Study: Algorithms for Rocket Engine Thermochemistry

UNIVERSITAT POLITÉCNICA DE CATALUNYA

Author: Caleb Fuster Jiménez
Supervisor: Manel Soria Guerrero
Co-supervisor: Arnau Miró Jané
Final master thesis - Thesis
Master's degree in Aerospace Engineering
Fall quarter 2020-2021

MAY 2021

ABSTRACT

This thesis compiles the process of upgrading two libraries (`INIST` and HGS) which provide to their users the thermodynamics properties and algorithms needed to analyse liquid rocket engine feed cycles. First, the Thermodynamic theory for real fluids, ideal fluids and mixtures is reviewed, as well as the algorithms and tools that the libraries implement. For ideal gases, the NASA 7 coefficient polynomials are used while for real fluids, data interpolated from NIST is applied. Then, five rocket engine feed cycles analyses are carried out using the libraries. Turbomachinery (pumps, turbines) as well as combustion chamber, nozzles, mixers and heat exchangers are studied with a thermodynamic analysis. The obtained results show good agrement with data obtained from the literature. Finally, some additional utilities of the potential of the library are given: (i) a chemical equilibrium algorithm based on the direct minimisation of Gibbs free energy that is used to obtain the combustion chamber mixture of dissociated species; (ii) a surrogate of RP1 for the simulation of RP1-LOX rocket cycles; (iii) a study on the effect of using ideal gas versus real fluid properties.

DEDICATION AND ACKNOWLEDGEMENTS

Dedicat a Don Fernando Chorro i Jesus Romero per la passió despertada en mi per estimar la Física i la Química que han sigut part d'aquesta tesi. I a tots els mestres i professors que han sigut particeps d'alguna forma.

Gràcies a Manel Soria per aquest viatge tan entretingut que espero repetir en alguna altra ocasió. També a tothom que d'alguna manera hagi participat siga amb anims, menjar o cervesa.

AUTHOR'S DECLARATION

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: ......CALEB FUSTER JIMÉNEZ .... DATE: ....27 APRIL 2021..............

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

FIGURE **Page**

CHAPTER 1

INTRODUCTION

1.1 General Introduction

Rocketry is a challenge for engineering due to the complex systems that have to perform well enough to defy gravity. One of the critical points of this process concerns to the propulsion system involving complex machinery with an extreme temperature range. Multiple analysis can be done for this system such as a CFD analysis of the nozzle, combustion analysis of the combustion chamber or the thermodynamic analysis of the machinery. This thesis will focus on the thermodynamic analysis of multiple rockets, with different turbomachinery systems and propellants. By and large, rockets are classified into two categories: solid rockets and liquid rockets. In this thesis, only liquid rockets are going to be reviewed as solid rockets are, basically, fuel stored which is burned without any preliminary process while liquid rockets require a cycle before burning the propellants (Fig. 1.2). As it can bee seen in the different author papers (Sec. A.3), cycles differ between missions, depending on the propellants, rocket engine stage and the technology and knowledge available at the time they were designed.

(a) Higher temperature in the RD-180 cycle (Fig. A.13)[1] **(b)** Lowest temperature in the Vinci cycle (Fig. A.9)[2]

Figure 1.1: Subsystems with extreme temperatures

The previous figure shows part of one LH2-LOX cycle (Fig. 1.1(b)) and RP1-LOX cycle 1.1(a)). These three propellants are the widely used while LOX and LH2 are cryogenic liquids as their temperature range in the cycle goes from 20K to 200K. With this special condition, they require special treatment to obtain their thermodynamic values. These values are tabulated and can be obtained from the NIST webpage [3]. RP1 and the post-combustion cycle run in a range from 300K to 3500K and HGS with polynomials could offer the thermodynamic variables.

Two libraries will be created to write codes for rocket analysis of different engines, presumably liquid ones. The purpose of these libraries is didactic, thus they are not optimised for speed of execution but rather for clarity in the code and ease of maintenance. As an example, a function called `HGSminG`, analogous to `Matlab`'s `fmincon` is developed (Sec. 4.27) just to illustrate the minimisation algorithm. With both codes, the analysis of turbomachinery is sought like the scheme below:

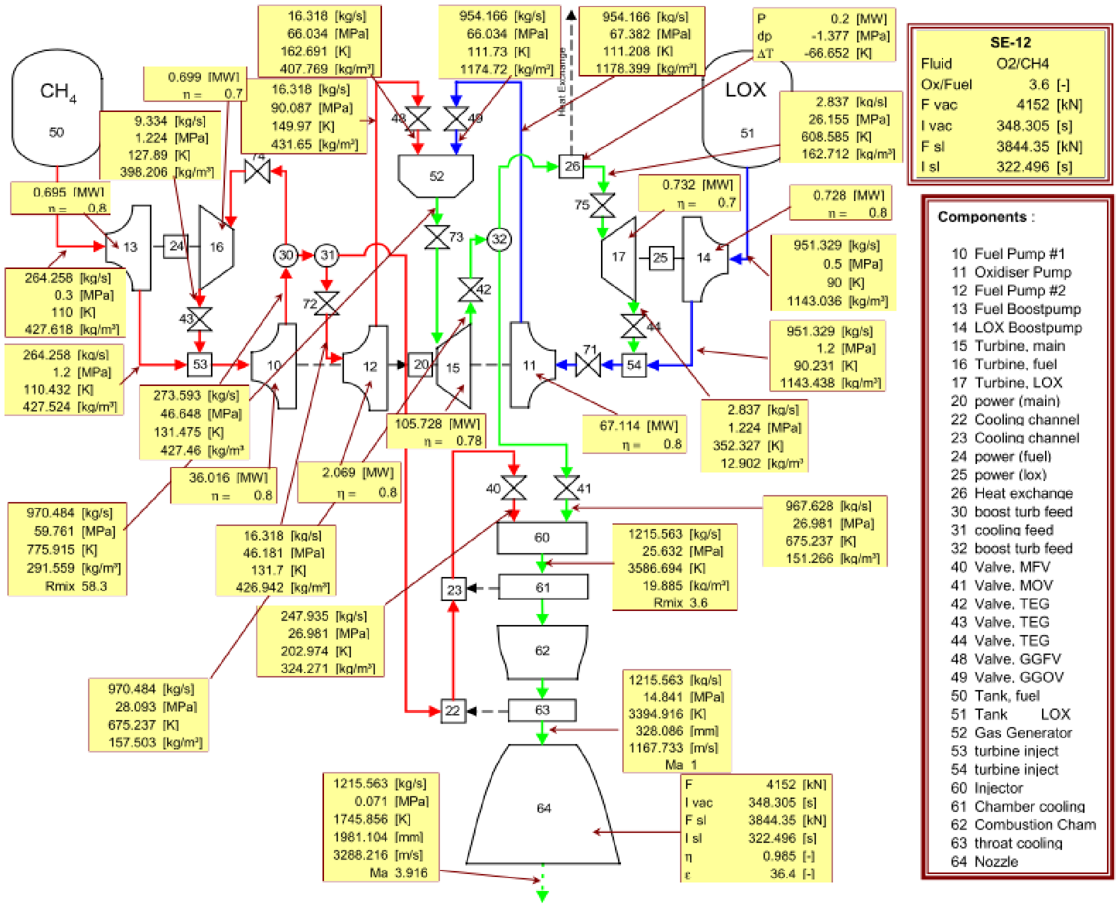


Figure 1.2: SE-12 engine Scheme by SEQ software of DLR [1]

1.2 Aim of the thesis

The master thesis' aim is to upgrade a `MATLAB` Thermochemistry library for analysis of liquid-fuel Rocket engines. The libraries are based on previous versions called `HGS` [4] and `INIST` [5] from professor *Manel Soria* and other authors [6]. To do it, the main aims of this Thesis are: *i*) to review the Thermodynamics and Chemistry basics in order to provide a solid background for the code, *ii*) to implement new functionalities not foreseen in the original codes and *iii*) to write a user manual.

The libraries, `HGS` and `INIST`, have a didactic purpose, as they are used by students as a library to create their own rocket analysis codes.

1.3 Justification

First of all, these libraries are going to be used to create other codes for engines and rocketry. These codes require a script that calculates thermodynamics properties of species or mixtures taking into account some assumptions that will be commented in Chapter 3. Besides, other assumptions will be used for chemical reaction estimations.

The previous versions of `HGS` [4] and `INIST` [5] were not optimised, they have some vicissitudes on the Thermodynamics code and lack clear documentation in some of the functions. There are alternatives from commercial codes (`RPA` [7]) and Open-Source codes (`CEA` [8]) but the license cost or interface limits their access. In Section 2.3, more detailed information is going to be provided in order to explain the main differences between the codes.

`HGS` [4] and `INIST` [5] databases are incomplete, poorly detailed and from an unknown origin, so they require an update with a clear process for further versions and accurate documentation. Burcat coefficients, for `HGS` the database, are periodically updated and new species are incorporated, so database requests a script to make it automatically if needed. In the case of `INIST`, **NIST** [3] changes sporadically their platform so a database is downloaded with more detailed data, even if a script is created to download a database automatically.

Finally, elaborate documentation, besides code comments, is requested for both codes as they are going to be used by inexperienced users in thermodynamics. Thermodynamics and scripts limits must be clearly detailed, especially in the thesis and user manual, to avoid misunderstandings that could end in errors.

1.4 Scope

The scope of this thesis is to make:

- In HGS

1. a review of thermochemistry: ideal gases assumptions and mixtures, evaluation of enthalpy, entropy and Gibbs free energy for substances with C_p function of T,
2. an implementation of the database for NASA 7 polynomials. characterisation of the species in terms of their chemical structure or empirical composition in the case of jet and rocket propellants.
3. an implementation, validation and optimisation of a single species property calculator,
4. an implementation, validation and optimisation of a gas mixture property calculator,
5. an implementation and validation of a gas equilibrium code based on a restricted minimisation algorithm, given T and P,
6. an extension of the previous code to constant enthalpy and constant entropy.
7. an application to rocket combustion chambers and nozzles.
8. and, finally, an elaboration of a user manual.

- In INIST
    1. an implementation of a new database with more accuracy.
    2. an optimisation of the overall performance of INIST.
    3. at last, some additional examples for didactic purposes.

1.5 Requirements

In order to succeed, a good assessment of the project relies (Sec. 1.4) in the following requirements:

- the comprehension of code development for projects,
- the review of gas thermodynamics for rocketry and engines,
- the review of code optimisation and optimisation tools,
- and clear documentation and code comments for the last users.

CHAPTER 2

STATE OF THE ART

In this chapter, the state of the art will cover a historical review of thermodynamics, thermodynamic and chemistry concepts and current sources that are helpful when calculating thermodynamics properties. This chapter does not cover thermodynamics development for the codes, since this is furtherly analyzed in Chapter 3.

2.1 Review of Thermodynamics

Thermodynamics is early related to chemistry and physics but it was not until 1656 when ***Boyle*** and ***Hooke*** created the first air pump that allowed to carry out the experiments described in *New Experiments physico-mechanical, Touching the Spring of the Air and its Effects*. which inspired Boyle-Mariotte's law [9]. But, it was not until the Fahrenheit thermometer appeared, that other gas laws could be formulated with accuracy [10]. Charles and Gay-Lussac's laws were formulated in 1802 when ***Gay-Lussac*** presented his results to the French National Institute [11]. ***Clapeyron*** used these three laws plus Avogadro's law to present the ideal gas law equation state model in 1835 [12].

In parallel to the gas theory, the first engine was built in 1697: the Savery engine [13]. Under the shelter of the Industrial Revolution (1760-1840) [14], thermodynamics experienced great growth due to the underlying economic interests of research and innovation. With ***Carnot***, the father of modern thermodynamics, significant advances were made in the field. ***Carnot*** introduced in 1824 the concepts of useful work, efficiency (Carnot's rule), outline of the thermodynamic second law,... in his publication "*Réflexions sur la puissance motrice du feu et sur les machines propres à développer cette puissance*" [15].

Using Carnot's work as reference, ***Joule***, ***Clausius***, ***Rankine***, ***Gibbs***, ***Nernst*** and ***Lord Kelvin*** developed modern thermodynamics. Joule's work "*On the Mechanical Equivalent of Heat*" connected work and heat, this concept is related with the Energy Conservation law, and led into the first law of thermodynamics [16], which was introduced by ***Clausius***, and ***Rankine***. Separately, in 1850 [17]. ***Clausius*** also introduced the entropy concept, in rigorous form, and the second law of thermodynamics in 1850 [18]. The third law is the Nernst's Heat theorem, introduced in 1912 by Walter ***Nernst*** [19]. Zeroth's law was enunciated the latest, in 1931, by ***R. H. Fowler*** as it was an obvious idea that can not be deduced from the other three [20]. Gibbs presented the concept of available energy (also known as free Gibbs' energy) that links enthalpy entropy with a variable useful for chemical equilibrium [21]. Lord Kelvin collaborated with the previous scientist, he even postulated his thermodynamics second law and the term thermo-dynamics in 1854 [22].

With the basis established, thermodynamics became a very open discipline to investigate. Thermodynamics branches are varied, related to other disciplines, like thermochemistry, thermodynamics engineering, biological thermodynamics, quantum thermodynamics or statistical thermodynamics. Most of them are out of the scope, however, this thesis and contents can be explained with basic thermodynamics and chemical thermodynamics. Previous concepts are explained in the following section (Sec. 2.2) when used in the thesis.

2.2 Thermodynamic and Chemistry Concepts

Some concepts and ideas are going to be explained as they are used often during the thesis. This thesis refers to two types of fluid models: *i*) Ideal gas model *ii*) and real fluid model. Rockets could use both models for different parts of their machinery, and even both can be applied to the same part. To distinguish between the two models, the differences in usage can be reflected in the following figure:

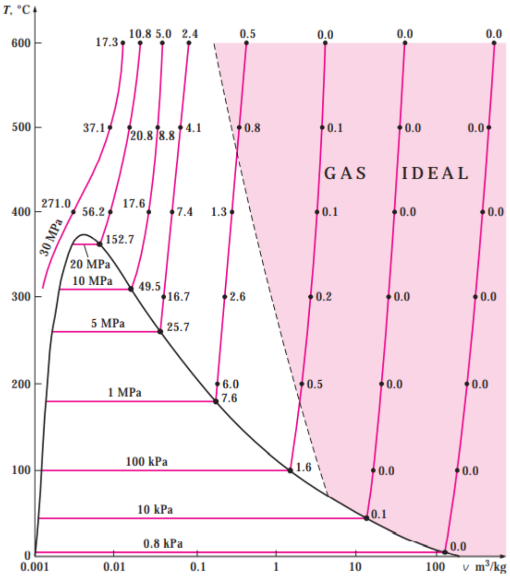


Figure 2.1: Ideal gas model acceptability with less of 1% error the specific volume [20]

As seen, the ideal gas model could be used away from the saturation bell (Sec. 3.1.4) where the pressure effect is minimum and there is no possibility of it becoming a condensable fluid.

2.2.1 NIST Tabulated Data

`INIST` is based on the National Institute of Standards and Technology (NIST) tabulated data. This database offers thermodynamic data among other data, (μ , sound speed, ρ,...) from some species like O_2, N_2, R-113, CH_4... The data can be obtained under three references, near boiling point (NBP), ASHRAE or IIR, by default each element has its own reference. As the data downloaded could be done by default, here is the definition of the possible references.

- NBP reference sets the enthalpy and entropy to zero for the saturated liquid at the normal boiling point temperature.
- ASHRAE reference sets to zero for enthalpy and entropy for the saturated liquid at –40 °C.
- IIR reference sets to 200 kJ/kg and 1 kJ/(kg-K) for enthalpy and entropy, respectively, for the saturated liquid at 0°C.

2.2.2 Ideal Gas

The ideal gas model state equation is broadly used to link pressure, volume and temperature for any gas substances. As mentioned before in Section 2.1, Clapeyron combines the three gas laws of Boyle-Mariotte, Gay-Lussac and Charles' with Avogadro's laws to enunciate the following state equation [20]:

$$
\begin{aligned}
P \cdot V &= k_1 \text{ (Boyle-Mariotte's law)}\\
\frac{V}{T} &= k_2 \text{ (Charles' law)}\\
\frac{P}{T} &= k_3 \text{ (Gay-Lussac's law)}\\
\frac{V}{n} &= k_4 \text{ (Avogadro's law)}\\
P \cdot V &= n \cdot R \cdot T \text{ (State equation for ideal gas)}
\end{aligned}
\tag{2.1}
$$

Where

$$
\begin{aligned}
&k_n \text{ is a constant relation for each gas mixture}\\
&R = 8.31447 \frac{\text{J}}{\text{mol} \cdot \text{K}} \text{ universal gas constant}
\end{aligned}
\tag{2.2}
$$

Some properties depend on temperature plus other variables (pressure, volume,...), but if the ideal gas model is applied, U, H, C_p and C_v only depend on the temperature [23]. Starting from the first law of thermodynamics (Eq. 2.23), for a closed system ($\Delta E_{sys} = 0$), without changes in kinetic or potential energy ($K_e = 0$ and $P_e = 0$) which has a substance with a constant mass and compressible. It is expressed for an internally reversible process as [20]:

$$\delta Q_{\text{int rev}} - \delta W_{\text{int rev}} = dU \tag{2.3}$$

The definition of entropy for reversible process and boundary work is:

$$
\begin{aligned}
\delta Q_{\text{int rev}} &= T\,dS\\
\delta W_{\text{int rev}} &= P\,dv
\end{aligned}
\tag{2.4}
$$

The first $T\,dS$ (or Gibbs') equation is obtained:

$$T \cdot dS = dU + P\,dV \tag{2.5}$$

The first specific Gibbs' equation is:

$$T\,ds = du + P\,dv \tag{2.6}$$

Throughout the definition of enthalpy [24]:

$$h \equiv u + P\,v \tag{2.7}$$

In a differential form:

$$dh = du + v\,dP + P\,dv \tag{2.8}$$

With the specific first Gibbs' equation and the previous differential enthalpy definition, the second Gibbs' equation is obtained:

$$T \cdot ds = dh - v\,dP \tag{2.9}$$

From the first Gibbs' equation (Eq. 2.5):

$$\begin{aligned} dU &= T\,dS - P\,dV \\ \left.\frac{\partial U}{\partial V}\right|_T &= T\left.\frac{\partial S}{\partial V}\right|_T - P \end{aligned} \tag{2.10}$$

The Maxwell relation [20] obtained from Helmholtz function $\left.\frac{\partial S}{\partial V}\right|_T = \left.\frac{\partial P}{\partial T}\right|_V$ can be substituted in the previous equation obtaining:

$$\left.\frac{\partial U}{\partial V}\right|_T = T\left.\frac{\partial P}{\partial T}\right|_T - P \tag{2.11}$$

Then, applying the ideal gas state equation:

$$\begin{aligned} \left.\frac{\partial U}{\partial V}\right|_T &= T\left.\frac{R}{V}\right|_T - \frac{R \cdot T}{V} \\ \left.\frac{\partial U}{\partial V}\right|_T &= 0 \end{aligned} \tag{2.12}$$

As the derivation of the internal energy by the volume is 0, it depends only on temperature ($U = U(T)$).

For the enthalpy case, it can be explained from the second Gibbs' equation 2.9:

$$\begin{aligned} dH &= -T\,dS + V\,dP \\ \left.\frac{\partial H}{\partial P}\right|_T &= T\left.\frac{\partial S}{\partial P}\right|_T + V \end{aligned} \tag{2.13}$$

The Maxwell relation [20] obtained from Gibbs' function $\left.\frac{\partial S}{\partial P}\right|_T = -\left.\frac{\partial V}{\partial T}\right|_P$ can be substituted in the previous equation obtaining:

$$\left.\frac{\partial H}{\partial P}\right|_T = -T\left.\frac{\partial V}{\partial T}\right|_T + V \tag{2.14}$$

Then, applying the ideal gas state equation:

$$\begin{aligned} \left.\frac{\partial H}{\partial P}\right|_T &= T\left.\frac{R}{P}\right|_T - \frac{R \cdot T}{P} \\ \left.\frac{\partial H}{\partial P}\right|_T &= 0 \end{aligned} \tag{2.15}$$

Meanwhile, the partial derivative becomes 0 and H depends only on temperature ($H = H(T)$).

For the specific heat at a constant volume and pressure as they are defined in the equations 3.5 and 3.6 , they depend on U and H respectively which in turn is only determined by temperature so $C_v \propto U(T) \rightarrow C_v = C_v(T)$ and $C_p \propto H(T) \rightarrow C_p = C_p(T)$.

2.2.3 Mixture Properties

This section explains the calculation for mixtures based on the single properties and the ideal gas model. There are four ways to define the composition of a mixture [20]:

- Mass
- Mass fraction
- Molar concentration
- Molar fraction

They are defined as follows:

The mass (m): of a mixture is the summation of each one of the species.

$$m = \sum_{i=1}^{N} m_i \tag{2.16}$$

The mass fraction (mf): is the ratio of the mass of a species to the mass of the mixture.

$$mf_i = \frac{m_i}{\sum_{i=1}^{N} m_i} \tag{2.17}$$

The molar concentration (n): of a mixture is the summation of the mols of each species.

$$n = \sum_{i=1}^{N} n_i \tag{2.18}$$

The molar fraction (y): is the ratio of the mols of a species to the mols of the mixture.

$$y_i = \frac{n_i}{\sum_{i=1}^{N} n_i} \tag{2.19}$$

There are two types of properties depending on how the property modifies itself to adjust to the system: intensive properties, like density, do not change when the amount of substance varies; and extensive properties which vary proportionally when the substance amount changes. [20].

Extensive properties mixture calculation [20]:

$$\Phi_T = \frac{\sum_{i=1}^{n} n_i \cdot \phi_i}{\sum_{i=1}^{n} n_i} = \sum_{i=1}^{n} \frac{n_i}{n} \cdot \phi_i = \sum_{i=1}^{n} y_i \cdot \phi_i \tag{2.20}$$

Extensive properties mixture calculation [20]:

$$\Phi_T = \sum_{i=1}^{n} n_i \cdot \phi_i \tag{2.21}$$

Additionally, entropy and Gibbs' free energy require Dalton's law that is enunciated as: "the total pressure exerted is equal to the sum of the partial pressures of the individual gases". This law is only applicable if the gases follow the ideal gas behaviour explained in Section 2.2.2. So [20]:

$$P_i = P_t \cdot \frac{n_i}{n_t} \tag{2.22}$$

2.2.4 Burcat's Coefficients

`HGS` is based on Burcat's coefficients, also known as the NASA 7 polynomial. These coefficients allow calculating C_p, H and S values of any species in the database. They appear as an alternative to tabulated data from JANAF and other sources. Burcat's coefficients are more restricted than the 9 NASA polynomials, as their limits are up to 6000K while 9 coefficients are up to 20000K.

Polynomials were a bad alternative before the appearance of computers and, even with the actual kinetic programs, they still are a good way to calculate thermodynamic properties, assuming their temperature and hypothesis limitation. As the range is large (from 200K to 6000K), each species has two branches, from 200K to 1000K and 1000K to 6000K.

2.2.5 Thermodynamic Laws

Thermodynamics is based on four basic laws enunciated during the 19th and 20th centuries (Sec. 2.1).

2.2.5.1 Law Zero of Thermodynamics

It is a phenomenological law stated by ***R.H. Fowler*** that could be uttered as:

> *"if two bodies are in thermal equilibrium with a third, they are in equilibrium with each other."*[20]

This sentence is the base of thermal measurement as the third body could be a thermometer.

2.2.5.2 First Law of Thermodynamics

It is the first law to be stated by ***Clausius*** in 1850 that could be uttered as:

> *"in all cases in which work is produced by the agency of heat, a quantity of heat is consumed which is proportional to the work done; and conversely, by the expenditure of an equal quantity of work an equal quantity of heat is produced."*[17]

This law (also known as Energy conservation) has different interpretations depending on whether the system type is open or closed and how the variables that take actions over the system, but it can be stated mathematically in a generalised form as:

$$E_{in} - E_{out} = \Delta E_{sys} \tag{2.23}$$

Both, inlet and outlet energies can be categorised in Heat transfer (Q), Work transfer (W) and Mass transfer, which includes the internal, kinetic and potential energy (U, K_e and P_e).

2.2.5.3 Second Law of Thermodynamics

The second law is a phenomenological negative statement. The law can not be proved. However, no experiment has demonstrated the opposite so it should be taken as sufficient validation proof. One of the first scientist to state it was Clausius in 1850 and uttered as:

> *"heat can never pass from a colder to a warmer body without some other change, connected therewith, occurring at the same time."*[18]

Kevin-Plank's statement is equivalent to this one. The different statements were made due to the approach of the scientists, e. g. Clausius was focused on refrigerators and heat pumps while Kevin-Plank were focused on heat engines[20]. The mathematical state is:

$$\Delta S \geqslant \oint \frac{\delta Q}{T} \tag{2.24}$$

2.2.5.4 Third Law of Thermodynamics

As the previous laws, multiple statements have been done to express the phenomenological law, ***Nernst*** stated it in 1912 and uttered as:

> *"it is impossible for any process, no matter how idealised, to reduce the entropy of a system to its absolute-zero value in a finite number of operations."*[19]

Moreover, entropy for a crystalline substance is 0 when it reaches the absolute 0. Mathematically, it can be expressed as:

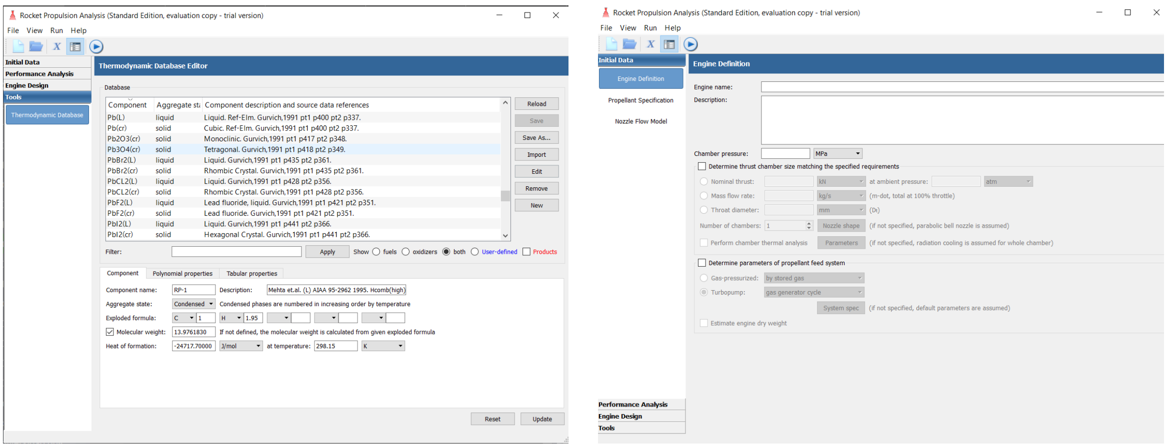
$$\lim_{x \to 0} \Delta S = 0 \tag{2.25}$$

2.3 Other Codes for Rocket Thermochemistry

Chemistry reactions and thermodynamic data are of special interest in aerospace engineering and rocketry, but are essential in chemistry and other branches of engineering. Consequently, some scripts, web app and programs have appeared to fulfil the need for general or specific purposes.

2.3.1 RPA

RPA and RPA-C are two programs from ***Alexander Ponomarenko*** that are written in C++ and Java. RPA simulates the combustion of a rocket from the turbo-machinery to the nozzle outlet, while RPA-C calculates combustion reaction for general purposes as automotive or energy in engineering. Thermodynamics data is obtained from NASA Glenn's thermodynamic database [25] and Grurvich's thermodynamic database [26], using 9 coefficients and 7 coefficients depending on the species. For chemical reactions, RPA uses Gibbs' energy minimisation with a Newton-Raphson's method after linearization of the governing equations [7].

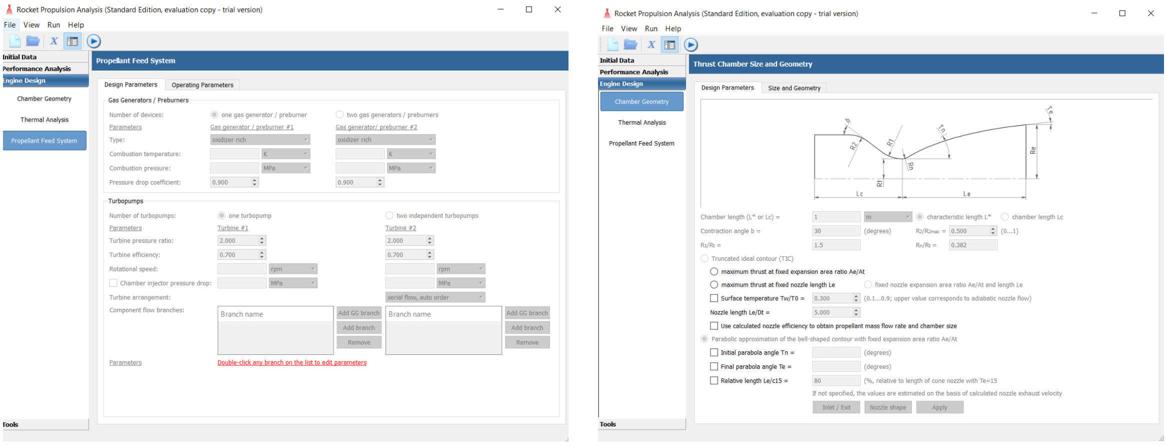


Component	Aggregate st	Component description and source data references
Pb(L)	liquid	Liquid. Ref-Elm. Gurvich,1991 pt1 p400 pt2 p337.
Pb(cr)	solid	Cubic. Ref-Elm. Gurvich,1991 pt1 p400 pt2 p337.
Pb2O3(cr)	solid	Monoclinic. Gurvich,1991 pt1 p417 pt2 p348.
Pb3O4(cr)	solid	Tetragonal. Gurvich,1991 pt1 p418 pt2 p349.
PbBr2(L)	liquid	Liquid. Gurvich,1991 pt1 p435 pt2 p361.
PbBr2(cr)	solid	Rhombic Crystal. Gurvich,1991 pt1 p435 pt2 p361.
PbCL2(L)	liquid	Liquid. Gurvich,1991 pt1 p428 pt2 p356.
PbCL2(cr)	solid	Rhombic Crystal. Gurvich,1991 pt1 p428 pt2 p356.
PbF2(L)	liquid	Lead fluoride, liquid. Gurvich,1991 pt1 p421 pt2 p351.
PbF2(cr)	solid	Lead fluoride. Gurvich,1991 pt1 p421 pt2 p351.
PbI2(L)	liquid	Liquid. Gurvich,1991 pt1 p441 pt2 p366.
PbI2(cr)	solid	Hexagonal Crystal. Gurvich,1991 pt1 p441 pt2 p366.

Component name: RP-1 Description: Mehta et.al. (L) AIAA 95-2962 1995. Hcomb(high)
Aggregate state: Condensed Condensed phases are numbered in increasing order by temperature
Exploded formula: C 1 H 1.95
☑ Molecular weight: 13.9761830 If not defined, the molecular weight is calculated from given exploded formula
Heat of formation: -24717.70000 J/mol at temperature: 298.15 K

Engine name:
Description:
Chamber pressure: MPa
☐ Determine thrust chamber size matching the specified requirements
○ Nominal thrust: kN at ambient pressure: atm
○ Mass flow rate: kg/s (m-dot, total at 100% throttle)
○ Throat diameter: mm (Dt)
Number of chambers: 1 (if not specified, parabolic bell nozzle is assumed)
☐ Perform chamber thermal analysis (if not specified, radiation cooling is assumed for whole chamber)
☐ Determine parameters of propellant feed system
○ Gas-pressurized: by stored gas
● Turbopump: gas generator cycle
(if not specified, default parameters are assumed)
☐ Estimate engine dry weight

(a) Thermodynamic data interface **(b)** Engine Data interface



Gas Generators / Preburners
Number of devices: ● one gas generator / preburner ○ two gas generators / preburners
Parameters Gas generator / preburner #1 Gas generator/ preburner #2
Type: oxidizer rich oxidizer rich
Combustion temperature: K K
Combustion pressure: MPa MPa
Pressure drop coefficient: 0.900 0.900

Turbopumps
Number of turbopumps: ● one turbopump ○ two independent turbopumps
Parameters Turbine #1 Turbine #2
Turbine pressure ratio: 2.000 2.000
Turbine efficiency: 0.700 0.700
Rotational speed: rpm rpm
☐ Chamber injector pressure drop: MPa MPa
Turbine arrangement: serial flow, auto order
Component flow branches: Branch name Branch name
Parameters Double-click any branch on the list to edit parameters

Chamber length (L* or Lc) = 1 m ● characteristic length L* ○ chamber length Lc
Contraction angle b = 30 (degrees) R_2/R_{2max} = 0.500 (0...1)
R_1/R_t = 1.5 R_n/R_t = 0.382
○ Truncated ideal contour (TIC)
○ maximum thrust at fixed expansion area ratio Ae/At
○ maximum thrust at fixed nozzle length Le ○ fixed nozzle expansion area ratio Ae/At and length Le
☐ Surface temperature Tw/T0 = 0.300 (0.1...0.9; upper value corresponds to adiabatic nozzle flow)
Nozzle length Le/Dt = 5.000
☐ Use calculated nozzle efficiency to obtain propellant mass flow rate and chamber size
● Parabolic approximation of the bell-shaped contour with fixed expansion area ratio Ae/At
☐ Initial parabola angle Tn = (degrees)
☐ Final parabola angle Te = (degrees)
☐ Relative length Le/c15 = 80 (%, relative to length of cone nozzle with Te=15
If not specified, the values are estimated on the basis of calculated nozzle exhaust velocity

(c) Machinery interface creation **(d)** Nozzle design interface

Figure 2.2: RPA interfaces

Due to their module coding, their scripts can be used to create new ones by the users, however, it is not of free access.

2.3.2 NASA Software

NASA, with its independent laboratories, created some softwares for rocketry thermodynamics and chemical reaction. CEA was the first code for chemical reactions based on Fortran's developed in [27]. CEARUN [8] is a web-based user-interface from the Fortran's code of ***Gordon-McBride***. There are some variation of this code for example CEAM for Matlab[28]. CEARUN is free to access while its variants for `Fortran` or `Matlab` are under request to the NASA research centers (Marshal space flight and Glenn respectively) and probably with a cost.

Some codes developed by third-parties based on CEA can be found on the internet. They are developed in multiples languages like `Python` [29] or `Matlab` [30] that wraps the results from CEARUN or make an interface from `Fortran's` code in `Matlab` [31].

2.3.3 CHEMKIN

CHEMKIN is one of the best known thermodynamic codes regarding kinetic reaction [32]. It is a widely used code by researchers and companies to develop their work on CFD, chemistry, automotive,... It is not an open-source code but there is some capacity to be adapt to the user's needs. Its major inconvenient is the cost, as is not a free.

2.3.4 TEA

This `Python` code is developed by Jasmina Blecic et al. [33]. It uses NASA polynomial combined with their Gibbs' energy minimisation (GEM) based on Lagrange's multipliers to obtain chemical equilibrium. As it is a modular and open-source code, it could be used by any user with the corresponding citation.

2.3.5 Other Codes

There are more codes to obtain thermodynamic data and chemical equilibrium. However, it is not the main point of this thesis. But there is an interesting point with how they calculate the chemical equilibrium. ***Markus Hans Alexander*** made a summary of the software, the purpose and methodology of his thesis [34]. The following table will classify part of these codes oriented to general, combustion or aerospace utility:

Code/Authors	**Objective**	**Methodology for CE**
CEA	Combustion, rocket design	GEM
ChemApp	General	GEM
ChemSage	General	GEM
Dantzig	Combustion	Simplex
FACT	General	GEM
FactSage	General	GEM
HSC	General	GEM
MTDATA	General	GEM
Schnedler	Thermochemistry for FEM on CFD	GEM
SOLGASMIX and derivateds	General	GEM
STANJAN/EQUIL	Combustion	PFE
Storey et al	Combustion	GEM
THERIAK	Combustion	GEM
ThermoCalc and derivateds	General	GEM
White et al	Combustion	GEM
Zeleznik et al	Combustion, rocket	LMA

Table 2.1: Code analysis from Markus Hans Alexander thesis [34]

Most of the codes use Gibbs' Energy Minimisation (GEM) (Sec.3.2.7) but some of them use other interesting ways like the Simplex method to approximate the GEM, law of mass action (LMA) or partitioning of free energy (PFE). From this list, chemical kinetic programs are excluded like `CHEMKIN` or `COSILAB` as their idea is based on understanding the rates of the chemical reactions. For other purposes, there are several methods like Brinkley-Newton-Raphson's as cosmic nebular gases enriched in dust or Helmholtz' energy minimising for fluid-phase equilibrium [34].

CHAPTER

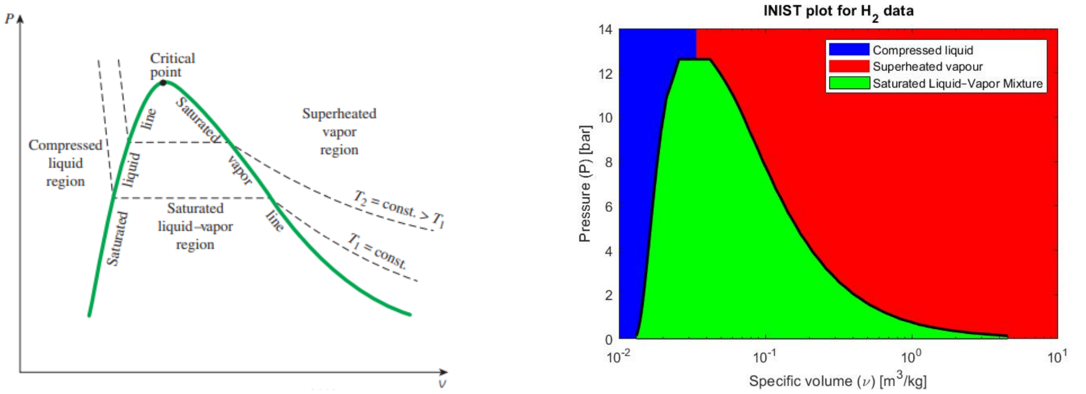
3

THERMODYNAMICS OF ROCKET ENGINES

In this chapter, thermodynamics equations and concepts, which are used in the `HGS` library, are summarised besides their assumptions to calculate thermodynamic properties of a species. Moreover, the assumptions and calculation for mixtures are going to be explained. Furthermore, assumptions for chemical equilibrium, chemical reaction and isentropic expansion are going to be detailed. `INIST` has not thermodynamic theory development as it uses tabulated data but some insights are explained in order to clarify the provenance of the data.

3.1 INIST Model

`INIST` does not need a thermodynamic development as the data comes tabulated from NIST [3]. However, some explanations about this data must be given in order to understand its suitability. NASA polynomials were proved to be useful in a huge range of temperatures and species but they have limitations: low temperature (under 200K depending on the species) and ideal gas behaviour. As it is seen in the figure 1.1(b), rocketry requires cryogenic data from some propellants and, at these conditions, if it remains still a gas, ideal gas behaviour can not be supposed. When low temperatures are reached, it is found that the state of the species acts in accordance with a *P-V-T* diagram as the following image:



(a) Generic *P-V-T* diagram [20]

(b) `INIST` for H2 data with the regions delimited

Figure 3.1: *P-V-T* diagrams

These diagrams show the three regions where a cooled-off fluid could be. Each substance has its curve but `INIST` could not offer a good approximation near the critical point. It is required two intensive properties to determine the state of the substance.

3.1.1 Saturated Liquid and Saturated Vapour

This line is known as saturation bell. On this line, the substance is the limit to be defined as a single state (vapour or liquid). It can be found the critical point, it is the lowest temperature where the substance appears as a unique state, below this temperature the isotherms have a range where liquid and vapour coexist (Sec. 3.1.4).[20]

3.1.2 Superheated Vapour Area

The right side of the saturation bell is known as superheated vapour. This region is characterised by a single state (vapour), temperature and pressure are independent properties of the substance and the properties (h, v and u) are higher than the saturated vapour at a given temperature or pressure.[20]

3.1.3 Compressed Liquid Area

The left side of the saturation bell shows the compressed liquids. This region is characterised by a single state (liquid), the substance properties depend primarily on the temperature and the fact that properties (h, v and u) are lower than the saturated liquid at a given temperature or pressure.[20]

3.1.4 Saturated Liquid–Vapour Mixture Area

Under the saturation bell, during the vaporisation of the substance, both states coexist. Extra information is needed to determine the balance between both states, it is known as quality (χ).

This parameter is defined as:

$$\chi \equiv \frac{m_{vapour}}{m_{vapour} + m_{liquid}} \tag{3.1}$$

Quality could be used as one of the two properties that define a state. Alternatively, density could be used instead of mass to calculate the quality. To determine the properties of a substance in the saturation bell, it is required the property levels as a saturated liquid and as a saturated vapour.[20]

$$\phi = (1 - \chi) \cdot \phi_{liquid} + \chi \cdot \phi_{vapour} \tag{3.2}$$

From the previous equation, it is demonstrated that any property are between both saturated states being quality 0 for a saturated liquid and quality 1 for a saturated vapour.

3.2 HGS Model

As mentioned in the previous chapter, `HGS` is based on the NASA 7 polynomials. These polynomials range from 200K to 6000K, with some exceptions that are explained in the database. Polynomials are used to calculate c_p, h and s assuming that the first two properties depend only on temperature, pressure effect is neglected [35]. For the entropy, HGS assumes that all components are in a gas state, as condensed phases require more extensive analysis and other considerations. Like the Gibbs free energy, the reaction of thermodynamic equilibrium and the calculations reaction, combustion and isentropic expansion depend on the entropy, the gas state is extensive to these parts. It is assumed, for all these points, ideal gas behaviour (Sec. 2.2.2).

Starting from the specific heat (c) defined as $c = \frac{dq}{dT}$ [24], it is replaced dq from the equation of energy conservation (Eq. 2.3):

$$\begin{aligned} \delta Q - \delta W &= dU \\ dq - dw &= du \\ dq &= P\,dv + du \end{aligned} \tag{3.3}$$

$$c = \frac{P\,dv + du}{dT} \tag{3.4}$$

This coefficient results in two other coefficients. The first one is the specific heat at constant volume (c_v), where $dV = 0$:

$$c_v = \left.\frac{du}{dT}\right|_v \tag{3.5}$$

And the specific heat at constant pressure (c_p). For this coefficient, it is combined the specific heat equation (Eq. 3.4) with the definition of enthalpy in the differential form (Eq. 2.8), as $dP = 0$:

$$c_p = \left.\frac{dh}{dT}\right|_p \tag{3.6}$$

3.2.1 Constant Pressure Specific Heat

As mentioned earlier in this chapter, c_p is calculated with the 7-NASA polynomials compiled by ***Burcat*** in the *Third Millennium Ideal Gas and Condensed Phase Thermochemical Database for Combustion with Updates from Active Thermochemical Tables* [35].

The c_p equation is defined with these 5 coefficients as:

$$\frac{c_p}{R} = a_1 + a_2 \cdot T + a_3 \cdot T^2 + a_4 \cdot T^3 + a_5 \cdot T^4 \tag{3.7}$$

3.2.2 Enthalpy

For the enthalpy calculation, c_p equation (Eq. 3.6) is integrated:

$$\begin{gathered} dh = c_p \cdot dT \\ h = \int c_p \cdot dT \end{gathered} \tag{3.8}$$

The enthalpy is defined at the temperature reference framework of 298K, so the enthalpy has two terms: formation enthalpy and c_p integration:

$$h = \Delta h_f^0 + \int_{298}^{T} c_p \cdot dT \tag{3.9}$$

The equation 3.7 is integrated replacing the integral term. Additionally, the sixth coefficient has to be used to reference the enthalpy framework as $a_6 \cdot R = \Delta h_f^0$.

$$\frac{h}{R} = a_6 + a_1 \cdot T + \frac{a_2 \cdot T^2}{2} + \frac{a_3 \cdot T^3}{3} + \frac{a_4 \cdot T^4}{4} + \frac{a_5 \cdot T^5}{5} \tag{3.10}$$

3.2.3 Entropy

For the entropy's calculation, two cases are distinguished, gas and condensables. As it was previously detailed, condensable phases would not be covered by `HGS` for the entropy and deductions. For the gas case, the ideal gas model is assumed for the equation usage. Entropy is calculated from the integration of the second Gibbs' equation (Eq2.9). It is assumed that $V = \frac{RT}{P}$ derivated from the ideal gas assumption (Sec. 2.2.2) and the differential enthalpy definition (Eq. 3.8)

$$\begin{aligned} ds &= \frac{dh}{T} - \frac{V}{T} \cdot dP \\ ds &= \frac{c_p}{T} \cdot dT - \frac{R}{P} \cdot dP \\ \Delta s &= \int \frac{c_p}{T} \cdot dT - \int \frac{R}{P} \cdot dP \end{aligned} \tag{3.11}$$

Two terms are distinguished in the previous equation, the temperature contribution and the pressure contribution. The contribution of the temperature follows the same procedure of the enthalpy, a term of formation and the integrated term referenced to 298 K.

$$s_t = \Delta s_f^0 + \int_{298}^{T} c_p \cdot dT \tag{3.12}$$

In this case, the seventh coefficient is used as $a_7 \cdot R = \Delta S_f^0$:

$$\frac{s^0}{R} = a_7 + a_1 \cdot ln(T) + a_2 \cdot T + \frac{a_3 \cdot T^2}{2} + \frac{a_4 \cdot T^3}{3} + \frac{a_5 \cdot T^4}{4} \tag{3.13}$$

Pressure contribution establishes the reference pressure as 1 bar.

$$\Delta s_p = R \cdot ln(\frac{P}{P_{ref}}) \tag{3.14}$$

The gas entropy calculated as:

$$s = s_t - s_p \tag{3.15}$$

3.2.4 Gibbs' Free Energy

Gibbs' free energy is calculated from the following equation [36]:

$$g = h - T \cdot s \tag{3.16}$$

3.2.5 Specific Gas Constant

Even if the universal gas constant (R) is widely used, some equations, e. g. Mayer's relation (Eq. 3.18), usually require the use of specific gas constant (R_g). [20]

$$R_g = \frac{R}{Mm} \tag{3.17}$$

3.2.6 Constant Volume Specific Heat and Heat Capacity Ratio

Constant volume specific heat (c_v) and heat capacity ratio (γ) are other properties that `HGS` are calculated. Both are related with c_p and they need it in order to calculate the following equations. To determine c_v , it is used Mayer's relation deduced from the ideal gas assumptions (Sec. 2.2.2) and the definition of enthalpy (Eq.2.7):

$$\begin{aligned}
&h \equiv u + P \cdot v \text{ (Enthalpy definition)}\\
&P \cdot v = R_g T \text{ (Specific ideal gas state equation)}\\
&h = u + R_g T\\
&dh = du + R_g dT\\
&c_p = c_v + R_g\\
&R_g = c_p - c_v
\end{aligned} \tag{3.18}$$

The heat capacity ratio is calculated from the following equation [37]:

$$\gamma = \frac{c_p}{c_v} \tag{3.19}$$

3.2.7 Gibbs' Free Energy Minimisation

Gibbs' free energy minimisation (*GEM*) is one of the methods to describe the chemical equilibrium if the temperature and pressure are constant, see Section 2.3.5 to see other methods to achieve chemical equilibrium. This method is based on the following thermodynamic development [20].

From first and second thermodynamic laws applied to a closed system (Eq. 2.23 and Eq. 2.24):

$$\left.\begin{aligned} &\delta Q - P\,dV = dU\\ &dS \geqslant \tfrac{\delta Q}{T}\end{aligned}\right\} dU + P\,dV - T\,dS \leqslant 0 \tag{3.20}$$

The differential form of Gibbs' free energy from equation 3.16 and differential form of the enthalpy definition (Eq. 2.8):

$$(dG)_{T,P} = dH - T\,dS - S\,dT \tag{3.21}$$

$$(dG)_{T,P} = dU + V\,dP + P\,dV - T\,dS - S\,dT \tag{3.22}$$

As pressure and temperature are constant:

$$(dG)_{T,P} = dU + P\,dV - T\,dS \tag{3.23}$$

From the equations 3.20 and 3.23, it is given that $(dG)_{T,P} \leqslant 0$ which means that equilibrium occurs whenever G decreases. The result of this equation is achieved when the equilibrium equals to 0:

$$(dG)_{T,P} = 0 \tag{3.24}$$

When a derivative is equal to zero means that it is a local maximum or minimum, in this case, a minimum. Furthermore, this methodology takes into account the mass conservation of the different elements.

3.2.8 Combustion

Combustion has multiple treatments, constant pressure, constant volume, isentropic, multiple phases of combustion,... In `HGS`, the combustion is calculated in `HGStp`, where it is assumed ideal gas properties, as it uses entropy, carried out at a constant pressure, in equilibrium (dG = 0) and an isentropic behaviour, without any loss of energy ($\Delta H = 0$).

The method carried out by HGS consists in:

- H_{in} calculation
- Chemical Equilibrium at a T* and P
- H_{out} calculation
- Modify T* until $H_{in} = H_{out}$

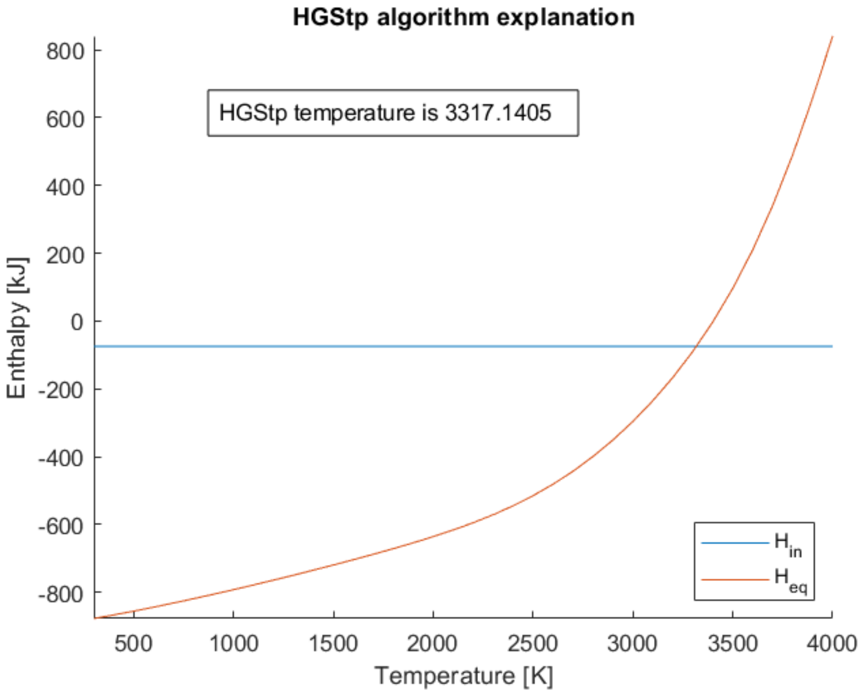


Figure 3.2: HGStp demonstration A.2.3

3.2.9 Isentropic Expansion

In `HGS`, the isentropic expansion is calculated in `HGSisentropic`, where it is assumed ideal gas properties, as it uses entropy, and in equilibrium (dG = 0) .

The method carried out by HGS consist in:

- S_{in} calculation
- Chemical Equilibrium at a T* and P
- S_{out} calculation
- Modify T* until $S_{in} = S_{out}$

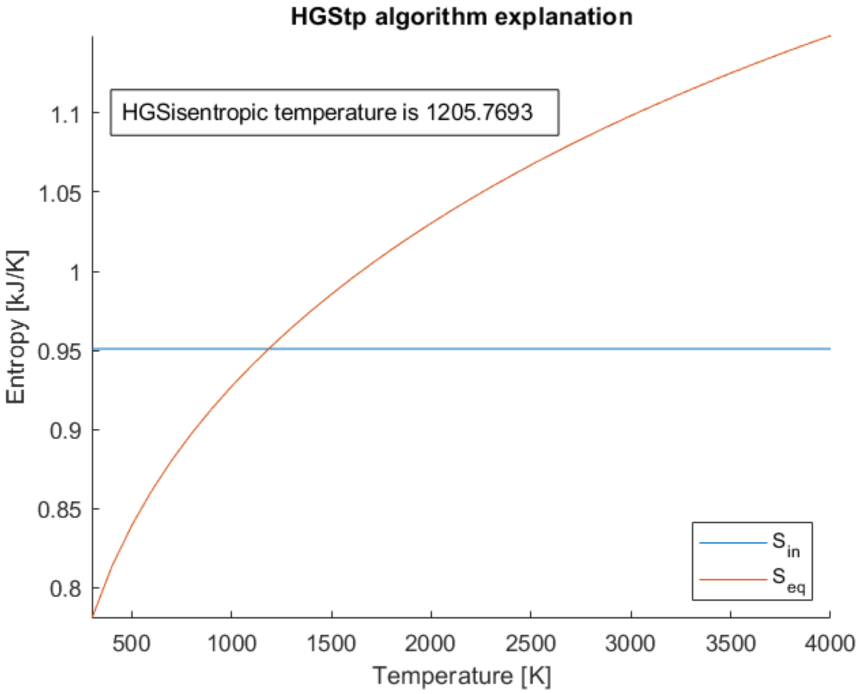


Figure 3.3: HGSisentropic demonstration A.2.3

3.2.10 Mixtures Properties

In the previous sections, related to properties, the calculation for a single species was described. This section will explain the calculation for mixtures based on the single properties.

3.2.10.1 Molecular Mass

The molecular mass of a mixture follows the next equation, as an intensive property [20]:

$$Mm = \frac{\sum_{i=1}^{n} n_i \cdot Mm_i}{\sum_{i=1}^{n} n_i} = \sum_{i=1}^{n} \frac{n_i}{n} \cdot Mm_i \tag{3.25}$$

3.2.10.2 H, S, G, $\mathbf{C_p}$ and $\mathbf{C_v}$

These properties are extensive which means that they are calculated as [20]:

$$\Phi_T = \sum_{i=1}^{n} n_i \cdot \phi_i \tag{3.26}$$

where ϕ can be h, s, g, c_p or c_v and Φ is H, S, G, C_p or C_v.

3.2.10.3 Heat Capacity Ratio, Specific Gas Constant and Sound Speed

These three properties depend on other properties previously calculated so they do not follow any particular mixture rule. They are calculated as follows:

$$\gamma = \frac{C_p}{C_v} \tag{3.27}$$

$$R_g = \frac{R}{Mm} \tag{3.28}$$

$$a = \sqrt{\gamma \cdot R_g \cdot T} \tag{3.29}$$

where C_p, C_v and Mm are the ones calculated from the Section 3.2.10.1 and Section 3.2.10.2

CHAPTER 4

HGS

This chapter explains the upgrades from the previous version and the functions that constitute `HGS`. For each function, their features are described besides their modelled calculations if they were not explained in the previous chapter (Chap. 3) and their units. `HGS` code is intended to be modular based on the idea that each function has to refer to other smaller functions that develop a simpler idea. The general view of the code works while the yellow functions are being used by the user while blues are assisting the yellow ones:

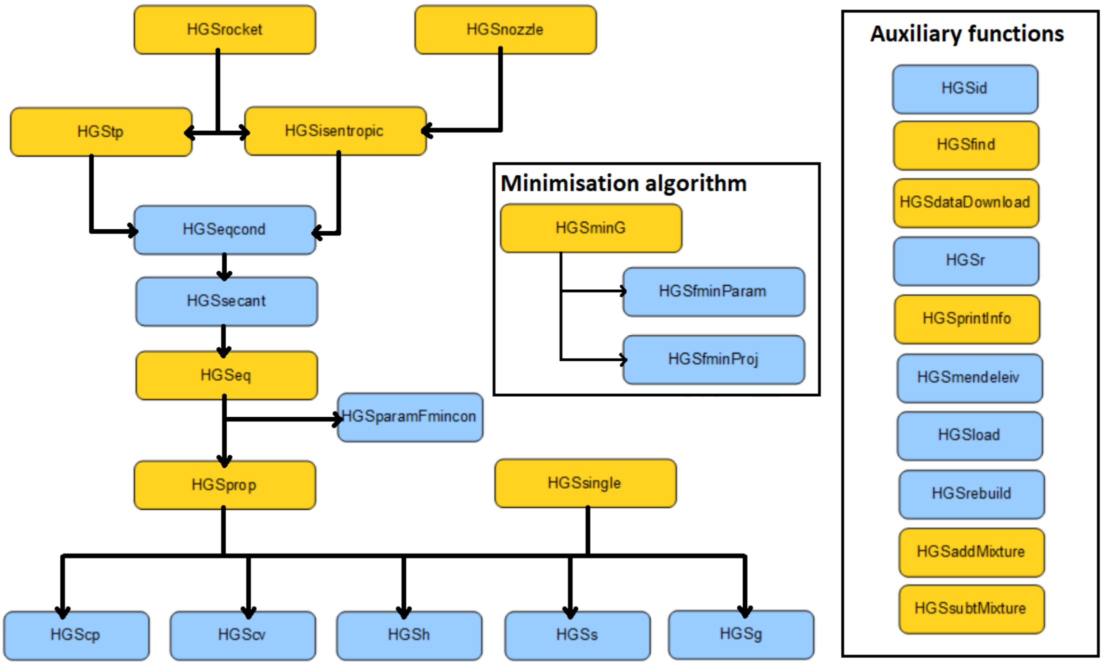
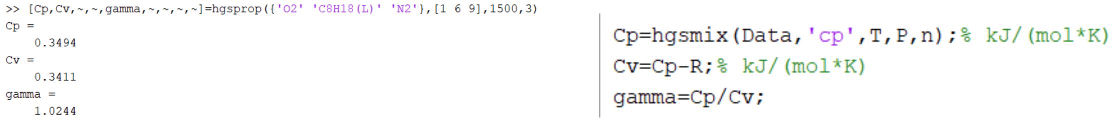


Figure 4.1: Structure of the `HGS` code

The previous HGS has some minor inconveniences that have been solved in this new version of the library, they are listed as follows:

- Incomplete data and from an unknown source.
- Some elements, containing additional state information or special characters, are not eligible for some functions.
- Unclear code that could lead to errors.
- γ has an error for mixtures.
- `hgsTp` has limitations in their inputs that made it useless in most of the cases.

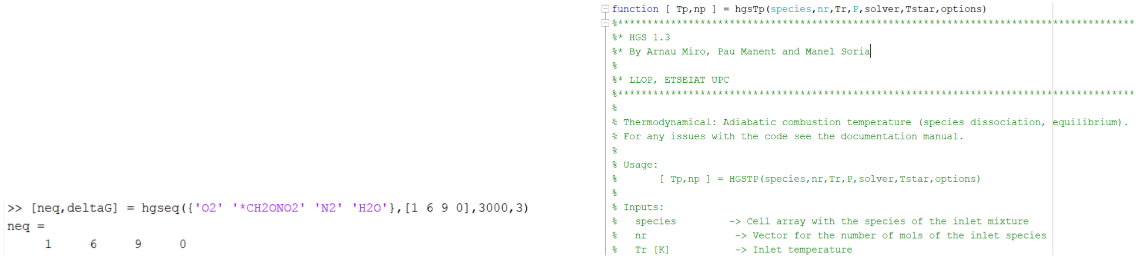


```
>> [Cp,Cv,~,~,gamma,~,~,~,~]=hgsprop({'O2' 'C8H18(L)' 'N2'},[1 6 9],1500,3)
Cp =
    0.3494
Cv =
    0.3411
gamma =
    1.0244
```

```
Cp=hgsmix(Data,'cp',T,P,n);% kJ/(mol*K)
Cv=Cp-R;% kJ/(mol*K)
gamma=Cp/Cv;
```

(a) γ error

(b) Unclear code that leads to γ error



```
>> [neq,deltaG] = hgseq({'O2' '*CH2ONO2' 'N2' 'H2O'},[1 6 9 0],3000,3)
neq =
     1     6     9     0
```

```
function [ Tp,np ] = hgsTp(species,nr,Tr,P,solver,Tstar,options)
%******************************************************************************************
%* HGS 1.3
%* By Arnau Miro, Pau Manent and Manel Soria
%
%* LLOP, ETSEIAT UPC
%******************************************************************************************
%
% Thermodynamical: Adiabatic combustion temperature (species dissociation, equilibrium).
% For any issues with the code see the documentation manual.
%
% Usage:
%       [ Tp,np ] = HGSTP(species,nr,Tr,P,solver,Tstar,options)
%
% Inputs:
%   species         -> Cell array with the species of the inlet mixture
%   nr               -> Vector for the number of mols of the inlet species
%   Tr [K]           -> Inlet temperature
```

(c) Special char. and additional info that lead to errors

(d) Tr allows one inlet temperature that is a special case. Propellants work usually in a different temperature

Figure 4.2: hgs errors [4]

In order to solve these errors from the previous version, some new features and limitations were introduced. The database was downloaded in the past and converted to an Excel file. However, it was not available during this master thesis, probably lost, so a new database had to be found. Original database [35] has a web where the database can be found to be parsed. It is created a function (Sec. 4.25) that downloads the database whenever it is needed. With the creation of this function, it has been detected that the amount of each element atoms present in the species is provided. This new information solves the problems with special characters with appropriate treatment of this data. Unclear code has been solved with modular codification, as explained previously. γ error were solved with this modular code and with the revision of mixtures thermodynamic properties (Sec. 3.2.10). `hgsTp` was rewritten to solve this major inconvenient allowing enthalpy input among the temperature. This new function solves the different temperature of the inlet propellants and, as a side-effect, the vaporisation enthalpy can be used in combustion.

4.1 HGScp

This function returns the c_p of a species following the equation 3.7. It requires the auxiliary function `HGSr`. The output units are $\frac{\mathrm{kJ}}{\mathrm{mol} * \mathrm{K}}$.

HGScp	
Requires:	
HGSr	
Used in:	
HGSsingle	HGSprop

Table 4.1: HGScp function relations

The function variables are:

```
% Inputs:
%----------------------------------------------------------------------
% a --> Burcat's coefficients
% T --> [K] Temperature
%
% Outputs:
%----------------------------------------------------------------------
% Cp --> [kJ/(mol*K)] Constant pressure coefficient
```

Listing 4.1: `HGScp` variables

4.2 HGScv

This function returns the c_v of a species following the equation 3.5. It does not require any additional functions. The output units are $\frac{\mathrm{kJ}}{\mathrm{mol} * \mathrm{K}}$.

HGScv	
Requires:	
HGSr	
Used in:	
HGSsingle	HGSprop

Table 4.2: HGScv function relations

The function variables are:

```
% Inputs:
%----------------------------------------------------------------------
% Cp --> [kJ/(mol*K)] Constant pressure coefficient
```

```
% Mm --> [g/mol] Molar Mass
%
% Outputs:
%---------------------------------------------------------------------
% Cv --> [kJ/(mol*K)] Constant volume coefficient
```

Listing 4.2: `HGScv` variables

4.3 HGSh

This function returns the *h* of a species following the equation 3.10. It requires the auxiliary function `HGSr`. The output units are $\frac{\text{kJ}}{\text{mol}}$.

HGSh	
Requires:	
HGSr	
Used in:	
HGSsingle	HGSprop

Table 4.3: HGSh function relations

The function variables are:

```
% Inputs:
%---------------------------------------------------------------------
% a --> Burcat coefficients
% T --> [K] Temperature
%
% Outputs:
%---------------------------------------------------------------------
% H --> [kJ/mol] Enthalpy
```

Listing 4.3: `HGSh` variables

4.4 HGSs

This function returns the *s* of a species following the equation 3.15. It requires the auxiliary function `HGSr`, also there is a pressure reference of 1 bar. The output units are $\frac{\text{kJ}}{\text{mol}*\text{K}}$.

HGSs	
Requires:	
HGSr	
Used in:	
HGSsingle	HGSprop

Table 4.4: HGSs function relations

The function variables are:

```
% Inputs:
%-------------------------------------------------------------
% a --> Burcat's coefficients
% T --> [K] Temperature
% P --> [bar] Pressure
% state --> State of the species
%
% Outputs:
%-------------------------------------------------------------
% S --> [kJ/(mol*K)]Entropy
```

Listing 4.4: `HGSs` variables

4.5 HGSg

This function returns the g of a species following the equation 3.16. It does not require any additional functions. The output units are $\frac{\text{kJ}}{\text{mol}}$.

HGSg
Requires:

Used in:	
HGSsingle	HGSprop

Table 4.5: HGSg function relations

The function variables are:

```
% Inputs:
%-------------------------------------------------------------
% S --> [kJ/(mol*K)] Entropy
% H --> [kJ/mol] Enthalpy
% T --> [K] Temperature
```

```
%
% Outputs:
%-------------------------------------------------------------------------
% G --> [kJ/mol] Gibbs's Free Energy
```

Listing 4.5: `HGSg` variables

4.6 HGSsingle

This function returns the specific property desired from a single species for temperature and pressure. It requires the modules for the properties calculations mentioned before. The output units are the ones of the previous modules.

HGSsingle		
Requires:		
HGSload	HGSid	HGScp
HGScv	HGSh	HGSs
HGSg		
Used in:		

Table 4.6: HGSsingle function relations

The function variables are:

```
% Inputs:
%-------------------------------------------------------------------------
% species --> String or numbers of species
% property --> Property requested (see below)
% T --> [K] Temperature
% P --> [bar] Pressure
%
% Outputs:
%-------------------------------------------------------------------------
% Res --> Property result
%           Mm [g/mol]
%           Cp [kJ/(mol*K)]
%           Cv [kJ/(mol*K)]
%           h [kJ/mol]
%           s [kJ/(mol*K)]
%           g [kJ/mol]
```

Listing 4.6: `HGSsingle` variables

4.7 HGSprop

This function returns the property or properties desired from a mixture for temperature and pressure following the concepts of Section 3.2.10. It requires the modules for the properties calculations mentioned before and auxiliary functions. The output units are the ones of the previous modules. This function allows to request multiple properties at the same time or in case to do not request any of them, it returns all of them. The output units are:

HGSprop		
Requires:		
HGSload	HGSr	HGSid
HGSrebuild	HGScp	HGScv
HGSh	HGSs	HGSg
Used in:		
HGSrocket	HGSnozzle	HGSisentropic
HGStp	HGSeqcond	HGSeq

Table 4.7: HGSprop function relations

The function variables are:

```
% Inputs:
%-----------------------------------------------------------------------
% species --> String or numbers of species
% n --> [mols] Number of mols per species
% T --> [K] Temperature
% P --> [bar] Pressure
% varargin --> Expected return: 'Mm' 'Cp' 'Cv' 'H' 'S' 'G' 'Rg' 'gamma'
    'a' If it is empty, all the properties will be  return
%
% Outputs:
%-----------------------------------------------------------------------
% varargout --> Property result
%                Mm [g/mol]
%                Cp [kJ/K]
%                Cv [kJ/K]
%                H [kJ]
%                S [kJ/K]
```

```
%                    G [kJ]
%                    Rg [kJ/(kg*K)]
%                    gamma
%                    a [m/s]
```

Listing 4.7: `HGSprop` variables

4.8 HGSparamFmincon

This function returns the `fmincon` parameter for the minimisation of the Gibbs' free energy. It requires `HGSload` to perform the calculation. The parameters are:

- A from $A \cdot x <= b$, in this case empty.
- b from $A \cdot x <= b$, in this case empty.
- Aeq from $A_{eq} \cdot x = b_{eq}$, from mass conservation restriction.
- beq from $A_{eq} \cdot x = b_{eq}$, from mass conservation restriction.
- lb lower boundary, in this case 0 for all the species.
- lb upper boundary, in this case Inf for all the species.

A_{eq} and b_{eq} matrices are formed by the mass conservation of each element, as examplified:

$$\mathrm{H_2} + \mathrm{O_2} + \mathrm{H_2O} + \mathrm{H} + \mathrm{O} + \mathrm{OH}$$

$$n = [3 \quad 5 \quad 0 \quad 0 \quad 0 \quad 0]$$

$$A_{eq} = \begin{bmatrix} 2 & 0 & 2 & 1 & 0 & 1 \\ 0 & 2 & 1 & 0 & 1 & 1 \end{bmatrix} \begin{array}{|l} H \\ O \end{array}$$

$$b_{eq} = A_{eq} \cdot n' = \begin{bmatrix} 3*2 \\ 5*2 \end{bmatrix} = \begin{bmatrix} 6 \\ 10 \end{bmatrix} \begin{array}{|l} H \\ O \end{array}$$

$$\mathrm{CH_4} + \mathrm{O_2} + \mathrm{H_2O} + \mathrm{H_2} + \mathrm{H} + \mathrm{O} + \mathrm{OH} + \mathrm{CO} + \mathrm{CO_2}$$

$$n = [\ 1 \quad 2 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

$$A_{eq} = \begin{bmatrix} 4 & 0 & 2 & 2 & 1 & 0 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 & 1 & 1 & 1 & 2 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{array}{|l} H \\ O \\ C \end{array}$$

$$b_{eq} = A_{eq} \cdot n' = \begin{bmatrix} 4*2 \\ 2*2 \\ 1*1 \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \\ 1 \end{bmatrix} \begin{array}{|l} H \\ O \\ C \end{array}$$

HGSparamFmincon
Requires:
HGSload
Used in:
HGSeqcond

Table 4.8: HGSparamFmincon function relations

The function variables are:

```
% Inputs:
%--------------------------------------------------------------------
% id --> Id of species
% n0 --> [mol] Species mols
%
% Outputs:
%--------------------------------------------------------------------
% A --> fmincon A inequality matrix A*x < b
% b --> fmincon b inequality vector A*x < b
% A --> fmincon Aeq equality matrix Aeq*x = beq
% b --> fmincon beq equality vector Aeq*x = beq
% lb --> fmincon lb lower boundary limit
% ub --> fmincon ub upper boundary limit
```

Listing 4.8: `HGSprop` variables

4.9 HGSeq

This function returns the equilibrium of a mixture at a fixed pressure and temperature, using the minimisation of Gibbs' free energy explained in Section 3.2.7. It requires `HGSparamFmincon` to minimise it with the `Matlab` function `fmincon` and auxiliary functions. The output units are mols and kJ.

HGSeq		
Requires:		
HGSparamFmincon	HGSid	HGSrebuild
HGSprop		
Used in:		
HGSeqcond		

Table 4.9: HGSeq function relations

The function variables are:

```
% Inputs:
%----------------------------------------------------------------
% species --> String or numbers of species
% n0 --> [mol] Initial mixture
% T --> [K] Temperature. Could be a single value or an array.
% P --> [bar] Pressure
% options --> Options for the fmincon
%
% Outputs:
%----------------------------------------------------------------
% species --> Species
% n --> [mol] Final mixture
% Gmin --> [kJ] Minimum Gibbs free energy
```

Listing 4.9: `HGSeq` variables

4.10 HGSsecant

This function returns the point where a function becomes zero by the secant method only for monotone functions. This method is developed as follows:
1 - Check change sign with the limits. If there is not a sign change, there is no zero.
2 - If there is a difference greater than 1500K between limits, add it into a parabola with a third point following the script below:

$$
\begin{aligned}
&a*x^2+b*x+c=0\\
&a=\frac{f(x_1)-\frac{f(x_2)-f(x_3)}{x_2-x_3}*x_1-f(x_3)+x_3*\frac{f(x_2)-f(x_3)}{x_2-x_3}}{x_2-x_3}\\
&b=\frac{f(x_2)-f(x_3)+a*x_3^2-a*x_2^2}{x_2-x_3}\\
&c=f(x_3)-a*x_3^2-b*x_3
\end{aligned}
$$

3 - Secant Method to find zero. However, if the change between values is under a tolerance, it is used bisection over secant.

Bisection method for next step:

$$x_c = \frac{x_2 + x_1}{2}$$

Secant method for next step:

$$x_c = x1 - f(x_1) * \frac{x2 - x1}{f(x_2) - f(x_1)}$$

4 - Repeat until tolerance for x or/and f(x) are satisfied.

HGSsecant
Requires:

Used in:
HGSeqcond

Table 4.10: HGSsecant function relations

The function variables are:

```
% Inputs:
%--------------------------------------------------------------------------
% f --> Function
% n0 --> [mol] Initial mixture
% options --> Structure with the options for the secant method.
%                   .xmin [K] Temperature minimum for the solver;
%                   .xmax [K] Temperature maximum for the solver;
%                   .maxiter Max iterations for the solver;
%                   .epsx Differential T where the solver reaches the
    solution;
%                   .epsy Differential S where the solver reaches the
    solution;
%                   .fchange T difference where secant method is changed
    by bisection method;
%                   .type Select between: 'Frozen' for frozen flow
%                                         'Shifting' for shifting flow
%                   .info Detailed info == 1; No info == 0.
%                   .dTp Improve the velocity with the approximation of
%                   parabola. +- dTp
%
% Outputs:
%--------------------------------------------------------------------------
```

```
% Tp --> [K] Final temperature
% n --> [mol] Final mixture
% flag --> Solver error detection:
%                 1  Solver has reached the solution
%                -1  Solver failed. Maximum iterations
%                -2  Solver failed. Initial sign change not found
```

Listing 4.10: `HGSsecant` variables

4.11 HGSeqcond

This function intermediates between `HGStp` and `HGSisentropic` with `HGSsecant`. It requires `HGSsecant`, `HGSeq` and `HGSprop`. It returns the temperature and the mols of the combustion or isentropic expansion.

HGSeqcond		
Requires:		
HGSload	HGSid	HGSrebuild
HGSprop	HGSeq	HGSsecant
Used in:		
HGStp	HGSisentropic	

Table 4.11: HGSeqcond function relations

The function variables are:

```
% Inputs:
%--------------------------------------------------------------------------
% species --> Species name, ids are also accpeted as entry
% n0 --> [mols] Number of mols of each species
% type --> Entry type. It could be 'H' or 'S'
% V0 --> Entry that should be for type:'H'   V0=H [kJ]
%                                      'S'   V0=S [kJ/K]
% P --> [bar] Mixture pressure
% options --> Structure with the options for the secant method.
%                 .xmin [K] Temperature minimum for the solver;
%                 .xmax [K] Temperature maximum for the solver;
%                 .maxiter Max iterations for the solver;
%                 .epsx Differential T where the solver reaches the
    solution;
%                 .epsy Differential S where the solver reaches the
    solution;
```

```
%                    .fchange T difference where secant method is
%                              changed by bisection method;
%                    .type Select between: 'Frozen' for frozen flow
%                                          'Shifting' for shifting flow
%                    .info Detailed info == 1; No info == 0.
%                    .dTp Improve the velocity with the approximation of
%                    parabola. +- dTp
%
% Outputs:
%-------------------------------------------------------------------------------
% Tp --> [K] Exit temperature
% n --> [mols] Species resultant mols
% species --> String or numbers of species
% flag --> Solver error detection:
%                    1  Solver has reached the solution
%                   -1  Solver failed. Maximum iterations
%                   -2  Solver failed. Initial sign change not found
```

Listing 4.11: `HGSeqcond` variables

4.12 HGStp

This function calculates the temperature and the mixture of a combustion as explained in Section 3.2.8. It requires `HGSeqcond` and auxiliary functions. The outputs units are K and mols.

HGStp		
Requires:		
HGSload	HGSid	HGSrebuild
HGSprop	HGSeqcond	
Used in:		
HGSrocket		

Table 4.12: HGStp function relations

The function variables are:

```
% Inputs:
%-------------------------------------------------------------------------------
% species --> String or numbers of species
% n0 --> [mols] Number of mols of each species
% type --> Entry type that defines the state of the input.
```

```
%        It can be 'T' or 'H'
% V0 --> Entry that should be for type:'T'   V0=T [K] input temperature
%                                     'H'   V0=H [kJ] input enthalpy
% P --> [bar] Pressure
% options --> Structure with the options for the secant method.
%                  .xmin [K] Temperature minimum for the solver;
%                  .xmax [K] Temperature maximum for the solver;
%                  .maxiter Max iterations for the solver;
%                  .epsx Differential T where the solver reaches the
   solution;
%                  .epsy Differential S where the solver reaches the
   solution;
%                  .fchange T difference where secant method is changed
   by bisection method;
%                  .type Select between: 'Frozen' for frozen flow
%                                        'Shifting' for shifting flow
%                  .info Detailed info == 1; No info == 0.
%                  .dTp Improve the velocity with the approximation of
%                  parabola. +- dTp
%
% Outputs:
%-------------------------------------------------------------------------
% Tp --> [K] Exit temperature
% n --> [mols] Species resultant mols
% species --> String or numbers of species
% flag --> Solver error detection:
%                  1  Solver has reached the solution
%                 -1  Solver failed. Maximum iterations
%                 -2  Solver failed. Initial sign change not found
```

Listing 4.12: `HGStp` variables

4.13 HGSisentropic

This function determines the temperature and the mixture of an isentropic expansion as explained in Section 3.2.9. It requires `HGSeqcond` and auxiliary functions. The outputs units are K and mols.

HGSisentropic		
Requires:		
HGSload	HGSid	HGSrebuild
HGSprop	HGSeqcond	
Used in:		
HGSrocket	HGSnozzle	

Table 4.13: HGSisentropic function relations

The function variables are:

```
% Inputs:
%-------------------------------------------------------------------------
% species --> String or numbers of species
% n0 --> [mols] Number of mols of each species
% T0 --> [K] Initial temperature
% P0 --> [bar] Inlet pressure
% P1 --> [bar] Exit pressure
% options --> Structure with the options for the secant method.
%                  .xmin [K] Temperature minimum for the solver;
%                  .xmax [K] Temperature maximum for the solver;
%                  .maxiter Max iterations for the solver;
%                  .epsx Differential T where the solver reaches the solution;
%                  .epsy Differential S where the solver reaches the solution;
%                  .fchange T difference where secant method is changed by bisection method;
%                  .type Select between: 'Frozen' for frozen flow
%                                        'Shifting' for shifting flow
%                  .info Detailed info == 1; No info == 0.
%                  .dTp Improve the velocity with the approximation of
%                  parabola. +- dTp
%
% Outputs:
%-------------------------------------------------------------------------
% Tp --> [K] Exit temperature
% n --> [mols] Species resultant mols
% species --> String or numbers of species
% v2 --> [m/s] Velocity of the mixture
% M2 --> [Mach] Mach of the mixture
```

```
% flag --> Solver error detection:
%                  1  Solver has reached the solution
%                 -1  Solver failed. Maximum iterations
%                 -2  Solver failed. Initial sign change not found
```

Listing 4.13: `HGSisentropic` variables

4.14 HGSnozzle

This function calculates the outlet properties of a nozzle besides the mixture and temperature. It requires `HGSisentropic` and auxiliary functions.

HGSnozzle	
Requires:	
HGSisentropic	HGSprop
Used in:	

Table 4.14: HGSnozzle function relations

The function variables are:

```
% Inputs:
%----------------------------------------------------------------------
% species --> String or numbers of species
% n0 --> [mols] Number of mols of each species
% T0 --> [K] Initial temperature
% P0 --> [bar] Inlet pressure
% P1 --> [bar] Exit pressure
% A --> [m^2] Exit area
% options --> Structure with the options for the secant method.
%                  .xmin [K] Temperature minimum for the solver;
%                  .xmax [K] Temperature maximum for the solver;
%                  .maxiter Max iterations for the solver;
%                  .epsx Differential T where the solver reaches the
    solution;
%                  .epsy Differential S where the solver reaches the
    solution;
%                  .fchange T difference where secant method is changed
    by bisection method;
%                  .type Select between: 'Frozen' for frozen flow
```

```
%                                   'Shifting' for shifting flow
%
% Outputs:
%-----------------------------------------------------------------
% Tp --> [K] Exit temperature
% n --> [mols] Species resultant mols
% species --> String or numbers of species
% F --> [N] Thrust
% Isp --> [s^-1]Specific impulse, g0 = 9.807 m/s^2
% flag --> Solver error detection:
%                1 Solver has reached the solution
%               -1 Solver failed. Maximum iterations
%               -2 Solver failed. Initial sign change not found
```

Listing 4.14: `HGSnozzle` variables

4.15 HGSrocket

This function calculates the properties of a rocket with a combustion chamber and nozzle. It requires `HGSisentropic` and `HGSisentropic` and auxiliary functions. The outputs are, where temperature and mixture are from the outlet of the combustion chamber and nozzle outlet:

- Temperature [K]
- Mixture [Mols]
- Thrust [N]
- Specific impulse [$\frac{1}{s}$]
- Velocity [$\frac{m}{s}$]
- Mach []

HGSrocket		
Requires:		
HGStp	HGSisentropic	HGSprop
Used in:		

Table 4.15: HGSrocket function relations

The function variables are:

```
% Inputs:
%-----------------------------------------------------------------
```

```
% species --> String or numbers of species
% n0 --> [mols] Number of mols of each species
% type -->
% V0 --> [K] Initial temperature
% P0 --> [bar] Inlet pressure
% P1 --> [bar] Nozzle exit pressure
% A --> [m^2] Nozzle exit area
% options --> Structure with the options for the secant method.
%                    .xmin [K] Temperature minimum for the solver;
%                    .xmax [K] Temperature maximum for the solver;
%                    .maxiter Max iterations for the solver;
%                    .epsx Differential T where the solver reaches the
    solution;
%                    .epsy Differential S where the solver reaches the
    solution;
%                    .fchange T difference where secant method is
%                             changed by bisection method;
%                    .type Select between: 'Frozen' for frozen flow
%                                          'Shifting' for shifting flow
%
% Outputs:
%-------------------------------------------------------------------
% Tp --> [K] Exit temperature
% n --> [mols] Species resultant mols
% species --> String or numbers of species
% param --> Exit parameter: param(1) = v [m/s]
%                           param(2) = Mach
%                           param(3) = Thrust [N]
%                           param(4) = Isp
% flag --> Solver error detection:
%                    1  Solver has reached the solution
%                   -1  Solver failed. Maximum iterations
%                   -2  Solver failed. Initial sign change not found
```

Listing 4.15: `HGSrocket` variables

4.16 HGSid

This auxiliary function speeds up `HGS` as it converts species into ids, that are easily found in the database. Most of the main `HGS` functions contain this one to convert species and pass it through different levels.

HGSid		
Requires:		
HGSload		
Used in:		
HGSsingle	HGSprop	HGSeq
HGSisentropic	HGStp	HGSeqcond
HGSprintInfo	HGSaddMixture	HGSsubtMixture
HGSrebuild		

Table 4.16: HGSid function relations

The function variables are:

```
% Inputs:
%------------------------------------------------------------
% name --> String or numbers of species
%
% Outputs:
%------------------------------------------------------------
% id --> Species id
```

Listing 4.16: `HGSid` variables

4.17 HGSfind

This auxiliary function allows the user to find if an species exist in the database or to find species with a concrete element. This function has an auxiliary input to return the complete name of a species that have the same name, more details in Section 4.26.

HGSfind
Requires:
HGSload
Used in:

Table 4.17: HGSfind function relations

The function variables are:

```
    % Inputs:
```

```
%-------------------------------------------------------------------------
% name --> Name to find
% complete --> Returns the complete name if complete = 1; No complete = 0.
%
% Outputs:
%-------------------------------------------------------------------------
% Command Window Print
```

Listing 4.17: `HGSfind` variables

4.18 HGSload

This auxiliary function loads the database to the current workspace as a global variable (Sec. 4.26). All the main and some auxiliary functions use it as they need the database to work.

HGSload
Requires:

Used in:		
HGSdataDownload	HGSrebuild	HGSid
HGSsubtMixture	HGSprintInfo	HGStp
HGSaddMixture	HGSfind	HGSsingle
HGSisentropic	HGSeqcond	HGSprop
HGSparamFmincon		

Table 4.18: HGSload function relations

The function variables are:

```
% Inputs:
%-------------------------------------------------------------------------
%
% Outputs:
%-------------------------------------------------------------------------
% HGSdata is load in the global workspace
```

Listing 4.18: `HGSload` variables

4.19 HGSr

This auxiliary function loads the ideal gas constant to the current workspace as a global variable (Eq. 2.2). Some of the `HGS` function need it to provide properties calculations.

HGSr		
Requires:		

Used in:		
HGScp	HGScv	HGSh
HGSs	HGSprop	

Table 4.19: HGSr function relations

The function variables are:

```
% Inputs:
%---------------------------------------------------------------
%
% Outputs:
%---------------------------------------------------------------
% R is load in the global workspace
```

Listing 4.19: `HGSload` variables

4.20 HGSprintInfo

This auxiliary function shows the information of a species that includes name, id, composition and molar mass.

HGSprintInfo	
Requires:	
HGSload	HGSid
Used in:	

Table 4.20: HGSprintInfo function relations

The function variables are:

```
% Inputs:
%---------------------------------------------------------------
% species --> A single species
%
% Outputs:
%---------------------------------------------------------------
% Command Window Print
```

Listing 4.20: `HGSprintInfo` variables

4.21 HGSmendeleiev

This auxiliary function returns a structure with the periodic table molar mass [38] to calculate it when the database is created (Sec. 4.25). Some of the `HGS` function need it to provide properties calculations.

HGSmendeleiev
Requires:

Used in:
HGSdataDownload

Table 4.21: HGSmendeleiev function relations

The function variables are:

```
% Inputs:
%----------------------------------------------------------------------
%
% Outputs:
%----------------------------------------------------------------------
% Mendeleiev --> [g/mol] Molar mass of each substance of the periodic
   table
```

Listing 4.21: `HGSmendeleiev` variables

4.22 HGSrebuild

This auxiliary function breaks up a mixture in their species with their respective mols. Some of the `HGS` use it as part of their code to be able to perform their calculation.

HGSrebuild	
Requires:	
HGSload	HGSid
Used in:	

Table 4.22: HGSrebuild function relations

The function variables are:

```
% Inputs:
%----------------------------------------------------------------------
% species --> String or numbers of species
% n0 --> [mols] Mols
```

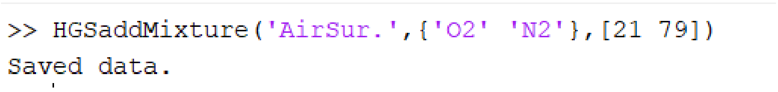
```
%
% Outputs:
%-----------------------------------------------------------------------
% species --> String of species
% n0 --> [mols] Mols
```

Listing 4.22: `HGSrebuild` variables

4.23 HGSaddMixture

This auxiliary function adds a mixture to the database (Sec. 4.26). These mixtures have to be from species that are contained in the database, e.g.:

`HGSdata` contains two mixtures pre-established: RP1 surrogate and Air surrogate. For the Air case, it was decided to create the surrogate based in the simplest combination O_2 and N_2. The surrogate is 21% and 79% respectively.



```
>> HGSaddMixture('AirSur.',{'O2' 'N2'},[21 79])
Saved data.
```

Figure 4.3: How to add a mixture

HGSaddMixture	
Requires:	
HGSload	HGSid
Used in:	

Table 4.23: HGSaddMixture function relations

The function variables are:

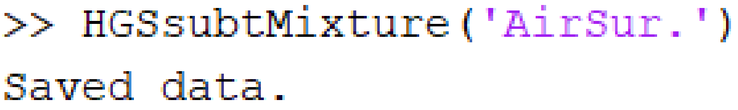
```
% Inputs:
%-----------------------------------------------------------------------
% name --> Name of the mixture f.e. 'Air'
% species --> species from the mixture f.e. {,...}
% percent --> Percentage of each species in the mixture [,...]
%
% Outputs:
%-----------------------------------------------------------------------
% HGSdata update
```

Listing 4.23: `HGSaddMixture` variables

4.24 HGSsubtMixture

This auxiliary function substracts a mixture to the database (Sec 4.26). This mixtures have to be from species that are contained in the database,e. g. :

In case any mixture created will not be useful anymore:



```
>> HGSsubtMixture('AirSur.')
Saved data.
```

Figure 4.4: How to delete a mixture

<table>
<tr><th colspan="2">HGSsubtMixture</th></tr>
<tr><td colspan="2"><i>Requires:</i></td></tr>
<tr><td>HGSload</td><td>HGSid</td></tr>
<tr><td colspan="2"><i>Used in:</i></td></tr>
</table>

Table 4.24: HGSsubtMixture function relations

The function variables are:

```
% Inputs:
%-------------------------------------------------------------------------
% name --> Name of the mixture f.e. 'Air'
%
% Outputs:
%-------------------------------------------------------------------------
% HGSdata update
```

Listing 4.24: `HGSsubtMixture` variables

4.25 HGSdataDownload

This auxiliary function creates the `HGS` database downloading it from the mirror page of ***Burcat*** [35]. The database is downloaded in a txt from the webpage and parsed by this function. This function requires `HGSmendeliev` to calculate the molar mass of each species. The txt downloaded (DATA_7_coef.txt) has some patterns to parse it but some exceptions have to be taken into account to create it (Sec. 4.26). As a property of this database, it rewrites the contents when you download it so the mixtures are not affected.

HGSdataDownload	
Requires:	
HGSload	HGSmendeliev
Used in:	

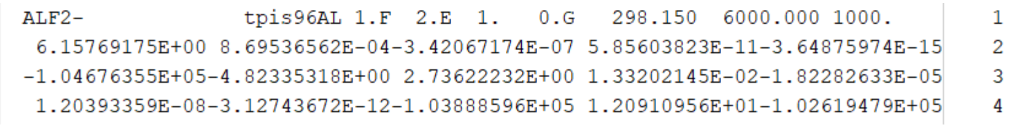
Table 4.25: HGSdataDownload function relations

The function variables are:

```
% Inputs:
%---------------------------------------------------------------------------
%
% Outputs:
%---------------------------------------------------------------------------
% Creates HGSdata.mat
```

Listing 4.25: `HGSdataDownload` variables

4.26 HGS database



```
ALF2-             tpis96AL 1.F  2.E  1.   0.G   298.150  6000.000 1000.        1
 6.15769175E+00 8.69536562E-04-3.42067174E-07 5.85603823E-11-3.64875974E-15    2
-1.04676355E+05-4.82335318E+00 2.73622232E+00 1.33202145E-02-1.82282633E-05    3
 1.20393359E-08-3.12743672E-12-1.03888596E+05 1.20910956E+01-1.02619479E+05    4
```

Figure 4.5: Standard species format

Standard species format:

- First row:
  - The 1-18 characters refer to the name that will be converted to a shorter name.
  - The 19-24 characters refer to the provenance that is not stored.
  - The 25-44 characters refer to the elements present in the species.
    - The 25-29 characters refer to the first element.
      - 25-26 element's name.
      - 26-28 number of atoms.
      - 29 should be a point but there are some exceptions.
    - The 30-34 characters refer to the second element.
      - 30-31 element's name.
      - 32-33 number of atoms.
      - 34 should be a point but there are some exceptions.
    - The 35-39 characters refer to the third element.

      - 35-36 element's name.
      - 37-38 number of atoms.
      - 39 should be a point but there are some exceptions.
    - The 40-44 characters refer to the fourth element.
      - 40-41 element's name.
      - 42-43 number of atoms.
      - 44 should be a point but there are some exceptions.
  - The 45 character refers to the state
    - G for gas state
    - L for liquid state
    - S for solid state
  - The 49-75 characters refer to the limits of the coefficients' calculation
    - The 46-55 refer to the temperature lower limit.
    - The 56-65 refer to the temperature upper limit.
    - The 66-75 refer to the temperature where it changes between low temperature coefficients and high temperature coefficients.
- Second row:
  - The 1-15 characters refer to the highest temperature coefficient.
  - The 16-30 characters refer to the second highest temperature coefficient.
  - The 31-45 characters refer to the third highest temperature coefficient.
  - The 46-60 characters refer to the fourth highest temperature coefficient.
  - The 60-75 characters refer to the fifth highest temperature coefficient.
- Third row:
  - The 1-15 characters refer to the sixth highest temperature coefficient.
  - The 16-30 characters refer to the seventh highest temperature coefficient.
  - The 31-45 characters refer to the lowest temperature coefficient.
  - The 46-60 characters refer to the second lowest temperature coefficient.
  - The 60-75 characters refer to the third lowest temperature coefficient.
- Fourth row:
  - The 1-15 characters refer to the fourth lowest temperature coefficient.
  - The 16-30 characters refer to the fifth lowest temperature coefficient.
  - The 31-45 characters refer to the sixth low temperature coefficient.
  - The 46-60 characters refer to the seventh lowest temperature coefficient.
  - The 60-75 characters refer to formation enthalpy at 298 K but its not stored.

Some exceptions have been found to the previous scheme that is listed below:

- Air as a species is in the database and breaks multiple rules so it is not stored.

- CH2O HCOH triplet add 1 the first row character values.

- Some species have not points in the elements so there is added 1 to the number return character.

The database of `HGS`, named `HGSdata.m`, has the following shape:

HGSdata	
.name{}	Row of species names
.state{}	Row of species state
.lim()	Matrix with temperature limits. First column lower limit Second column change between High and low coefficients Third column upper limit
.HV()	Matrix with high temperature coefficients
.LV()	Matrix with low temperature coefficients
.Mm()	Row with the species molar mass
.ena{}	Row with species elements
.nat{}	Row with species number of elements
.nameback{}	Row of species complete names
.comb{}	Row of mixture names
.cspec{}	Row of mixture species
.cper{}	Row of mixture species percentage

Table 4.26: `HGSdata` structure

As mentioned before names are converted to shorter names so it is easier to work with them. However, it is distinguished between two cases: additional useless data and usefully data. Data that is not necessary is deleted from the name as it could be the laboratory where the coefficients come from or the name of the organic species (Carbon or Toluene). Other name data is converted to be shorter like the state. Some of the names formulations have a errors as they save letters in an uppercase when it should be lowercase, species with chlorine and aluminium elements essentially. A small script checks these sensitive cases and changes them if needed.

Most of the species names are the formula, but some cases have their common names like RDX, Mustard gas or Jet-A. Finally, if there are some species with the same formulation (e. g. C_2H_2 that can be vinylidene or acetylene), mainly organic species, they are shortened, conserving the formulation but adding an roman number between brackets. If the complete information is needed, it can be checked with `HGSfind` with the additional input as it checks the name in the `HGSdata.nameback`.

4.27 Minimisation Algorithm

As mentioned before, an algorithm has been developed to avoid the dependence of the `fmicon` `Matlab` function. The minimisation algorithm is not integrated in the `HGS` scheme (Fig. 4.1), as it can not be called by function like `HGStp` or `HGSisentropic`. `HGSminG` is the main function of the algorithm, and equivalent to `HGSeq` as both have the same inputs and purposes. In this section, the new algorithm are described while in the chapter 7 the results are presented.

The minimisation algorithm has been developed in three separate functions so it can be reused in the future by other students to improve the algorithm or to be used in other topics.

- `HGSminG`: main function for minimisation algorithm equivalent to `HGSeq`.
- `HGSfminParam`: creates the inequality matrices for the minimisation with constrains.
- `HGSfminProj`: the minimisation algorithm.
- `HGSgenerate`: generates the complete mixture of species from the reduced mixture.

4.27.1 Constrains

The Gibbs' free energy minimisation, thus the thermodynamic equilibrium, is subjected to two types of constraints as it is exposed in Section 4.8. For this algorithm, equality restrictions derived from the mass conservation have to be converted to inequality constraints. To do this, the functions `HGSgenerate` and `HGSfminParam` are needed.

As a prelude of the minimisation algorithm, the number of unknown (the number of species in this case) is reduced to the minimum. In this case, it is reduced by one for each mass equality constrain meaning that there is a species depending on the rest for each element present in the equilibrium. As an example:

$$\mathrm{H_2 + O_2 + H_2O + H + O + OH}$$

This equilibrium has six species with two elements so the reduced number of species will be four. Before determining which species will become free of degrees, it is required to create the mass constrain matrices as in Section 4.8. The A_{eq} (see Sec. 4.8) is separated in two matrices A_1 with the dependant species and A_2 with the free species, turning out:

$$A_1 \cdot x_1 + A_2 \cdot X_2 = b_{eq} \tag{4.1}$$

So,

$$x_1 = A_1^{-1} \cdot (b_{eq} - A_2 \cdot x_2) \tag{4.2}$$

That will be the main calculation of the nested function `HGSgenerate`.

To determine which species is dependant on the rest, it is ensured that there are at least one more species with each element in the free species matrix. In the example of this section, the first two elements are the dependant, so the equation 4.1 of this example will be:

$$\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \cdot x_1 + \begin{bmatrix} 2 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \cdot x_2 = b_{eq}$$

After minimising the number of unknowns, it is needed to convert mass equality restrictions to inequalities. From the equation 4.1:

$$\begin{aligned} &x1 + A1^{-1} \cdot A2 \cdot x2 = A1^{-1} \cdot beq \\ &x1 + Q \cdot x2 = P \end{aligned} \tag{4.3}$$

Part of the inequality matrix ($A \cdot x_2 \leqslant b$) is that any of the reduced unknowns should be zero or greater so:

$$-I \cdot x_2 \leqslant 0 \rightarrow \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \cdot x_2 \leqslant 0 \tag{4.4}$$

In the limit case, where $x_1 = 0$ (Eq. 4.3), it is obtained so that the maximum value of the reduced unknowns are:

$$Q \cdot x_2 \leqslant P \tag{4.5}$$

Combining both matrices, inequality matrices are obtained for the minimisation algorithm:

$$\begin{bmatrix} -I \\ Q \end{bmatrix} \cdot x_2 \leqslant \begin{bmatrix} 0 \\ P \end{bmatrix}$$

All this development is done in the function `HGSparamFmin`.

4.27.2 Minimisation Algorithm

The algorithm can be found in the function `HGSfminProj` with nested functions inside it. The scheme of the algorithm, based on a gradient descent algorithm, is as follows:

1. Check if the initial point is in the boundaries.
    - If not, error message.
    - Else, continue.
2. Calculate gradient.
3. Check the restrictions.
    - If there are affected restrictions, transform the gradient
4. Calculate next point.
5. Repeat until tolerance or maximum iteration is satisfied

4.27.2.1 Gradient Calculation

The gradient has to be numerical for this algorithm as the analytic gradient can not be done. This numerical gradient is done with a central difference unless one of the limits is violated so forward/backward difference is done. This gradient has to be done by each one of the unknowns: Central difference

$$\nabla F = \frac{F(x+\epsilon) - F(x-\epsilon)}{2 \cdot \epsilon} \tag{4.6}$$

Forward difference

$$\nabla F = \frac{F(x+\epsilon) - F(x)}{\epsilon} \tag{4.7}$$

Backward difference

$$\nabla F = \frac{F(x) - F(x-\epsilon)}{\epsilon} \tag{4.8}$$

4.27.2.2 Restriction Check

The restriction has to be checked in order to adapt the gradient to not violate the boundaries. When the current point is over a boundary, three cases are possible: *i*) Next step goes away from the boundaries, *ii*) Next step violates boundaries and *iii*) Next step goes parallel to the boundaries.. The nested function `activeRestrictions` check if the point is at a distance minor or equal to epsilon, and if it ignores the effect of the first case converting as a non-active.

4.27.2.3 Projection of the Gradient on the Boundaries

If there is an active restriction, the gradient has to be projected over the boundaries. The boundaries active are passed to the nested function `projectd`, under the matrix G. To not violate the boundaries, $G \cdot \nabla F = 0$ has to been respected. Again, G has to be split into two parts:

$$G_1 \cdot d_1 + G_2 \cdot d_2 = 0 \tag{4.9}$$

$$d_1 = G_1^{-1} \cdot G_2 \cdot d_2 \tag{4.10}$$

where G_1 is a non-singular matrix part of the G matrix.

A new base formed from the subspace determined by the active restriction matrix (n), is created using the previous equation.

```
% Base for the subspace of linear independent vectors
B=zeros(n,n-na);
for ii=1:n-na
    x2=zeros(n-na,1);
```

```
    x2(ii)=1;
    x1=-(G1^-1)*G2*x2;
    % Comutate again to the original distribution of G
    x(colsg1) = x1;
    x(colsg2) = x2;
    B(:,ii)=x';
end
```

Listing 4.26: New base of linear independent vectors

Before the projection can be done, this linear independent base has to be orthogonalised. In this code, the base is orthonormalised by the Gram-Schmidt method. The next step is the projection over each component of the base:

$$\sum_{i=1}^{n} proj_{B_i}(d) = \sum_{i=1}^{n} \frac{\overrightarrow{d} \cdot \overrightarrow{B_i}}{|\overrightarrow{B_i}|^2} \overrightarrow{B_i} = \sum_{i=1}^{n} (\overrightarrow{d} \cdot \overrightarrow{B_i}) * \overrightarrow{B_i} \tag{4.11}$$

Gram-Schmidt algorithm is applied as follows [39]:

$$\begin{aligned} u_i &= v_i - \sum_{j=1}^{i-1} proj_{u_j}(v_i) \\ B_i &= \frac{u_i}{|u_i|} \end{aligned} \tag{4.12}$$

4.27.2.4 Next Step Calculation

With a corrected gradient, the gradient descent scheme can be applied:

$$x_n = x_1 - \lambda \cdot \nabla F \tag{4.13}$$

Where λ is a value between 0 and the maximum lambda value before leaving the feasible zone.

The first step to determinate λ is to calculate the value for all the boundaries and pick up the minimum positive value. This calculation is done by the nested function `lambdaLimit` and `lambdaStop` following:

$$\begin{aligned} \lambda_i &= \frac{b_i - A_i \cdot x0}{A_i \cdot \nabla F} \\ \lambda &= min(\lambda_i > 0) \end{aligned} \tag{4.14}$$

With this, it is determined that the lambda value is comprised between zero and the lambda previously calculated. For the next step calculation of the lambda value that minimises F is done with a bisection method. This method requires also a gradient calculation towards λ. The gradient of $F(x0-\lambda\nabla F)$ is calculated numerically with the same rules exposed in Section 4.27.2.1. The Bisection method is carried out as: *i*) calculates the midpoint between two, *ii*) detects the sign of the gradient of this new point, *iii*) and replaces the point with the same sign until tolerance is achieved.

```
% Check the value
fa=f(a);
fb=f(b);

% Check if it changes sign the derivation
f1a=f1(a);
f1b=f1(b);

% If there is no change in the sign  the step is one of the limits
if sign(f1a) == sign(f1b)
    if fa<fb
        xmin=a;
    else
        xmin=b;
    end
    return
end

% If not , we look for the minimum
for i=1:itemax
    c=(a+b)/2;
    if b-a<epsilon
        xmin=c;
        break;
    end
    f1c=f1(c);
    if f1a*f1c>0
        a=c;
        f1a=f1c;
    else
        b=c;
        f1b=f1c;
    end
end
```

Listing 4.27: Bisection method

4.27.3 HGSminG

This function is equivalent to `HGSeq` but instead of `fmincon` uses the previously explain minimisation method. To be analogue to the main HGS function inputs and outputs are identical.

HGSminG		
Requires:		
Nested function HGSgenerate	HGSid	HGSfminProj
HGSprop	HGSparamFmin	
Used in:		

Table 4.27: HGSminG function relations

The function variables are:

```
% Inputs:
%----------------------------------------------------------------
% species --> String or id of species
% n0 --> [mols] Number of mols per species
% T --> [K] Temperature
% P --> [bar] Pressure
%
% Outputs:
%----------------------------------------------------------------
% species --> String or numbers of species
% n --> [mol] Final mixture
% Gmin --> [kJ] Minimum Gibbs free energy
```

Listing 4.28: `HGSminG` variables

4.27.4 HGSfminParam

This function is analogue to `HGSparamFmin` as it returns the constraints to the minimisation algorithm. Moreover, it returns parameters to the reduction of the number of variables.

HGSfminParam	
Requires:	
HGSid	HGSid
Used in:	
HGSminG	

Table 4.28: HGSfminParam function relations

The function variables are:

```
% Inputs:
%-----------------------------------------------------------------------------
% species --> String or numbers of species
% n --> [mol] Initial mixture
%
% Outputs:
%-----------------------------------------------------------------------------
% idnew --> species ids strings commute so A1 is a squared non-singular
%               matrix
% x02 --> [mol] Reduced mols vector corresponding to A2
% A1 --> Squared non-singular matrix from Aeq
% A2 --> Matrix resultant from the non-used rows of Aeq  by A1
% beq --> Matrix from the equality mass restrictions (Aeq*x = beq)
% A --> Inequality extended matrix with mass restrictions (A*x <= b)
% b --> Inequality extended matrix with mass restrictions (A*x <= b)
```

Listing 4.29: `HGSfminParam` variables

4.27.5 HGSfminProj

This function minimises a function with inequalities constraints, similar to `fmincon Matlab's` function. It contains nested functions to develop small parts of the code:

- `gradf`: calculates the gradient of the function
- `activeRestrictions`: check the active restrictions
- `projectd`: project in the subspace of the active restrictions
- `minDir`: next step calculation
- `gramschmidt`: Gram-Schmidt algorithm
- `lambdaLimit`: calculates the distance to the boundaries
- `lambdaStop`: returns the minimum lambda distance
- `fmin1D`: minimise a the function with the descent gradient
- `gradflambda`: calculates the gradient of the function with the gradient descent

HGSfminProj
Requires:
Nested functions
Used in:
HGSminG

Table 4.29: HGSfminProj function relations

The function variables are:

```
% Inputs:
%----------------------------------------------------------------------------
% x0 --> Initial value
% F --> Function to minimise
% A --> Inequality matrix with mass restrictions (A*x <= b)
% b --> Inequality matrix with mass restrictions (A*x <= b)
% options --> Structure with the options for the gradient descend method
%             subjected to inequality constrains.
%                 .maxite Maximum number of iterations for the gradient
%                  descent
%                 .tolx Tolerance for exit for the gradient descent;
%                 .info Detailed info == 1; No info == 0.
%                 .epsGrad Epsilon for the gradient
%                 .epsRes Epsilon to accept the active restriction
%                 .epsLam Epsilon to lambda modification
%                 .eps1D Epsilon to gradient of the function with
   gradient
%                 descent
%
%
% Outputs:
%----------------------------------------------------------------------------
% x --> Minimum of the function F
```

Listing 4.30: `HGSfminProj` variables

CHAPTER 5

INIST

This chapter will explain the minor updates from the previous versions of `INIST` besides the documentation required for its use. In contrast to `HGS`, `INIST` database origin is known but, from time to time, NIST changes its webpage [3], where the data come from so, it is required to download a permanent database. Previous iterations of the code had a script to parse the data, however, some adjustments have been done to get a more complete database and some new species have been added. Also, a new script has been added to plot isobars. Finally, the main function has been changed to load internally the database.

5.1 INIST database

The database downloaded from NIST is divided into three parts for each element; data parameters, the saturation bell (Fig. 3.1(b)) and the isobars. The parameters are basic information data as the name, the molar mass, critical pressure and temperature and the id, are necessary to access the NIST database. The saturation bell and isobars contain the same properties information, even though isobars have a constant pressure while saturation pressure varies. Properties are:

- Temperature (K)
- Density ($\frac{\mathrm{kg}}{\mathrm{m}^3}$)
- Enthalpy ($\frac{\mathrm{kJ}}{\mathrm{kg}}$)
- Internal energy ($\frac{\mathrm{kJ}}{\mathrm{kg}}$)
- Sound speed ($\frac{\mathrm{m}}{\mathrm{s}}$)
- Specific volume ($\frac{\mathrm{m}^3}{\mathrm{kg}}$)
- Entropy ($\frac{\mathrm{J}}{\mathrm{g \cdot K}}$)
- c_p ($\frac{\mathrm{J}}{\mathrm{g \cdot K}}$)
- c_v ($\frac{\mathrm{J}}{\mathrm{g \cdot K}}$)
- Viscosity ($\mathrm{Pa \cdot s}$)
- Thermal cond. ($\frac{\mathrm{W}}{\mathrm{m \cdot K}}$)
- Phase (Liquid or Vapour)

In case it is required to use the data directly from the database, the following table could be used as a guideline:

Acces	Propierty	Acces	Propierty
.name	name	.sl	Entropy for liquid saturated
.idcast	Id for NIST search	.sv	Entropy for vapour saturated
.MM	Molar mass	.cpl	Const. pressure coef. for liquid saturated
.Tsat	Saturated temperature	.cpv	Const. pressure coef. for vapour saturated
.Psat	Saturated pressure	.cvl	Const. volume coef. for liquid saturated
.rl	Density for liquid saturated	.cvv	Const. volume coef. for vapour saturated
.rv	Density for vapour saturated	.al	Sound speed for liquid saturated
.vl	Specific volume for liquid saturated	.av	Sound speed coef. for vapour saturated
.vv	Specific volume for vapour saturated	.mul	Viscosity for liquid saturated
.ul	Internal energy for liquid saturated	.muv	Viscosity for vapour saturated
.uv	Internal energy for vapour saturated	.kl	Thermal conductivity for liquid saturated
.hl	Enthalpy for liquid saturated	.kv	Thermal conductivity for vapour saturated
.hv	Enthalpy for vapour saturated	.isoP	Access to isobars
.Ref	Reference of the data		

.isoP			
Acces	**Propierty**	**Acces**	**Propierty**
{}.P	Isobar pressure	{}.T	Temperature isobar range
{}.r	Density isobar range	{}.v	Specific volume isobar range
{}.u	Internal energy isobar range	{}.h	Enthalpy isobar range
{}.cp	Const. pressure coef. isobar range	{}.cv	Const. volume coef. isobar range
{}.s	Entropy isobar range	{}.a	Sound speed isobar range
{}.mu	Viscosity isobar range	{}.k	Thermal conductivity isobar range
{}.phase	Phase isobar range		

Table 5.1: Database from INIST

As mentioned before, a script parses the previous properties from NIST webpage [3]. First of all, the idcast from this thesis interest are summarised below. As it is impossible to get information on every single point, a selection has been made to interpolate between points. Each species have their minimums and maximums so the numbers provided are to take the limits whatever they are.

Name	**Idcast**	**Reference**
H_2O	C7732185	Self-Reference
N_2	C7727379	Self-Reference
H_2	C1333740	NBP
CO	C630080	NBP
CO_2	C124389	IIR
N_2O	C124389	NBP
CH_4O	C67561	NBP
CH_4	C74828	NBP
C_2H_6	C74840	NBP
C_2H_4	C74851	NBP
C_3H_8	C74986	IIR
C_3H_6	C115071	IIR
C_3H_4	C74997	IIR
C_4H_{10}	C106978	IIR
C_5H_{12}	C109660	NBP
C_6H_{14}	C110543	NBP
C_6H_{12}	C110827	NBP
C_6H_6	C71432	NBP
NH_3	C7664417	Self-Reference
He	C7440597	NBP
O_2	C7782447	Self-Reference
R134a	C811972	IIR

Table 5.2: Name, idcast and reference relation

For the saturation bell, the values are $T_{min} = 0$ and $T_{max} = 1000$, with a step of 1 degree. For the rest, the isobars are (MPa): from 0.001 to 0.01,steps of 0.001, from 0.02 to 0.1, steps of 0.01, from 0.2 to 10, steps of 0.1, from 10.5 to 100, steps of 0.5, from 101 to 250, steps of 1 and from 255 to 500, steps of 5. The values are $T_{min} = 0$ and $T_{max} = 5000$, with a step of 2 degrees.

5.2 INIST

`INIST` is a library that was not intended to be modular as `HGS`. It is based in `XSteam Matlab` library [40], this library calculates the properties of vapour steam, interesting for thermodynamic calculations. `INIST` extends this idea to the cryogenic state of other species. Inside the function multiple options are given:

- Special functions
- Non-saturated properties as function of temperature and pressure
- Saturated liquid as function of
  - Pressure
  - Temperature
- Saturated vapour as function of
  - Pressure
  - Temperature

The function can be called with multiple inputs depending on the needs `INIST(1,2,...)`. First, the entry has to be the species name (Tab. 5.2) or *’Database’*. If it is a species, the second input is the property required within the previous list of possibilities. Following inputs are the minimum properties needed for the calculation.

5.2.1 Special functions

If a species is the first input, some special function can be requested. They are tabulated below with their duty:

Special functions	
'tcrit'	Returns the critical temperature of the species
'pcrit'	Returns the critical pressure of the species
'vcrit'	Returns the critical volume of the species
'MM'	Returns the molar mass of the species
'tsat_p'	Returns the saturation temperature at pressure
'psat_t'	Returns the saturation pressure at temperature
'minp'	Returns the minimum isobar available
'maxp'	Returns the maximum isobar available
'mint'	Returns the minimum temperature available
'maxt'	Returns the maximum temperature available
'isobars'	Returns a vector with the available isobars
'add_p'	Disabled. Add new isobars to the database

Table 5.3: `INIST` special functions

5.2.2 Saturated Vapour/Liquid as Function of Temperature

For the case where data is requested for the saturation bell, or in the bell applying equation 3.2 that requires the quality (Eq. 3.1). Additionally, the only requirement is the temperature. It is checked that temperature is between minimum saturation temperature and critic temperature, and for the cases of c_p, c_v and a it is different to the critical temperature. After, if the temperature is not tabulated, a linear interpolation (`Matlab` function `interp1`) is carried.

$$\phi = \phi_1 + \frac{T - T_1}{T_2 - T_1} \cdot (\phi_2 - \phi_1) \tag{5.1}$$

where ϕ is any of the properties of the list in Section 5.1.

5.2.3 Saturated Vapour/Liquid as Function of Pressure

For the case where data required for the saturation bell, or in the bell applying equation 3.2 that requires quality (Eq. 3.1). Additionally, the only requirement is the pressure. It is checked that pressure is between minimum saturation pressure and critic pressure, and for the cases of c_p, c_v and a it is different to critic temperature. After, if the pressure is not tabulated, a linear interpolation (`Matlab` function `interp1`) is carried.

$$\phi = \phi_1 + \frac{P - P_1}{P_2 - P_1} \cdot (\phi_2 - \phi_1) \tag{5.2}$$

where ϕ is any of the properties of the list in Section 5.1.

5.2.4 Non-Saturated Properties as Function of Temperature and Pressure

Out of the saturation bell (Fig. 3.1(b)), it needs two properties to determine the state of the fluid, generally temperature and pressure. Any of the properties of the list in Section 5.1 can be estimated with the temperature and pressure as follows:

1. Check that temperature is within the limits.
2. Check that pressure is within the limits.
3. Check that pressure and temperature are not saturated values.
4. Find the isobar minor or equal to the pressure
5. Linear interpolation of the property in the isobar below and next isobar using the temperature (Eq. 5.1).
6. Linear interpolation with the previous values using the pressure (Eq. 5.2).

```
val1=myi(dat.isoP{p1}.T,dat.isoP{p1}.property,T);
   val2=myi(dat.isoP{p1+1}.T,dat.isoP{p1+1}.property,T);

ret=interp1([dat.isoP{p1}.P(1) dat.isoP{p1+1}.P(1)],[val1 val2],p);

    function y=myi(X,Y,x) % interpolates avoiding l-v phase change singularity
        if x<X(1)
            error('Too low: Splendiferous error');
        end
        if x>X(end)
            error('Too high: Splendiferous error');
        end
        q=find(diff(X)==0);
        if isempty(q)
            y=interp1(X,Y,x);
            return
        end
        if (x<=X(q))
            y=interp1(X(1:q),Y(1:q),x);
        else
            y=interp1(X(q+1:end),Y(q+1:end),x);
        end
    end
```

Listing 5.1: 's_pt' case

`INIST` offers the possibility to calculate the temperature in terms of entropy and pressure as it is explained below:

1. Check that pressure is within the limits.
2. Check that pressure is below critic pressure.
    a) Check if the entropy is in the saturated bell, between saturated liquid and saturated vapour, so the temperature is saturated.
3. Use fsolve to find the temperature as the function is @(x) INIST(varargin1,'t_sp',p,x)-s

```
checkp(p);
if (p<dat.Pcrit)
    sl=INIST(varargin{1},'sl_p',p);
    sv=INIST(varargin{1},'sv_p',p);
    tsat=INIST(varargin{1},'tsat_p',p);
    if s>=sl && s<=sv
        ret=tsat;
    else
        eq=@(x) INIST(varargin{1},'s_pt',p,x)-s;
        options=optimset('Display','none');
        ret=fsolve(eq,tsat,options);
    end
else
    eq=@(x) INIST(varargin{1},'s_pt',p,x)-s;
    options=optimset('Display','none');
    ret=fsolve(eq,dat.Tcrit*1.1,options);
end
```

Listing 5.2: 't_sp' case

CHAPTER 6

TURBOMACHINERY ANALYSIS

Analysis of the turbomachinery is required to check the correct behaviour of the `INIST` and `HGS` codes. Moreover, this topic is an interesting field to be reviewed in a class. Multiple types of turbomachinery cycles are depending on the aim mission, the stage of the cycle, propellants,... This chapter collects the examples of rocketry cycles from other authors and compare results with the codes resultant.

Before creating a complete analysis of a rocket cycle turbomachinery, all components are tested separately to ensure that thermodynamic codes and rocket machinery codes work correctly. The machinery present in the schemes are:

- Pump
- Turbine
- Combustion Chamber
- Nozzle
- Tank
- Mixer
- Heat exchanger
- Valves

Valves, mixer and tanks are not interesting from a thermodynamic point of view as their analyses are simple. Nozzles and heat exchangers are interesting to be analysed but in the schemes analysed (An. A.3) they are connected between them, so any analysis developed will not be accurate or realistic. Most of the nozzle contains an injector of propellant or a heat exchanger in the throat to reduce the temperature. Therefore, it misses information like the equilibrium in the nozzle, whether it is frozen or shifting, and the geometric information. The following sections will analyse pumps, turbines and combustion chamber.

6.1 Pump

Multiple analyses can be done on a pump depending on the data known from it. For this thesis, it is assumed that input and output are known, so work and efficiency are calculated and compared. These calculation are done by a separated code of `HGS` and `INIST`, using them as a library for thermodynamic data (Sec. A.2.3). There are four cases of pumping: *i)* CH_4, *ii)* H_2, *iii)* RP_1 and *iv)* O_2.

The analysis is performed as follows:

1 - Enthalpy calculation of the inlet and outlet using `HGS` or `INIST` depending on the conditions and propellant.

$$\begin{aligned} H_i &= HGSprop(...,'H') \quad [\text{kJ}] \quad \text{For } HGS \\ H_i &= INIST() \cdot \frac{n * 1000}{Mm} \quad [\text{kJ}] \quad \text{For } INIST \end{aligned} \tag{6.1}$$

2 - Work calculation, derived from the closed system energy conservation (Eq. 2.3) with no heat losses, no kinetic variation and no potential energy variation:

$$\begin{aligned} &\cancelto{0}{\dot{Q}} - \dot{W} = \dot{m} \cdot \Delta h \\ &\dot{W} = -\dot{m} \cdot \Delta h \\ &\dot{W} = \dot{m} \cdot (h_{in} - h_{out}) \end{aligned} \tag{6.2}$$

3 - Calculation of inlet entropy as isentropic work conserves the pressure ($S_{in} = S_{out}$).

$$\begin{aligned} S_{in} &= HGSprop(...,'S') \quad [\frac{\text{kJ}}{\text{K}}] \quad \text{For } HGS \\ S_{in} &= INIST() \quad [\frac{\text{kJ}}{\text{mol} \cdot \text{K}}] \quad \text{For } INIST \end{aligned} \tag{6.3}$$

4 - Calculation of the outlet temperature for an isentropic process, thus enthalpy calculation of the mentioned temperature.

$$\begin{aligned} T &= fsolve(@(T_x)HGSprop(...,T_x,'S') - S_{in} == 0) \quad [\text{K}] \quad \text{For } HGS \\ T &= INIST(species,'t_ps',P,S_{in}) \quad [\text{K}] \quad \text{For } INIST \end{aligned} \tag{6.4}$$

5 - Isentropic work calculation and efficiency as:

$$\begin{aligned} \dot{W}_s &= \dot{m} \cdot (h_{in} - h_{outs}) \\ \eta &= \frac{\dot{W}_s}{\dot{W}} \end{aligned} \tag{6.5}$$

The following subsection will show the analysed pump parameters and the results obtained from the previously explained analysis. In Section 6.1.12, an overall analysis will be explained.

6.1.1 SE-22 engine O2

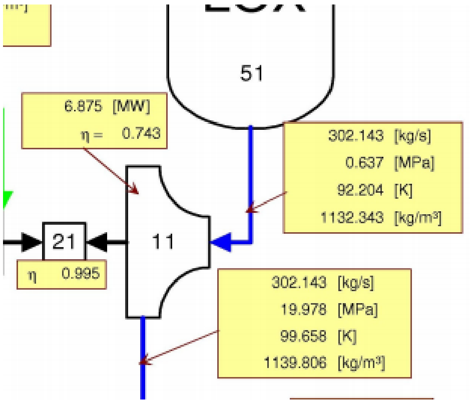


Figure 6.1: SE-22 engine O_2 pump (Fig.A.12)

SE-22 engine O_2		
	Inlet:	***Outlet:***
Species	O_2	
Mols [mols]	9.442e+03	
Temperature [K]	92.204	99.658
Pressure [bar]	6.37	199.78
	Paper	***Calculated***
η	0.743	0.743
W [kJ/s]	6.875e+03	6.875e+03

Table 6.1: Calculated parameters for SE-22 engine O_2 pump

The calculations seem to agree with the data obtained from the papers.

6.1.2 SE-22 Engine H2

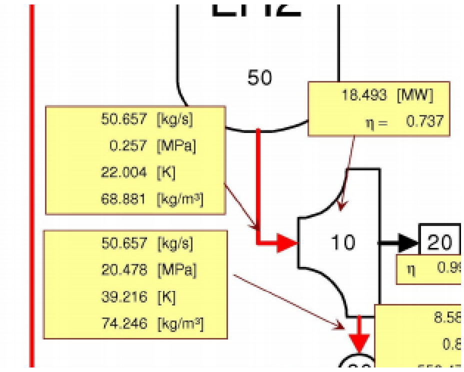


Figure 6.2: SE-22 engine H_2 pump (Fig.A.12)

SE-22 engine H_2		
	Inlet:	***Outlet:***
Species	H_2	
Mols [mols]	2.5127e+04	
Temperature [K]	22.004	39.216
Pressure [bar]	2.57	204.78
	Paper	***Calculated***
η	0.737	0.7040
W [kJ/s]	1.8493e+04	1.8387e+04

Table 6.2: Calculated parameters for SE-22 engine H_2 pump

The calculations seem to agree with the data obtained from the papers.

6.1.3 Vinci Engine O2

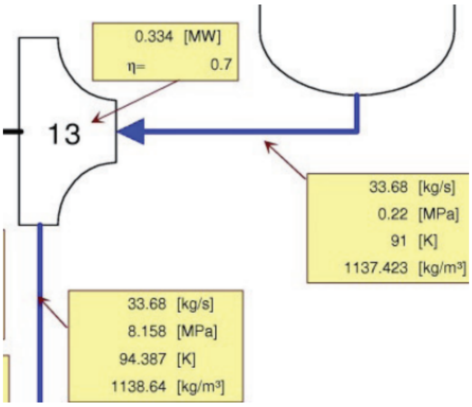


Figure 6.3: Vinci engine O_2 pump (Fig.A.9)

Vinci engine O_2		
	Inlet:	***Outlet:***
Species	O_2	
Mols [mols]	1.0525e+03	
Temperature [K]	91	94.387
Pressure [bar]	2.2	81.58
	Paper	***Calculated***
η	0.7	0.699
W [kJ/s]	3.34e+02	3.3431e+02

Table 6.3: Calculated parameters for Vinci engine H_2 pump

The calculations seem to agree with the data obtained from the papers.

6.1.4 Vinci Engine H2

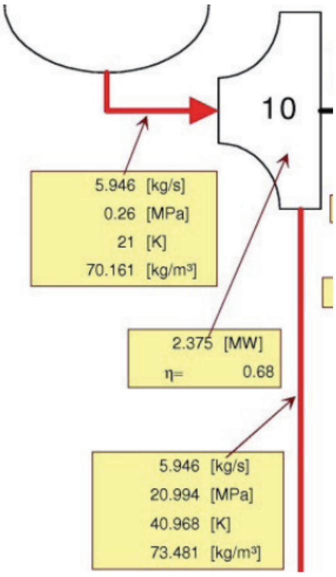


Figure 6.4: VINICI engine H_2 pump (Fig.A.9)

VINCI engine H_2		
	Inlet:	***Outlet:***
Species	H_2	
Mols [mols]	2.9494e+03	
Temperature [K]	21	40.968
Pressure [bar]	2.6	209.9
	Paper	***Calculated***
η	0.68	0.684
W [kJ/s]	2.375e+03	2.360e+03

Table 6.4: Calculated parameters for VINCI engine H_2 pump

The calculations seem to agree with the data obtained from the papers.

6.1.5 SE 12 Engine CH4 (13)

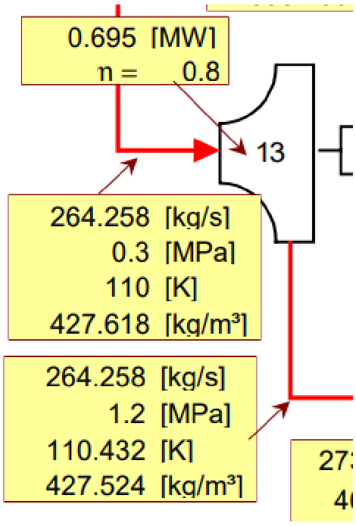


Figure 6.5: SE-12 engine CH_4 pump (13) (Fig.A.14)

SE-12 engine CH_4 (13)		
	Inlet:	***Outlet:***
Species	CH_4	
Mols [mols]	1.647e+04	
Temperature [K]	110	110.432
Pressure [bar]	3	12
	Paper	***Calculated***
η	0.8	0.749
W [kJ/s]	695	745.098

Table 6.5: Calculated parameters for SE-12 engine CH_4 pump (13)

The calculations seem to agree with the data obtained from the papers.

6.1.6 SSME Block II H2 (10)

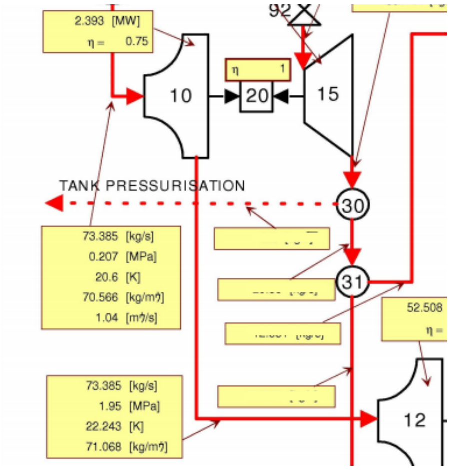


Figure 6.6: SSME Block II H_2 pump (10) (Fig.A.11)

SSME Block II engine H_2 (10)		
	Inlet:	***Outlet:***
Species	H_2	
Mols [mols]	3.640e+04	
Temperature [K]	20.6	22.243
Pressure [bar]	2.07	19.5
	Paper	***Calculated***
η	0.75	0.756
W [kJ/s]	2.393e+03	2.366e+03

Table 6.6: Calculated parameters for SSME Block II engine H_2 pump (10)

The calculations seem to agree with the data obtained from the papers.

6.1.7 SSME Block II H2 (12)

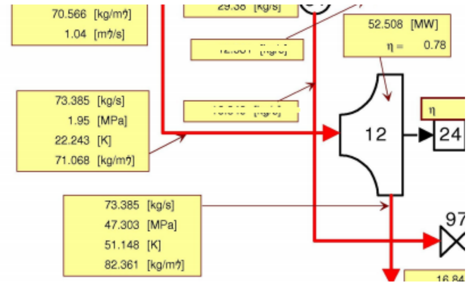


Figure 6.7: SSME Block II H_2 pump (12) (Fig.A.11)

SSME Block II engine H_2 (12)		
	Inlet:	***Outlet:***
Species	H_2	
Mols [mols]	3.6401e+04	
Temperature [K]	22.243	51.148
Pressure [bar]	19.5	473.03
	Paper	***Calculated***
η	0.78	0.782
W [kJ/s]	5.2508e+04	5.2362e+04

Table 6.7: Calculated parameters for SSME Block II engine H_2 pump (12)

The calculations seem to agree with the data obtained from the papers.

6.1.8 SSME Block II O2 (11)

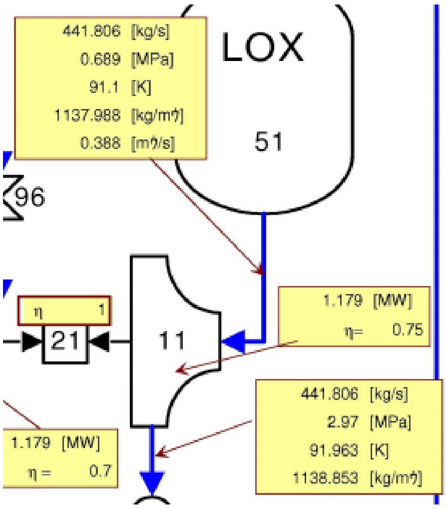


Figure 6.8: SSME Block II O_2 pump (11) (Fig.A.11)

SSME Block II engine O_2 (11)		
	Inlet:	***Outlet:***
Species	O_2	
Mols [mols]	1.380e+04	
Temperature [K]	91.1	91.963
Pressure [bar]	6.89	29.7
	Paper	***Calculated***
η	0.75	0.75
W [kJ/s]	1.179e+03	1.179e+03

Table 6.8: Calculated parameters for SSME Block II engine O_2 pump (11)

The calculations seem to agree with the data obtained from the papers.

6.1.9 SSME Block II O2 (13)

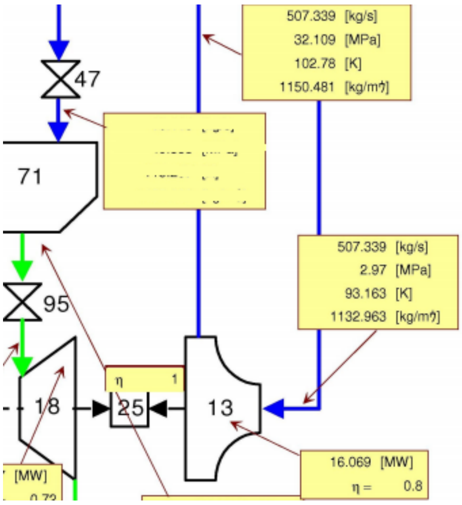


Figure 6.9: SSME Block II O_2 pump (13) (Fig.A.11)

SSME Block II engine O_2 (13)		
	Inlet:	***Outlet:***
Species	O_2	
Mols [mols]	1.585e+04	
Temperature [K]	93.163	102.78
Pressure [bar]	29.7	321.09
	Paper	***Calculated***
η	0.8	0.799
W [kJ/s]	1.6069e+04	1.6069e+04

Table 6.9: Calculated parameters for SSME Block II engine O_2 pump (13)

The calculations seem to agree with the data obtained from the papers.

6.1.10 SSME Block II O2 (14)

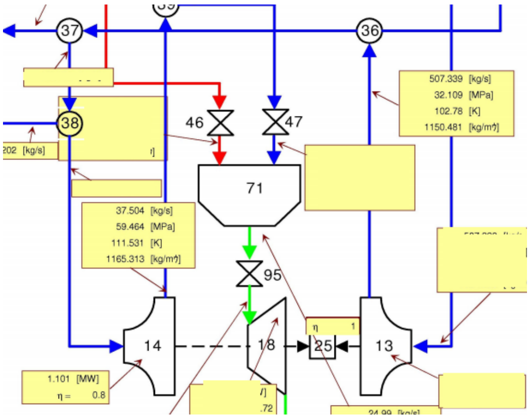


Figure 6.10: SSME Block II O_2 pump (14) (Fig.A.11)

SSME Block II engine O_2 (14)		
	Inlet:	***Outlet:***
Species	O_2	
Mols [mols]	1.172e+03	
Temperature [K]	1027.82	111.531
Pressure [bar]	321.09	594.64
	Paper	***Calculated***
η	0.8	0.80
W [kJ/s]	1.101e+03	1.101e+03

Table 6.10: Calculated parameters for SSME Block II engine O_2 pump (14)

The calculations seem to agree with the data obtained from the papers.

6.1.11 RD-180 Engine O2

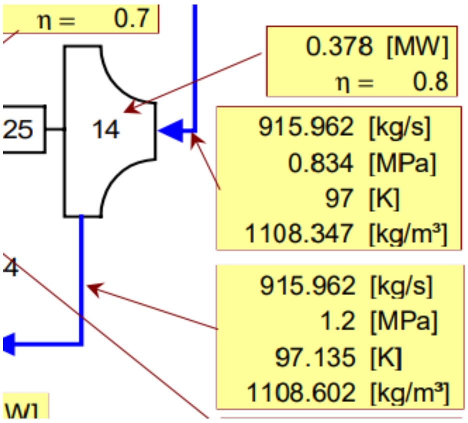


Figure 6.11: RD-180 engine O_2 pump (Fig.A.13)

RD-180 engine O_2		
	Inlet:	***Outlet:***
Species	O_2	
Mols [mols]	2.8625e+04	
Temperature [K]	97	97.135
Pressure [bar]	8.34	12
	Paper	***Calculated***
η	0.7	0.80
W [kJ/s]	378	377.744

Table 6.11: Calculated parameters for RD-180 engine O_2 pump

The calculations seem to agree with the data obtained from the papers.

6.1.12 Pump Overall Analysis

First of all, the RP1 pumps are excluded from the analysis because of the fact that kerosene is in a liquid state. `HGS` is not recommended when the mixture is in a liquid state and when the effect of the pressure might be not neglected.

For the rest of the pump, the following figures illustrate the results:

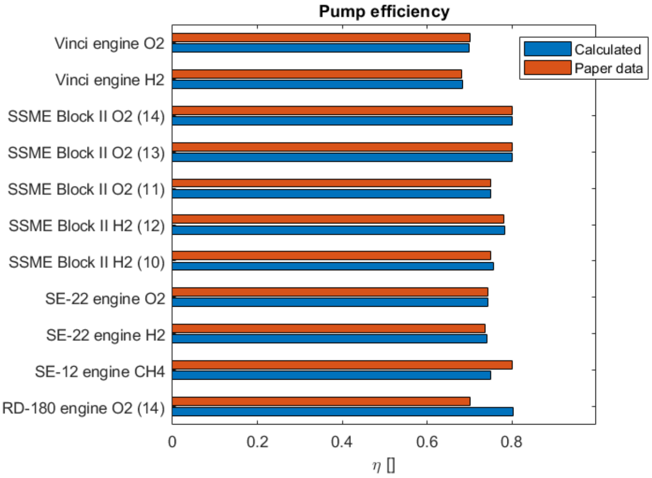


Figure 6.12: Efficiency comparison between papers data and calculation

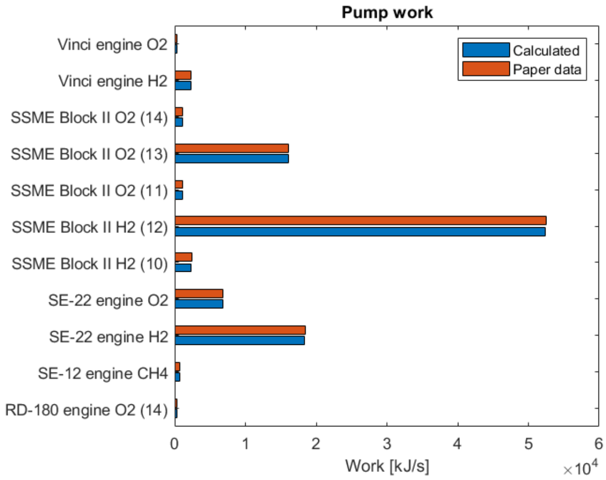


Figure 6.13: Work comparison between papers data and calculation

All the pumps reflected are simulated with `INIST` and their results seem to be accurate with the data obtained from the authors. The worst case is from the pump of RD-180 (6.1.11) where the error is 0.1 in the efficiency coefficient.

6.2 Combustion Chamber

The combustion chamber is tested with `HGStp` with both input propellant properties known. It is assumed that both propellants enter at the same pressure even if the scheme specifies otherwise. This is because pressure should be similar to avoid problems in the injector. If the inlet enthalpies are calculated with `HGS` the effect of the pressure is neglected. Additionally, the scheme reflects an efficiency that will not be taken into account.

The analysis is performed as follows:

1 - Enthalpy calculation of the propellants to be burned using `HGS` or `INIST` depending on the conditions and propellants.

$$\begin{aligned} H_i &= HGSprop(...,'H') \quad \text{[kJ]} \quad \text{For } HGS \\ H_i &= INIST() \cdot \frac{n * 1000}{Mm} \quad \text{[kJ]} \quad \text{For } INIST \end{aligned} \tag{6.6}$$

1.1 - For RP-1, vaporisation enthalpy is also calculated and subtracted from the enthalpy calculation.

2 - Apply `HGStp` to calculate the combustion temperature and outlet mixture. Assume that this reaction occurs at the outlet pressure without losses

3 - Calculation of the gas mass flow using `HGSprop`.

The following subsection will show the analysed combustion chamber parameters and the results obtain from the previously explained analysis. In Section 6.2.13, an overall analysis will be made.

6.2.1 Vinci Engine Combustion Chamber (LH2-LOX)

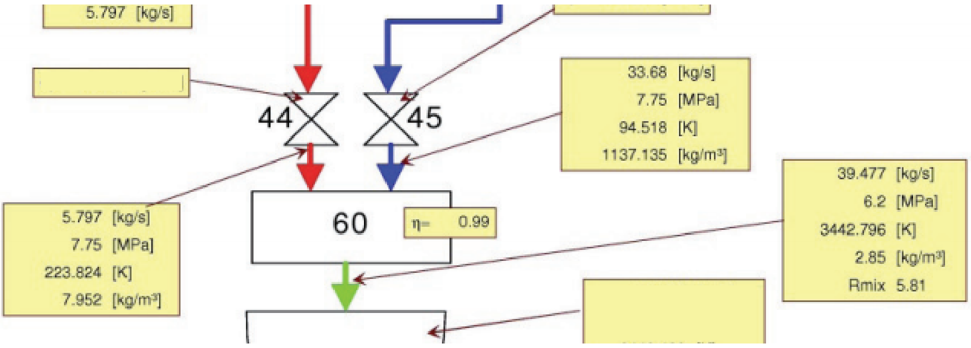


Figure 6.14: Vinci engine combustion chamber (LH2-LOX) (Fig.A.9)

Vinci engine combustion chamber (LH2-LOX)		
	Propelant 1	***Propelant 2***
Species	H_2	O_2
Mols [mols]	2.875e+03	1.053e+03
Temperature [K]	223.824e+02	94.518
Pressure inlet [bar]	77.5	
	Combustion	
Pressure outlet [bar]	695	
Species	[H_2 O_2 H_2O H O OH]	
Mols [mols]	[8.1612e+02 , 11.119 , 1.922e+03 , 1.251e+02 , 12.276 , 1.476e+02]	
	Paper	***Calculated***
Temperature [K]	3442.796	3525.952

Table 6.12: Calculated parameters for Vinci engine combustion chamber (LH2-LOX)

6.2.2 SE-22 Engine Main Combustion Chamber (LH2-LOX)

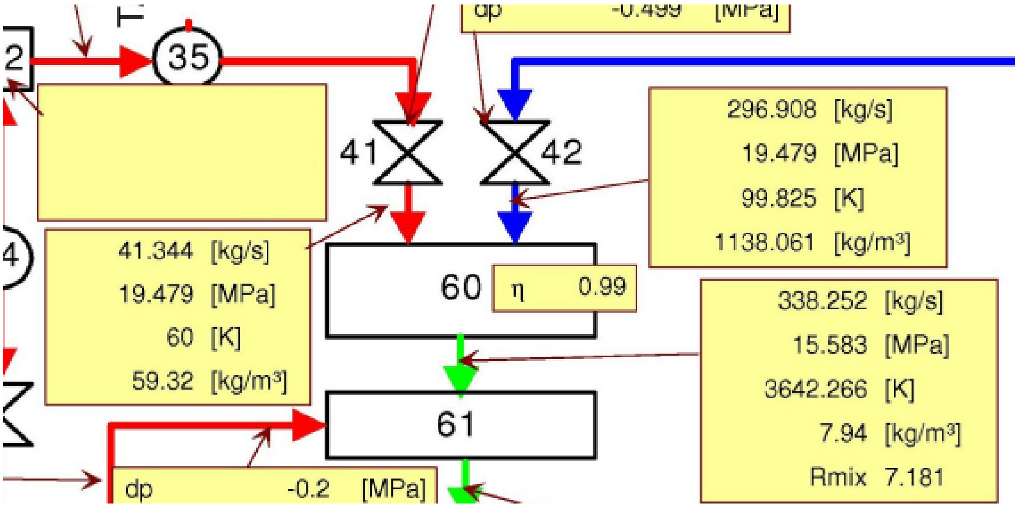


Figure 6.15: SE-22 engine main combustion chamber (LH2-LOX) (Fig.A.12)

SE-22 engine main combustion chamber (LH2-LOX)		
	Propelant 1	***Propelant 2***
Species	H_2	O_2
Mols [mols]	2.051e+04	9.278e+03
Temperature [K]	60	99.825
Pressure inlet [bar]	194.79	
	Combustion	
Pressure outlet [bar]	155.83	
Species	[H_2 O_2 H_2O H O OH]	
Mols [mols]	[3.521e+03 , 3.8049e+02 , 1.563e+04 , 7.678e+02 , 2.255e+02 , 1.936 e+03]	
	Paper	***Calculated***
Temperature [K]	3642.266	3786.97

Table 6.13: Calculated parameters for SE-22 engine main combustion chamber (LH2-LOX)

6.2.3 SE-22 Engine Secondary Combustion Chamber (LH2-LOX)

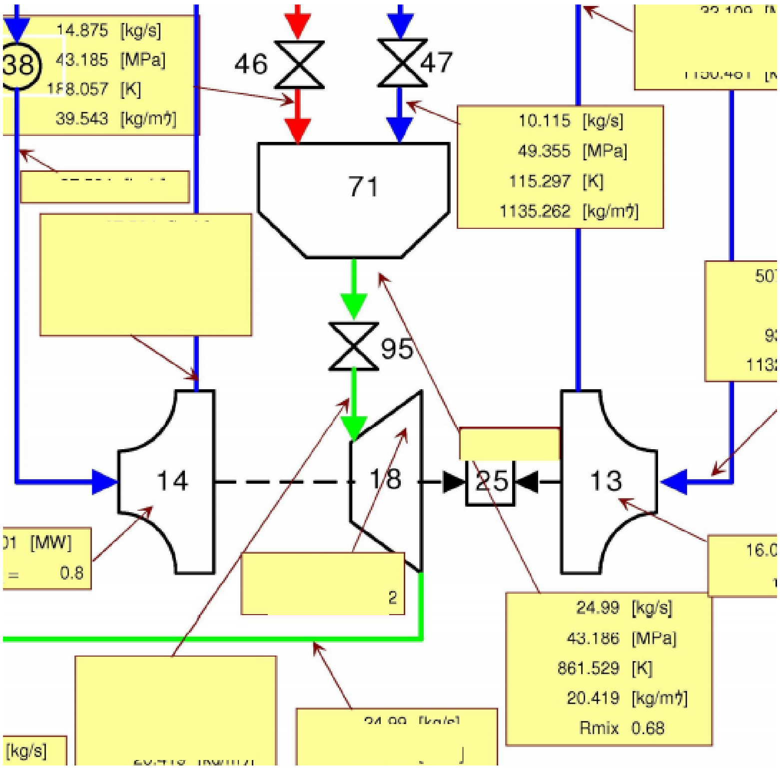


Figure 6.16: SE-22 engine secondary combustion chamber (LH2-LOX) (Fig.A.12)

SE-22 engine secondary combustion chamber (LH2-LOX)		
	Propelant 1	***Propelant 2***
Species	H_2	O_2
Mols [mols]	3.2455e+03	163.60
Temperature [K]	40.56	100.068
Pressure inlet [bar]	187.5	
	Combustion	
Pressure outlet [bar]	150	
Species	[H_2 O_2 H_2O H O OH]	
Mols [mols]	[2.918e+03 , 1.916e-09 , 3.272e+02 , 1.169e-05 , 2.192e-09 , 3.040e-08]	
	Paper	***Calculated***
Temperature [K]	810.298	1118.6e+03

Table 6.14: Calculated parameters for SE-22 engine secondary combustion chamber (LH2-LOX)

6.2.4 SSME Block II Engine Secondary Combustion Chamber for O2 Cycle (LH2-LOX)

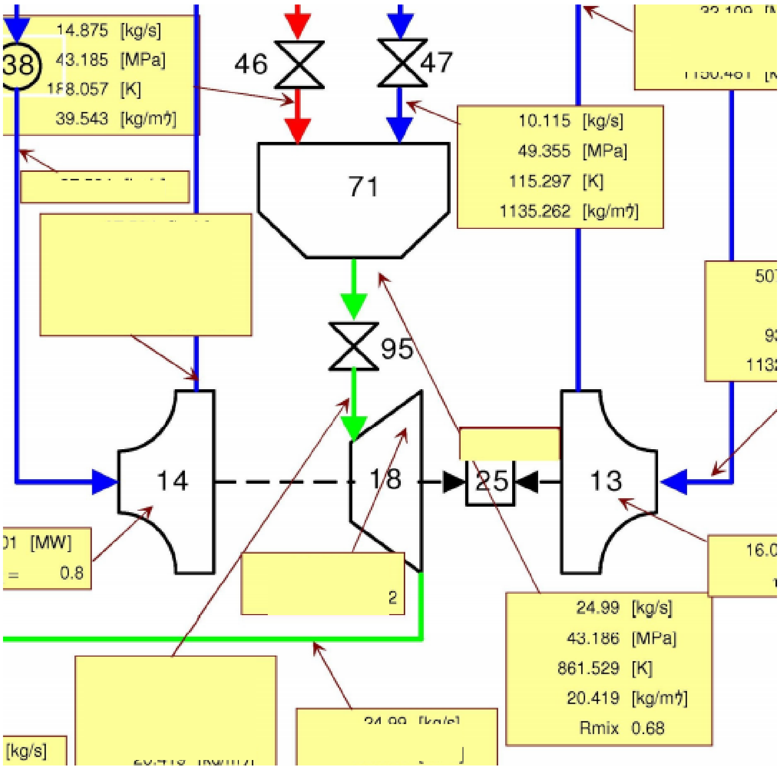


Figure 6.17: SSME Block II engine secondary combustion chamber for O_2 cycle (LH2-LOX) (Fig.A.11)

SSME Block II engine secondary combustion chamber for O_2 cycle (LH2-LOX)		
	Propelant 1	***Propelant 2***
Species	H_2	O_2
Mols [mols]	7.3784e+03	3.1611e+02
Temperature [K]	188.057	115.297
Pressure inlet [bar]	493.55	
	Combustion	
Pressure outlet [bar]	431.86	
Species	[H_2 O_2 H_2O H O OH]	
Mols [mols]	[6.746+03 , 4.420e-08 , 6.322e+02 , 2.611e-05 , 5.041e-08 , 3.490e-07]	
	Paper	***Calculated***
Temperature [K]	861.529	1149.55

Table 6.15: Calculated parameters for SSME Block II engine secondary combustion chamber for O_2 cycle (LH2-LOX)

6.2.5 SSME Block II Engine Secondary Combustion Chamber for H2 Cycle (LH2-LOX)

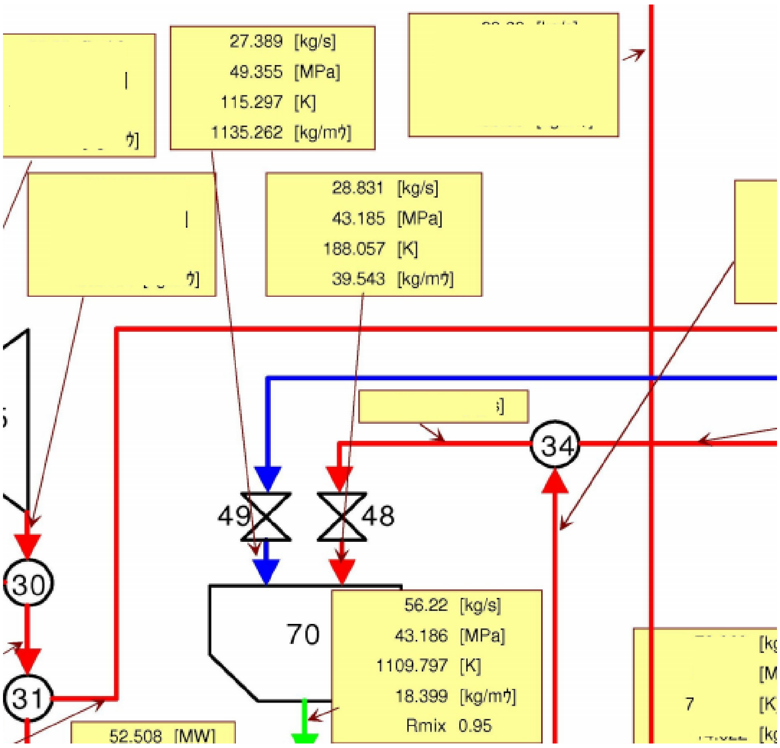


Figure 6.18: SSME Block II engine secondary combustion chamber for H_2 cycle (LH2-LOX) (Fig.A.11)

SSME Block II engine secondary combustion chamber for H_2 cycle (LH2-LOX)		
	Propelant 1	***Propelant 2***
Species	H_2	O_2
Mols [mols]	1.4301e+04	8.559e+02
Temperature [K]	188.057	115.297
Pressure inlet [bar]	493.55	
	Combustion	
Pressure outlet [bar]	431.86	
Species	[H_2 O_2 H_2O H O OH]	
Mols [mols]	[1.258e+04 ,6.385e-08 , 1.711e+03 , 0.0027 , 7.8605e-08 , 9.535e-06]	
	Paper	***Calculated***
Temperature [K]	1109.797	1390.49

Table 6.16: Calculated parameters for SSME Block II engine secondary combustion chamber for H_2 cycle (LH2-LOX)

6.2.6 SLME Engine Secondary Combustion Chamber for O2 Cycle (LH2-LOX)

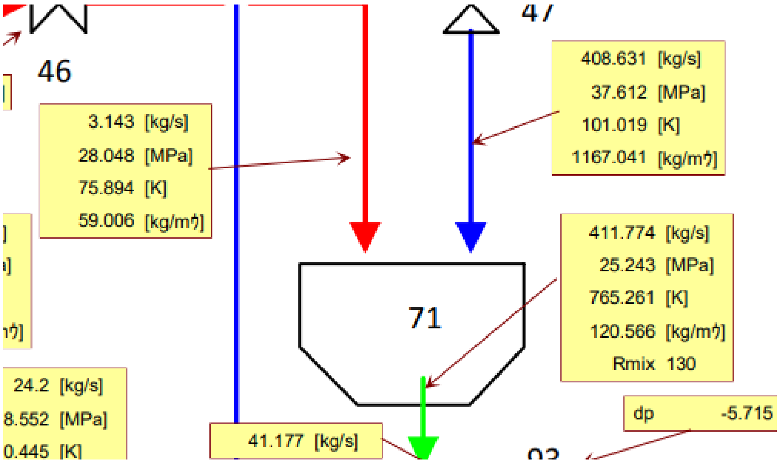


Figure 6.19: SLME engine secondary combustion chamber for O_2 cycle (LH2-LOX) (Fig.A.10)

SLME engine secondary combustion chamber for O_2 cycle (LH2-LOX)		
	Propelant 1	***Propelant 2***
Species	H_2	O_2
Mols [mols]	1.4301e+04	8.559597474842177e+02
Temperature [K]	75.894	101.019
Pressure inlet [bar]	280.48	
	Combustion	
Pressure outlet [bar]	252.43	
Species	[H_2 O_2 H_2O H O OH]	
Mols [mols]	[4.682e-07 , 1.199e+04 , 1.559e+03 , 1.240e-07 , 1.3729e-06 , 0.0020]	
	Paper	***Calculated***
Temperature [K]	765.261	1053.7

Table 6.17: Calculated parameters for SLME engine secondary combustion chamber for O_2 cycle (LH2-LOX)

6.2.7 SLME Engine Secondary Combustion Chamber for H2 Cycle (LH2-LOX)

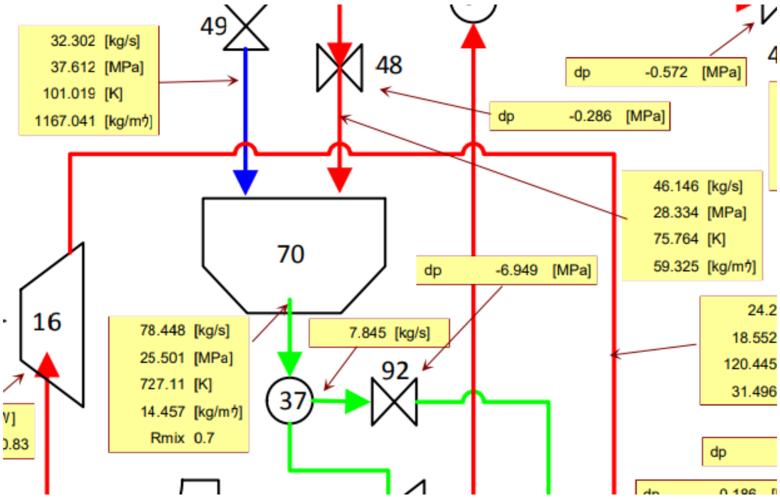


Figure 6.20: SLME engine secondary combustion chamber for H_2 cycle (LH2-LOX) (Fig.A.10)

SLME engine secondary combustion chamber for H_2 cycle (LH2-LOX)		
	Propelant 1	***Propelant 2***
Species	H_2	O_2
Mols [mols]	2.2889e+04	1.0095e+03
Temperature [K]	1.88057e+02	1.15297e+02
Pressure inlet [bar]	283.34	
	Combustion	
Pressure outlet [bar]	255.01	
Species	[H_2 O_2 H_2O H O OH]	
Mols [mols]	[2.0870e+04 , 8.5311e-07 , 2.019e+03 , 1.4232e-04 , 9.6755e-07 , 5.1722e-06]	
	Paper	***Calculated***
Temperature [K]	727.11	1159.501

Table 6.18: Calculated parameters for SLME engine secondary combustion chamber for H_2 cycle (LH2-LOX)

6.2.8 SLME Engine Main Combustion Chamber (LH2-LOX)

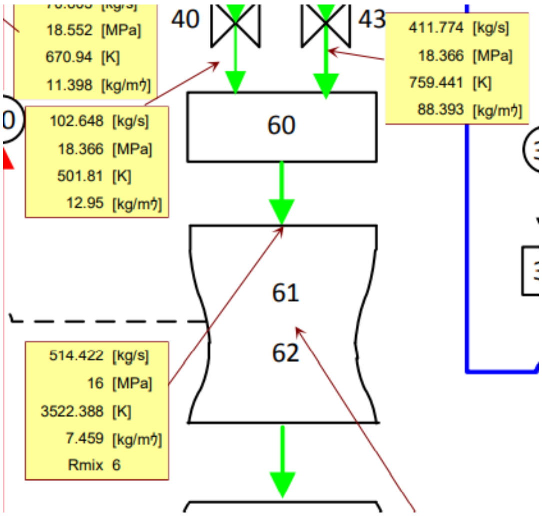


Figure 6.21: SLME engine main combustion chamber (LH2-LOX) (Fig.A.10)

SLME engine main combustion chamber (LH2-LOX)		
	Propelant 1	***Propelant 2***
Species	Comb. species from table 6.18	Comb. species from table 6.17
Mols [mols]	Comb. mols from table 6.18	Comb. mols from table 6.17
Temperature [K]	670.940	759.441
Pressure inlet [bar]	185.52	
	Combustion	
Pressure outlet [bar]	160	
Species	[H_2 O_2 H_2O H O OH]	
Mols [mols]	[1.9166e+03 , 1.7021e+03 , 2.0817e+04 , 5.00027e+02 , 4.1051e+02 , 2.927e+03]	
	Paper	***Calculated***
Temperature [K]	3522.388	3673.57

Table 6.19: Calculated parameters for SLME engine mian combustion chamber (LH2-LOX)

6.2.9 SE-12 Engine Secondary Combustion Chamber (CH4-LOX)

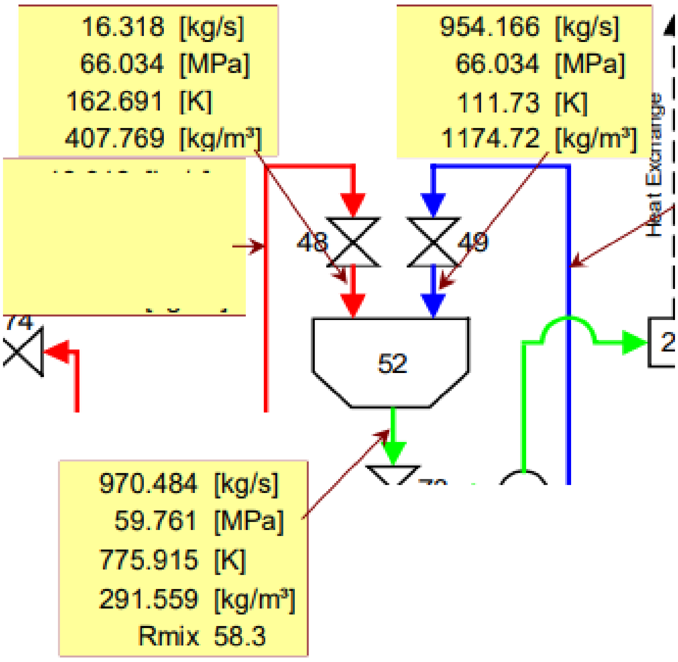


Figure 6.22: SE-12 engine secondary combustion chamber (CH4-LOX) (Fig.A.14)

SE-12 engine secondary combustion chamber (CH4-LOX)		
	Propelant 1	***Propelant 2***
Species	CH_4	O_2
Mols [mols]	1.0171e+03	2.9819e+04
Temperature [K]	162.691	111.73
Pressure inlet [bar]	660.34	
	Combustion	
Pressure outlet [bar]	597.761	
Species	[CH_4 O_2 H_2O H O OH CO_2 CO H_2]	
Mols [mols]	[3.0866e-07 , 2.7785e+04 , 2.0342e+03 , 2.12438e-06 , 1.49228e-05 , 0.0070 , 1.0171e+03 , 5.0939e-06 2.0730e-05]	
	Paper	***Calculated***
Temperature [K]	775.915	1107.523

Table 6.20: Calculated parameters for SE-12 engine secondary combustion chamber (CH4-LOX)

6.2.10 SE-12 Engine Main Combustion Chamber (CH4-LOX)

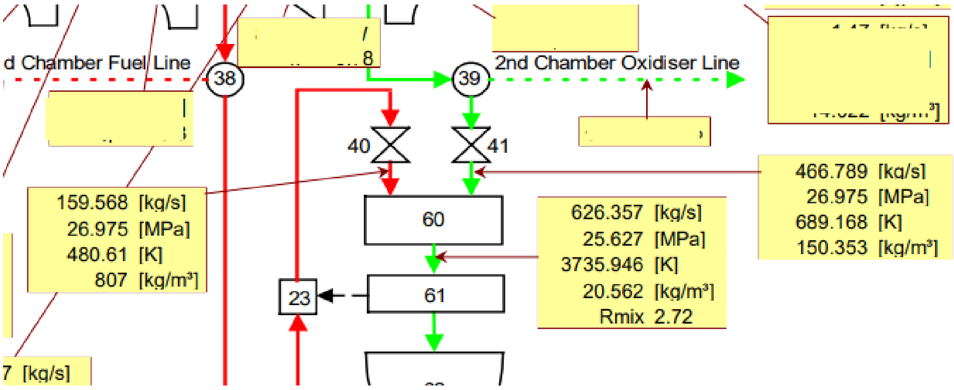


Figure 6.23: SE-12 engine main combustion chamber (CH4-LOX) (Fig.A.14)

SE-12 engine main combustion chamber (CH4-LOX)		
	Propelant 1	***Propelant 2***
Species	Comb. species from table 6.20	CH_4
Mols [mols]	$\frac{967.628}{970.484}$ · Comb. mols from table 6.20	2.9819e+04
Temperature [K]	675.237	202.974
Pressure inlet [bar]	269.81	
	Combustion	
Pressure outlet [bar]	256.32	
Species	[CH_4 O_2 H_2O H O OH CO_2 CO H_2]	
Mols [mols]	[2.5232e-05 , 1.663e+03 , 2.721e+04 , 8.546e+02 , 5.3604e+02 , 3.9109e+03 , 8.002e+03 , 8.4666e+03 , 3.3357e+03]	
	Paper	***Calculated***
Temperature [K]	3586.694	3758.607

Table 6.21: Calculated parameters for SE-12 engine main combustion chamber (CH4-LOX)

6.2.11 RD-180 Engine Secondary Combustion Chamber (RP1-LOX)

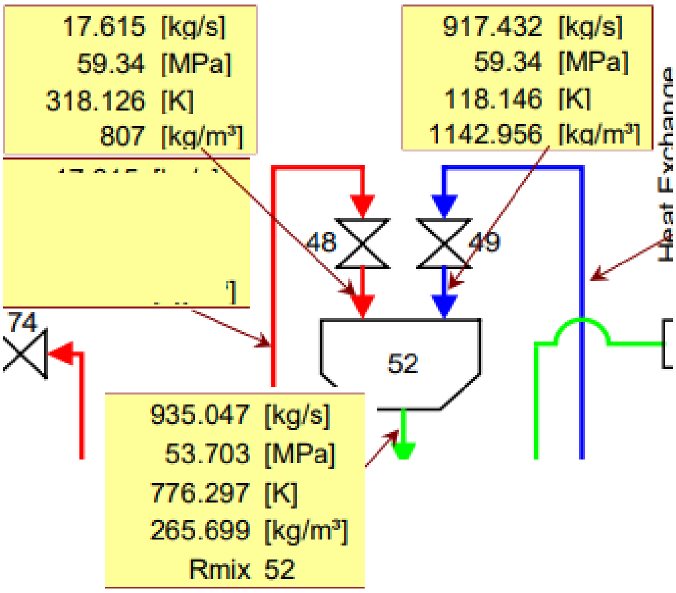


Figure 6.24: RD-180 engine secondary combustion chamber (RP1-LOX) (Fig.A.13)

RD-180 engine secondary combustion chamber (RP1-LOX)		
	Propelant 1	***Propelant 2***
Species	O_2	RP_1
Mols [mols]	2.8672e+04	1.254e+02
Temperature [K]	118.146	318.126
Pressure inlet [bar]	593.4	
	Combustion	
Pressure outlet [bar]	537.03	
Species	[$C_{10}H_{22}$ $C_{10}H_{18}$ O_2 H_2O H O OH CO_2 CO]	
Mols [mols]	[1.421e-09 , 1.535e-09 , 2.6776e+04 , 1.282e+03 , 1.170e-07 , 1.447e-06 , 0.0018 , 3.719e-07 , 3.534e-07 , 1.254e+03]	
	Paper	***Calculated***
Temperature [K]	3735.946	3901.625

Table 6.22: Calculated parameters for RD-180 engine secondary combustion chamber (RP1-LOX)

6.2.12 RD-180 Engine Main Combustion Chamber (RP1-LOX)

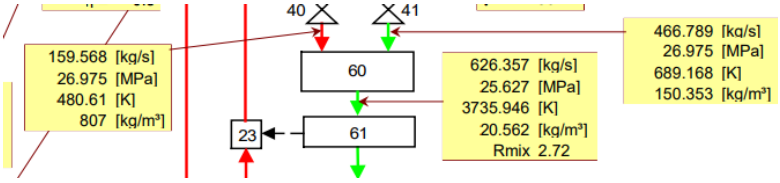


Figure 6.25: RD-180 engine main combustion chamber (RP1-LOX) (Fig.A.13)

RD-180 engine main combustion chamber (RP1-LOX		
	Propelant 1	***Propelant 2***
Species	Comb. species from table 6.21	RP_1
Mols [mols]	$\frac{466.789}{970.484}\cdot$ Comb. mols from table 6.21	1.136e+03
Temperature [K]	689.168	480.651
Pressure inlet [bar]	269.75	
	Combustion	
Pressure outlet [bar]	256.27	
Species	[$C_{10}H_{22}$ $C_{10}H_{18}$ O_2 H_2O H O OH H_2 CO_2 CO]	
Mols [mols]	[3.508e-10 , 5.373e+02 , 9.235e+03 , 5.842e+02 , 2.871e+02 , 1.726e+03 , 1.865e+03 , 7.672e+03 , 4.314e+03]	
	Paper	***Calculated***
Temperature [K]	3735.946	3901.625

Table 6.23: Calculated parameters for RD-180 engine main combustion chamber (RP1-LOX)

6.2.13 Combustion Chamber Overall Analysis

From the previous data, it can be highlighted that the calculation is above the authors' data (Fig. 6.26) and it is due to multiple factors. Some figures (e.g. A.12 or A.9) include efficiency in their chart, so it is assumed that all the combustion chamber got one even if it is not included, nevertheless, these coefficient are not explained so it is impossible to simulate with these factors. Additionally, in a subsequent paper of ***Herbertz***, their analysis of the combustion chamber are cooled off by reducing the temperature [2].

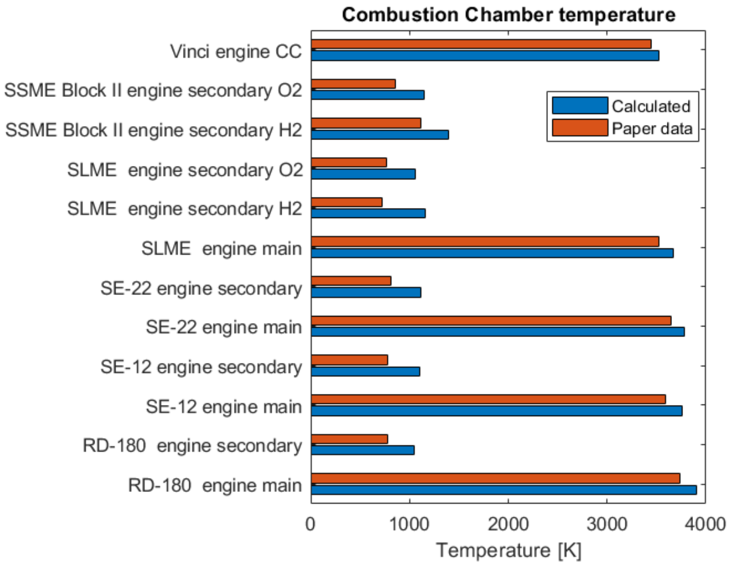
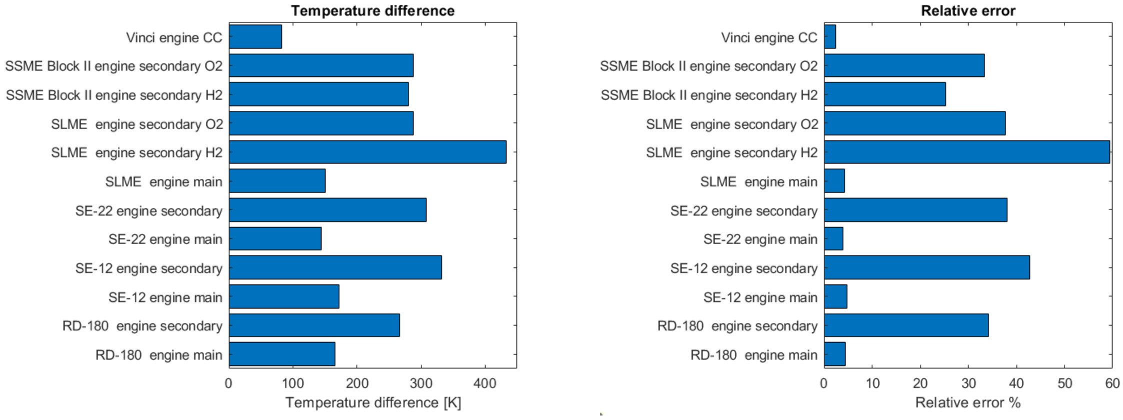


Figure 6.26: Combustion temperature comparison between papers and calculated

It can be noted a strange behaviour where lower combustion temperatures have more gap between the author data and the calculated one. These difference are computed and included in the figures below. This effect could be attributed to the combustion pressure effect as secondary combustion chambers are at higher pressures, that `HGS` neglects, but there are other possibilities. The main possibility suggested is that main combustion chambers are followed by a nozzle that does not restrict the flux to leave the combustion chamber while other possibilities are linked to other machines that have restrictions in term of flux. This effect could affect the heat transferred out and how the combustion is developed.



(a) Difference between temperatures **(b)** Relative error of the temperature

Figure 6.27: Differences between data and calculation

6.3 Turbine

Turbine analysis is the same as the pumps, to calculate the work and efficiency, one must acknowledge the inputs and outputs. In this case, turbines use after-combustion gases or H_2 to perform. It will require for after-combustion gases for the output from some combustion chambers of the previous section.

The analysis is performed as follows:

1 - Enthalpy calculation of the inlet and outlet using `HGS` or `INIST` depending on the conditions and propellant.

$$\begin{aligned} H_i &= HGSprop(...,'H') \quad [\mathrm{kJ}] \quad \text{For } HGS \\ H_i &= INIST() \cdot \frac{n * 1000}{Mm} \quad [\mathrm{kJ}] \quad \text{For } INIST \end{aligned} \tag{6.7}$$

2 - Work calculation, derived from the closed system energy conservation (Eq. 2.3) with no heat losses, no kinetic variation and no potential energy variation:

$$\begin{aligned} \cancelto{0}{\dot{Q}} - \dot{W} &= \dot{m} \cdot \Delta h \\ \dot{W} &= -\dot{m} \cdot \Delta h \\ \dot{W} &= \dot{m} \cdot (h_{in} - h_{out}) \end{aligned} \tag{6.8}$$

3 - Calculation of inlet entropy as isentropic work conserves the pressure ($S_{in} = S_{out}$).

$$\begin{aligned} S_{in} &= HGSprop(...,'S') \quad [\frac{\mathrm{kJ}}{\mathrm{K}}] \quad \text{For } HGS \\ S_{in} &= INIST() \quad [\frac{\mathrm{kJ}}{\mathrm{mol} \cdot \mathrm{K}}] \quad \text{For } INIST \end{aligned} \tag{6.9}$$

4 - Calculation of the outlet temperature for an isentropic process, thus enthalpy calculation of the mentioned temperature.

$$\begin{aligned} T &= fsolve(@(T_x)HGSprop(...,T_x,'S') - S_{in} == 0) \quad [\mathrm{K}] \quad \text{For } HGS \\ T &= INIST(species,'t_p s',P,S_{in}) \quad [\mathrm{K}] \quad \text{For } INIST \end{aligned} \tag{6.10}$$

5 - Isentropic work calculation and efficiency as:

$$\begin{aligned} \dot{W}_s &= \dot{m} \cdot (h_{in} - h_{outs}) \\ \eta &= \frac{\dot{W}}{\dot{W}_s} \end{aligned} \tag{6.11}$$

The following subsection will show the analysed turbine parameters and the results obtained from the previously explained analysis. In Section 6.3.13, an overall analysis will be presented.

6.3.1 Vinci O2 Supplier Turbine

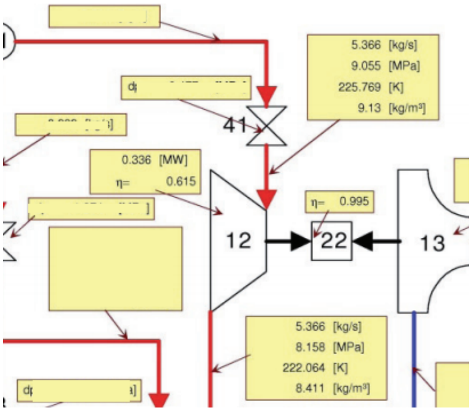


Figure 6.28: Vinci O_2 supplier turbine (Fig.A.9)

Vinci O_2 supplier turbine		
	Inlet:	***Outlet:***
Species	H_2	
Mols [mols]	2.6617e+03	
Temperature [K]	225.769	222.064
Pressure [bar]	90.55	81.58
	Paper	***Calculated***
η	0.615	0.533
W [kJ/s]	336	275.7

Table 6.24: Calculated parameters for Vinci O_2 supplier turbine

The results are slightly below the authors' data.

6.3.2 Vinci H2 Supplier Turbine

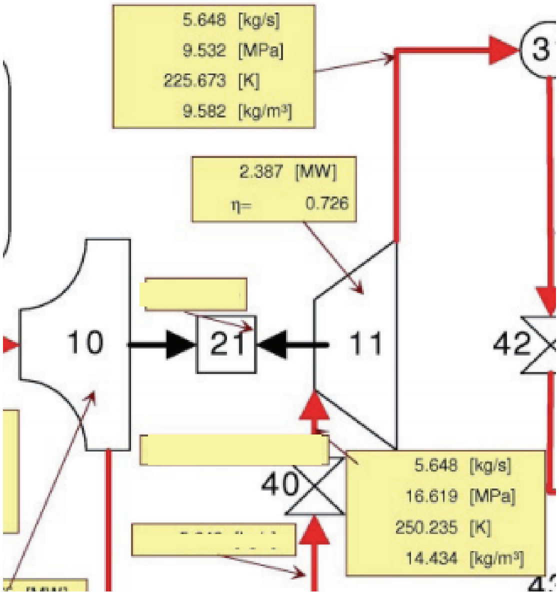


Figure 6.29: Vinci H_2 supplier turbine (Fig.A.9)

Vinci H_2 supplier turbine		
	Inlet:	***Outlet:***
Species	H_2	
Mols [mols]	2.8016e+03	
Temperature [K]	250.235	225.673
Pressure [bar]	166.19	95.32
	Paper	***Calculated***
η	0.726	0.645
W [kJ/s]	2387	1938.404

Table 6.25: Calculated parameters for Vinci H_2 supplier turbine

6.3.3 SE-22 H2 Supplier Turbine

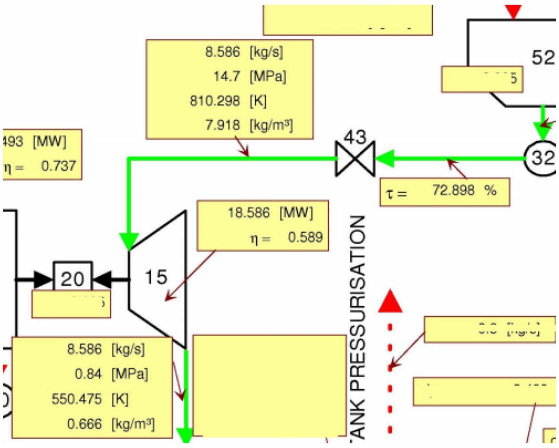


Figure 6.30: SE-22 H_2 supplier turbine (Fig.A.12)

SE-22 H_2 supplier turbine		
	Inlet:	***Outlet:***
Species	Comb. species (tab. 6.14)	
Mols [mols]	0.73· Comb. mols (tab. 6.14)	
Temperature [K]	810.298	550.475
Pressure [bar]	147	8.4
	Paper	***Calculated***
η	0.589	0.588
W [kJ/s]	1.8586e+04	1.8541e+04

Table 6.26: Calculated parameters for SE-22 H_2 supplier turbine

6.3.4 SE-22 O2 Supplier Turbine

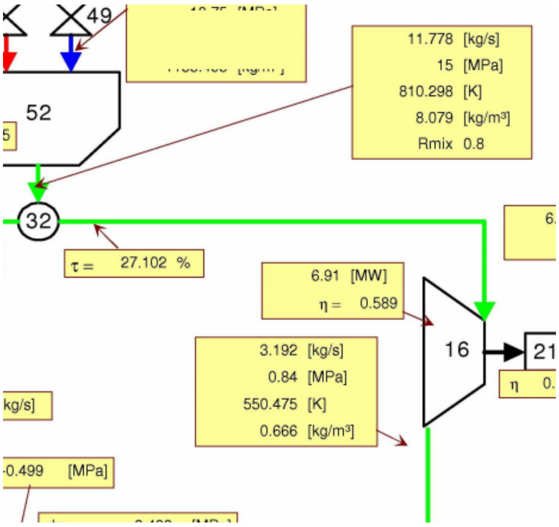


Figure 6.31: SE-22 O_2 supplier turbine (Fig.A.12)

SE-22 O_2 supplier turbine		
	Inlet:	***Outlet:***
Species	Comb. species (tab. 6.14)	
Mols [mols]	0.27· Comb. mols (tab. 6.14)	
Temperature [K]	810.298	550.475
Pressure [bar]	147	8.4
	Paper	***Calculated***
η	0.589	0.588
W [kJ/s]	6.910e+03	6.8931e+03

Table 6.27: Calculated parameters for SE-22 O_2 supplier turbine

6.3.5 SE-12 O2 Supplier Turbine (17)

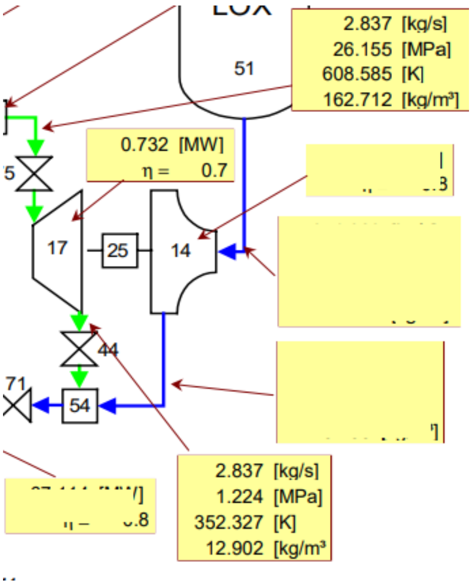


Figure 6.32: SE-12 O_2 supplier turbine (17) (Fig.A.14)

SE-12 O_2 supplier turbine (17)		
	Inlet:	***Outlet:***
Species	Comb. species (tab. 6.20)	
Mols [mols]	$\frac{2.84}{970.484}\cdot$ Comb. mols (tab. 6.20)	
Temperature [K]	608.585	352.327
Pressure [bar]	261.55	12.24
	Paper	***Calculated***
η	0.7	0.76
W [kJ/s]	732	730.1

Table 6.28: Calculated parameters for SE-12 O_2 supplier turbine (17)

6.3.6 SE-12 O2 Supplier Turbine (15)

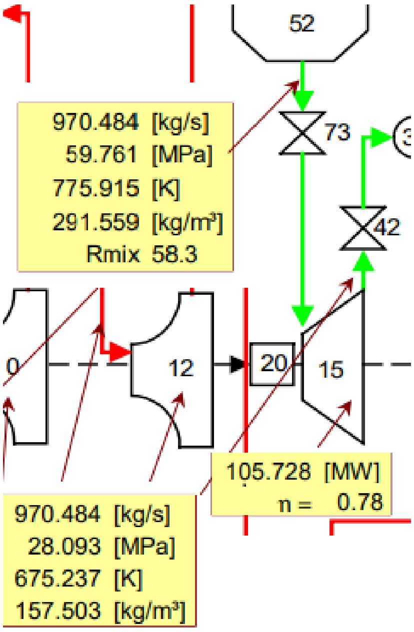


Figure 6.33: SE-12 O_2 supplier turbine (15) (Fig.A.14)

SE-12 O_2 supplier turbine (15)		
	Inlet:	***Outlet:***
Species	Comb. species (tab. 6.20)	
Mols [mols]	Comb. mols (tab. 6.20)	
Temperature [K]	775.915	675.237
Pressure [bar]	597.61	280.93
	Paper	***Calculated***
η	0.78	0.77
W [kJ/s]	1.05728e+05	1.0573e+05

Table 6.29: Calculated parameters for SE-12 O_2 supplier turbine (15)

6.3.7 SSME Block II O2 Supplier Turbine (16)

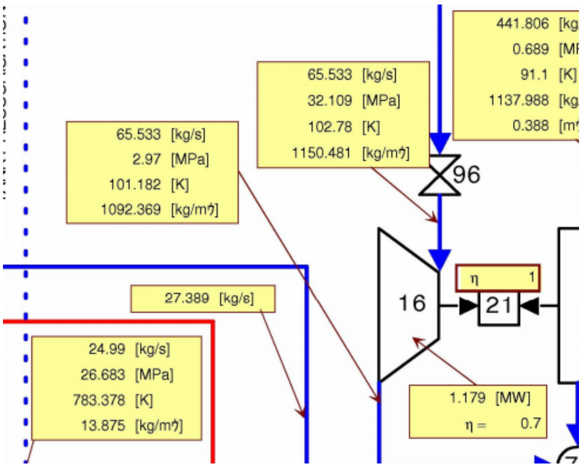


Figure 6.34: SSME Block II O_2 supplier turbine (16) (Fig.A.11)

SSME Block II O_2 supplier turbine (16)		
	Inlet:	***Outlet:***
Species	O_2	
Mols [mols]	2.048e+03	
Temperature [K]	102.78	101.182
Pressure [bar]	321.09	29.7
	Paper	***Calculated***
η	0.7	0.699
W [kJ/s]	1.179e+03	1.1792e+03

Table 6.30: Calculated parameters for SSME Block II O_2 supplier turbine (16)

6.3.8 SSME Block II O2 Supplier Turbine (18)

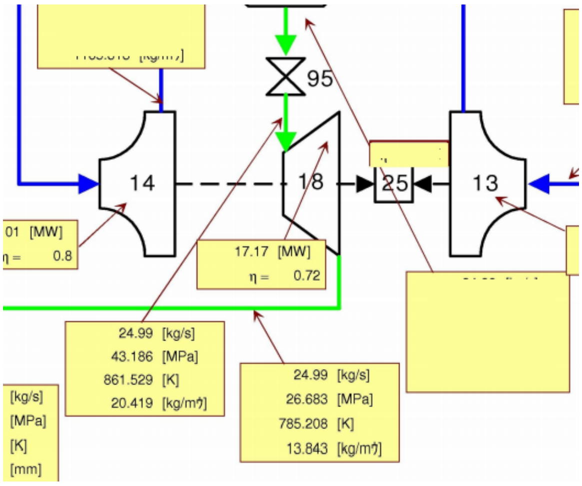


Figure 6.35: SSME Block II O_2 supplier turbine (18) (Fig.A.11)

SSME Block II O_2 supplier turbine (18)		
	Inlet:	***Outlet:***
Species	Comb. species (tab. 6.15)	
Mols [mols]	Comb. mols (tab. 6.15)	
Temperature [K]	861.529	785.208
Pressure [bar]	431.86	266.83
	Paper	***Calculated***
η	0.72	0.72
W [kJ/s]	1.7170e+04	1.7173e+04

Table 6.31: Calculated parameters for SSME Block II O_2 supplier turbine (18)

6.3.9 SSME Block II H2 Supplier Turbine (17)

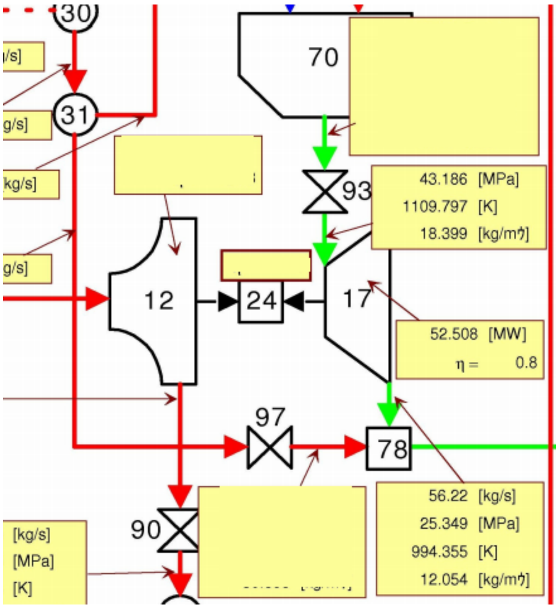


Figure 6.36: SSME Block II H_2 supplier turbine (17) (Fig.A.11)

SSME Block II H_2 supplier turbine		
	Inlet:	***Outlet:***
Species	Comb. species (tab. 6.16)	
Mols [mols]	Comb. mols (tab. 6.16)	
Temperature [K]	1109.797	994.355
Pressure [bar]	431.86	253.49
	Paper	***Calculated***
η	0.8	0.799
W [kJ/s]	5.2508e+04	5.2465e+04

Table 6.32: Calculated parameters for SSME Block II H_2 supplier turbine

6.3.10 SSME Block II H2 Supplier Turbine (15)

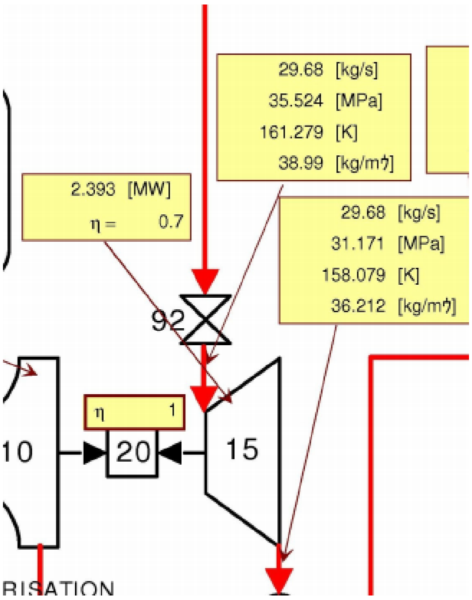


Figure 6.37: SSME Block II H_2 supplier turbine (15) (Fig.A.11)

SSME Block II H_2 supplier turbine (17)		
	Inlet:	***Outlet:***
Species	H_2	
Mols [mols]	1.4722+04	
Temperature [K]	161.279	158.079
Pressure [bar]	355.24	311.71
	Paper	***Calculated***
η	0.7	0.602
W [kJ/s]	2.393e+03	2.055e+03

Table 6.33: Calculated parameters for SSME Block II H_2 supplier turbine (17)

6.3.11 RD-180 O2 Supplier Turbine (17)

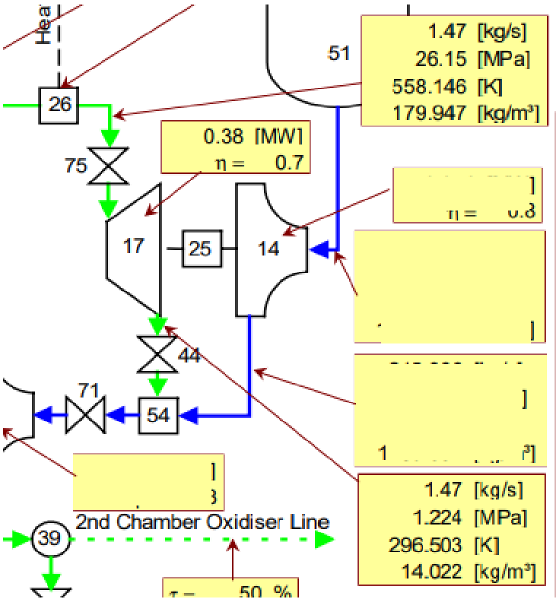


Figure 6.38: RD-180 O_2 supplier turbine (17) (Fig.A.13)

RD-180 O_2 supplier turbine (17)		
	Inlet:	***Outlet:***
Species	Comb. species (tab. 6.22)	
Mols [mols]	$\frac{1.47}{935.047}$Comb. mols (tab. 6.22)	
Temperature [K]	776.297	689.168
Pressure [bar]	537.03	280.88
	Paper	***Calculated***
η	0.7	0.83
W [kJ/s]	380	374.88

Table 6.34: Calculated parameters for RD-180 O_2 supplier turbine (17)

6.3.12 RD-180 O2 Supplier Turbine (15)

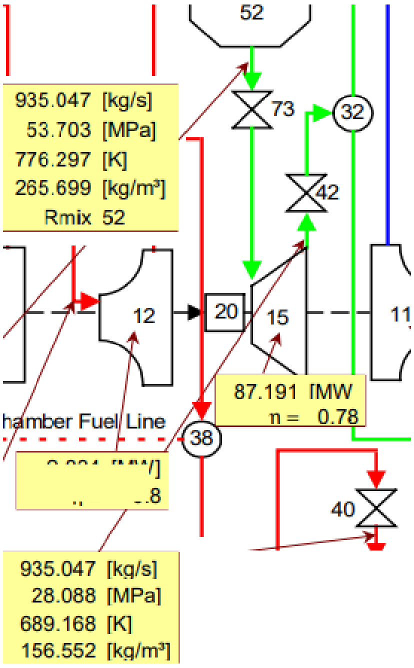


Figure 6.39: RD-180 O_2 supplier turbine (15) (Fig.A.13)

RD-180 O_2 supplier turbine (15)		
	Inlet:	***Outlet:***
Species	Comb. species (tab. 6.22)	
Mols [mols]	Comb. mols (tab. 6.22)	
Temperature [K]	776.297	675.237
Pressure [bar]	597.61	280.93
	Paper	***Calculated***
η	0.78	0.77
W [kJ/s]	8.7191e+04	8.719e+04

Table 6.35: Calculated parameters for RD-180 O_2 supplier turbine (15)

6.3.13 Turbine Overall Analysis

The following figures show the results of the previous analysis and, in most of the case, the results of the calculations agree with the authors' data. The case that is slightly different to the expected is analysed below.

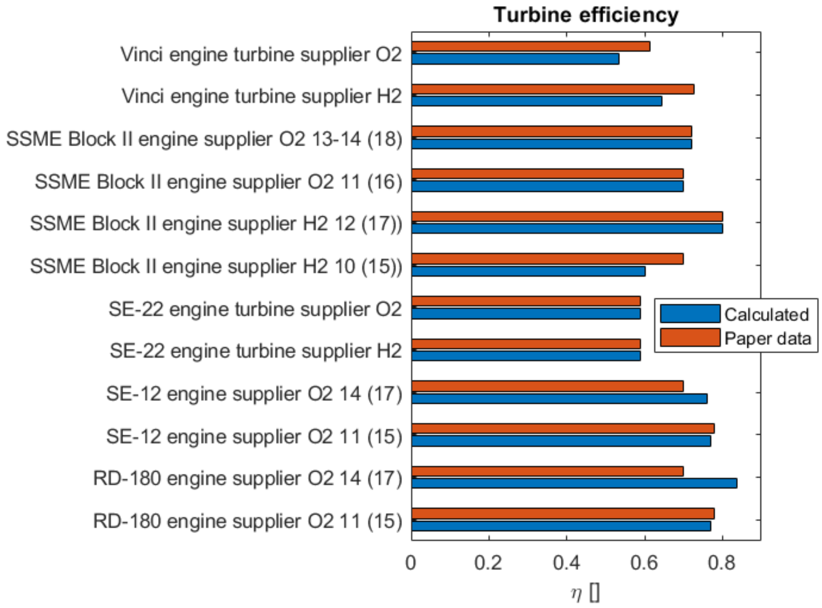


Figure 6.40: Turbines efficiencies comparison

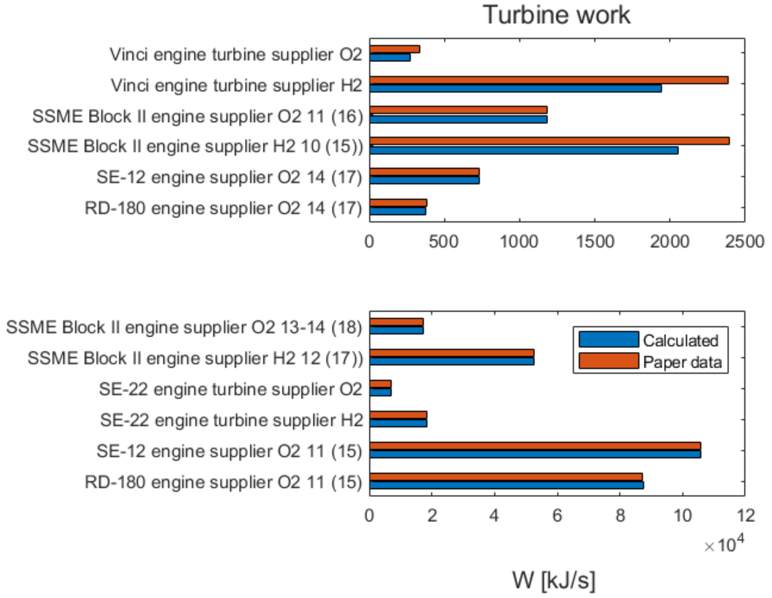


Figure 6.41: Turbines work comparison

The turbines from RD-180 that supplies energy to the oxygen pump (Sec. 6.3.11) has a discrepancy in the results that can be attributed to the omission of the valve before the turbines by lack of data. The drop of pressure can affect the efficiency, as the isentropic work depends on inlet entropy which, in turn, depends on the pressure (Sec. 3.2.3).

The turbines simulations carried out with `INIST` as code to supply thermodynamics data are slightly below the author's data. This can be due to multiples reasons but it can be attributed to the difference in the data used by the authors and `INIST`, the authors could use data that approximates these turbines to ideal fluid and not a real one. This idea is reinforced with the fact that pumps do not have this error while they use the inverse process and the ideal fluid turbines that use `HGS` do not have this difference in the result. This ideal vs real fluid will be reviewed in Chapter 7.

6.4 Vulcain 2

Finally, the complete rocket cycle, the Vulcain 2 cycle A.15, is analysed. This cycle requires some additional analysis to approximate valves, the mixer and the nozzle. As mentioned before, schemes have some lack of information, so to carry out the analysis, if there is no information, it is assumed the species, mols, temperature and pressure for the previous machine in the cycle, supposing there are no losses.

Valves are distinguished into two types: Post-combustion and cryogenic. With the post-combustion valves, the temperature remains equal after the pressure changes, see Sec. 2.2.2. For cryogenic valves, a script that recalculates the temperature has to be done as the enthalpy must remain constant.

```
function [P2,T2] = Valve(species,T1,P1,dp)
H1 = INIST(species{1},'h_pt',P1,T1);
P2 = P1 + dp;
options.Display = 'off';
T2 = fsolve(@(T)DeltaH(T),T1,options);

    function deltaH = DeltaH(T)
        H2 = INIST(species{1},'h_pt',P2,T);
        deltaH = H2-H1;
    end
end
```

Listing 6.1: Cryogenic valves

The Mixer of the Vulcain blends the post-combustion products after the energy is harnessed in the turbine. Enthalpy has to be conserved after the mixture following the script below.

```
H1 = HGSprop(species1,n1,T1,P,'H');
```

```
H2 = HGSprop(species2,n2,T2,P,’H’);
Hin = H1 + H2;
options.Display = ’off’;
T = fsolve(@(T)DeltaH(T),T1,options);

    function deltaH = DeltaH(T)
        Hout = HGSprop(species,n,T,P,’H’);
        deltaH = Hout-Hin;
    end
```

Listing 6.2: Mixer

The Nozzle, in the convergent part, assumes that the Mach number is one on the throat and the isentropic behaviour of the flux, using the following equations:

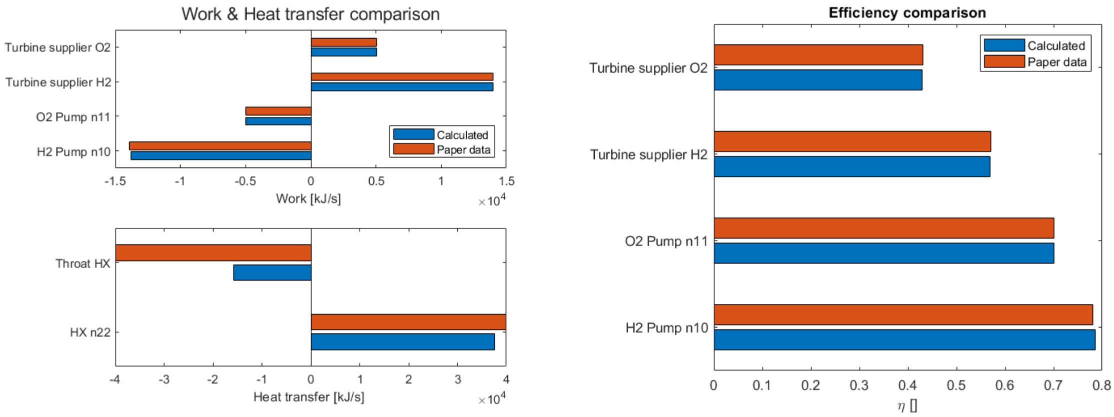
$$\begin{aligned} T_{throat} &= \frac{T_{inlet}}{1+\frac{\gamma-1}{2}} \\ P_{throat} &= \frac{P_{inlet}}{\frac{T_{inlet}}{T_{throat}}^{\frac{\gamma}{\gamma-1}}} \end{aligned} \tag{6.12}$$

The Heat exchanger calculates the heat transfer using the following equation:

$$\dot{Q} = \dot{m} \cdot (h_{in} - h_{out}) \tag{6.13}$$

6.4.1 Vulcain 2 Analysis Results

These are the results of the Vulcain cycle, excluding the divergent nozzle and its extension.



(a) Vulcain 2 Work comparison

(b) Vulcain 2 efficiencies comparison

Figure 6.42: Work, Heat Transfer and efficiency comparisons

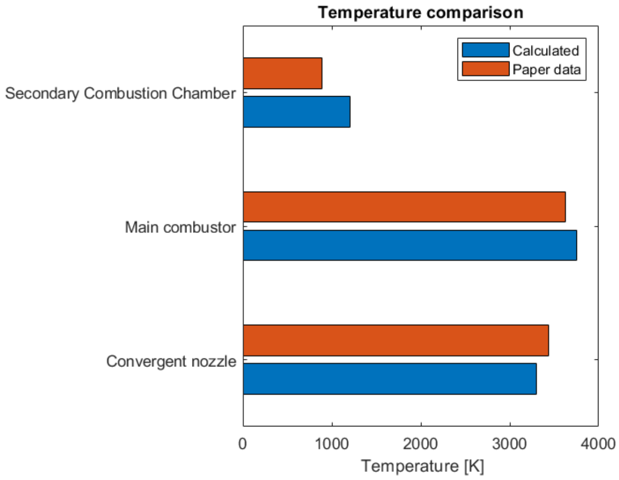


Figure 6.43: Vulcain 2 Temperature comparison

Results are accurate with the data from ***Herbertz*** [41] but there are some discrepancies. The temperature of the secondary combustion chamber has been explained in Section 6.2.13. Nozzle temperature calculation is slightly below the offered by the author, it could be explained as for our calculation of the velocity is zero at the combustion chamber exit, while it has not to be it. The work difference in the heat exchanger in the nozzle throat can not be explained without assuming a lack of information to do the analysis or an error on the data, this difference is the reason why the divergent nozzle and its extension has not been calculated.

CHAPTER 7

ADDITIONAL CAPABILITIES OF THE LIBRARIES

This chapter will present the other examples, besides turbomachinery analysis (Chap. 6), that are extracted from this thesis' development. The treatment of RP-1 as a propellant is overviewed in this section, as it not appears in `HGS` database. As mentioned before, the minimisation algorithm is one of the main outputs that can be used by professors to explain further why Gibbs' free energy is minimised. Moreover, the use of `GitHub` is going to be explained as a platform to do a code version control, due to its general advantages like the one mentioned above and many other possibilities.

7.1 RP-1

Rockets use multiple propellants depending on the stage and multiple parameters. One of the most complexes is RP-1 because there is no characterised formula, only an approximation, so a surrogate model is required to use it in rocket cycles. According to Sutton [42]'s RP-1 formula approximation is near to $CH_{1,97_n}$ with a mass of $175\frac{g}{mol}$, while RPA uses $CH_{1,95_n}$ [7] and Chickos et all propose $CH_{2}\cdot{}_{045_n}$ with a mass $148\frac{g}{mol}$ [43]. There is a consensus but with significant discrepancies that could affect significantly the calculations. Chickos' purpose is taken as reference because RPA is in-code data without reference and Sutton's book provides less information than Chickos' on his paper as it is a deep analysis of fuels.

RP-1 is an amalgam of hydrocarbonsthat, that is the reason why there is not a unique formula, furthermore, there are variations called RP-2, improved versions of RP-1, and RG-1, Soviet/Russian version. For this thesis, an approximation formed by only two hydrocarbons is done, these hydrocarbons are composed only of carbon and hydrogen, disregarding any sulphur or nitrogen trace, for simplicity.

The formula of the hydrocarbon that will be approximated is $C_{10\cdot 6}H_{21\cdot 7}$ and the ratio that will be maintained is the H/C ratio of 2.045. See the code in Section A.2.3 and the possible combination tested are:

- $0.6125 \cdot C_{10}H_{22} + 0.3875 \cdot C_{10}H_{18}$
- $0.8167 \cdot C_{10}H_{21} + 0.1833 \cdot C_{10}H_{18}$
- $0.5400 \cdot C_{12}H_{25} + 0.4600 \cdot C_{12}H_{24}$
- $0.8647 \cdot C_{12}H_{25} + 0.1353 \cdot C_{10}H_{18}$
- $0.2250 \cdot C_{10}H_{22} + 0.7750 \cdot C_{12}H_{24}$
- $0.4500 \cdot C_{10}H_{21} + 0.5500 \cdot C_{12}H_{24}$

These are the results obtained compared with the RP-1 results of RPA [7] that can be found in annex A.1.2:

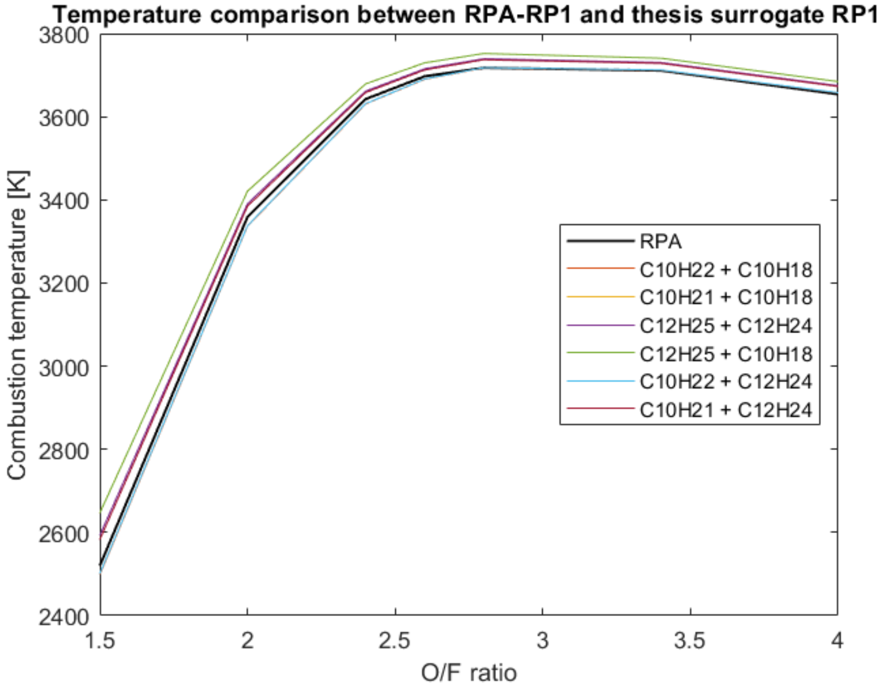


Figure 7.1: Temperature comparison

In terms of temperature, the best combination are $C_{10}H_{22}$ + $C_{10}H_{18}$ and $C_{10}H_{22}$ + $C_{12}H_{24}$. Both combinations have some discrepancies with resultant molar fraction with high ratios of oxidiser-fuel that can be appreciated in the annexes (An. A.1.1). According to Chickos, molar mass is approximately $148\frac{g}{mol}$ so $C_{10}H_{22}$ + $C_{10}H_{18}$ with $140.45\frac{g}{mol}$ is more accurate than $C_{10}H_{22}$ + $C_{12}H_{24}$ with $162.15\frac{g}{mol}$.

7.2 Analysis of HGSminG

`HGSminG` and all the minimisation algorithms are the relevant outcome, besides the *HGS* and *INIST* code. In Section 4.27, the scheme of this code is explained and this section will analyse the results of it comparing them with its counterpart `HGSeq` that use `fmincon`. The first analysis is to check the accuracy of the results for some equilibrium reactions that could be found in a combustion chamber of a rocket:

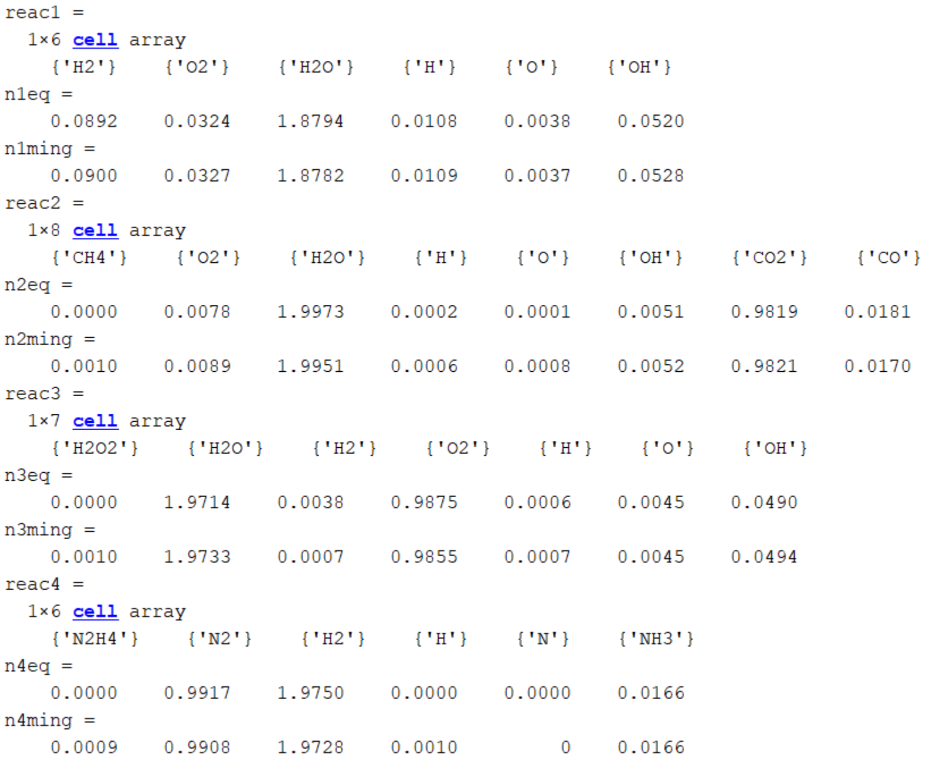
$$H_2 + O_2 + H_2O + H + O + OH \tag{7.1}$$

$$CH_4 + O_2 + H_2O + H + O + OH + CO_2 + CO \tag{7.2}$$

$$H_2O_2 + H_2 + O_2 + H_2O + H + O + OH \tag{7.3}$$

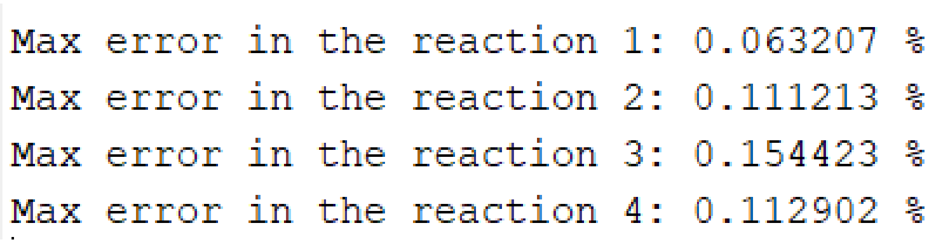
$$N_2H_4 + NH_3 + N_2 + H_2 \tag{7.4}$$

With the previous reaction, it has been tested comparing `HGSeq` and `HGSminG` with their maximum relative error for random temperatures.



```
reac1 =
  1×6 cell array
    {'H2'}    {'O2'}    {'H2O'}    {'H'}    {'O'}    {'OH'}
n1eq =
    0.0892    0.0324    1.8794    0.0108    0.0038    0.0520
n1ming =
    0.0900    0.0327    1.8782    0.0109    0.0037    0.0528
reac2 =
  1×8 cell array
    {'CH4'}    {'O2'}    {'H2O'}    {'H'}    {'O'}    {'OH'}    {'CO2'}    {'CO'}
n2eq =
    0.0000    0.0078    1.9973    0.0002    0.0001    0.0051    0.9819    0.0181
n2ming =
    0.0010    0.0089    1.9951    0.0006    0.0008    0.0052    0.9821    0.0170
reac3 =
  1×7 cell array
    {'H2O2'}    {'H2O'}    {'H2'}    {'O2'}    {'H'}    {'O'}    {'OH'}
n3eq =
    0.0000    1.9714    0.0038    0.9875    0.0006    0.0045    0.0490
n3ming =
    0.0010    1.9733    0.0007    0.9855    0.0007    0.0045    0.0494
reac4 =
  1×6 cell array
    {'N2H4'}    {'N2'}    {'H2'}    {'H'}    {'N'}    {'NH3'}
n4eq =
    0.0000    0.9917    1.9750    0.0000    0.0000    0.0166
n4ming =
    0.0009    0.9908    1.9728    0.0010         0    0.0166
```

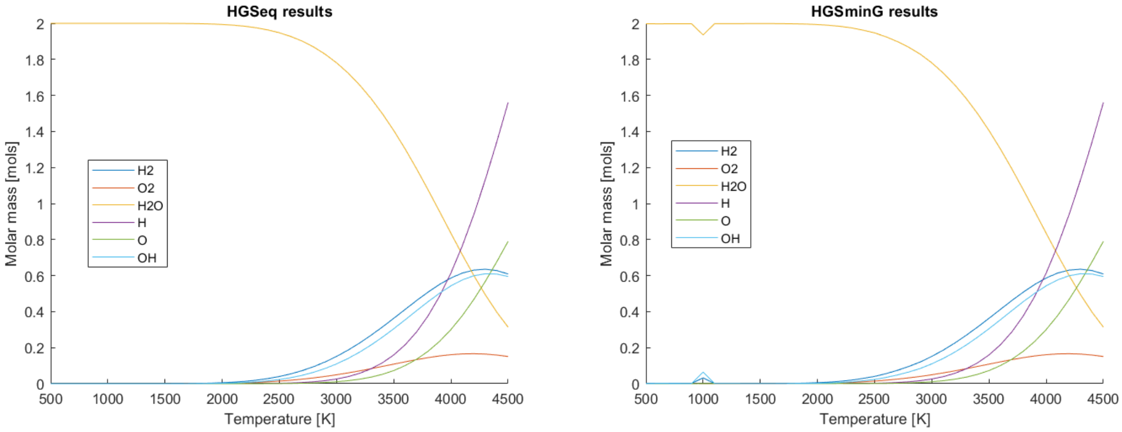
Figure 7.2: `HGSminG` vs `HGSeq`



```
Max error in the reaction 1: 0.063207 %
Max error in the reaction 2: 0.111213 %
Max error in the reaction 3: 0.154423 %
Max error in the reaction 4: 0.112902 %
```

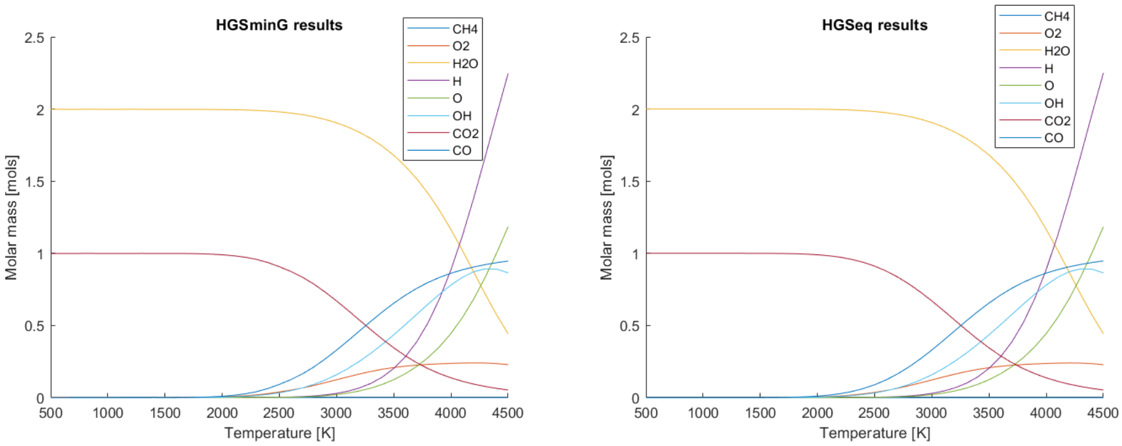
Figure 7.3: Maximum relative error

To avoid cherry-picking, it has been calculated a range of temperatures for each one, resulting in the following figures:



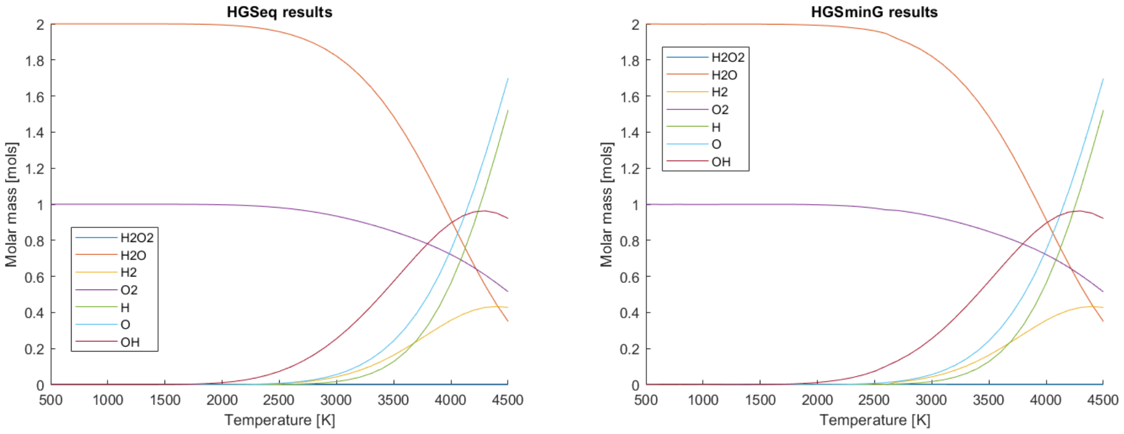
(a) `HGSeq` **(b)** `HGSminG`

Figure 7.4: Comparison for H_2 reaction



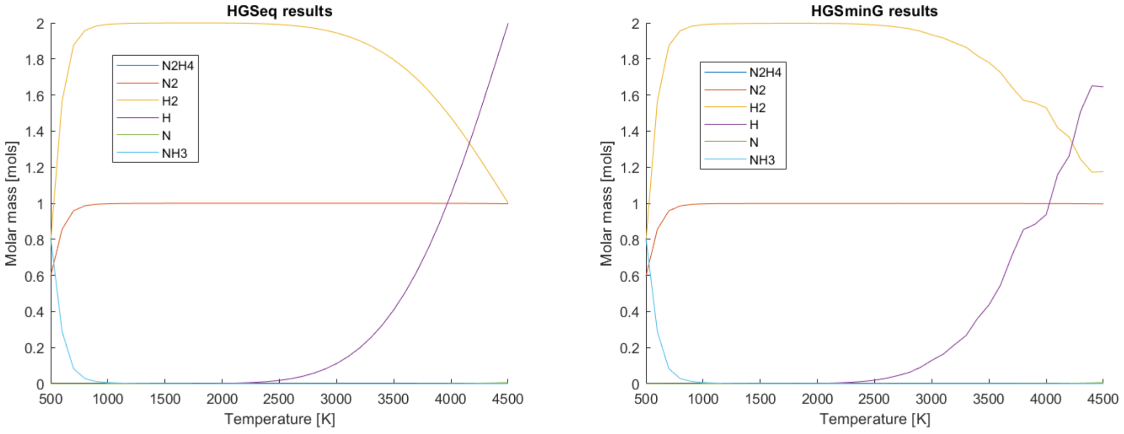
(a) `HGSeq` **(b)** `HGSminG`

Figure 7.5: Comparison for CH_4 reaction



(a) `HGSeq` **(b)** `HGSminG`

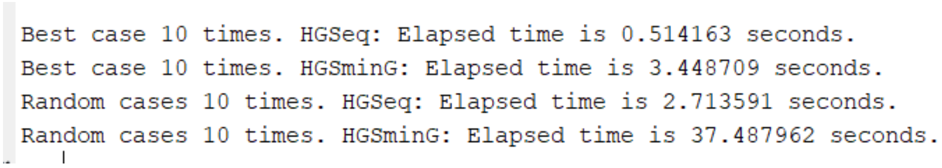
Figure 7.6: Comparison for H_2O_2 reaction



(a) `HGSeq` **(b)** `HGSminG`

Figure 7.7: Comparison for N_2H_4 reaction

As it can be seen, the minimisation with constraints developed is good enough even if there is a lack of accuracy for certain cases. The different cases are very sensible when a species is small enough to be in the range of the ϵ used to do the numerical acceptance, this should be thoughtfully considered in future iterations. Finally, after checking the accuracy, it is interesting to see the time performance difference even if it is not the point of this code. There is a clear disadvantage in time performance and it should be regarded in future iterations after solving the sensible cases. The time difference is due to the number of times that the minimisation algorithm requests the enthalpy from `HGSprop`. `HGSminG` algorithm request this property 20 times more than `fmincon`, which is more optimised.



```
Best case 10 times. HGSeq: Elapsed time is 0.514163 seconds.
Best case 10 times. HGSminG: Elapsed time is 3.448709 seconds.
Random cases 10 times. HGSeq: Elapsed time is 2.713591 seconds.
Random cases 10 times. HGSminG: Elapsed time is 37.487962 seconds.
```

Figure 7.8: Code comparison performance

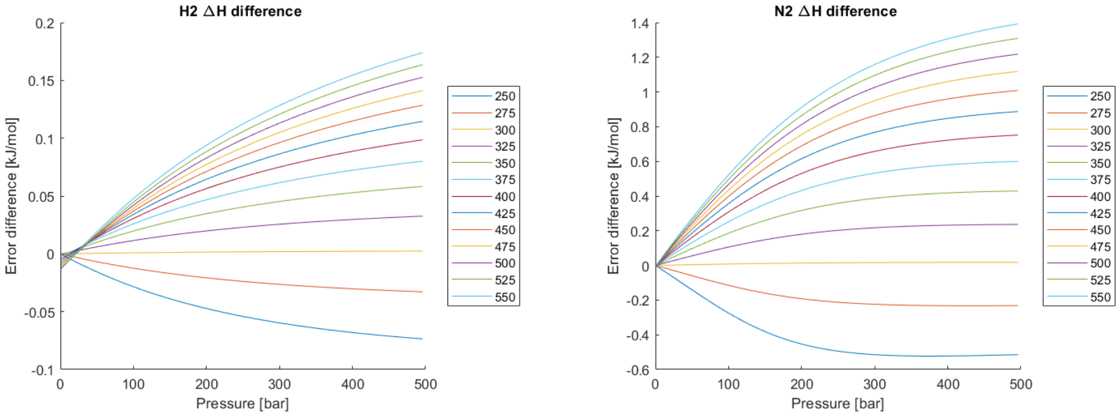
7.3 Effect of Pressure: Real fluid versus Ideal Fluid

During the development of this thesis, it has been pointed that pressure does not affect the enthalpy while in some case it might be incorrect, some lack of accuracy in the turbomachinery analysis could be an effect. It has been decided to check this topic with deeper insight.

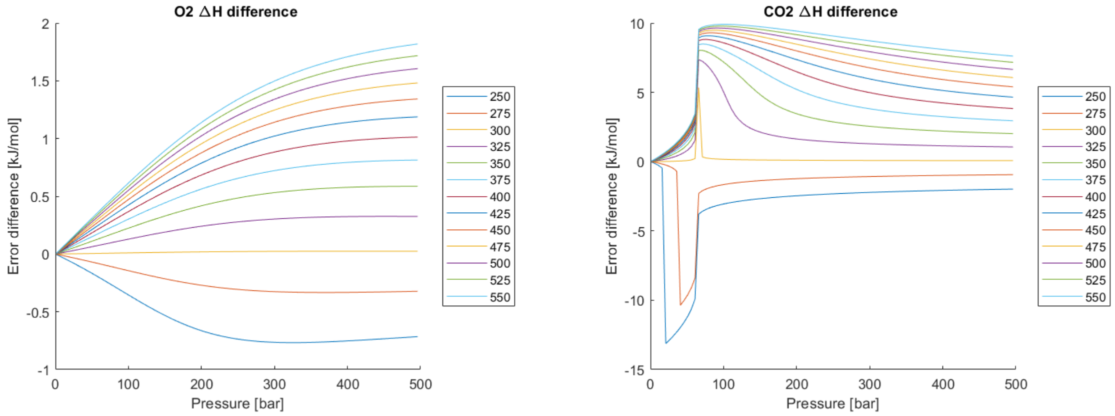
The main difference between `HGS` and `INIST` code lies on the fluid treatment being ideal and real respectively. As each code and elements have their own reference point, the comparison is carried out by variation of enthalpy. The species that are going to be used are:

- H_2
- O_2
- N_2
- CO_2
- CH_4
- C_2H_6

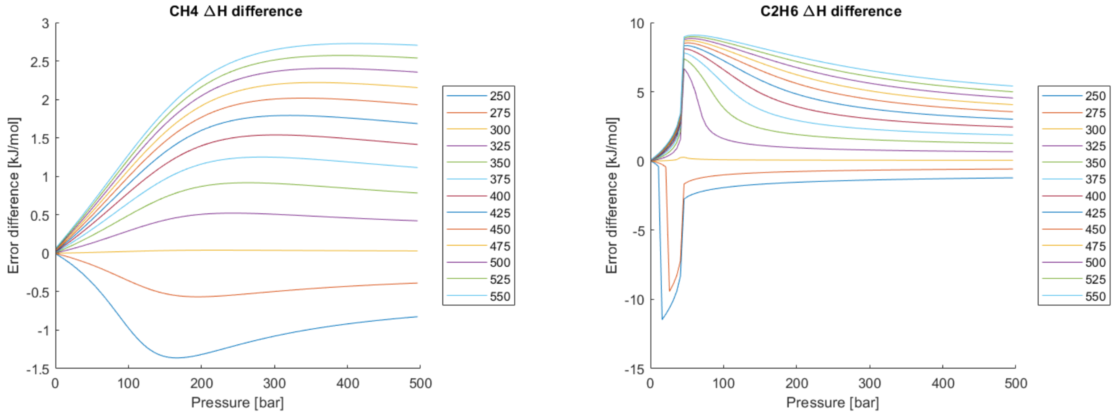
These species are compared in the range where both codes can provide data, so between 250 to 550 in steps of 25 K. Pressure calculation is done from 1 to 496 bars in steps of 5 bar. As the calculation of the variation of enthalpy requires a reference temperature and it wis used 298.15 K. The comparison between Δh are:



(a) H_2 **(b)** N_2



(c) O_2 **(d)** CO_2



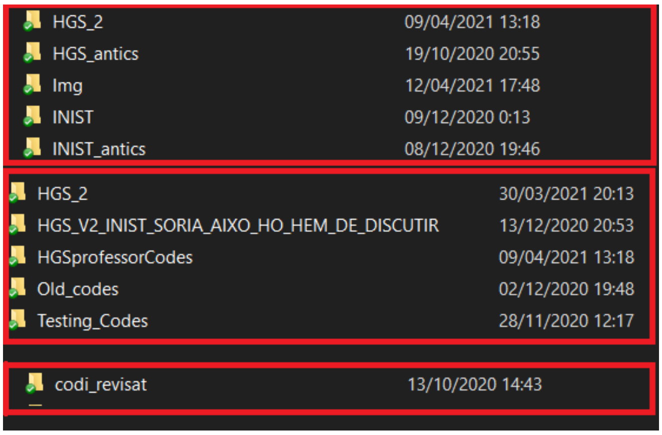
(e) CH_4 **(f)** C_2H_6

Figure 7.9: Error between ideal fluid and real fluid

At low pressure, the effect of it is negligible as difference is low per mol. However, when the pressure increases, and depending on the species, the effect grows to a certain point per mol. This could be one of the causes that altered the results in the previous chapters.

7.4 GitHub

Tracking the changes of the code is a quite challenging issue, therefore, this becomes vital when multiple editors are involved. One of the outcomes of this thesis is the use of `Github` and, as an extension, `SourceTree` for the tracking of the changes. During the development of the thesis, the first steps of control by the thesis director were chaotic as it can be seen in the figure below:



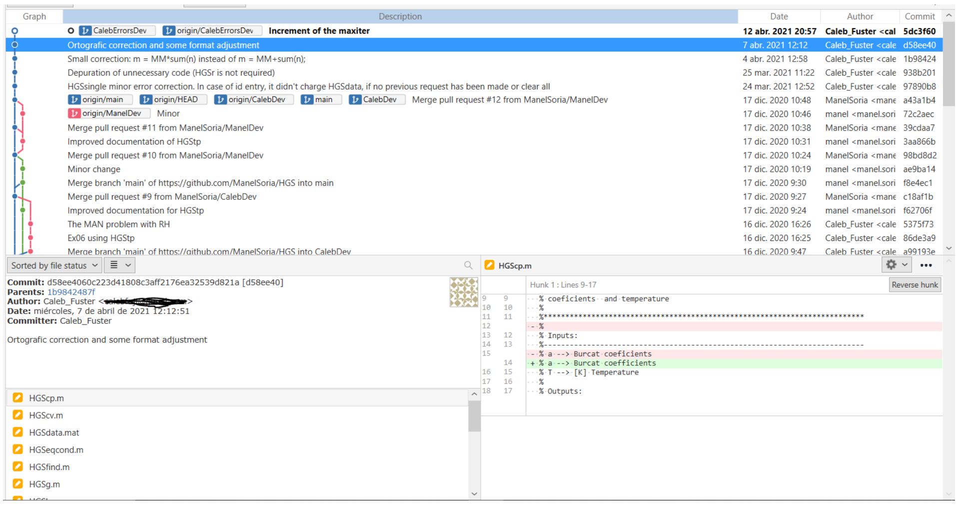
HGS_2	09/04/2021 13:18
HGS_antics	19/10/2020 20:55
Img	12/04/2021 17:48
INIST	09/12/2020 0:13
INIST_antics	08/12/2020 19:46

HGS_2	30/03/2021 20:13
HGS_V2_INIST_SORIA_AIXO_HO_HEM_DE_DISCUTIR	13/12/2020 20:53
HGSprofessorCodes	09/04/2021 13:18
Old_codes	02/12/2020 19:48
Testing_Codes	28/11/2020 12:17

codi_revisat	13/10/2020 14:43

Figure 7.10: Messy control of the versions

Most of the code is repeated, small changes become untracked as the editor will not create new folders and it can be confusing if multiple editors use this method. With the code uploaded in `Github`, `SourceTree` allows to track all the changes before merging it into the final code. Both codes were uploaded before the presentation of this thesis and used by some students during their lectures with Dr. Manel Soria. As it can see in the following figure, this software allows to track changes from the different editors:



Graph	Description	Date	Author	Commit
	CalebErrorsDev origin/CalebErrorsDev **Increment of the maxiter**	**12 abr. 2021 20:57**	**Caleb_Fuster <cal**	**5dc3f60**
	Ortografic correction and some format adjustment	7 abr. 2021 12:12	Caleb_Fuster <cale	d58ee40
	Small correction: m = MM*sum(n) instead of m = MM+sum(n);	4 abr. 2021 12:58	Caleb_Fuster <cale	1b98424
	Depuration of unnecessary code (HGSr is not required)	25 mar. 2021 11:22	Caleb_Fuster <cale	938b201
	HGSsingle minor error correction. In case of id entry, it didn't charge HGSdata, if no previous request has been made or clear all	24 mar. 2021 12:52	Caleb_Fuster <cale	97890b8
	origin/main origin/HEAD origin/CalebDev main CalebDev Merge pull request #12 from ManelSoria/ManelDev	17 dic. 2020 10:48	ManelSoria <mane	a43a1b4
	origin/ManelDev Minor	17 dic. 2020 10:46	manel <manel.sori	72c2aec
	Merge pull request #11 from ManelSoria/ManelDev	17 dic. 2020 10:38	ManelSoria <mane	39cdaa7
	Improved documentation of HGStp	17 dic. 2020 10:31	manel <manel.sori	3aa866b
	Merge pull request #10 from ManelSoria/ManelDev	17 dic. 2020 10:24	ManelSoria <mane	98bd8d2
	Minor change	17 dic. 2020 10:19	manel <manel.sori	ae9ba14
	Merge branch 'main' of https://github.com/ManelSoria/HGS into main	17 dic. 2020 9:30	manel <manel.sori	f8e4ec1
	Merge pull request #9 from ManelSoria/CalebDev	17 dic. 2020 9:27	ManelSoria <mane	c18af1b
	Improved documentation for HGStp	17 dic. 2020 9:24	manel <manel.sori	f62706f
	The MAN problem with RH	16 dic. 2020 16:26	Caleb_Fuster <cale	5375f73
	Ex06 using HGStp	16 dic. 2020 16:25	Caleb_Fuster <cale	86de3a9
	Merge branch 'main' of https://github.com/ManelSoria/HGS into CalebDev	16 dic. 2020 9:47	Caleb_Fuster <cale	a99193e

Commit: d58ee4060c223d41808c3aff2176ea32539d821a [d58ee40]
Parents: 1b9842487f
Author: Caleb_Fuster <[redacted]>
Date: miércoles, 7 de abril de 2021 12:12:51
Committer: Caleb_Fuster

Ortografic correction and some format adjustment

HGScp.m
HGScv.m
HGSdata.mat
HGSeqcond.m
HGSfind.m
HGSg.m

HGScp.m

Hunk 1 : Lines 9-17

```
   % coeficients  and temperature
   %
   %*********************************************************************
 - %
   % Inputs:
   %---------------------------------------------------------------------
 - % a --> Burcat coeficients
 + % a --> Burcat coefficients
   % T --> [K] Temperature
   %
   % Outputs:
```

Figure 7.11: Track changes with source tree

Some of the students and the professor, that are mentioned below, contributed to the code development either with *i*) new examples, or modification of the actual ones, *ii*) correction of minimal errors, *iii*) or code clarifications that were too obvious for the original writer and needs extra information.. Acknowledgement, from both codes, to: *i*) Manel Soria *ii*) Jordi Luque, *iii*) Mario Gayete, *iv*) Pau Nadal, *v*) and Paulino Gil.

Additionally, `Github` allows a bigger diffusion of these codes as it is a well-known site to find codes of any nature. This fact has provided a new example in `HGS` (`Ex10_The_MAN_problem.m`) from the doubts of a third-party entity.

CHAPTER 8

CONCLUSION

From the previous thesis development, it can be concluded that the main purposes of the thesis have been achieved. A review of Thermodynamic theory was done for real fluids and ideal fluids for `INIST` and `HGS`. Rocket engine analysis were carried out to validate the proper performance of the codes, obtaining good agreement with published results. A minimisation algorithm was developed to avoid `fmincon` dependence with good results. While this algorithm is not as fast as the original Matlab implementation, its code is easier to read and is useful to port the libraries to other languages if needed. Finally, some additional illustrative examples are considered with the RP1 surrogate model and the effect of the pressure on the enthalpy value.

Overall 66 `Matlab` functions with 7800 code lines were written, verified and documented. All the software is open-source and, like the previous versions of the libraries, it is expected to be used by students of Aerospace Engineering. The libraries allow users to write codes for the analysis of rocket engines (mainly liquid) but also other Thermochemical problems such as the example 'E10_The_MAN_problem.m', motivated by non-UPC user of the code.

Future enhancements of the present work could be: improving the user interface, allowing the use of the libraries without the need of writing any code; improving the database introducing NASA 9 polynomials that would expand their range of applicability; adding more turbomachinery examples; improving the graphic output of the code so that charts of rocket engines can be easily generated or the development of a heat transfer module allowing to perform a more detailed analysis of the nozzle cooling.

APPENDIX A

ANNEXES

A.1 RP-1

A.1.1 RP1 from RPA

RPA-RP1-O2-1.5

```
# Engine name: JDM
# lu. 12. apr. 13:26:37 2021
#
#****************************************************************************************************
# Propellant Specification
#----------------------------------------------------------------------------------------------------
#    Component  Temp.        Mass        Mole
#                [K]    fraction    fraction
#----------------------------------------------------------------------------------------------------
#         RP-1  350.0   0.4000000   0.6041719
#        O2(L)   90.0   0.6000000   0.3958281
#----------------------------------------------------------------------------------------------------
#              Total:   1.0000000   1.0000000
#----------------------------------------------------------------------------------------------------
#    Exploded formula: (O)0.792 (C)0.604 (H)1.178
#                O/F:   1.5000000
#              O/F O:   3.4056663 (stoichiometric)
#              alpha:   0.4404425 (oxidizer excess coefficient)
#----------------------------------------------------------------------------------------------------
#
# Table 1. Thermodynamic properties
#----------------------------------------------------------------------------------------------------
#          Parameter    Injector  Nozzle inl  Nozzle thr  Nozzle exi        Unit
#----------------------------------------------------------------------------------------------------
            Pressure      8.0000      0.0000      0.0000      0.0000         MPa
         Temperature   2519.6674      0.0000      0.0000      0.0000           K
            Enthalpy   -903.5924      0.0000      0.0000      0.0000       kJ/kg
             Entropy     12.2703      0.0000      0.0000      0.0000   kJ/(kg·K)
     Internal energy  -2088.9305      0.0000      0.0000      0.0000       kJ/kg
Specific heat (p=const)      2.3628      0.0000      0.0000      0.0000   kJ/(kg·K)
Specific heat (V=const)      1.8820      0.0000      0.0000      0.0000   kJ/(kg·K)
               Gamma      1.2554      0.0000      0.0000      0.0000
```

```
 Isentropic exponent      1.2548      0.0000      0.0000      0.0000
        Gas constant      0.4704      0.0000      0.0000      0.0000   kJ/(kg·K)
Molecular weight (M)     17.6740      0.0000      0.0000      0.0000
Molecular weight (MW)      0.0177      0.0000      0.0000      0.0000
             Density      6.7491      0.0000      0.0000      0.0000       kg/m³
      Sonic velocity   1219.5638      0.0000      0.0000      0.0000         m/s
           Viscosity      0.0001      0.0000      0.0000      0.0000    kg/(m·s)
Conductivity, frozen      0.3304      0.0000      0.0000      0.0000     W/(m·K)
Specific heat (p=const), frozen      2.2340      0.0000      0.0000      0.0000   kJ/(kg·K)
Prandtl number, frozen      0.5250      0.0000      0.0000      0.0000
Conductivity, effective      0.3806      0.0000      0.0000      0.0000     W/(m·K)
Specific heat (p=const), effective      2.3630      0.0000      0.0000      0.0000   kJ/(kg·K)
Prandtl number, effective      0.4821      0.0000      0.0000      0.0000
#----------------------------------------------------------------------------------------------------
#
#----------------------------------------------------------------------------------------------------
#
# Table 2. Fractions of the combustion products
#----------------------------------------------------------------------------------------------------
#           Species    Injector    Injector  Nozzle inl  Nozzle inl  Nozzle thr  Nozzle thr  Nozzle exi  Nozzle exi
#                    mass fract  mole fract  mass fract  mole fract  mass fract  mole fract  mass fract  mole fract
#----------------------------------------------------------------------------------------------------
                CH3   0.0000003   0.0000004   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                CH4   0.0000036   0.0000039   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                 CO   0.7582965   0.4784758   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                CO2   0.0680655   0.0273348   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
               COOH   0.0000015   0.0000006   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                  H   0.0001054   0.0018477   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                 H2   0.0413674   0.3626842   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                H2O   0.1319809   0.1294809   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
     HCHO,formaldehy   0.0000096   0.0000057   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                HCO   0.0000159   0.0000097   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
              HCOOH   0.0000048   0.0000018   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                  O   0.0000004   0.0000005   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                 OH   0.0001480   0.0001538   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
#----------------------------------------------------------------------------------------------------
#
#----------------------------------------------------------------------------------------------------
#
# Table 3. Theoretical (ideal) performance
#----------------------------------------------------------------------------------------------------
#
#----------------------------------------------------------------------------------------------------
#
# Table 4. Estimated delivered performance
#----------------------------------------------------------------------------------------------------
#
#<b>Ambient condition for optimum expansion:</b>  --
#
#
#----------------------------------------------------------------------------------------------------
#
#
#----------------------------------------------------------------------------------------------------
```

A.1.1.1 RPA-RP1-O2-2

```
# Engine name: JDM
# lu. 12. apr. 13:24:29 2021
#
#****************************************************************************************************
# Propellant Specification
#----------------------------------------------------------------------------------------------------
#    Component  Temp.        Mass        Mole
```

```
#                [K]    fraction    fraction
#---------------------------------------------------------------------------------------------------
#         RP-1  350.0   0.3333333   0.5337478
#        O2(L)   90.0   0.6666667   0.4662522
#---------------------------------------------------------------------------------------------------
#              Total:   1.0000000   1.0000000
#---------------------------------------------------------------------------------------------------
#    Exploded formula: (O)0.933 (C)0.534 (H)1.041
#                O/F:   2.0000000
#              O/F O:   3.4056663 (stoichiometric)
#              alpha:   0.5872566 (oxidizer excess coefficient)
#---------------------------------------------------------------------------------------------------
#
# Table 1. Thermodynamic properties
#---------------------------------------------------------------------------------------------------
#          Parameter    Injector  Nozzle inl  Nozzle thr  Nozzle exi        Unit
#---------------------------------------------------------------------------------------------------
            Pressure      8.0000      0.0000      0.0000      0.0000             MPa
         Temperature   3358.6350      0.0000      0.0000      0.0000               K
            Enthalpy   -819.5029      0.0000      0.0000      0.0000           kJ/kg
             Entropy     11.8146      0.0000      0.0000      0.0000       kJ/(kg·K)
     Internal energy  -2159.6587      0.0000      0.0000      0.0000           kJ/kg
Specific heat (p=const)      3.4583      0.0000      0.0000      0.0000       kJ/(kg·K)
Specific heat (V=const)      2.9106      0.0000      0.0000      0.0000       kJ/(kg·K)
               Gamma      1.1882      0.0000      0.0000      0.0000
 Isentropic exponent      1.1767      0.0000      0.0000      0.0000
        Gas constant      0.3990      0.0000      0.0000      0.0000       kJ/(kg·K)
Molecular weight (M)     20.8373      0.0000      0.0000      0.0000
Molecular weight (MW)      0.0208      0.0000      0.0000      0.0000
             Density      5.9695      0.0000      0.0000      0.0000           kg/m³
      Sonic velocity   1255.7463      0.0000      0.0000      0.0000             m/s
           Viscosity      0.0001      0.0000      0.0000      0.0000        kg/(m·s)
Conductivity, frozen      0.3631      0.0000      0.0000      0.0000         W/(m·K)
Specific heat (p=const), frozen      2.1490      0.0000      0.0000      0.0000       kJ/(kg·K)
Prandtl number, frozen      0.5974      0.0000      0.0000      0.0000
Conductivity, effective      0.8209      0.0000      0.0000      0.0000         W/(m·K)
Specific heat (p=const), effective      3.4580      0.0000      0.0000      0.0000       kJ/(kg·K)
Prandtl number, effective      0.4252      0.0000      0.0000      0.0000
#---------------------------------------------------------------------------------------------------
#
#---------------------------------------------------------------------------------------------------
#
# Table 2. Fractions of the combustion products
#---------------------------------------------------------------------------------------------------
#            Species    Injector    Injector  Nozzle inl  Nozzle inl  Nozzle thr  Nozzle thr  Nozzle exi  Nozzle exi
#                     mass fract  mole fract  mass fract  mole fract  mass fract  mole fract  mass fract  mole fract
#---------------------------------------------------------------------------------------------------
                  CO   0.5645024   0.4199459   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                 CO2   0.1625696   0.0769724   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                COOH   0.0000218   0.0000101   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                   H   0.0009950   0.0205707   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                  H2   0.0182496   0.1886384   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                 H2O   0.2409259   0.2786664   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                H2O2   0.0000025   0.0000015   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
     HCHO,formaldehy   0.0000038   0.0000027   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                 HCO   0.0000527   0.0000379   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
               HCOOH   0.0000079   0.0000036   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                 HO2   0.0000063   0.0000040   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                   O   0.0006340   0.0008257   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                  O2   0.0007255   0.0004724   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                  OH   0.0113029   0.0138483   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
#---------------------------------------------------------------------------------------------------
#
#---------------------------------------------------------------------------------------------------
#
```

```
# Table 3. Theoretical (ideal) performance
#---------------------------------------------------------------------------------------------------
#
#---------------------------------------------------------------------------------------------------
#
# Table 4. Estimated delivered performance
#---------------------------------------------------------------------------------------------------
#
#<b>Ambient condition for optimum expansion:</b>  --
#
#
#---------------------------------------------------------------------------------------------------
#
#
#---------------------------------------------------------------------------------------------------
```

A.1.1.2 RPA-RP1-O2-2.4

```
# Engine name: JDM
# lu. 12. apr. 13:36:22 2021
#
#***************************************************************************************************
# Propellant Specification
#---------------------------------------------------------------------------------------------------
#    Component  Temp.        Mass        Mole
#               [K]    fraction    fraction
#---------------------------------------------------------------------------------------------------
#         RP-1  350.0   0.2941176   0.4882209
#        O2(L)   90.0   0.7058824   0.5117791
#---------------------------------------------------------------------------------------------------
#             Total:   1.0000000   1.0000000
#---------------------------------------------------------------------------------------------------
#    Exploded formula: (O)1.024 (C)0.488 (H)0.952
#               O/F:   2.4000000
#             O/F 0:   3.4056663 (stoichiometric)
#             alpha:   0.7047079 (oxidizer excess coefficient)
#---------------------------------------------------------------------------------------------------
#
# Table 1. Thermodynamic properties
#---------------------------------------------------------------------------------------------------
#          Parameter    Injector  Nozzle inl  Nozzle thr  Nozzle exi        Unit
#---------------------------------------------------------------------------------------------------
            Pressure      8.0000      0.0000      0.0000      0.0000         MPa
         Temperature   3642.3558      0.0000      0.0000      0.0000           K
            Enthalpy   -770.0385      0.0000      0.0000      0.0000       kJ/kg
             Entropy     11.4070      0.0000      0.0000      0.0000   kJ/(kg·K)
     Internal energy  -2100.9757      0.0000      0.0000      0.0000       kJ/kg
Specific heat (p=const)      5.4213      0.0000      0.0000      0.0000   kJ/(kg·K)
Specific heat (V=const)      4.5967      0.0000      0.0000      0.0000   kJ/(kg·K)
               Gamma      1.1794      0.0000      0.0000      0.0000
 Isentropic exponent      1.1446      0.0000      0.0000      0.0000
        Gas constant      0.3654      0.0000      0.0000      0.0000   kJ/(kg·K)
Molecular weight (M)     22.7541      0.0000      0.0000      0.0000
Molecular weight (MW)      0.0227      0.0000      0.0000      0.0000
             Density      6.0108      0.0000      0.0000      0.0000       kg/m³
      Sonic velocity   1234.2467      0.0000      0.0000      0.0000         m/s
           Viscosity      0.0001      0.0000      0.0000      0.0000    kg/(m·s)
Conductivity, frozen      0.3612      0.0000      0.0000      0.0000     W/(m·K)
Specific heat (p=const), frozen      2.0730      0.0000      0.0000      0.0000   kJ/(kg·K)
Prandtl number, frozen      0.6355      0.0000      0.0000      0.0000
Conductivity, effective      1.2510      0.0000      0.0000      0.0000     W/(m·K)
Specific heat (p=const), effective      5.4210      0.0000      0.0000      0.0000   kJ/(kg·K)
Prandtl number, effective      0.4799      0.0000      0.0000      0.0000
#---------------------------------------------------------------------------------------------------
```

```
#
#---------------------------------------------------------------------------------------------------
#
# Table 2. Fractions of the combustion products
#---------------------------------------------------------------------------------------------------
#            Species    Injector    Injector  Nozzle inl  Nozzle inl  Nozzle thr  Nozzle thr  Nozzle exi  Nozzle exi
#                     mass fract  mole fract  mass fract  mole fract  mass fract  mole fract  mass fract  mole fract
#---------------------------------------------------------------------------------------------------
                  CO   0.4311774   0.3502683   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                 CO2   0.2485598   0.1285121   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                COOH   0.0000406   0.0000205   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                   H   0.0012889   0.0290975   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                  H2   0.0092283   0.1041632   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                 H2O   0.2565968   0.3240931   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                H2O2   0.0000156   0.0000104   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
     HCHO,formaldehy   0.0000016   0.0000012   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                 HCO   0.0000459   0.0000360   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
               HCOOH   0.0000072   0.0000035   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                 HO2   0.0000804   0.0000554   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                   O   0.0050602   0.0071965   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                  O2   0.0120283   0.0085532   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                  OH   0.0358687   0.0479887   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
#---------------------------------------------------------------------------------------------------
#
#---------------------------------------------------------------------------------------------------
#
# Table 3. Theoretical (ideal) performance
#---------------------------------------------------------------------------------------------------
#
#---------------------------------------------------------------------------------------------------
#
# Table 4. Estimated delivered performance
#---------------------------------------------------------------------------------------------------
#
#<b>Ambient condition for optimum expansion:</b>  --
#
#
#---------------------------------------------------------------------------------------------------
#
#
#---------------------------------------------------------------------------------------------------
```

A.1.1.3 RPA-RP1-O2-2.6

```
# Engine name: JDM
# lu. 12. apr. 13:37:03 2021
#
#***************************************************************************************************
# Propellant Specification
#---------------------------------------------------------------------------------------------------
#    Component  Temp.        Mass        Mole
#                 [K]    fraction    fraction
#---------------------------------------------------------------------------------------------------
#         RP-1  350.0   0.2777778   0.4682508
#        O2(L)   90.0   0.7222222   0.5317492
#---------------------------------------------------------------------------------------------------
#             Total:   1.0000000   1.0000000
#---------------------------------------------------------------------------------------------------
#    Exploded formula: (O)1.063 (C)0.468 (H)0.913
#               O/F:   2.6000000
#             O/F 0:   3.4056663 (stoichiometric)
#             alpha:   0.7634336 (oxidizer excess coefficient)
#---------------------------------------------------------------------------------------------------
#
```

```
# Table 1. Thermodynamic properties
#---------------------------------------------------------------------------------------------------
#          Parameter    Injector  Nozzle inl  Nozzle thr  Nozzle exi        Unit
#---------------------------------------------------------------------------------------------------
            Pressure      8.0000      0.0000      0.0000      0.0000         MPa
         Temperature   3697.1131      0.0000      0.0000      0.0000           K
            Enthalpy   -749.4283      0.0000      0.0000      0.0000       kJ/kg
             Entropy     11.2156      0.0000      0.0000      0.0000   kJ/(kg·K)
     Internal energy  -2057.0522      0.0000      0.0000      0.0000       kJ/kg
Specific heat (p=const)      6.2643      0.0000      0.0000      0.0000   kJ/(kg·K)
Specific heat (V=const)      5.2897      0.0000      0.0000      0.0000   kJ/(kg·K)
               Gamma      1.1842      0.0000      0.0000      0.0000
 Isentropic exponent      1.1380      0.0000      0.0000      0.0000
        Gas constant      0.3537      0.0000      0.0000      0.0000   kJ/(kg·K)
Molecular weight (M)     23.5079      0.0000      0.0000      0.0000
Molecular weight (MW)      0.0235      0.0000      0.0000      0.0000
             Density      6.1180      0.0000      0.0000      0.0000      kg/m^3
      Sonic velocity   1219.8759      0.0000      0.0000      0.0000         m/s
           Viscosity      0.0001      0.0000      0.0000      0.0000    kg/(m·s)
Conductivity, frozen      0.3562      0.0000      0.0000      0.0000     W/(m·K)
Specific heat (p=const), frozen      2.0360      0.0000      0.0000      0.0000   kJ/(kg·K)
Prandtl number, frozen      0.6476      0.0000      0.0000      0.0000
Conductivity, effective      1.4160      0.0000      0.0000      0.0000     W/(m·K)
Specific heat (p=const), effective      6.2640      0.0000      0.0000      0.0000   kJ/(kg·K)
Prandtl number, effective      0.5015      0.0000      0.0000      0.0000
#---------------------------------------------------------------------------------------------------
#
#---------------------------------------------------------------------------------------------------
#
# Table 2. Fractions of the combustion products
#---------------------------------------------------------------------------------------------------
#            Species    Injector    Injector  Nozzle inl  Nozzle inl  Nozzle thr  Nozzle thr  Nozzle exi  Nozzle exi
#                     mass fract  mole fract  mass fract  mole fract  mass fract  mole fract  mass fract  mole fract
#---------------------------------------------------------------------------------------------------
                  CO   0.3765599   0.3160341   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                 CO2   0.2829323   0.1511300   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                COOH   0.0000443   0.0000232   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                   H   0.0012273   0.0286237   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                  H2   0.0068953   0.0804088   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                 H2O   0.2520267   0.3288668   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                H2O2   0.0000241   0.0000166   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
     HCHO,formaldehy   0.0000011   0.0000009   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                 HCO   0.0000380   0.0000308   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
               HCOOH   0.0000064   0.0000033   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                 HO2   0.0001520   0.0001082   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                   O   0.0082595   0.0121357   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                  O2   0.0257444   0.0189132   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                  O3   0.0000003   0.0000001   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                  OH   0.0460883   0.0637043   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
#---------------------------------------------------------------------------------------------------
#
#---------------------------------------------------------------------------------------------------
#
# Table 3. Theoretical (ideal) performance
#---------------------------------------------------------------------------------------------------
#
#---------------------------------------------------------------------------------------------------
#
# Table 4. Estimated delivered performance
#---------------------------------------------------------------------------------------------------
#
#<b>Ambient condition for optimum expansion:</b>  --
#
#
#---------------------------------------------------------------------------------------------------
```

```
#
#
#---------------------------------------------------------------------------------------------------
```

A.1.1.4 RPA-RP1-O2-2.8

```
# Engine name: JDM
# lu. 12. apr. 11:04:27 2021
#
#***************************************************************************************************
# Propellant Specification
#---------------------------------------------------------------------------------------------------
#    Component  Temp.        Mass        Mole
#                 [K]    fraction    fraction
#---------------------------------------------------------------------------------------------------
#         RP-1  298.0   0.2631579   0.4498503
#        O2(L)   90.0   0.7368421   0.5501497
#---------------------------------------------------------------------------------------------------
#             Total:   1.0000000   1.0000000
#---------------------------------------------------------------------------------------------------
#    Exploded formula: (O)1.100 (C)0.450 (H)0.877
#                O/F:   2.8000000
#              O/F 0:   3.4056663 (stoichiometric)
#              alpha:   0.8221592 (oxidizer excess coefficient)
#---------------------------------------------------------------------------------------------------
#
# Table 1. Thermodynamic properties
#---------------------------------------------------------------------------------------------------
#          Parameter    Injector  Nozzle inl  Nozzle thr  Nozzle exi        Unit
#---------------------------------------------------------------------------------------------------
            Pressure      8.0000      0.0000      0.0000      0.0000          MPa
         Temperature   3717.5388      0.0000      0.0000      0.0000            K
            Enthalpy   -756.8740      0.0000      0.0000      0.0000        kJ/kg
             Entropy     11.0289      0.0000      0.0000      0.0000    kJ/(kg·K)
     Internal energy  -2035.0015      0.0000      0.0000      0.0000        kJ/kg
Specific heat (p=const)      6.7703      0.0000      0.0000      0.0000    kJ/(kg·K)
Specific heat (V=const)      5.6983      0.0000      0.0000      0.0000    kJ/(kg·K)
               Gamma      1.1881      0.0000      0.0000      0.0000
 Isentropic exponent      1.1344      0.0000      0.0000      0.0000
        Gas constant      0.3438      0.0000      0.0000      0.0000    kJ/(kg·K)
Molecular weight (M)     24.1833      0.0000      0.0000      0.0000
Molecular weight (MW)      0.0242      0.0000      0.0000      0.0000
             Density      6.2592      0.0000      0.0000      0.0000        kg/m³
      Sonic velocity   1204.0948      0.0000      0.0000      0.0000          m/s
           Viscosity      0.0001      0.0000      0.0000      0.0000     kg/(m·s)
Conductivity, frozen      0.3498      0.0000      0.0000      0.0000      W/(m·K)
Specific heat (p=const), frozen      2.0010      0.0000      0.0000      0.0000    kJ/(kg·K)
Prandtl number, frozen      0.6569      0.0000      0.0000      0.0000
Conductivity, effective      1.5110      0.0000      0.0000      0.0000      W/(m·K)
Specific heat (p=const), effective      6.7700      0.0000      0.0000      0.0000    kJ/(kg·K)
Prandtl number, effective      0.5144      0.0000      0.0000      0.0000
#---------------------------------------------------------------------------------------------------
#
#---------------------------------------------------------------------------------------------------
#
# Table 2. Fractions of the combustion products
#---------------------------------------------------------------------------------------------------
#            Species    Injector    Injector  Nozzle inl  Nozzle inl  Nozzle thr  Nozzle thr  Nozzle exi  Nozzle exi
#                     mass fract  mole fract  mass fract  mole fract  mass fract  mole fract  mass fract  mole fract
#---------------------------------------------------------------------------------------------------
                  CO   0.3293180   0.2843262   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                 CO2   0.3111347   0.1709692   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                COOH   0.0000448   0.0000241   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                   H   0.0011075   0.0265711   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
```

```
                  H2   0.0053179   0.0637954   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                 H2O   0.2451589   0.3290961   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                H2O2   0.0000315   0.0000224   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
     HCHO,formaldehy   0.0000008   0.0000006   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                 HCO   0.0000304   0.0000253   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
               HCOOH   0.0000056   0.0000029   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                 HO2   0.0002295   0.0001682   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                   O   0.0110920   0.0167658   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                  O2   0.0435678   0.0329267   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                  O3   0.0000006   0.0000003   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                  OH   0.0529599   0.0753055   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
#---------------------------------------------------------------------------------------------------
#
#---------------------------------------------------------------------------------------------------
#
# Table 3. Theoretical (ideal) performance
#---------------------------------------------------------------------------------------------------
#
#---------------------------------------------------------------------------------------------------
#
# Table 4. Estimated delivered performance
#---------------------------------------------------------------------------------------------------
#
#<b>Ambient condition for optimum expansion:</b>  --
#
#
#---------------------------------------------------------------------------------------------------
#
#
#---------------------------------------------------------------------------------------------------
```

A.1.1.5 RPA-RP1-O2-3.4

```
# Engine name: JDM
# lu. 12. apr. 13:48:54 2021
#
#***************************************************************************************************
# Propellant Specification
#---------------------------------------------------------------------------------------------------
#    Component  Temp.        Mass        Mole
#                [K]    fraction    fraction
#---------------------------------------------------------------------------------------------------
#         RP-1  350.0   0.2272727   0.4024104
#        O2(L)   90.0   0.7727273   0.5975896
#---------------------------------------------------------------------------------------------------
#             Total:   1.0000000   1.0000000
#---------------------------------------------------------------------------------------------------
#    Exploded formula: (O)1.195 (C)0.402 (H)0.785
#                O/F:   3.4000000
#              O/F 0:   3.4056663 (stoichiometric)
#              alpha:   0.9983362 (oxidizer excess coefficient)
#---------------------------------------------------------------------------------------------------
#
# Table 1. Thermodynamic properties
#---------------------------------------------------------------------------------------------------
#          Parameter    Injector  Nozzle inl  Nozzle thr  Nozzle exi        Unit
#---------------------------------------------------------------------------------------------------
            Pressure      8.0000      0.0000      0.0000      0.0000         MPa
         Temperature   3711.0357      0.0000      0.0000      0.0000           K
            Enthalpy   -685.7242      0.0000      0.0000      0.0000       kJ/kg
             Entropy     10.5660      0.0000      0.0000      0.0000   kJ/(kg·K)
     Internal energy  -1884.5004      0.0000      0.0000      0.0000       kJ/kg
Specific heat (p=const)      7.0409      0.0000      0.0000      0.0000   kJ/(kg·K)
Specific heat (V=const)      5.9139      0.0000      0.0000      0.0000   kJ/(kg·K)
```

```
               Gamma      1.1906      0.0000      0.0000      0.0000
 Isentropic exponent      1.1302      0.0000      0.0000      0.0000
        Gas constant      0.3230      0.0000      0.0000      0.0000   kJ/(kg·K)
Molecular weight (M)     25.7390      0.0000      0.0000      0.0000
Molecular weight (MW)      0.0257      0.0000      0.0000      0.0000
             Density      6.6735      0.0000      0.0000      0.0000       kg/m³
      Sonic velocity   1163.9615      0.0000      0.0000      0.0000         m/s
           Viscosity      0.0001      0.0000      0.0000      0.0000    kg/(m·s)
Conductivity, frozen      0.3304      0.0000      0.0000      0.0000     W/(m·K)
Specific heat (p=const), frozen      1.9110      0.0000      0.0000      0.0000   kJ/(kg·K)
Prandtl number, frozen      0.6736      0.0000      0.0000      0.0000
Conductivity, effective      1.5500      0.0000      0.0000      0.0000     W/(m·K)
Specific heat (p=const), effective      7.0410      0.0000      0.0000      0.0000   kJ/(kg·K)
Prandtl number, effective      0.5289      0.0000      0.0000      0.0000
#---------------------------------------------------------------------------------------------------
#
#---------------------------------------------------------------------------------------------------
#
# Table 2. Fractions of the combustion products
#---------------------------------------------------------------------------------------------------
#            Species    Injector    Injector  Nozzle inl  Nozzle inl  Nozzle thr  Nozzle thr  Nozzle exi  Nozzle exi
#                     mass fract  mole fract  mass fract  mole fract  mass fract  mole fract  mass fract  mole fract
#---------------------------------------------------------------------------------------------------
                  CO   0.2271201   0.2087049   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                 CO2   0.3587404   0.2098097   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                COOH   0.0000387   0.0000221   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                   H   0.0007770   0.0198408   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                  H2   0.0028598   0.0365148   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                 H2O   0.2202021   0.3146097   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                H2O2   0.0000463   0.0000351   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
     HCHO,formaldehy   0.0000003   0.0000003   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                 HCO   0.0000157   0.0000140   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
               HCOOH   0.0000037   0.0000021   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                 HO2   0.0004420   0.0003447   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                   O   0.0169108   0.0272052   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                  O2   0.1109717   0.0892627   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                  O3   0.0000024   0.0000013   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                  OH   0.0618689   0.0936327   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
#---------------------------------------------------------------------------------------------------
#
#---------------------------------------------------------------------------------------------------
#
# Table 3. Theoretical (ideal) performance
#---------------------------------------------------------------------------------------------------
#
#---------------------------------------------------------------------------------------------------
#
# Table 4. Estimated delivered performance
#---------------------------------------------------------------------------------------------------
#
#<b>Ambient condition for optimum expansion:</b>  --
#
#
#---------------------------------------------------------------------------------------------------
#
#
#---------------------------------------------------------------------------------------------------
```

A.1.1.6 RPA-RP1-O2-4

```
# Engine name: JDM
# lu. 12. apr. 13:22:29 2021
#
#***************************************************************************************************
```

```
# Propellant Specification
#---------------------------------------------------------------------------------------------------
#    Component  Temp.        Mass        Mole
#               [K]    fraction    fraction
#---------------------------------------------------------------------------------------------------
#         RP-1  350.0   0.2000000   0.3640218
#        O2(L)   90.0   0.8000000   0.6359782
#---------------------------------------------------------------------------------------------------
#             Total:   1.0000000   1.0000000
#---------------------------------------------------------------------------------------------------
#    Exploded formula: (O)1.272 (C)0.364 (H)0.710
#               O/F:   4.0000000
#             O/F 0:   3.4056663 (stoichiometric)
#             alpha:   1.1745132 (oxidizer excess coefficient)
#---------------------------------------------------------------------------------------------------
#
# Table 1. Thermodynamic properties
#---------------------------------------------------------------------------------------------------
#          Parameter    Injector  Nozzle inl  Nozzle thr  Nozzle exi        Unit
#---------------------------------------------------------------------------------------------------
            Pressure      8.0000      0.0000      0.0000      0.0000         MPa
         Temperature   3653.9636      0.0000      0.0000      0.0000           K
            Enthalpy   -651.3239      0.0000      0.0000      0.0000       kJ/kg
             Entropy     10.1849      0.0000      0.0000      0.0000   kJ/(kg·K)
     Internal energy  -1779.1148      0.0000      0.0000      0.0000       kJ/kg
Specific heat (p=const)      6.5231      0.0000      0.0000      0.0000   kJ/(kg·K)
Specific heat (V=const)      5.5075      0.0000      0.0000      0.0000   kJ/(kg·K)
               Gamma      1.1844      0.0000      0.0000      0.0000
 Isentropic exponent      1.1291      0.0000      0.0000      0.0000
        Gas constant      0.3086      0.0000      0.0000      0.0000   kJ/(kg·K)
Molecular weight (M)     26.9383      0.0000      0.0000      0.0000
Molecular weight (MW)      0.0269      0.0000      0.0000      0.0000
             Density      7.0935      0.0000      0.0000      0.0000       kg/m³
      Sonic velocity   1128.4467      0.0000      0.0000      0.0000         m/s
           Viscosity      0.0001      0.0000      0.0000      0.0000    kg/(m·s)
Conductivity, frozen      0.3119      0.0000      0.0000      0.0000     W/(m·K)
Specific heat (p=const), frozen      1.8400      0.0000      0.0000      0.0000   kJ/(kg·K)
Prandtl number, frozen      0.6837      0.0000      0.0000      0.0000
Conductivity, effective      1.4080      0.0000      0.0000      0.0000     W/(m·K)
Specific heat (p=const), effective      6.5230      0.0000      0.0000      0.0000   kJ/(kg·K)
Prandtl number, effective      0.5369      0.0000      0.0000      0.0000
#---------------------------------------------------------------------------------------------------
#
#---------------------------------------------------------------------------------------------------
#
# Table 2. Fractions of the combustion products
#---------------------------------------------------------------------------------------------------
#            Species    Injector    Injector  Nozzle inl  Nozzle inl  Nozzle thr  Nozzle thr  Nozzle exi  Nozzle exi
#                     mass fract  mole fract  mass fract  mole fract  mass fract  mole fract  mass fract  mole fract
#---------------------------------------------------------------------------------------------------
                  CO   0.1600243   0.1539011   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                 CO2   0.3783041   0.2315608   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                COOH   0.0000298   0.0000178   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                   H   0.0005270   0.0140859   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                  H2   0.0017403   0.0232554   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                 H2O   0.1988047   0.2972734   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                H2O2   0.0000516   0.0000408   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
     HCHO,formaldehy   0.0000001   0.0000001   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                 HCO   0.0000082   0.0000076   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
               HCOOH   0.0000024   0.0000014   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                 HO2   0.0005717   0.0004666   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                   O   0.0185241   0.0311891   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                  O2   0.1808088   0.1522146   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                  O3   0.0000046   0.0000026   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
                  OH   0.0605981   0.0959827   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
```

```
#---------------------------------------------------------------------------------------------------
#
#---------------------------------------------------------------------------------------------------
#
# Table 3. Theoretical (ideal) performance
#---------------------------------------------------------------------------------------------------
#
#---------------------------------------------------------------------------------------------------
#
# Table 4. Estimated delivered performance
#---------------------------------------------------------------------------------------------------
#
#<b>Ambient condition for optimum expansion:</b>  --
#
#
#---------------------------------------------------------------------------------------------------
#
#
#---------------------------------------------------------------------------------------------------
```

A.1.2 Molar fraction comparisons

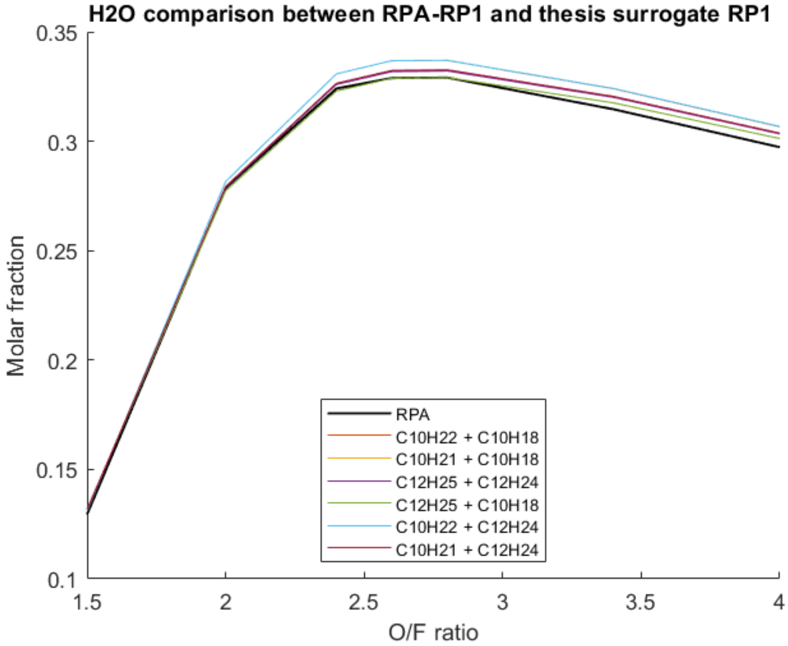


Figure A.1: H2O comparison

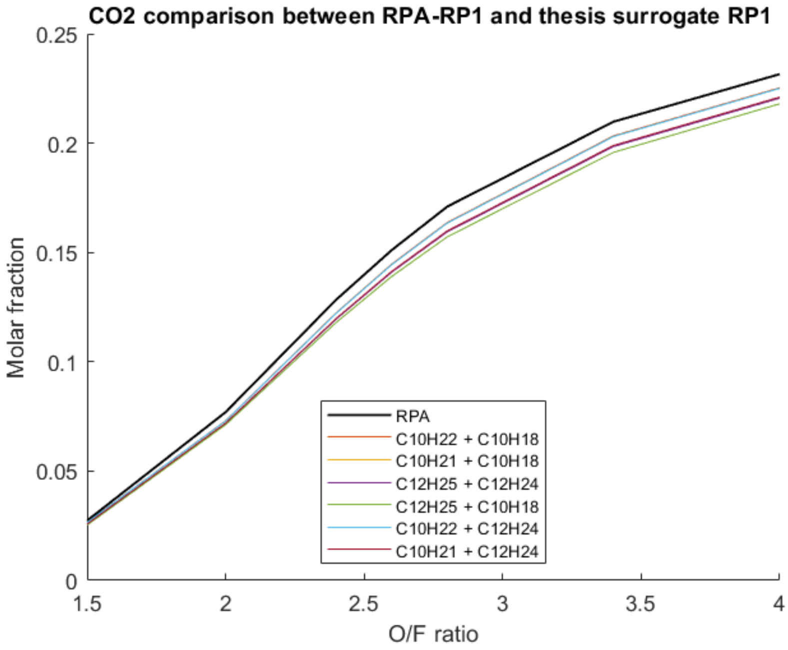


Figure A.2: CO_2 comparison

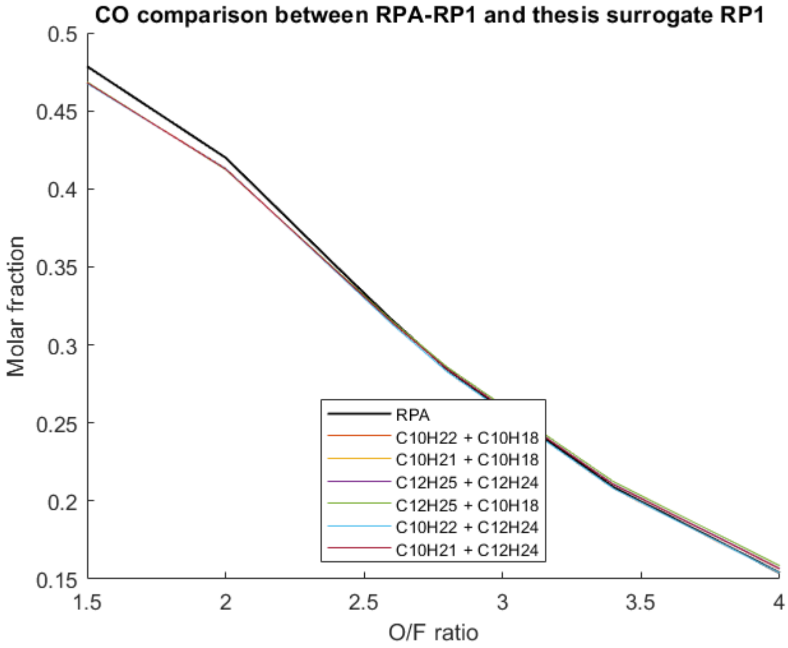


Figure A.3: CO comparison

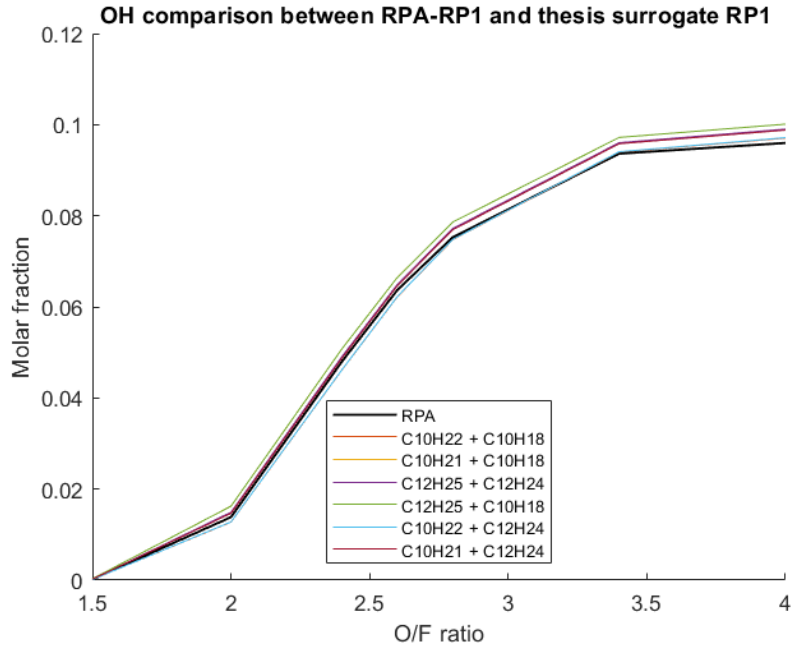


Figure A.4: H2O comparison

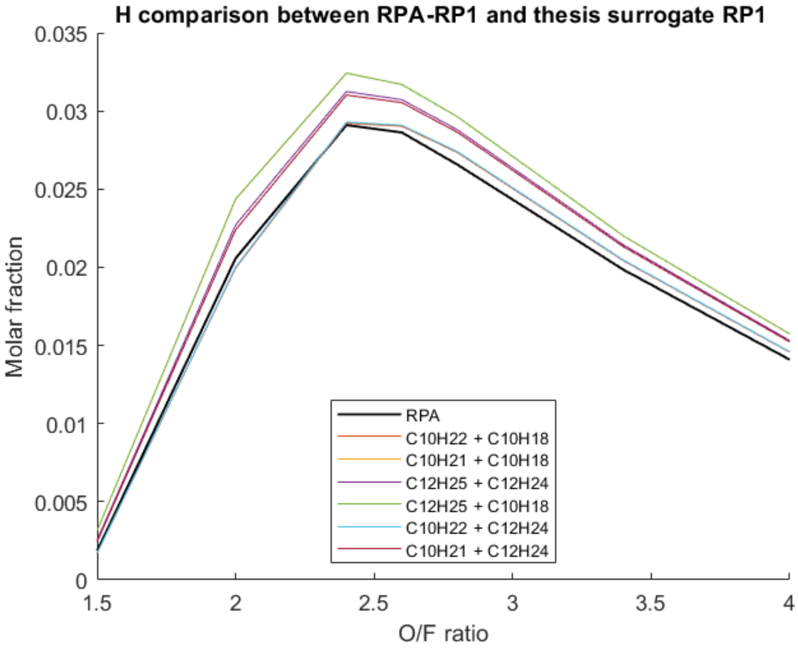


Figure A.5: H comparison

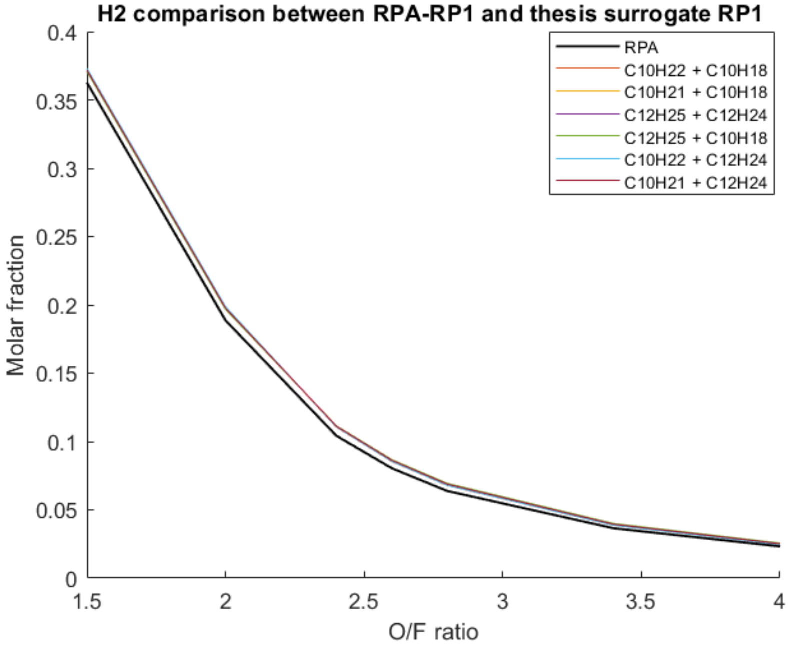


Figure A.6: H2 comparison

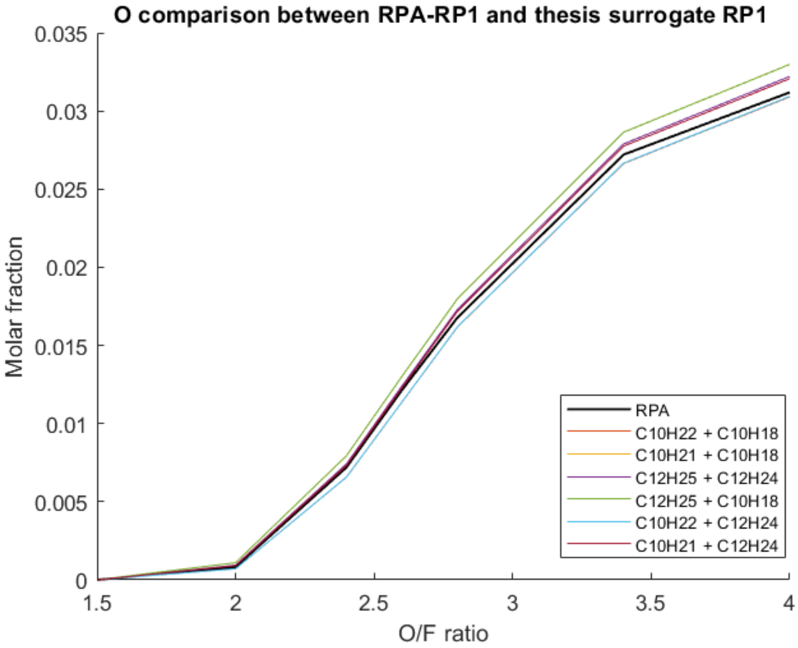


Figure A.7: O comparison

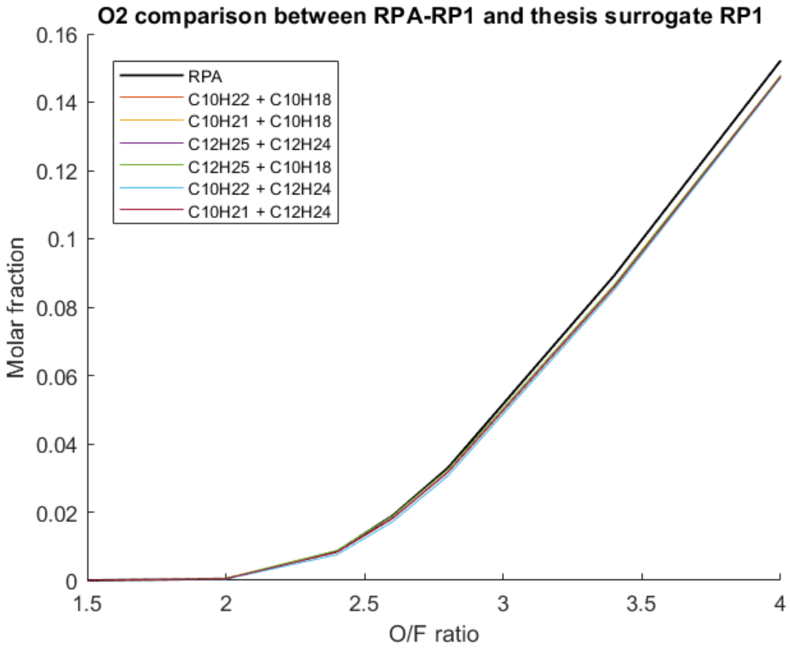


Figure A.8: O2 comparison

A.2 Codes

A.2.1 HGS

`HGS` code can be found and download for free in the following link: `https://github.com/ManelSoria/HGS`

A.2.2 INIST

`INIST` code can be found and download for free in the following link: `https://github.com/ManelSoria/INIST`

A.2.3 Other codes

The codes developed to the simulation of turbomachinery, RP1 example and Gibbs free energy minimisation are not uploaded with the thesis, contact with the author.

A.3 Rocket Cycles

This annexes shows the complete schemes of the turbomachinery that are analysed in Chapter 6.

A.3.1 Vinci LH2-LOX

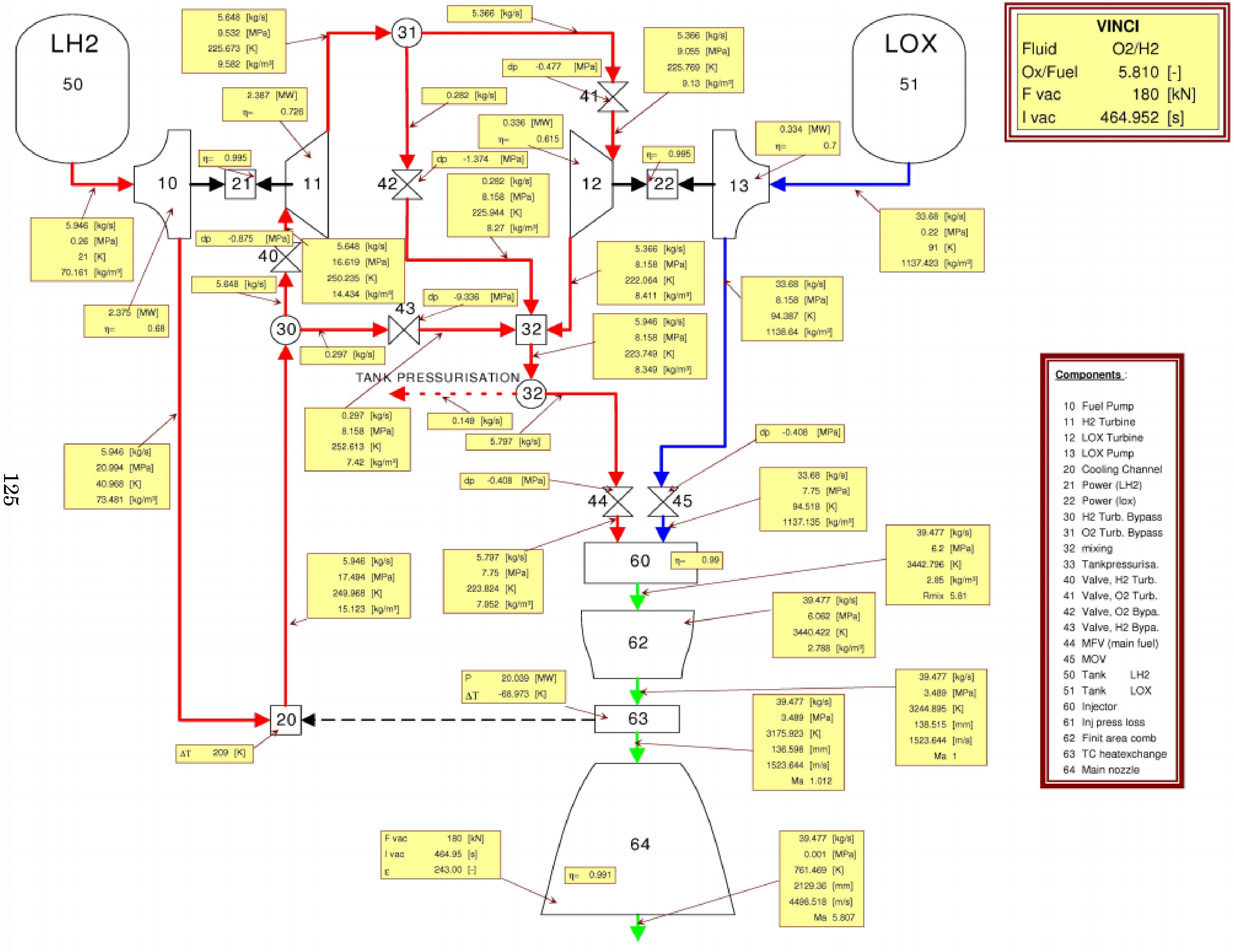


Figure A.9: Vinci Cycle [41]

A.3.2 SLME LH2-LOX

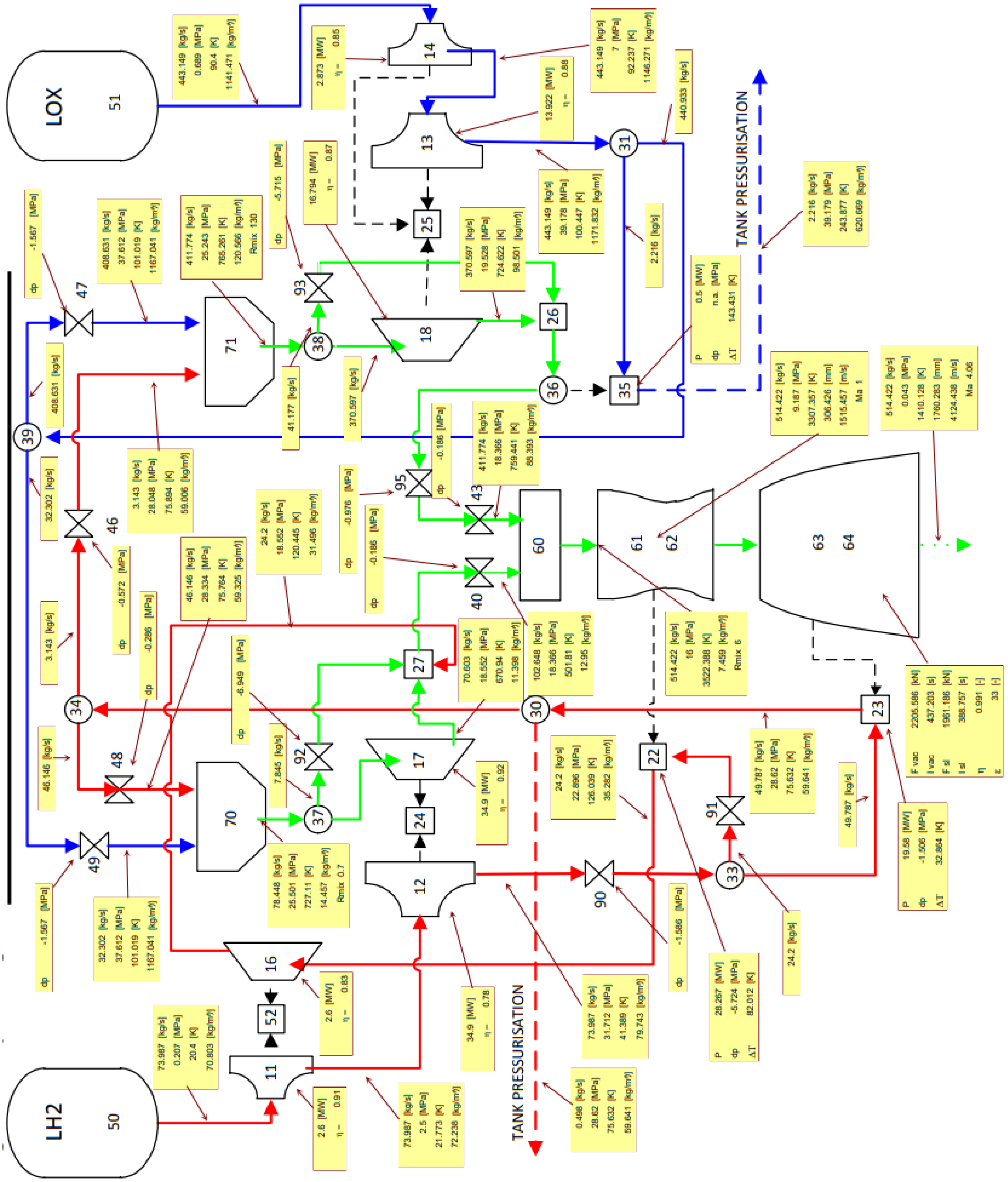


Figure A.10: SLME Cycle [44]

A.3.3 SSME Block II LH2-LOX

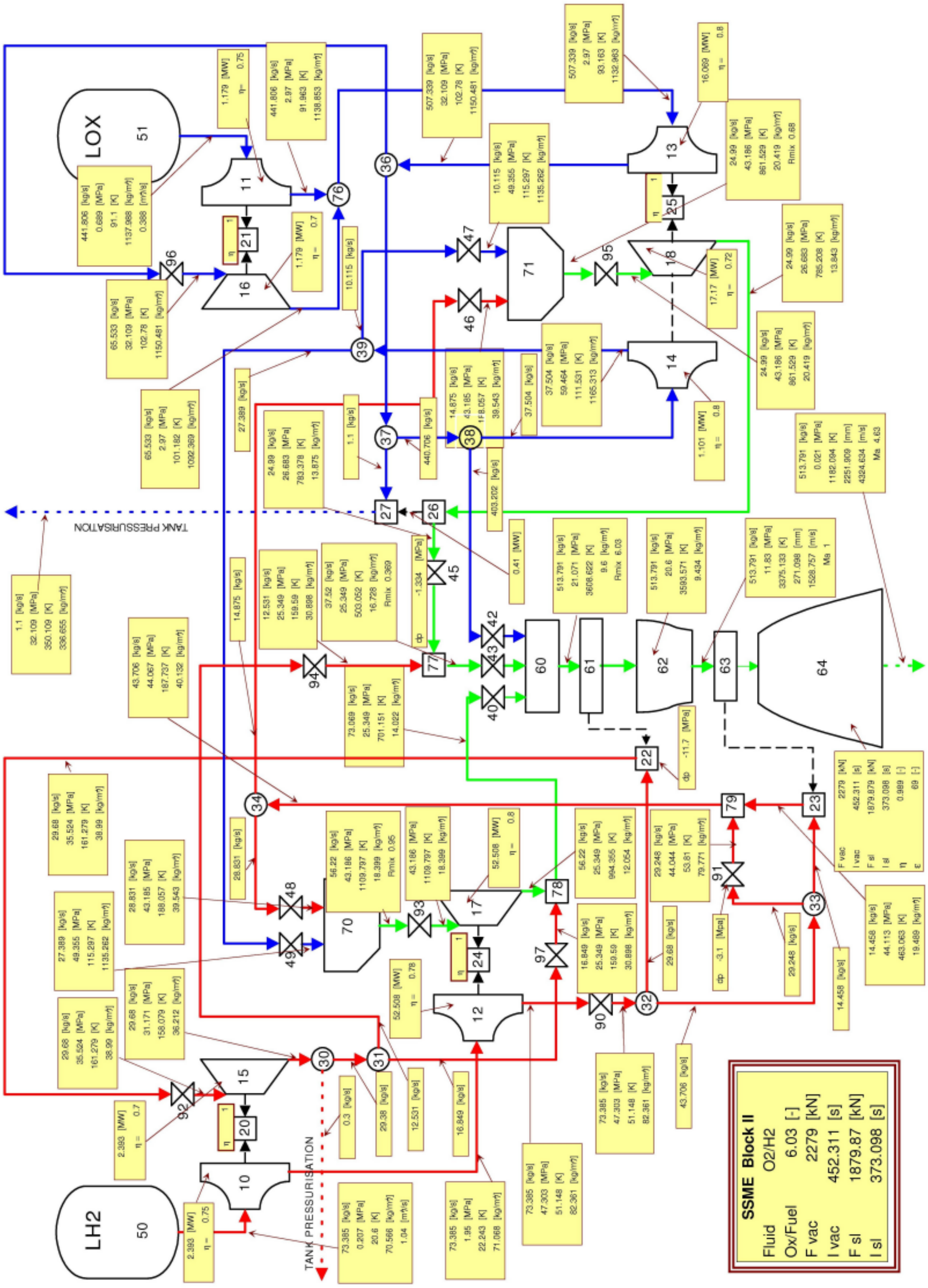


Figure A.11: SSME Block II Cycle[41]

A.3.4 SE-22 LH2-LOX

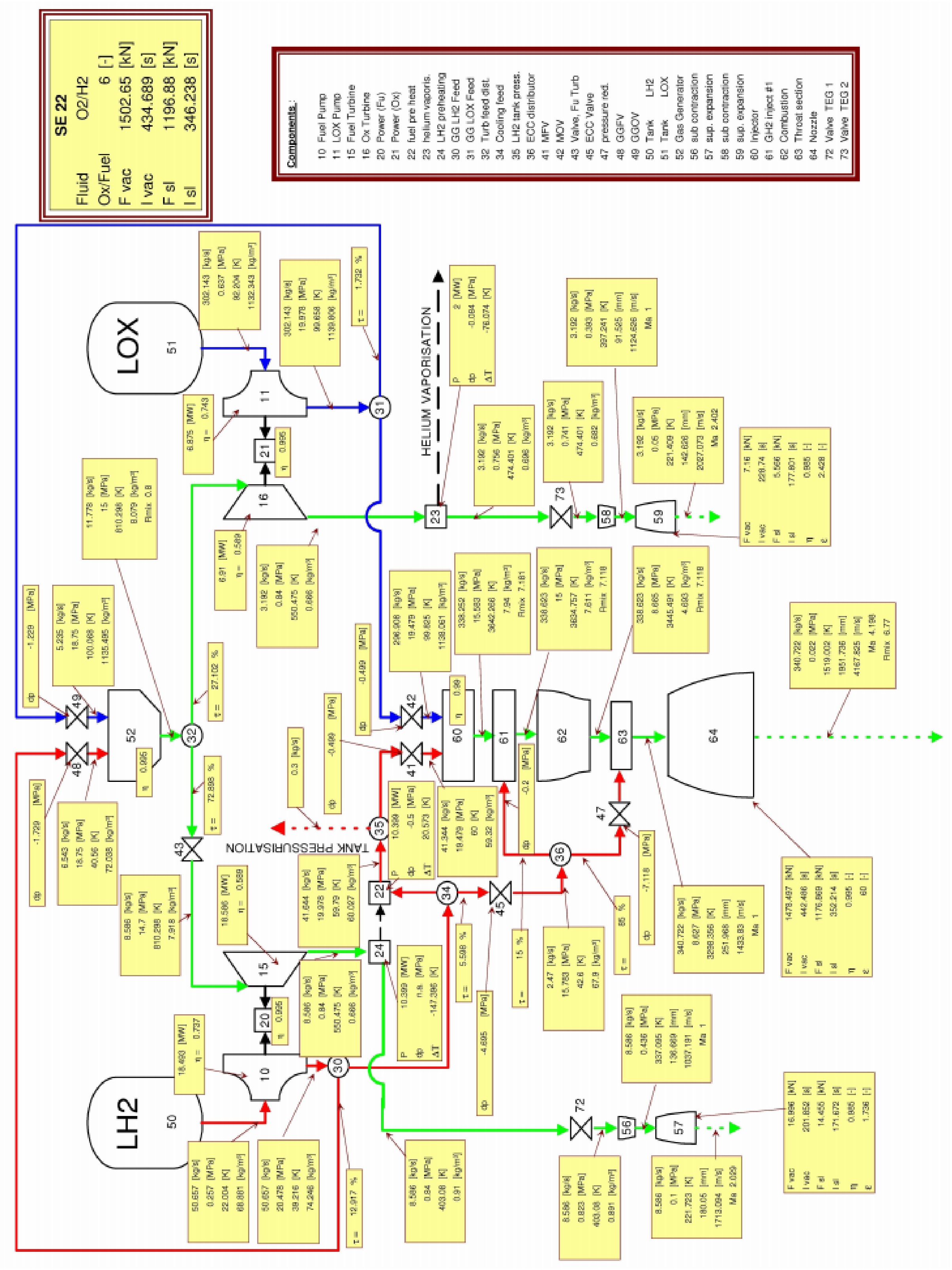


Figure A.12: SE-22 Cycle [41]

A.3.5 RD-180 RP1-LOX

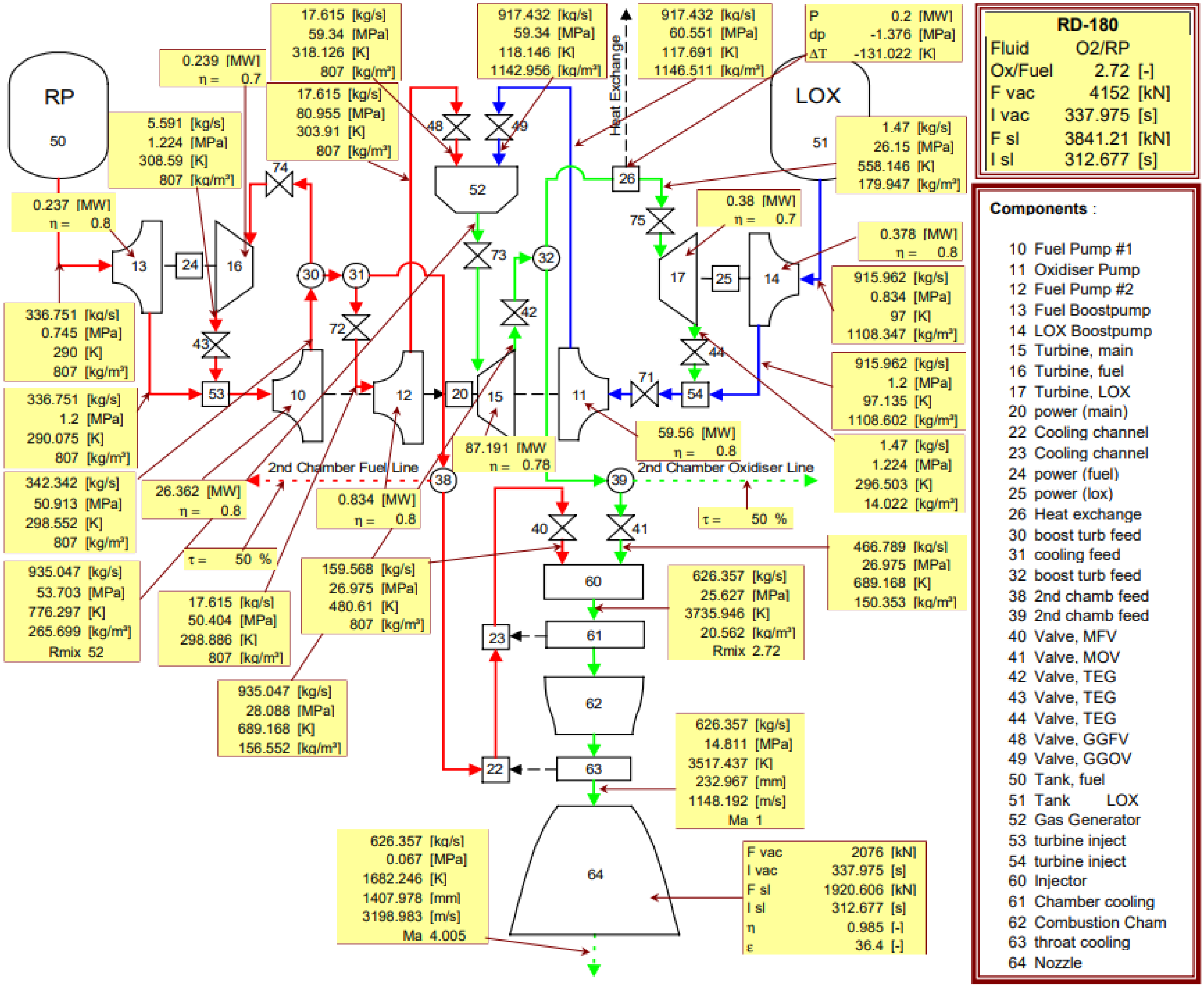


Figure A.13: RD-180 Cycle [1]

A.3.6 SE-12 CH4-LOX

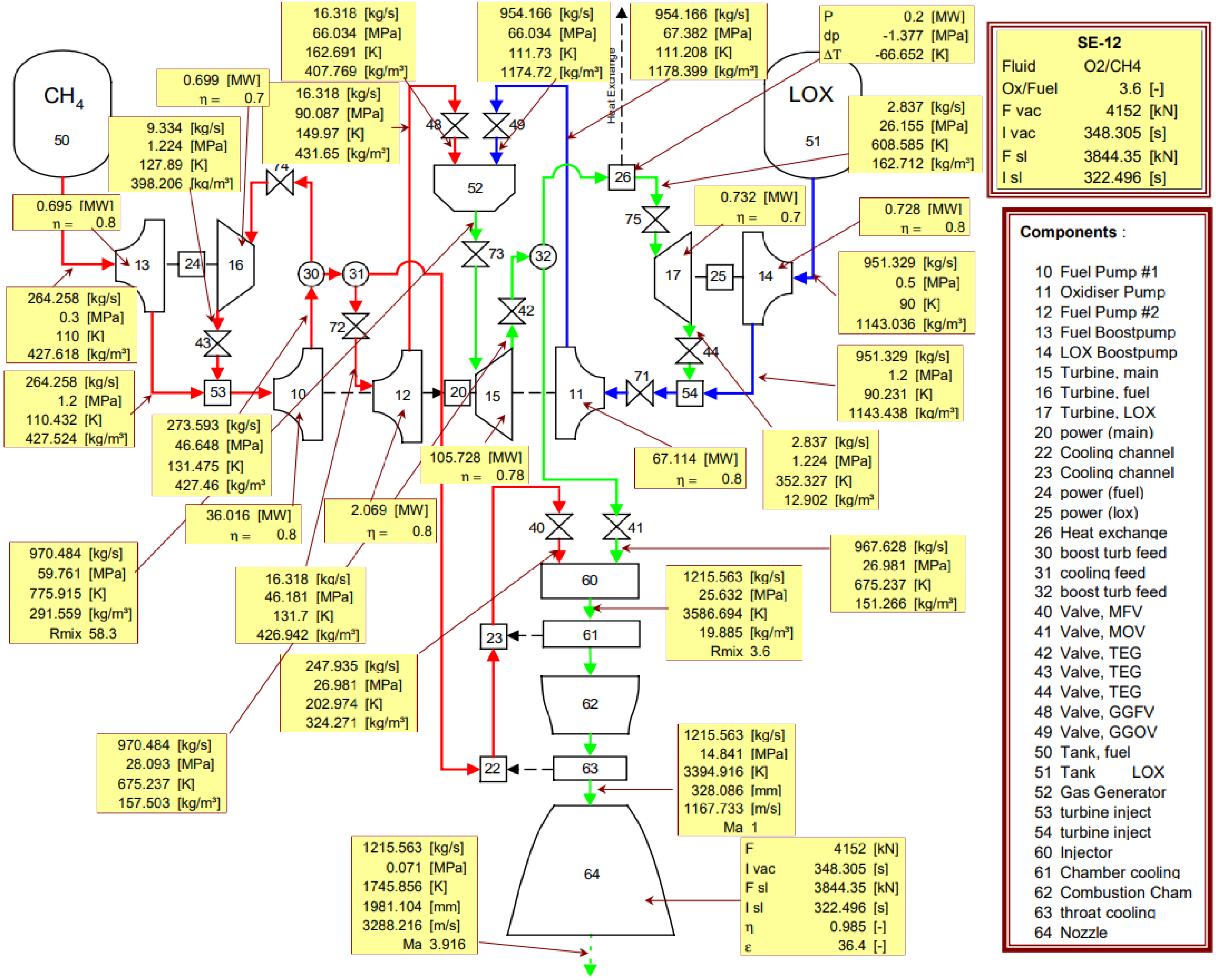


Figure A.14: SE-12 Cycle [1]

A.3.7 Vulcain 2

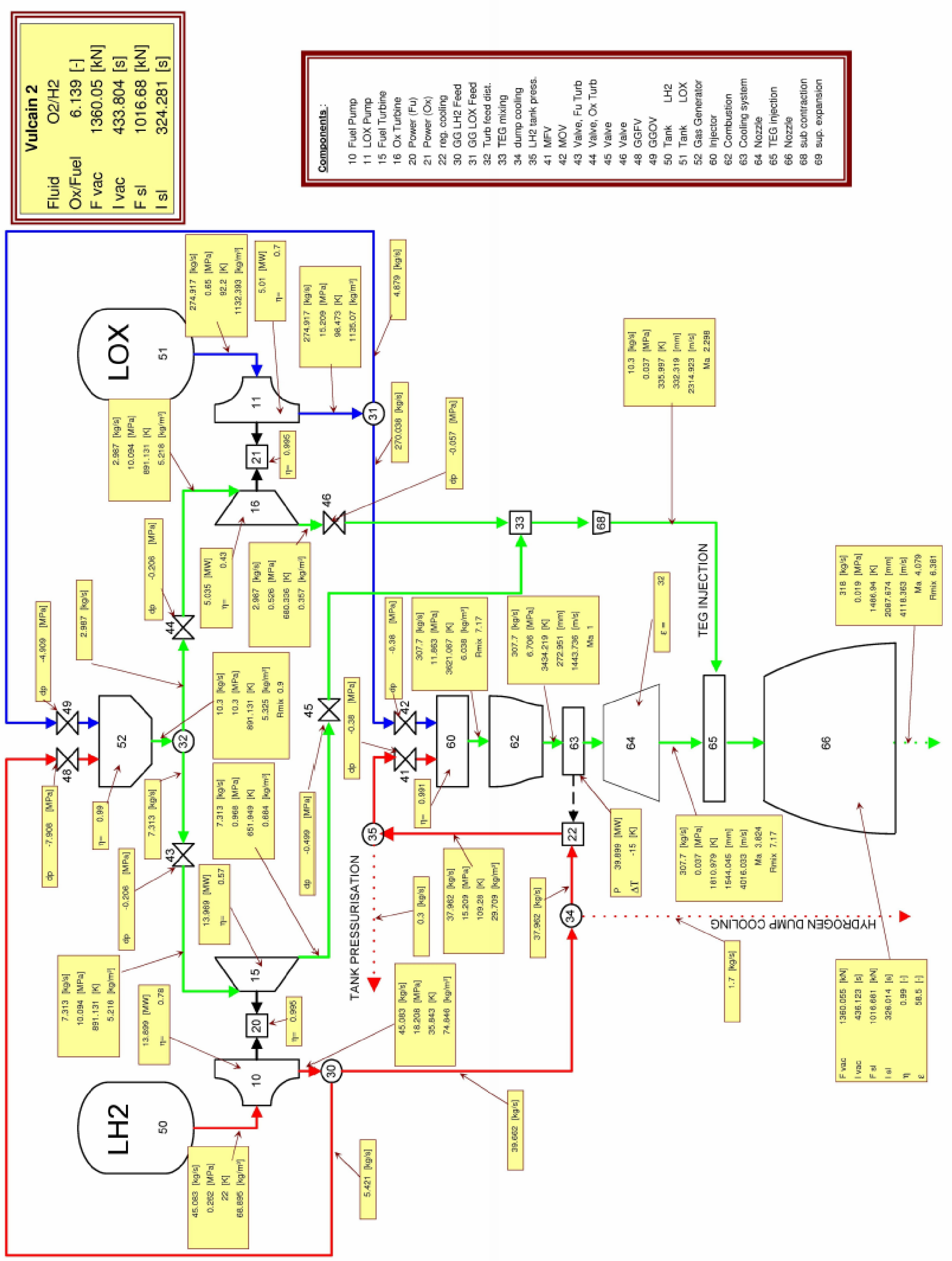


Figure A.15: Vulcain 2 Cycle [41]

BIBLIOGRAPHY

[1] Holger Burkhardt et al. “Comparative study of kerosene and methane propellant engines for reusable liquid booster stages”. In: *4th International Conference on Launcher Technology* December (2002). URL: http://www.dlr.de/Portaldata/55/Resources/dokumente/sart/0095-0212prop.pdf.

[2] Armin Herbertz. “Component Modeling for Rocket Engine Cycle Analysis”. In: *Transactions of the Japan Society for Aeronautical and Space Sciences, Aerospace Technology Japan* 14.ists30 (2016), Pa_119–Pa_127. ISSN: 1884-0485. DOI: 10.2322/tastj.14.pa{_}119.

[3] NIST. *NIST Chemistry WebBook - SRD 69*. 2017. DOI: https://doi.org/10.18434/T4D303. URL: https://webbook.nist.gov/chemistry/.

[4] Manel; Soria and Arnau; Miro. *HGS 1.3*. Terrassa, 2014.

[5] Manel Soria and Frezza Lorenzzo. *INIST*. Terrassa, 2014.

[6] Caleb Fuster et al. *HGS Chemical Equation Solver*. 2021. URL: https://ww2.mathworks.cn/matlabcentral/fileexchange/40862-hgs-chemical-equation-solver?s_tid=FX_rc2_behav.

[7] Alexander Ponomarenko. *RPA*. 2016. URL: https://www.rocket-propulsion.com/index.htm.

[8] NASA Glenn Research Center. *CEARUN*. URL: https://cearun.grc.nasa.gov/.

[9] Institute and Florence Museum of History of Science. *Robert Boyle*. URL: http://www.imss.fi.it/vuoto/eboyle.html.

[10] Sizes. *Fahrenheit temperature scale*. 2011. URL: https://www.sizes.com/units/temperature_Fahrenheit.htm.

[11] Joseph Louis Gay-Lussac. “Recherches sur la dilatation des gaz et des vapeurs”. In: *Annales de Chimie* 43 (1802), p. 350. URL: https://books.google.es/books?id=Z6ctSn3TIeYC&pg=PA137&redir_esc=y#v=onepage&q&f=false.

[12] Benoît Paul Émile Clapeyron. “Mémoire sur la puissance motrice de la chaleur”. In: *Journal de l’École Polytechnique* (1835). URL: https://gallica.bnf.fr/ark:/12148/bpt6k4336791/f173.item.

[13] R Nave. *Savery Engine*. URL: `http://hyperphysics.phy-astr.gsu.edu/hbase/Kinetic/psav.html`.

[14] The Editors of Encyclopaedia Britannica. *Industrial Revolution*. 2020. URL: `https://www.britannica.com/event/Industrial-Revolution`.

[15] Sadi Carnot. *Réflexions sur la puissance motrice du feu et sur les machines propres à développer cette puissance*. Paris: Bachélier, 1824. URL: `https://books.google.es/books?id=QX9iIWF3yOMC&pg=PP11&redir_esc=y#v=onepage&q&f=false`.

[16] James Prescott Joule. "On the mechanical equivalent of heat". In: *Philosophical Transactions of the Royal Society of London.* 140 (1850), pp. 61–82. DOI: `doi:10.1098/rstl.1850.0004`. URL: `https://archive.org/stream/philtrans00608634/00608634#mode/2up`.

[17] Clifford Truesdell. *The tragicomical history of thermodynamics 1822-1854*. Vol. 53. 9. New York [etc.] : Springer Verlag, 1980, pp. 1689–1699. ISBN: 9788578110796.

[18] R Clausius. "On a Modified Form of the Second Fundamental Theorem in the Mechanical Theory of Heat". In: *London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2 (1856), p. 86. DOI: `10.1002/andp.18541691202`. URL: `https://www.biodiversitylibrary.org/item/20044#page/100/mode/1up`.

[19] M Bailyn. *A Survey of Thermodynamics*. New York : American Institute of Physics, 1994, p. 342. ISBN: 0-88318-797-3.

[20] Yunus A. Çengel et al. *Termodinámica*. Berlin [etc.] : McGraw-Hill, 2006. ISBN: 9701056116. URL: `https://discovery.upc.edu/iii/encore/record/C__Rb1303347?lang=cat`.

[21] Josiah Willard. Gibbs. "A Method of Geometrical Representation of the Thermodynamic Properties of Substances by Means of Surfaces". In: *Transactions of the Connecticut Academy* 2 (1873), pp. 382–404.

[22] W. Thomson. "On the Dynamical Theory of Heat Part V. Thermo-electric Currents". In: *Transactions of the Royal Society of Edinburgh* 21 (1854). DOI: `10.1017/s0080456800032014`. URL: `https://www.biodiversitylibrary.org/item/126375#page/147/mode/1up`.

[23] John R Howell and Richard O. Buckius. *Principios de termodinámica para ingenieros*. Mexico: Mc-Graw Hill. ISBN: 9781626239777.

[24] Anil Waman Date. *Analytic combustion: With thermodynamics, chemical kinetics, and mass transfer*. Vol. 9781107002. New York : Cambridge University Press, 2011, pp. 1–345. ISBN: 9780511976759. DOI: `10.1017/CBO9780511976759`. URL: `https://discovery.upc.edu/iii/encore/record/C__Rb1460743?lang=cat`.

[25] Kelly Sands. *Chemical Equilibrium with Applications*. 2020. URL: `https://www1.grc.nasa.gov/research-and-engineering/ceaweb/`.

[26] Iorish V.S. and Yungman V.S. *ChemNet*. URL: `http://www.chem.msu.ru/cgi-bin/tkv.pl?show=welcome.html`.

[27] Gordon-McBride. *CEA*. 1996. URL: `https://software.nasa.gov/software/LEW-17687-1`.

[28] Terry Taylor. *CEAM*. URL: `https://federallabs.org/technology/chemical-equilibrium-with-applications-in-matlab-ceam`.

[29] Charlie Taylor. *RocketCEA*. 2020. URL: `https://pypi.org/project/rocketcea/`.

[30] Purdue Hydrogen Systems Lab. *MatlabCEA*. 2014. URL: `https://github.com/PurdueH2Lab/MatlabCEA`.

[31] Rene Leveille. *CEAMatlabAPI*. 2017. URL: `https://github.com/Progdrasil/CEAMatlabAPI`.

[32] ANSYS. *CHEMKIN*. 2021. URL: `https://www.ansys.com/products/fluids/ansys-chemkin-pro`.

[33] Jasmina Blecic, Joseph Harrington, and M. Oliver Bowman. "Tea: a Code Calculating Thermochemical Equilibrium Abundances". In: *The Astrophysical Journal Supplement Series* 225.1 (2016), p. 4. ISSN: 1538-4365. DOI: 10.3847/0067-0049/225/1/4. URL: `http://dx.doi.org/10.3847/0067-0049/225/1/4`.

[34] Markus Hans Alexander Piro. "Calculs par des Codes de Multi-Physique D'equilibres Thermodynamiques Pertinents aux Materiaux Nucleaires". PhD thesis. Division of Graduate Studies of the Royal Military College of Canada, 2011. ISBN: 9780494822272.

[35] a Burcat and R Branko. "Third millennium ideal gas and condensed phase thermochemical database for combustion with updates from active thermochemical tables". In: *Technical Report* ANL-05/20.September (2005), ANL–05/20 TAE 960. URL: `http://garfield.chem.elte.hu/Burcat/Introduction.pdf`.

[36] Kenneth K Kuo. *Principles of Combustion*. New York : John Wiley & Sons, 2005, p. 760. ISBN: 0471046892. URL: `https://discovery.upc.edu/iii/encore/record/C__Rb1277156?lang=cat`.

[37] M.White Frank. *Fluid mechanics*. 7th ed. Rhode Island: Mc-Graw Hill, 1979. ISBN: 0070622329.

[38] Lenntech. *Chemical elements listed by atomic mass*. URL: `https://www.lenntech.com/periodic/mass/atomic-mass.htm`.

[39] P Sam Johnson. *Gram-Schmidt Orthogonalization Process*. Karnataka, India., 2014. URL: `https://sam.nitk.ac.in/courses/MA904/Gram_Schmidt_Orthogonalization_Process.pdf`.

[40] Magnus Holmgren. *X Steam, Thermodynamic properties of water and steam.* 2008. URL: `https://www.mathworks.com/matlabcentral/fileexchange/9817-x-steam-thermodynamic-properties-of-water-and-steam`.

[41] A Herbertz. "Systemanalytische Untersuchung einer Brennkammer in faserkeramischer Bauweise von Raketenantrieben". PhD thesis. Rheinisch-Westf, 2008, p. 134.

[42] George P. Sutton and Oscar Biblarz. *Rocket Propulsion Element*. 9th ed. Hoboken: John Wiley & Sons, Inc., 2017. ISBN: 9781118753651.

[43] James S. Chickos and Hui Zhao. "Measurement of the vaporization enthalpy of complex mixtures by correlation-gas chromatography. The vaporization enthalpy of RP-1, JP-7, and JP-8 rocket and jet fuels at T = 298.15 K". In: *Energy and Fuels* 19.5 (2005), pp. 2064–2073. ISSN: 08870624. DOI: `10.1021/ef050116m`.

[44] Martin Sippel and Francesco Cremaschi. "Staged Combustion Cycle Rocket Engine Design Trade-Offs for Future Advanced Passenger Transport". In: May (2012), pp. 7–10. URL: `http://elib.dlr.de/78208/1/Prop2012-2.pdf`.

[45] Stephen R. Turns. *An introduction to combustion: concepts and applications*. Vol. 499. New York : McGraw-Hill International Edition, 2000, p. 411. ISBN: 0677026900. DOI: `10.1016/j.ijhydene.2008.07.121`. URL: `https://discovery.upc.edu/iii/encore/record/C__Rb1405476?lang=cat%20http://sutlib2.sut.ac.th/sut_contents/H89404.pdf`.

[46] J. Warnatz. *Combustion : physical and chemical fundamentals, modelling and simulation, experiments, pollutant formation / J. Warnatz, U. Maas, R. W. Dibble*. Berlin [etc.] : Springer, 2006. ISBN: 9783540259923. URL: `https://discovery.upc.edu/iii/encore/record/C__Rb1307542__SWarnatz__Orightresult__U__X4?lang=cat`.

[47] Jack D. Mattingly. *Elements of propulsion : gas turbines and rockets*. Reston : American Institute of Aeronautics and Astronautics, 2006, p. 867. ISBN: 1563477793. URL: `https://discovery.upc.edu/iii/encore/record/C__Rb1304582?lang=cat`.

[48] Michael J. Moran et al. *Fundamentals of Engineering Thermodynamics*. 8th. Vol. 29. 1. Willey, 2014, p. 1062. ISBN: 9781118412930. DOI: `10.7227/ijmee.29.1.2`. URL: `https://discovery.upc.edu/iii/encore/record/C__Rb1260524?lang=cat`.

[49] Martin Sippel et al. *High Thrust Bleed Rocket Engines for a Fly-Back Booster Application*. Cologne.

[50] Colin R. Ferguson and Allan. Kirkpatrick. *Internal combustion engines : applied thermosciences*. New York [etc.] : John Wiley & Sons, 2001, p. 369. ISBN: 0471356174. URL: `https://discovery.upc.edu/iii/encore/record/C__Rb1207367?lang=cat`.

[51] J.M. Smith, H. C. Van Ness, and M. M. Abbott. *Introducción a la Termodinámica en Ingeniería Química*. Mexico: McGraw-Hill. ISBN: 9701013336.

[52] Chris Snyder. *ThermoBuild*. 2020. URL: `https://cearun.grc.nasa.gov/ThermoBuild/index_ds.html`.

[53] Martin Sippel, Ryoma Yamashiro, and Francesco Cremaschi. *Staged Combustion Cycle Rocket Engine Design Trade-Offs for Future Advanced Passenger Transport*. Tech. rep. Bordeaux: DLR, 2012, p. 11.