozzle). Tolerance values under 10^{4} have not been found to work successfully. Furthermore, if the number of points used to described the nozzle is high, stricter tolerances may not work, as the increased number of points hinders convergence.

6.2.6 Set of reactions under consideration

For all ODK calculations a set of basic reactions have been considered. The user may add or remove reactions from this set, either using the reactions database already present in ODEKOpt or including new reactions manually or from external files. The set of species considered for ODK calculations are: $C, CO, CO_2, Cl, Cl_2, H, H_2, H_2O, HCl, N, N_2, NO, O, O_2, OH$.

And the considered reactions are listed in the following table:

$$CO_2 + M \rightleftharpoons CO + O + M$$
$$Cl_2 + M \rightleftharpoons 2Cl + M$$
$$H_2 + M \rightleftharpoons 2H + M$$
$$NO + M \rightleftharpoons N + O + M$$
$$O_2 + M \rightleftharpoons 2O + M$$
$$CO_2 + O \rightleftharpoons CO + O_2$$
$$H_2O + H \rightleftharpoons OH + H_2$$
$$CO + NO \rightleftharpoons CO_2 + N$$
$$HCl + Cl \rightleftharpoons H + Cl_2$$
$$H_2 + Cl \rightleftharpoons HCl + H$$
$$H_2 + O_2 \rightleftharpoons 2OH$$
$$NO + H \rightleftharpoons N + OH$$
$$O_2 + H \rightleftharpoons OH + O$$
$$H_2O + M \rightleftharpoons OH + H + M$$
$$HCl + M \rightleftharpoons H + Cl + M$$
$$N_2 + M \rightleftharpoons 2N + M$$
$$OH + M \rightleftharpoons O + H + M$$
$$CO_2 + H \rightleftharpoons CO + OH$$
$$H_2O + Cl \rightleftharpoons OH + HCl$$
$$H_2O + O \rightleftharpoons 2OH$$
$$CO + O \rightleftharpoons C + O_2$$
$$HCl + O \rightleftharpoons OH + Cl$$
$$H_2 + O \rightleftharpoons OH + H$$
$$N_2 + O \rightleftharpoons NO + N$$
$$NO + O \rightleftharpoons N + O_2$$

Table 6.7: Reactions considered in ODK calculations

An adequate selection of reactions is crucial in order to achieve correct results. The set of reactions was recommended for H,C,O,N,Cl systems in [4]. Only the most relevant reactions and species were considered, in order to reduce the number and achieve faster computations for ODK. The reaction parameters were obtained from references [10], [12], [8], [11] and [1].

If the user wants to perform a more precise computation for a propellant they should study in depth the relevance of different species and reactions in order to use all the relevant ones. Nonetheless, the reactions presented in Table 6.7 offer a good first approximation for most rocket engine propellants.

6.3 Examples

6.3.1 Raptor 2 engine

As a first example, SpaceX's Raptor engine was reproduced in ODEKO. Its relevant parameters are shwon in Table 6.8. The nozzle was reproduced with a Bezier curve; although, as will be later seen, the type of curve does not affect the rocket parameters in the ODE model. An image of the nozzle used in calculations can be seen in Figure 6.15.

Entrance diameter [m]	Throat diameter [m]	Exit diameter [m]	Convergent length [m]	Divergent length [m]	Oxidant	Reductor
0.32	0.2216	1.3	0.52	1.94	$78\% O_2(L)$	$22\% CH_4(L)$

Table 6.8: Parameters for the raptor engine

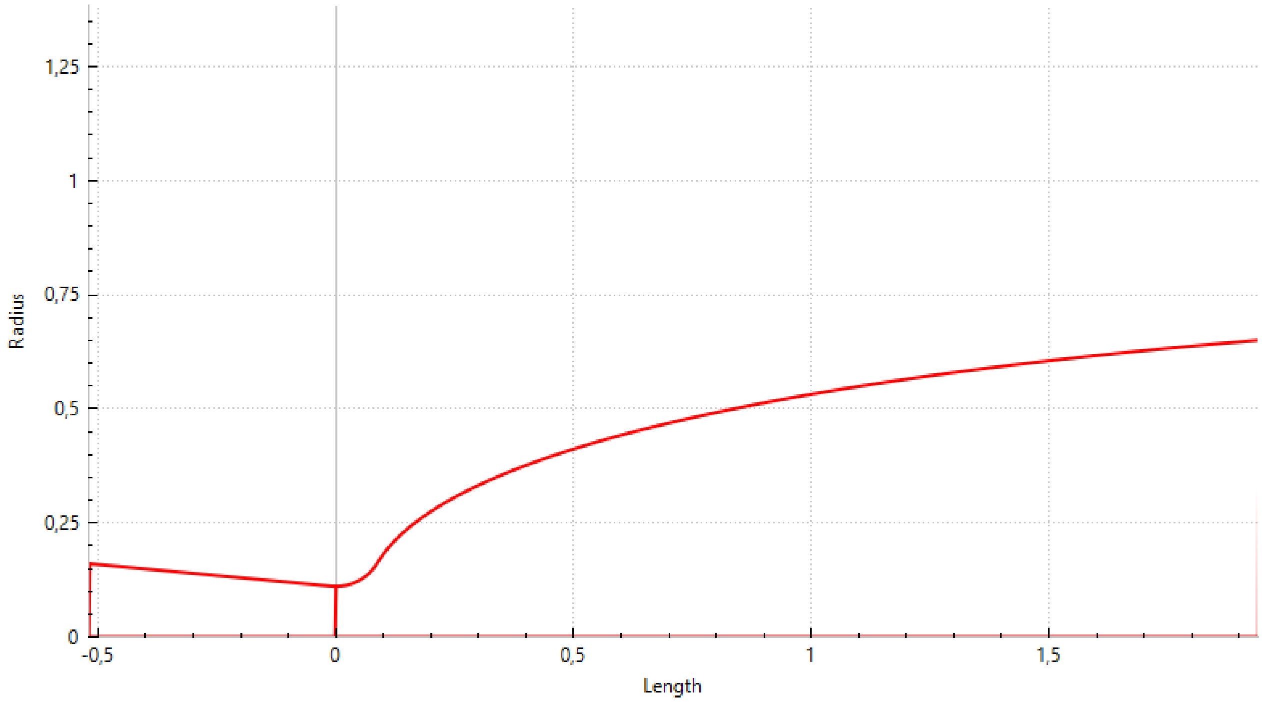


Figure 6.15: Nozzle of Raptor engine, using a Bezier curve

Considering a chamber pressure of 330bar and an external pressure of 1atm, a comparison between ODEKO's equilibrium computation and the reported values for the raptor engine is given in Table 6.9.

	Thrust (1atm)	$I_{sp}(1atm)$	$I_{sp}(vac)$
Real	2.30 MN	3.2 km/s	3.56 km/s
ODE	2.34 MN	3.42 km/s	3.61 km/s

Table 6.9: Raptor engine attitude parameters

The theoretical and experimental values are very close to each other, showing the validity of the applied model. The small differences can easily be attributed to two dimensional effects of the nozzle not considered by the model (note that ODE values are higher than experimental, therefore an efficiency could be introduced to account for these effects). Moreover, ODEKO does not account for the energy needed to power the turbo-pump of a liquid rocket engine. In Figure 6.16 a plot of the mach number is included.

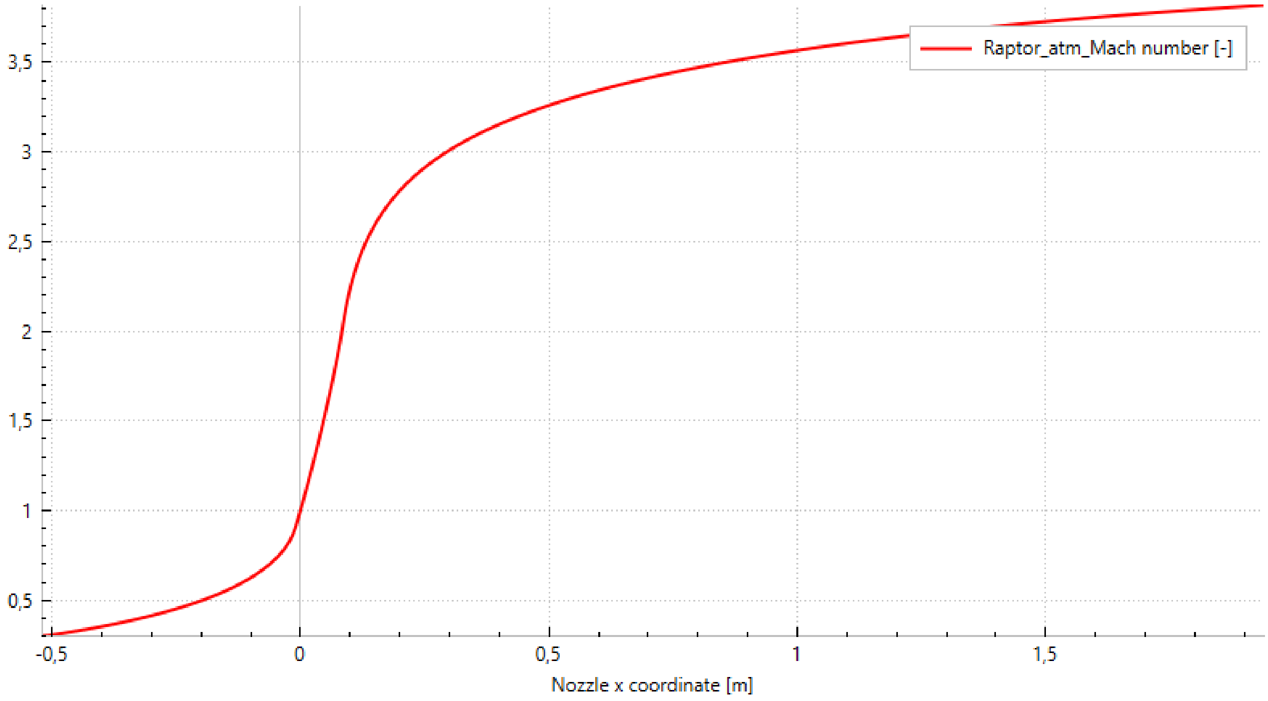


Figure 6.16: Mach along the nozzle of the Raptor engine

6.3.2 Vulcain 2 engine

As a second example, Vulcain 2 engine was studied. Its nozzle was also reproduced as a bell curve an its parameters are shown in Table 6.10, the value for the nozzle entrance was not known, so an assumption was made. A comparison between analytical and experimental results can be seen in Table 6.11, where a chamber pressure of 118.8bar has been used.

Entrance diamter [m]	Throat diameter [m]	Exit diameter [m]	Length [m]	Oxidant	Reductor
0.32	0.281	2.15	3.44	$85.8\% O_2(L)$	$14.2\% H_2(L)$

Table 6.10: Vulcain 2 engine parameters

	Thrust (Vacuum)	I_{sp} (vacuum)	$\dot{m}$
Real	1.35 MN	4.25 km/s	326 kg/s
ODE	1.44 MN	4.50 km/s	320 kg/s

Table 6.11: Vulcain 2 engine attitude parameters

The differences in between calculated values and values found in the engine's documentation are small for the thrust and specific impulse values, similar to the ones for the Raptor engine. In spite of this, the predicted mass flow is smaller than the real value (contrary to the simulation for SpaceX Raptor engine). This is easily explainable. Raptor engines are full-flow; therefore, all the mass flow passes through the nozzle (although it has passed through a pre-burner, therefore having less available energy). On the other hand, Vulcain 2 is a gas burner, therefore, some of the flow does not pass through the nozzle (which makes up for the additional mass flow), in term, this also affects the value for the specific impulse.

The raptor engine will be used as a benchmark in order to compare different variables of the engine using ODEKO. Following sections will either change the computation method or one of the aspects of the nozzle.

6.3.3 Grain combustion example

A double base propellant (91% NG, 9% NC) was used in the example for grain combustion. In order for the grain combustion to be computed, Vielle's parameters need to be defined, the following approximate values were used: $a = 0.0078 m/s, n = 0.67$. Note that the value of 'a' is valid for pressure in MPa. The grain geometry is a set of 5 cylindrical charges that burn at the internal surface and both bases. Numerical values for the charges are given in Table 6.13 and for the nozzle in Table 6.12. The results have been included in Table 6.14 and Figure 6.17.

Entrance diameter	Throat diameter	Exit diameter
35mm	20mm	40mm

Table 6.12: Parameters for the nozzle used in grain combustion example

Exterior diameter	Interior diameter	Charge length
35mm	14mm	55mm

Table 6.13: Parameters for the grain geometries used in grain combustion example

Thrust	I_{sp}	c^*	$\dot{m}$	t_b
854 N	2165 m/s	1522 m/s	0.394 kg/s	0.75 s

Table 6.14: Average values for the grain combustion example

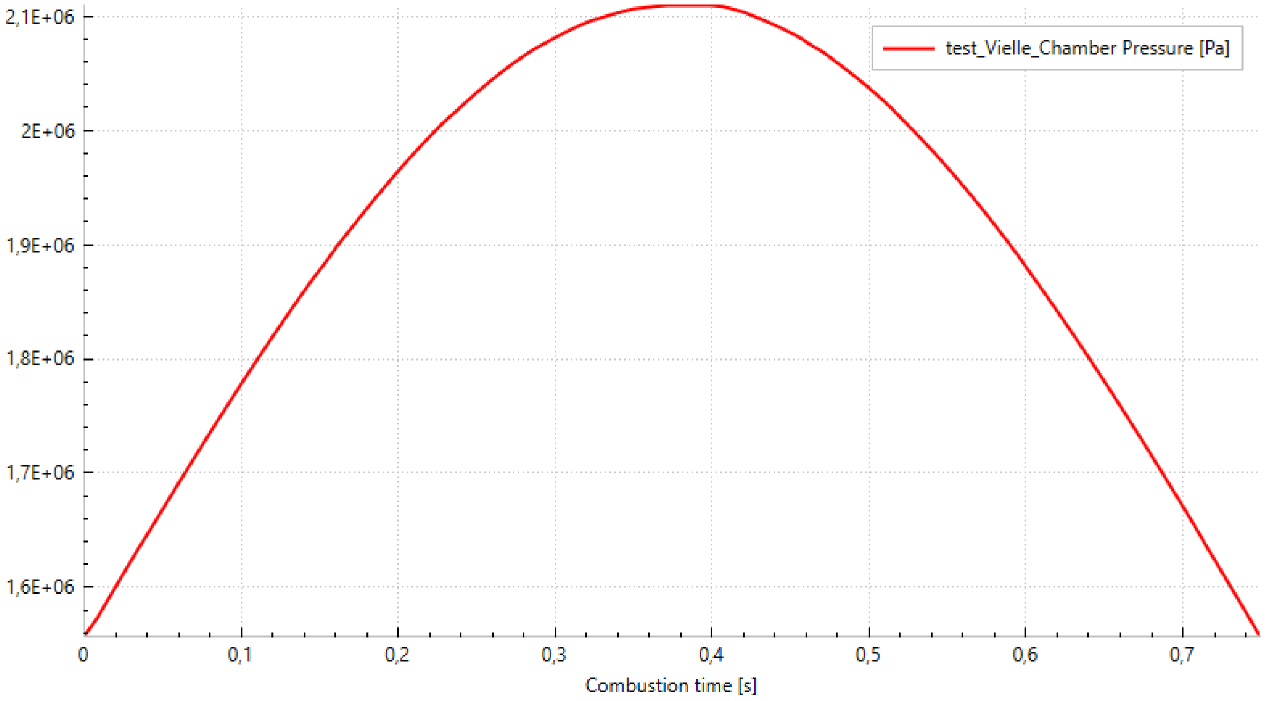


Figure 6.17: Time evolution for chamber pressure in grain combustion example

The main conclusion to be extracted from the obtained results is that the variation of the chamber pressure (and of the rocket parameters) follows a similar distribution to that of the area of combustion of the charges (which was to be expected). The main takeaway is that ODEKO allows the user to obtain the thrust evolution with respect to time for solid propellant rocket engines. This allows the user to clearly study the effect of different combustion geometries.

6.3.4 Nozzle optimization example

A double base propellant (91% NG, 9% NC) was used in order to show optimization examples that can be performed with ODEKO. The geometry of the cylindrical charges is described in Table 6.13. In Table 6.15 the resulting nozzles for different optimization targets have been presented. Note that the optimization was performed for a set range of acceptable chamber pressures: between 10bar and 100bar.

Target	Throat diameter [mm]	Exit diameter [mm]	Target value
P_c	22.1762	-	10 [bar]
P_c	17.1435	-	50 [bar]
P_c	15.3414	-	100 [bar]
t_b	18.1643	-	0.5 [s]
t_b	19.9712	-	0.75 [s]
t_b	21.3749	-	1 [s]
Thrust	14.8480	55.5543	3146 [N]
I_{sp}	14.8480	55.5543	2632 [m/s]
I_t	14.8480	55.5543	784.64 [Ns]

Table 6.15: Results of various optimizations

It is notable that all optimizations that look for the maximum target value (thrust, specific impulse and total impulse) have given the same result: a chamber pressure of 100bar, the maximum in the interval. This was to be expected due to the simple interval for Vielle's definition. Propellants with a more segmented definition and very different intervals may show a different behaviour. Additionally, as it was to be expected, increasing the chamber pressure or decreasing the combustion time result in throats of smaller diameter.

6.3.5 Comparison with CEA

Nasa's software CEA offers a set of examples. One of them will be used to validate ODEKO's calculations. The example consists of a mixture of liquid oxygen and hydrogen, with a oxidizer to fuel ratio of 5.55157 and a chamber pressure of 53.317 bar. Table 6.16 compares a variety of values given by both applications. It is clear that the values are in close accordance, further validating the software ODEKO.

	Temperature (combustion)	Temperature (throat)	Pressure (throat)	Entropy (constant)	gamma throat
CEA	3389.27 K	3190.21 K	30.652 bar	18.657 kJ/kg	1.1472
ODEKOpt	3383.84 K	3185.67 K	30.655 bar	18.658 kJ/kg	1.1468

Table 6.16: Comparison between CEA and ODEKOpt

6.3.6 Influence of chamber pressure on c^*

The same propellants studied in the previous subsection are present in Figure 6.18, where the value for c^* has been plotted relative to the chamber pressure. It can be seen that all propellants have a logarithmic-like curve. Therefore, once a certain chamber pressure has been reached, the effect of pressure on the value of c^* can be considered negligible.

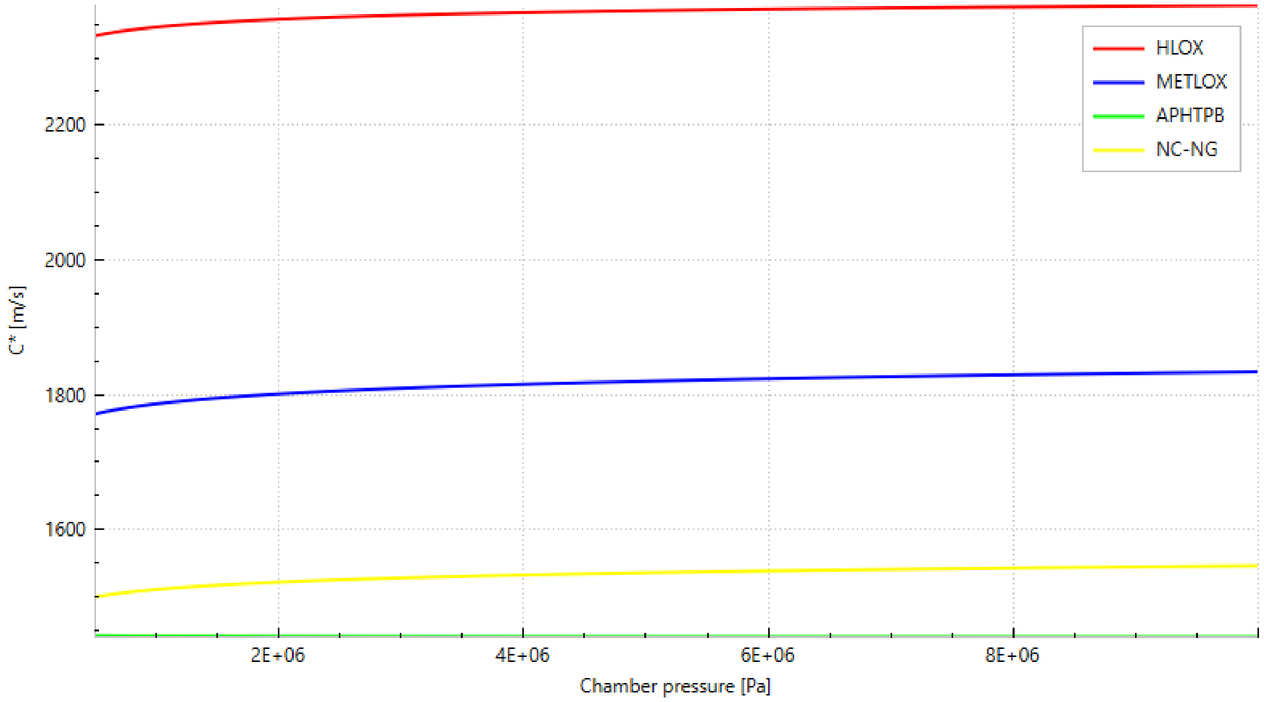


Figure 6.18: c^* dependence on chamber pressure for different propellants

6.4 IAC & FAC comparison

The IAC and FAC models were compared for the Raptor engine (Table 6.8). As seen in Figure 6.19 there is a significant drop in pressure at the start of the nozzle for the FAC model compared to IAC (due to the losses in the combustion chamber). In Table 6.17 it can be seen that the thrust is much lower for the FAC model (even below the real value), which is to be expected due to the lesser initial pressure (greater losses in the combustion chamber).

	IAC	FAC	Real
Thurst [MN]	2.34	2.23	2.3
I$_{sp}[km/s]$	3.42	3.41	3.2
$\dot{m}$ [kg/s]	685	654	-
c$^*[km/s]$	1.86	1.86	-

Table 6.17: Variable comparison for IAC and FAC models

SpaceX's raptor engine injectors are coaxial swirl injectors, known for their very low pressure losses. Since the hypothesis for the FAC model is that injectors inject the fluid at null velocity, it is clearly not valid for these type of injectors, characterized for elevated injection speeds. If the speed of injection was known, the FAC model could be corrected and better results would be achieved (since a higher initial speed would result in smaller losses, which is coherent with the described injector). Therefore, it is very clear that in this scenario, IAC offers a much better result than FAC, which could have been known beforehand due to the characteristics of the motor. It serves to highlight the importance for an appropriate choice of model for the given simulation, since it can have a very significant impact on the final results.

It is also important to note that the value for c^* is the same for both models, this is to be expected since the combustion chamber is modelled equally and c^* is only dependant on combustion variables. Moreover, the value for I_{sp} is also quite similar, since it is a product of c^* and C_E where the second is approximately only dependant on the nozzle geometry and the value for gamma (which is very close in both cases). On the other hand, the mass flow does suffer a significant change, which was to be expected since it can be written as a function of c^*, P_c and A_t, where the only value that differs between models is that of the throat area.

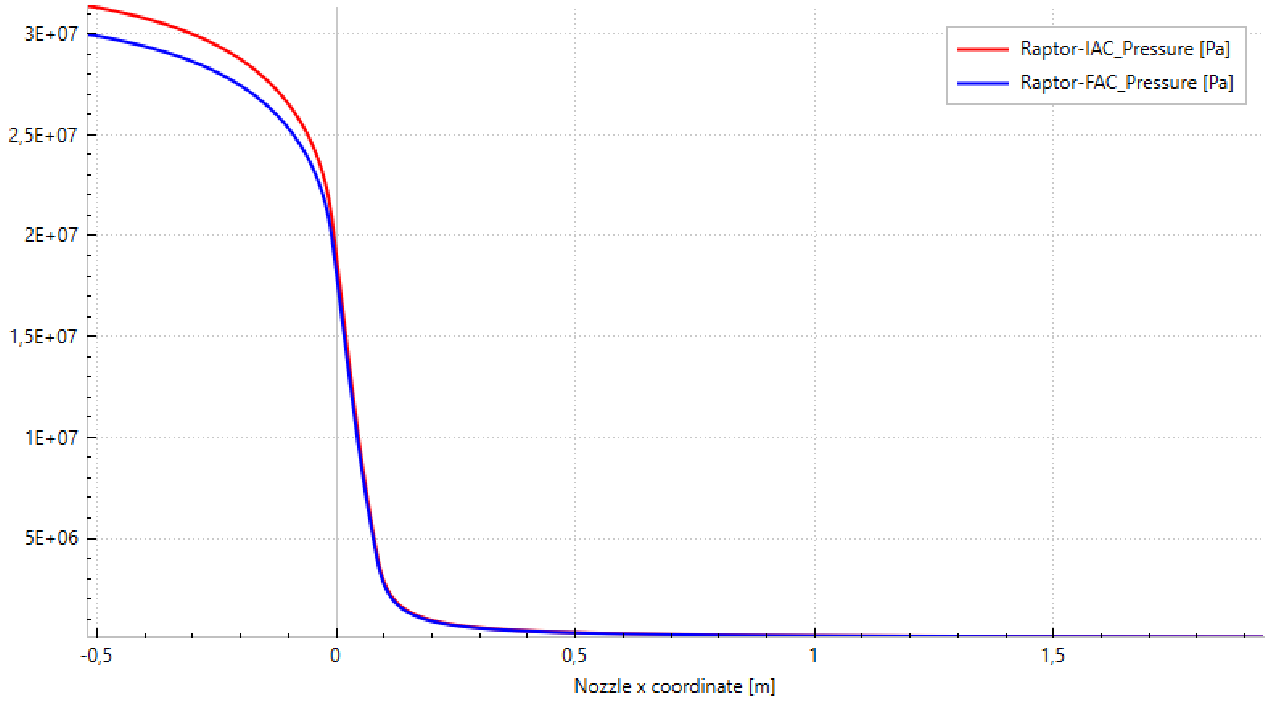


Figure 6.19: Pressure evolution in the nozzle for IAC and FAC models

6.5 ODE nozzle comparison

For the same propellant and chamber pressure as the raptor engine, a set of different nozzles were studied. Their parameters are shown in Table 6.18.

	Raptor	N1	N2	N3	N4	N5
Entrance diameter [m]	0.32	0.32	0.5	0.32	0.32	0.32
Throat diameter [m]	0.2216	0.2216	0.2216	0.2216	0.18	0.26
Exit diameter [m]	1.3	1.3	1.3	1.6	1.3	1.3
Curve	Bezier	Conical	Bezier	Bezier	Bezier	Bezier

Table 6.18: Parameters for the different nozzles to be compared

The resulting rocket parameters for each nozzle can be seen in Table 6.19. There are several details to be pointed out from the obtained results. Firstly, the results for the conical nozzle and Bezier nozzle are exactly the same. This is to be expected, since the mass flow is constant along the nozzle and determined by the throat, the thrust is determined by the values at the throat and exit and specific impulse is determined by the previous two variables. Both nozzles have the same values for throat and exit areas and ODE calculations are only dependent on the values of a given point (chemical equilibrium is established, the history of the mixture is not required). Therefore, it was to be expected that both nozzles would offer the same results. As will be seen in the subsequent section, this needs not be the case for the ODK calculation, since the previous values of the mixture in the nozzle play an important role. For the aforementioned reasons, changing the value for the entrance area also has no effect on the rocket parameters.

Increasing the exit area results in a slight decrease in thrust and specific impulse. This outcome is to be expected since the flow at the exit is no longer adapted to the ambient pressure. Changing the area of the throat has the most significant effect. As is to be expected, reducing it results in a smaller thrust and mass flow, while increasing it also increases said values. It is important to note that N5, N4 and the original nozzle have very similar values for the total impulse, despite their notable differences for thrust and mass flow. Therefore, the choice of throat area will be highly dependant on the thrust requirements of the mission and will also impact the burn time (for the same propellant amount). It is also worth pointing out that the specific impulse for the last two nozzles is slightly less than in the original case, meaning that they are not as efficient.

	Raptor	N1	N2	N3	N4	N5
Thrust [MN]	2.34	2.34	2.34	2.32	1.53	3.21
I_{sp} [km/s]	3.42	3.42	3.42	3.39	3.39	3.40
$\dot{m}$ [kg/s]	685	685	685	685	452	943

Table 6.19: Parameters for the different nozzles to be compared

In Figure 6.20 the pressure evolution for the different nozzles is represented. The original nozzle, along with N1 and N3 coincide in the convergent region, which is to be expected since in this region they are identical. Similarly, the original nozzle and N2 are superimposed in the divergent region. Since the IAC model assumes that the pressure at the combustion chamber corresponds to a section of infinite area, it is clear how an increased value for the entrance area of the nozzle would also increase the initial pressure at the entrance with respect to the original nozzle, as seen for N2. On the other hand, since all points of the nozzle are computed by establishing that the mass flow should equal that of the throat, it is easy to understand how this will also have effect on the pressure at the entrance, as seen in N4 and N5. Moreover, said effect is also slightly present at the divergent region of the throat. Finally, for N1 and N3 nozzles, the area distribution in the divergent region is different from the original nozzle, which explains the different curves. As it is to be expected, said difference is more prominent for the convergent nozzle, since the area distribution is noticeably different.

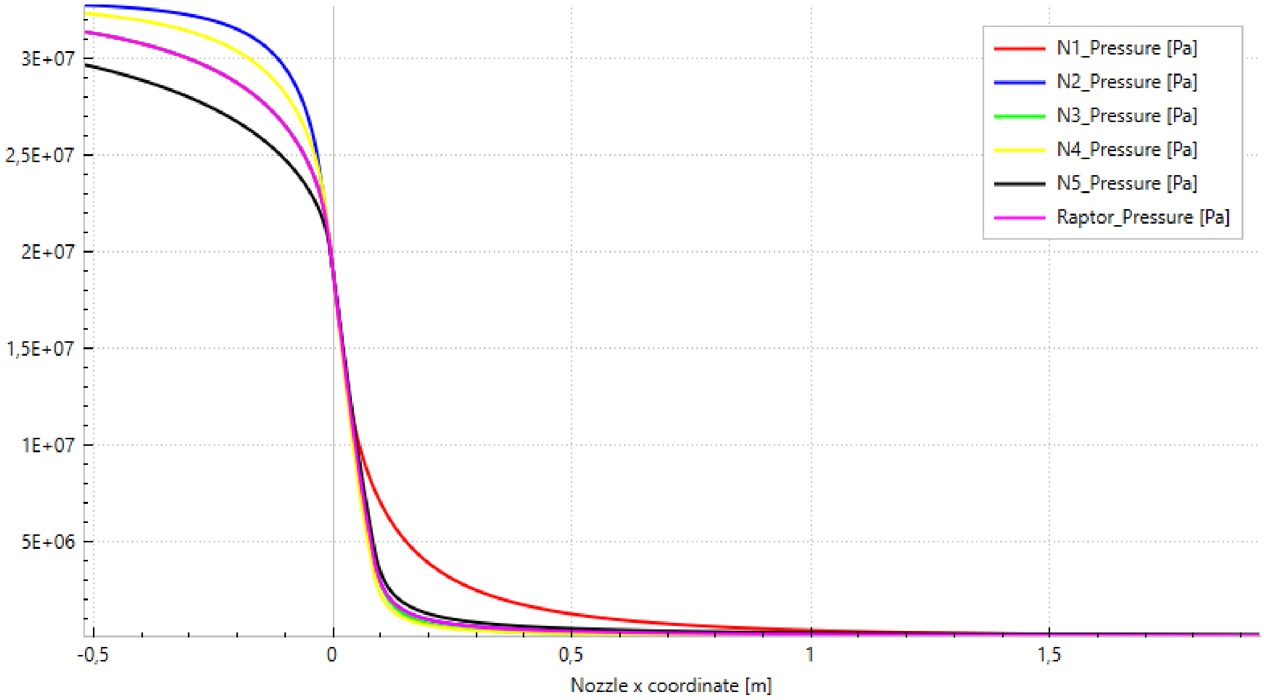


Figure 6.20: Pressure evolution for the nozzles under comparison

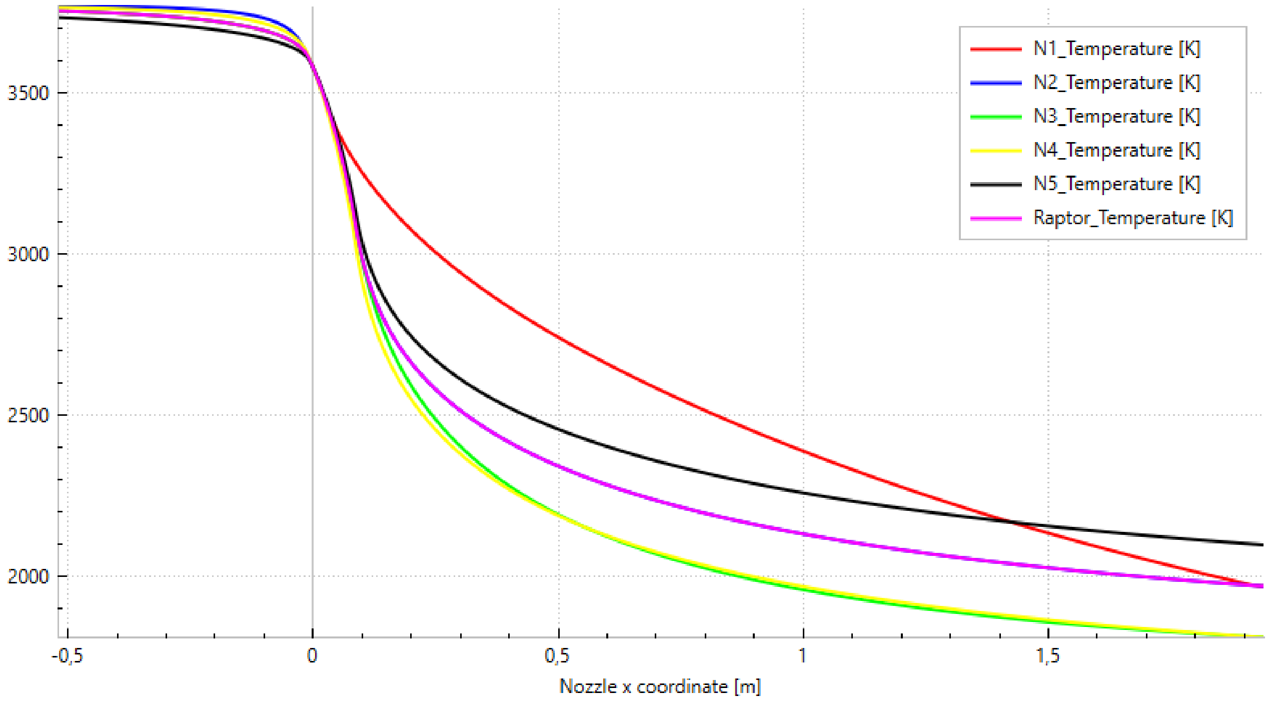


Figure 6.21: Temperature evolution for the nozzles under comparison

Finally, the temperature evolution has been plotted in Figure 6.21. Where the same conclusions can be extracted and the explanations given for the pressure are also valid. It is important to point out that, even though the divergent regions for the conical and original nozzle differ, their values coincide at the exit point. As was explained before, this is to be expected, since the area distribution is different along the divergent nozzle but equal at the exit, and ODE is only dependant on the value of a given point. Note that this also happened in the pressure plot, although it was not so clearly visible.

6.6 ODK nozzle comparison

The nozzles presented in Table 6.18 will now be compared using the ODK model. Note that the propellant will be changed to a version with less species considered, as explained in subsection 6.2.6.

In Figure 6.22 a very similar result to the one presented in Figure 6.20 can be seen, since the pressure values for both models are very close. The same conclusions may be extracted. On the other hand, Figure 6.23 has more visible differences, as the value for Temperature at the exit does not coincide for the original nozzle and N1 (conical). As was previously mentioned, ODK requires the history of the flow in order to compute a point, therefore, the different curves used in both nozzles impact the value at the exit area (even though it is the same area value in both cases). This effect is more clearly visible in the value for gamma, represented in Figure 6.24.

It is important to point out that, the differences in the value for gamma seen in Figure 6.24 are more prominent in the divergent region of the nozzle. Since CH4-LOX results in a fast reacting mixture; after the throat, due to the drop in temperature, reaction rates increase. Therefore, the differences between nozzles are more obvious in this region.

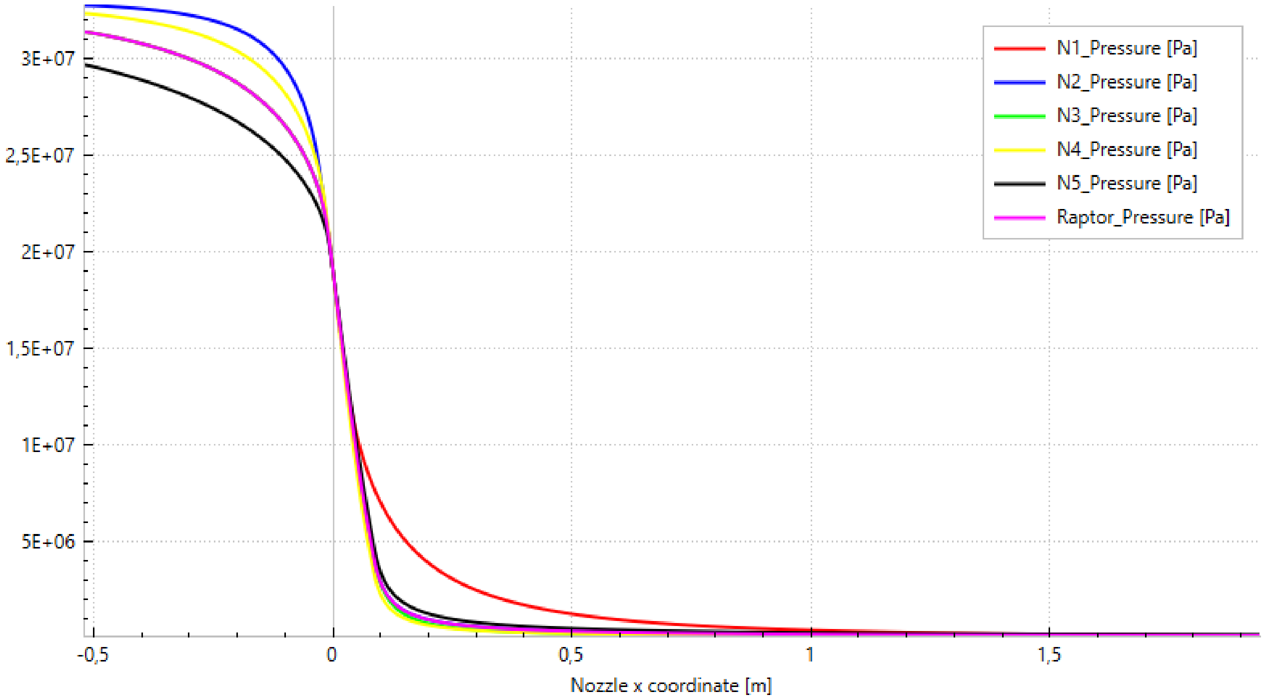


Figure 6.22: Pressure evolution for the nozzles under comparison using ODK

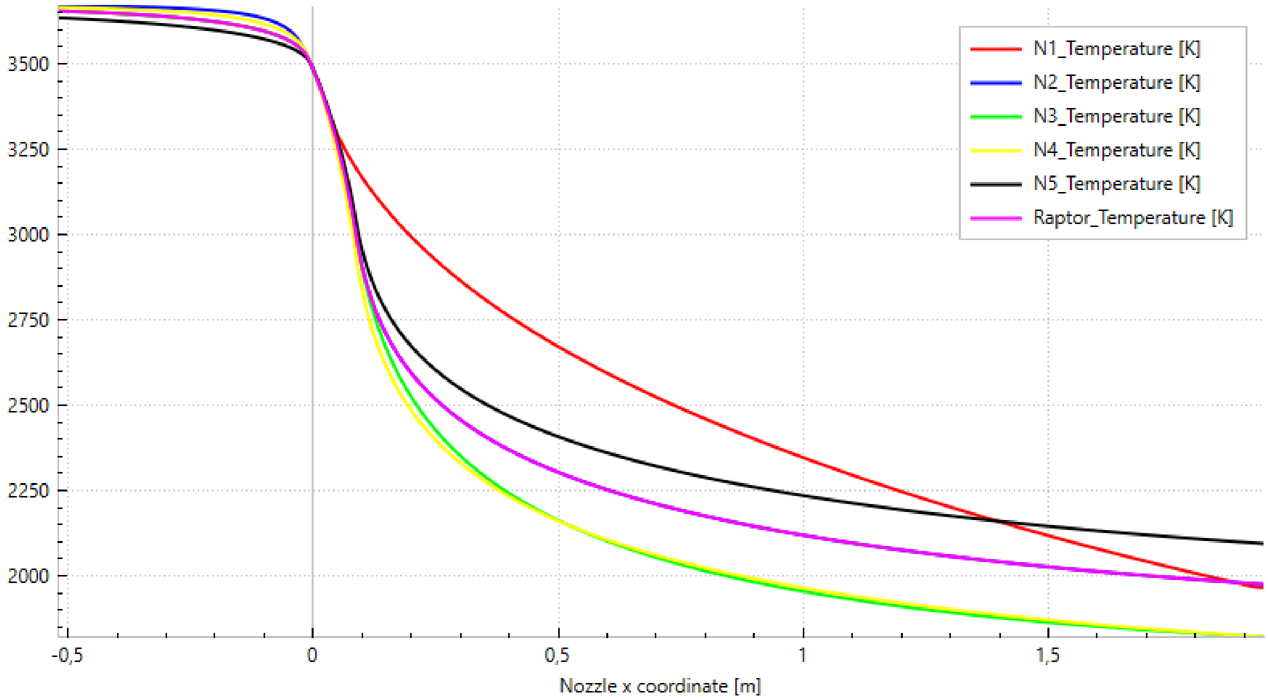


Figure 6.23: Temperature evolution for the nozzles under comparison using ODK

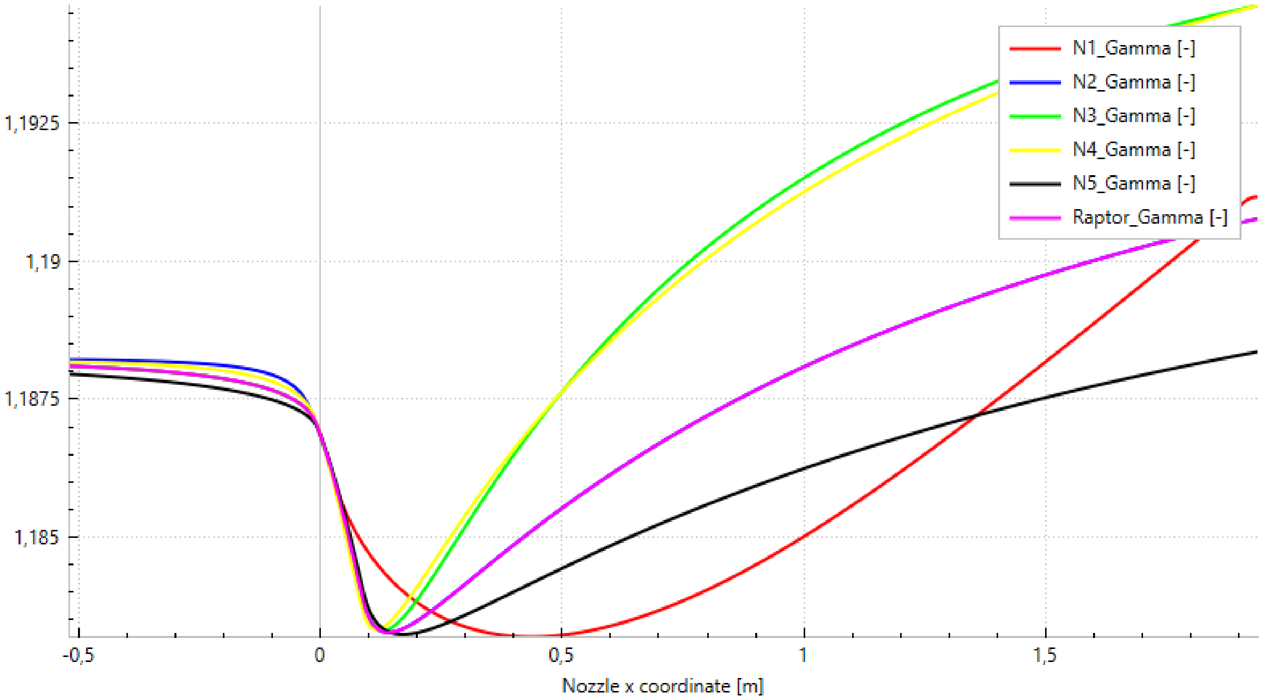


Figure 6.24: Gamma evolution for the nozzles under comparison using ODK

6.7 Effect of reactant temperature

The effect of the initial reactant temperature can only be seen in ODEKOpt if the C_p for reactants is defined in a range of temperatures (if it is single defined it will always be computed for that single temperature). HTPB-AP propellant was used in order to test the effect of the initial temperature. Note that the C_p definition for HTPB was not found, therefore, the effects of reactant temperature are only noticeable in the AP.

As seen in Figure 6.25, the effect of the reactant temperature is very slight, being the value of gamma (along with velocity of sound) one of the most affected by it (and the effect still is very slight in this case). Note that this only accounts for the effect of AP. Therefore, for other substances which may suffer a greater influence of temperature on their enthalpy, this effect might be more prominent.

Nonetheless, for a general scenario, it can be stated that the initial reactant temperature has no effect on rocket performance. Although, it is important to note that it may suppose a relevant variation on Vielle's parameter 'a', which would result in a noticeable difference in the behaviour of the engine.

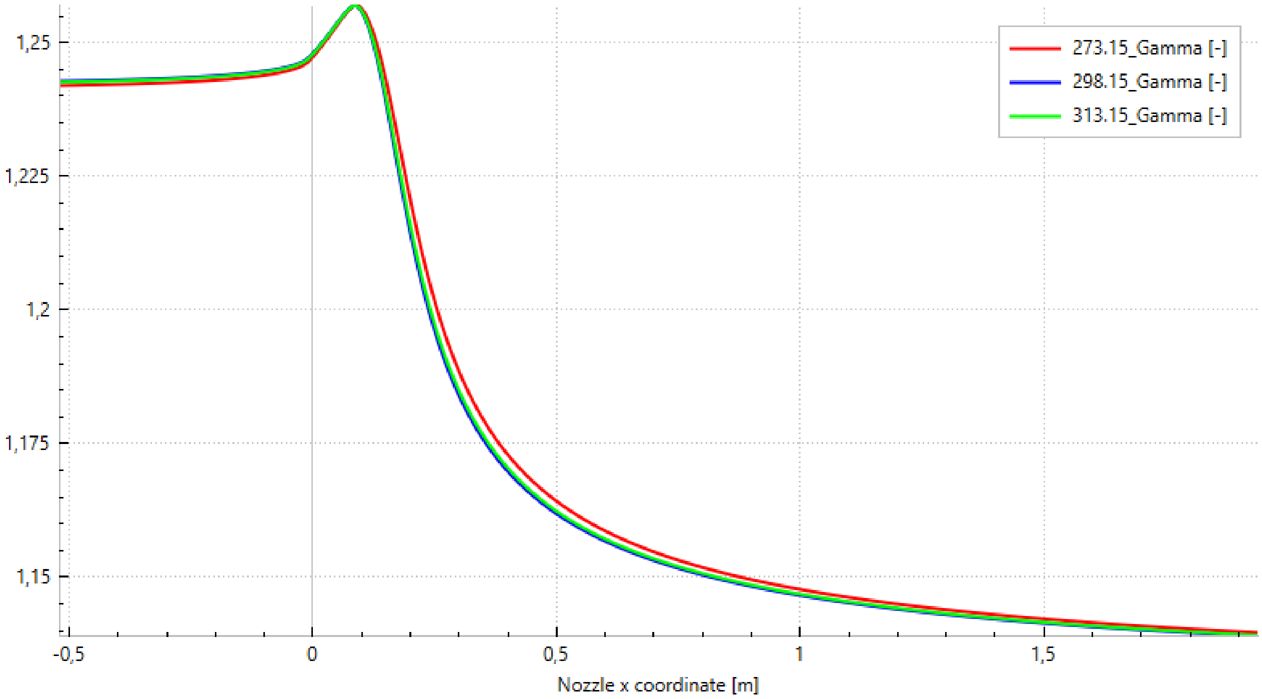


Figure 6.25: Reactant temperature effect on value of gamma throughout the nozzle

Bibliography

[1] Nuigmech1.1
http://c3.nuigalway.ie/combustionchemistrycentre/mechanismdownloads/.

[2] D. R. Bartz. A simple equation for rapid estimation of rocket nozzle convective heat transfer coefficients, 1957.

[3] Sanford Gordon (Sanford Gordon associates) Bonnie J. McBride (Lewis research center). *Computer program for calculation of complex chemical equilibrium compositions and applications. Volume I. Analysis.* 1994.

[4] D. E. Coats G. R. Nickerson, L. D. Dang. Two dimensional kinetic engineering and programming manual., 1985.

[5] Sanford Gordon. *Calculation of Theoretical Equilibrium Nozzle Throat Conditions When Velocity of Sound is Discontinuous.* 1971.

[6] Juan Manuel Tizón Pulido and Guillermo Jenaro de Mencos. *Propulsión de misiles tácticos.*

[7] R. J. Kee, F. M. Rupley, and J. A. Miller. *Chemkin-II: A Fortran Chemical Kinetics Package for the Analysis of Gas-Phase Chemical Kinetics.* 1989.

[8] A. A. Chernov I. D. Emel'yanov O. P. Korobeinichev, N. E. Ermolin. Flame structure, kinetics and mechanism of chemical reactions in flames of mixed composition base on ammonium perchlorate and polybutadiene rubber, 1992.

[9] Juan Manuel Tizón Pulido and Enrique Cabrera Revuelta. *Introducción a los Motores Cohete.*

[10] Victor Quan. Analytical predictions of delivered specific impulse., 1968.

[11] V. K. Jain T. S. Sheshadri. Propellant gas phase chemical kinetics., 1989.

[12] A. F. Kong V. P. Zhukov. A compact reaction mechanism of methane oxidation at high pressures.

A

System of equations for chemical equilibrium

A.1 Equilibrium calculation matrix

Note that for assigned temperature and volume problems, the last two rows and last column of the matrix are not used. For assigned enthalpy and pressure (chemical equilibrium at the combustion chamber), the last row is not used. Finally, for assigned entropy and pressure (chemical equilibrium at a point in the nozzle), the last row is used but not the one before it.

$$
\begin{bmatrix}
\sum_{j=1}^{NG} a_{1j}a_{1j}n_j & \cdots & \sum_{j=1}^{NG} a_{1j}a_{(NE)j}n_j & a_{1(NG+1)} & \cdots & a_{1(NS)} & \sum_{j=1}^{NG} a_{1j}n_j & \sum_{j=1}^{NG} a_{1j}n_j\frac{H_j^o}{RT} \\
\vdots & & \vdots & \vdots & & \vdots & \vdots & \vdots \\
\sum_{j=1}^{NG} a_{(NE)j}a_{1j}n_j & \cdots & \sum_{j=1}^{NG} a_{(NE)j}a_{(NE)j}n_j & a_{(NE)(NG+1)} & \cdots & a_{(NE)(NS)} & \sum_{j=1}^{NG} a_{(NE)j}n_j & \sum_{j=1}^{NG} a_{(NE)j}n_j\frac{H_j^o}{RT} \\
a_{1(NG+1)} & \cdots & a_{(NE)(NG+1)} & 0 & \cdots & 0 & 0 & \frac{H_{NG+1}^o}{RT} \\
\vdots & & \vdots & \vdots & & \vdots & \vdots & \vdots \\
a_{1(NS)} & \cdots & a_{(NE)(NS)} & 0 & \cdots & 0 & 0 & \frac{H_{NS}^o}{RT} \\
\sum_{j=1}^{NG} a_{1j}n_j & \cdots & \sum_{j=1}^{NG} a_{(NE)j}n_j & 0 & \cdots & 0 & \sum_{j=1}^{NG} n_j - n & \sum_{j=1}^{NG} n_j\frac{H_j^o}{RT} \\
\sum_{j=1}^{NG} a_{1j}n_j\frac{H_j^o}{RT} & \cdots & \sum_{j=1}^{NG} a_{(NE)j}n_j\frac{H_j^o}{RT} & \frac{H_{NG+1}^o}{RT} & \cdots & \frac{H_{NS}^o}{RT} & \sum_{j=1}^{NG} n_j\frac{H_j^o}{RT} & \sum_{j=1}^{NS} n_j\frac{C_{p,j}^o}{RT} + \sum_{j=1}^{NG} n_j\frac{H_j^o}{RT}\frac{H_j^o}{RT} \\
\sum_{j=1}^{NG} a_{1j}n_j\frac{S_j}{RT} & \cdots & \sum_{j=1}^{NG} a_{(NE)j}n_j\frac{S_j}{R} & \frac{S_{NG+1}}{R} & \cdots & \frac{S_{NS}}{R} & \sum_{j=1}^{NG} n_j\frac{S_j}{R} & \sum_{j=1}^{NS} n_j\frac{C_{p,j}^o}{RT} + \sum_{j=1}^{NG} n_j\frac{H_j^o}{RT}\frac{S_j^o}{R}
\end{bmatrix}
\begin{bmatrix}
\pi_1 \\ \vdots \\ \pi_{(NE)} \\ \Delta n_{NG+1} \\ \vdots \\ \Delta n_{NS} \\ \Delta \ln n \\ \Delta \ln T
\end{bmatrix}
=
\begin{bmatrix}
(b_1^o - b_1) + \sum_{j=1}^{NG} a_{1j}n_j\frac{\xi_j}{RT} \\
\vdots \\
(b_{NE}^o - b_{NE}) + \sum_{j=1}^{NG} a_{(NE)j}n_j\frac{\xi_j}{RT} \\
\frac{\xi_{NG+1}}{RT} \\
\vdots \\
\frac{\xi_{NS}}{RT} \\
n - \sum_{j=1}^{NG} n_j + \sum_{j=1}^{NG} n_j\frac{\xi_j}{RT} \\
\frac{h_0}{RT} - \frac{h}{RT} + \sum_{j=1}^{NG} n_j\frac{H_j^o}{RT}\frac{\xi_j}{RT} \\
\frac{s_0}{R} - \frac{s}{R} + n - \sum_{j=1}^{NG} n_j + \sum_{j=1}^{NG} n_j\frac{S_j}{R}\frac{\xi_j}{RT}
\end{bmatrix}
\tag{A.1}
$$

A.2 Temperature derivatives matrix

$$\begin{bmatrix} \sum_{j=1}^{NG} a_{1j}a_{1j}n_j & \cdots & \sum_{j=1}^{NG} a_{1j}a_{(NE)j}n_j & a_{1(NG+1)} & \cdots & a_{1(NS)} & \sum_{j=1}^{NG} a_{1j}n_j \\ \vdots & & \vdots & \vdots & & \vdots & \vdots \\ \sum_{j=1}^{NG} a_{(NE)j}a_{1j}n_j & \cdots & \sum_{j=1}^{NG} a_{(NE)j}a_{(NE)j}n_j & a_{(NE)(NG+1)} & \cdots & a_{(NE)(NS)} & \sum_{j=1}^{NG} a_{(NE)j}n_j \\ a_{1(NG+1)} & \cdots & a_{(NE)(NG+1)} & 0 & \cdots & 0 & 0 \\ \vdots & & \vdots & \vdots & & \vdots & \vdots \\ a_{1(NS)} & \cdots & a_{(NE)(NS)} & 0 & \cdots & 0 & 0 \\ \sum_{j=1}^{NG} a_{1j}n_j & \cdots & \sum_{j=1}^{NG} a_{(NE)j}n_j & 0 & \cdots & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{\delta \pi_1}{\delta \ln T} \\ \vdots \\ \frac{\delta \pi_{NE}}{\delta \ln T} \\ \frac{\delta n_{NG+1}}{\delta \ln T} \\ \vdots \\ \frac{\delta n_{NS}}{\delta \ln T} \\ \frac{\delta \ln n}{\delta \ln T} \end{bmatrix} = \begin{bmatrix} -\sum_{j=1}^{NG} a_{1j}n_j\frac{H_j^o}{RT} \\ \vdots \\ -\sum_{j=1}^{NG} a_{(NE)j}n_j\frac{H_j^o}{RT} \\ -\frac{H_{NG+1}^o}{RT} \\ \vdots \\ -\frac{H_{NS}^o}{RT} \\ -\sum_{j=1}^{NG} n_j\frac{H_j^o}{RT} \end{bmatrix} \tag{A.2}$$

A.3 Pressure derivatives matrix

$$\begin{bmatrix} \sum_{j=1}^{NG} a_{1j}a_{1j}n_j & \cdots & \sum_{j=1}^{NG} a_{1j}a_{(NE)j}n_j & a_{1(NG+1)} & \cdots & a_{1(NS)} & \sum_{j=1}^{NG} a_{1j}n_j \\ \vdots & & \vdots & \vdots & & \vdots & \vdots \\ \sum_{j=1}^{NG} a_{(NE)j}a_{1j}n_j & \cdots & \sum_{j=1}^{NG} a_{(NE)j}a_{(NE)j}n_j & a_{(NE)(NG+1)} & \cdots & a_{(NE)(NS)} & \sum_{j=1}^{NG} a_{(NE)j}n_j \\ a_{1(NG+1)} & \cdots & a_{(NE)(NG+1)} & 0 & \cdots & 0 & 0 \\ \vdots & & \vdots & \vdots & & \vdots & \vdots \\ a_{1(NS)} & \cdots & a_{(NE)(NS)} & 0 & \cdots & 0 & 0 \\ \sum_{j=1}^{NG} a_{1j}n_j & \cdots & \sum_{j=1}^{NG} a_{(NE)j}n_j & 0 & \cdots & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{\delta \pi_1}{\delta \ln P} \\ \vdots \\ \frac{\delta \pi_{NE}}{\delta \ln P} \\ \frac{\delta n_{NG+1}}{\delta \ln P} \\ \vdots \\ \frac{\delta n_{NS}}{\delta \ln P} \\ \frac{\delta \ln n}{\delta \ln P} \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{NG} a_{1j}n_j \\ \vdots \\ \sum_{j=1}^{NG} a_{(NE)j}n_j \\ 0 \\ \vdots \\ 0 \\ \sum_{j=1}^{NG} n_j \end{bmatrix} \tag{A.3}$$