

AE 5822: Master's Thesis

A Semi-Automated Framework for Launcher Remodelling that Handles Uncertain Data

Deutsche Zentrum für Luft- und Raumfahrt e. V.
Tiago Gracias Soares Rebelo

Delft University of Technology

AE 5822: Master's Thesis

A Semi-Automated Framework for Launcher
Remodelling that Handles Uncertain Data

by

Tiago Gracias Soares Rebelo

Student Number

6058787

Supervisors: Marc Naeije and Moritz Herberhold
Thesis Duration: February, 2025 - October, 2025
Faculty: Faculty of Aerospace Engineering, Delft

Cover: Space Launch System rocket by NASA retrieved from
<https://blog.w9-taxform.com/>

Preface

This document presents the master thesis "A Semi-Automated Framework for Launcher Remodelling that Handles Uncertain Data". With this thesis, I fulfil all requirements to graduate with my MSc in Aerospace Engineering diploma at TU Delft.

This master's thesis was carried out at the German Aerospace Center (DLR), within the Space Launcher Systems Analysis (SART) department, between February 2025 and October 2025. The work was conducted as part of the Space Sustainability and Sustainable Development (S3D) project, which aims to improve the understanding of the environmental impact of rocket launches by developing an accurate and comprehensive emissions inventory.

The motivation behind this research stems from the growing global awareness of the environmental footprint of space activities. Accurately remodelling rocket launches and their emissions is an essential step toward assessing and reducing their atmospheric impact. Within this context, the thesis focuses on developing a semi-automated framework for launcher remodelling capable of handling incomplete or uncertain input data. The outcome of this work contributes directly to DLR's ongoing efforts to make space transportation more sustainable.

Abstract

The increasing number of rocket launches has intensified concerns about the environmental effects of spaceflight. Quantifying rocket emissions at different altitudes and locations within the atmosphere is essential to assess these impacts. This requires realistic launcher models, but much of the publicly available technical data is incomplete or inconsistent. This research aimed to develop a semi-automated framework for launcher remodelling that can handle uncertain or missing input parameters while producing reliable and efficient results.

The framework was developed at the German Aerospace Center (DLR) and integrates a flexible input data structure, statistical estimation methods, and trajectory optimisation tools. Three statistical techniques — Monte Carlo, Latin Hypercube, and Approximate Bayesian Computation — were evaluated to estimate unknown parameters and define their valid ranges. Their performance was tested using several expendable launch vehicles with liquid propulsion.

Results showed that Latin Hypercube sampling achieved the best balance between accuracy and computational cost. When applied to real rockets, the framework produced configurations with payload estimates within 2% of reference values, even when up to five input parameters were uncertain.

Overall, the developed framework demonstrates that launcher remodelling can be automated while effectively handling uncertainty. It facilitates the generation of realistic launcher models and supports ongoing efforts to quantify the environmental impact of rocket emissions.

Contents

Nomenclature	xii
1 Introduction	1
1.1 Outline	3
1.2 Company Background	4
2 Literature Review and Project Plan	6
2.1 Remodelling Process	6
2.2 Statistical Methods to Handle Uncertainties	8
2.3 Research Gap	11
2.4 Research Goals	11
2.4.1 Research Objective	12
2.4.2 Research Question	12
2.5 Requirements	13
3 Methods	15
3.1 Theory	15
3.1.1 Rocketry Equations	15
3.1.2 Statistical Methods to Handle Uncertain Parameters	17
3.1.3 Calculation Models	21
3.1.4 Tools	22
3.2 Practical Implementation	27
3.2.1 Data Collection	28
3.2.2 CLAVA Conversion	31
3.2.3 Test Model	32
3.2.4 Complex Model	43
3.2.5 Statistical Methods to Handle Uncertain Parameters	46
3.2.6 Test Cases	49
3.3 Expected Results	51
3.4 Verification and Validation	52
4 Results and Discussion	54
4.1 Sensitivity Analysis	54
4.1.1 Test Model Results	55
4.1.2 Complex Model Results	58
4.2 Statistical Methods Performance	59
4.2.1 Monte Carlo and Variants	59
4.2.2 Latin Hypercube Performance	61
4.2.3 Approximate Bayesian Computation Performance	65
4.3 Comparison Between Latin Hypercube and Approximate Bayesian Computation	66

4.4	Complex Model Results	76
4.4.1	New Test Case - Long March 7	86
4.5	Concluding Discussion	87
4.6	Future Work	91
5	Conclusion	94
	References	96
A	Planning	104
B	Input Data Sheet	108
B.1	Required Parameters List	108
B.2	Excel Tabs Pictures	110
C	Test Model	112
C.1	Losses Results	112
C.2	Gravitational Loss Estimation	113
C.3	Drag Loss Estimation	114
C.4	Thrust Loss Estimation	115
D	Results and Discussion	117
D.1	Sensitivity Analysis	117
D.2	Statistical Methods Performance	118
D.3	Complex Model Results	119

List of Figures

1.1	Total propellant mass used by rockets per year for the five most used propellants.	1
1.2	Example of a test result for S3D: H2O emissions for different trajectories using a default launch vehicle [11].	3
1.3	Flowchart of the major steps of the remodelling framework. The Test Model has a different colour because it is not part of the final loop, only of the testing phase, although the flowchart is the same for both methods.	4
1.4	DLR facilities in Bremen (retrieved from [16]).	5
3.1	Monte Carlo-based sampling methods visualisation (retrieved from [52])	18
3.2	TOSCA output visualisation example - Long March 2C Low Earth Orbit (LEO) mission.	25
3.3	Excel stage template tab: contains the main technical data about a rocket stage needed for the remodelling process.	30
3.4	Structure of the conversion from Excel to CLAVA.	32
3.5	Excel to CLAVA conversion detailed in the two branches combined to generate the file: the rocket object and the trajectories.	33
3.6	Structure of the procedure to read values from the IDS and assign them to the CLAVA and UQ files.	34
3.7	Flowchart that demonstrates how the Test Model's algorithm is implemented.	35
3.8	Calculation algorithm to obtain the Δv that the rocket can produce.	36
3.9	Mass calculation procedure.	37
3.10	Drag Loss vs Ballistic Coefficient correlation.	39
3.11	Brent's method structure applied in this thesis. It calculates the maximum payload based on all velocity budget components.	42
3.12	Workflow of the CMA-ES optimiser wrapped around TOSCA as part of the Complex Model.	44
3.13	Flowchart of the integration process of the Complex Model in the overall framework.	45
3.14	Integration of the sampling process for Monte Carlo-based methods in the overall framework.	47
3.15	Integration of the sampling process for the Approximate Bayesian Computation method in the overall framework.	48
3.16	Images of the three rockets used as test cases throughout this thesis.	50
4.1	Sensitivity analysis first order results including confidence intervals for test case LM2C 5U using the Test Model.	56
4.2	Sensitivity analysis first order results including confidence intervals for test case LM4C 10U using the Test Model.	57

4.3	Sensitivity analysis first order results including confidence intervals for test case LM3BE 10U using the Test Model.	57
4.4	Sensitivity analysis first order results including confidence intervals for test case LM2C 5U using the Complex Model.	58
4.5	Comparison between the results achieved by using different Monte Carlo-based sampling methods with the Test Model - 100 total samples.	60
4.6	Comparison between the results achieved by using different Monte Carlo-based sampling methods with the Test Model - 500 total samples.	61
4.7	Convergence behaviour of Latin Hypercube sampling for the vacuum maximum I_{sp} of stage 2 in test case LM2C 2U. The curves show the evolution of three summary statistics (minimum, median, and maximum) of the valid results as a function of the number of samples (logarithmic scale).	62
4.8	Convergence behaviour of Latin Hypercube sampling for the structure mass of stage 2 in test case LM2C 2U. The curves show the evolution of three summary statistics (minimum, median, and maximum) of the valid results as a function of the number of samples (logarithmic scale).	63
4.9	Convergence behaviour of Latin Hypercube sampling for the vacuum maximum I_{sp} of stage 2 in test case LM2C 2U. The curves show the evolution of three summary statistics (minimum, median, and maximum) of the valid results as a function of the number of samples (logarithmic scale). Each simulation was repeated ten times to account for variability. The coloured regions represent the convergence bands, defined by the minimum and maximum values across the ten runs for each statistic.	64
4.10	Convergence behaviour of Latin Hypercube sampling for the structure mass of stage 2 in test case LM2C 2U. The curves show the evolution of three summary statistics (minimum, median, and maximum) of the valid results as a function of the number of samples (logarithmic scale). Each simulation was repeated ten times to account for variability. The coloured regions represent the convergence bands, defined by the minimum and maximum values across the ten runs for each statistic.	65
4.11	Convergence behaviour of Approximate Bayesian Computation (ABC) for the maximum vacuum I_{sp} of stage 2 in test case LM2C 2U. The curves show the evolution of three summary statistics (minimum, median, and maximum) of the valid results as a function of the number of samples (logarithmic scale). Each simulation was repeated ten times to account for variability. The coloured regions represent the convergence bands, defined by the minimum and maximum values across the ten runs for each statistic.	66

4.12	Convergence comparison of Latin Hypercube sampling and Approximate Bayesian Computation for the maximum vacuum I_{sp} of stage 2 in test case LM2C 2U. The curves show the evolution of three summary statistics (minimum, median, and maximum) of the valid results as a function of the number of samples (logarithmic scale). Each simulation was repeated ten times to account for variability. The coloured regions represent the convergence bands, defined by the minimum and maximum values across the ten runs for each statistic.	67
4.13	Convergence comparison of Latin Hypercube sampling and Approximate Bayesian Computation for the structure mass of stage 2 in test case LM2C 2U. The curves show the evolution of three summary statistics (minimum, median, and maximum) of the valid results as a function of the number of samples (logarithmic scale). Each simulation was repeated ten times to account for variability. The coloured regions represent the convergence bands, defined by the minimum and maximum values across the ten runs for each statistic.	68
4.14	Convergence comparison of Latin Hypercube sampling and Approximate Bayesian Computation for the maximum vacuum I_{sp} of stage 2 in test case LM2C 1U. The curves show the evolution of three summary statistics (minimum, median, and maximum) of the valid results as a function of the number of samples (logarithmic scale). Each simulation was repeated ten times to account for variability. The coloured regions represent the convergence bands, defined by the minimum and maximum values across the ten runs for each statistic.	69
4.15	Convergence comparison of Latin Hypercube sampling and Approximate Bayesian Computation for the maximum vacuum I_{sp} of stage 2 in test case LM2C 10U. The curves show the evolution of three summary statistics (minimum, median, and maximum) of the valid results as a function of the number of samples (logarithmic scale). Each simulation was repeated ten times to account for variability. The coloured regions represent the convergence bands, defined by the minimum and maximum values across the ten runs for each statistic.	70
4.16	Comparison of the final distributions for one run of Latin Hypercube sampling and Approximate Bayesian Computation for the maximum vacuum I_{sp} of stage 2 for test case LM2C 10U.	71
4.17	Convergence comparison of Latin Hypercube sampling and Approximate Bayesian Computation for the structure mass of stage 2 in test case LM2C 10U. The curves show the evolution of three summary statistics (minimum, median, and maximum) of the valid results as a function of the number of samples (logarithmic scale). Each simulation was repeated ten times to account for variability. The coloured regions represent the convergence bands, defined by the minimum and maximum values across the ten runs for each statistic.	72
4.18	Comparison of the final distributions for one run of Latin Hypercube sampling and Approximate Bayesian Computation for the structure mass of stage 2 for test case LM2C 10U.	72

- 4.19 Convergence comparison of Latin Hypercube sampling and Approximate Bayesian Computation for the ascent propellant mass of stage 3 in test case LM4C 3U. The curves show the evolution of three summary statistics (minimum, median, and maximum) of the valid results as a function of the number of samples (logarithmic scale). Each simulation was repeated ten times to account for variability. The coloured regions represent the convergence bands, defined by the minimum and maximum values across the ten runs for each statistic. 73
- 4.20 Comparison of the final distributions for one run of Latin Hypercube sampling and Approximate Bayesian Computation for the ascent propellant mass of stage 3 for test case LM4C 3U. 73
- 4.21 Convergence comparison of Latin Hypercube sampling and Approximate Bayesian Computation for the ascent propellant mass of stage 3 in test case LM4C 10U. The curves show the evolution of three summary statistics (minimum, median, and maximum) of the valid results as a function of the number of samples (logarithmic scale). Each simulation was repeated ten times to account for variability. The coloured regions represent the convergence bands, defined by the minimum and maximum values across the ten runs for each statistic. 74
- 4.22 Convergence comparison of Latin Hypercube sampling and Approximate Bayesian Computation for the vacuum I_{sp} of the boosters stage in test case LM3BE 3U. The curves show the evolution of three summary statistics (minimum, median, and maximum) of the valid results as a function of the number of samples (logarithmic scale). Each simulation was repeated ten times to account for variability. The coloured regions represent the convergence bands, defined by the minimum and maximum values across the ten runs for each statistic. 75
- 4.23 Convergence comparison of Latin Hypercube sampling and Approximate Bayesian Computation for the vacuum I_{sp} of the boosters stage in test case LM3BE 10U. The curves show the evolution of three summary statistics (minimum, median, and maximum) of the valid results as a function of the number of samples (logarithmic scale). Each simulation was repeated ten times to account for variability. The coloured regions represent the convergence bands, defined by the minimum and maximum values across the ten runs for each statistic. 76
- 4.24 Combination of unknown parameters for all the valid samples of the Complex Model with Latin Hypercube sampling for test case LM2C 2U with a payload margin of 7%. The highlighted samples correspond to the top 5% samples with the lowest summed mean distance to the others. 77
- 4.25 Combination of unknown parameters for all the valid samples of the Complex Model with Approximate Bayesian Computation for test case LM2C 1U with a payload margin of 7%. The highlighted samples correspond to the top 5% samples with the lowest summed mean distance to the others. 78

4.26	Combination of unknown parameters for all the valid samples of the Complex Model with Latin Hypercube sampling for test case LM4C 2U with a payload margin of 20%. The highlighted samples correspond to the top 5% samples with the lowest summed mean distance to the others.	79
4.27	Combination of unknown parameters for all the valid samples of the Complex Model with Latin Hypercube sampling for test case LM4C 2U with a payload margin of 2%. The highlighted samples correspond to the top 5% samples with the lowest summed mean distance to the others.	80
4.28	Combination of unknown parameters for all the valid samples of the Complex Model with Latin Hypercube sampling for test case LM4C 2U with a payload margin of 2% and a lower minimum payload value. The highlighted samples correspond to the top 5% samples with the lowest summed mean distance to the others.	81
4.29	Combination of unknown parameters for all the valid samples of the Complex Model with Latin Hypercube sampling for test case LM3BE 5U with a payload margin of 7%. The highlighted samples correspond to the top 5% samples with the lowest summed mean distance to the others.	82
4.30	Combination of unknown parameters for all the valid samples of the Complex Model with Latin Hypercube sampling for test case LM3BE 5U with a payload margin of 2%. The highlighted samples correspond to the top 5% samples with the lowest summed mean distance to the others.	83
4.31	Combination of the selected unknown parameters for all the valid samples of the Complex Model with Latin Hypercube sampling for test case LM2C 10U with a payload margin of 7%. The highlighted samples correspond to the top 25% samples with the lowest summed mean distance to the others.	85
4.32	Combination of unknown parameters for all the valid samples of the Complex Model with Latin Hypercube sampling for the Long March 7 with a payload margin of 1%. The highlighted samples correspond to the top 5% samples with the lowest summed mean distance to the others.	87
4.33	Preliminary results of rocket emissions in 2024 by latitude and longitude using remodelled launch trajectories (retrieved from [93]).	93
A.1	Gantt chart with all work packages and their duration and organisation in the overall planning structure.	104
A.2	Detailed algorithm flowchart containing the main components of the developed framework and their interconnections.	107
B.1	Component sketches of the rocket: fairing and booster	109
B.2	IDS general tab	110
B.3	IDS identifier tab	110
B.4	IDS thrust profile template tab	111
B.5	IDS validation cases tab	111
C.1	Gravitational Losses correlation with T/W	113

C.2	Gravitational Losses correlation with Burn Time	114
C.3	Thrust Losses correlation with T/W (2 nd Stage)	115
C.4	Gravitational Losses correlation with Burn Time (upper stages)	116
D.1	Sensitivity analysis second order results including confidence intervals for test case LM2C 5U using the Test Model.	117
D.2	Sensitivity analysis second order results including confidence intervals for test case LM2C 5U using the Complex Model.	118
D.3	Combination of unknown parameters for all the valid samples of the CM with LH sampling for test case LM3BE 5U with a payload margin of 7% intentionally deviated from the right values for comparison. The highlighted samples correspond to the top 5% samples with the lowest summed mean distance to the others.	119
D.4	Combination of the selected unknown parameters for all the valid samples of the CM with LH sampling for test case LM2C 10U with a payload margin of 2.5%. The highlighted samples correspond to the top 25% samples with the lowest summed mean distance to the others.	120

List of Tables

3.1	Validation results comparing TOSCA and Test Model outputs.	43
3.2	Summary of test cases and corresponding unknown parameters.	51
3.3	Overview of data types, collection methods, and processing details. . .	52
4.1	Comparison between the obtained results and the ones which are assumed to be real and thus were used in the manual remodelling process for test case LM2C 2U.	78
4.2	Comparison between the obtained results and the ones which are assumed to be real and thus were used in the manual remodelling process for test case LM3BE 5U.	84
4.3	Requirements with the corresponding sections where their VV is mentioned.	88
C.1	Gravitational, Drag, and Thrust Losses for Multiple Rockets and Orbits .	112
D.1	Summary statistics of the structure mass of stage 2 for test case LM2C 2U.	118

Nomenclature

Abbreviations

Abbreviation	Definition
ABC	Approximate Bayesian Computation
CAC	Calculation of Aerodynamic Coefficients
CLAVA	Concurrent Launch Vehicle Analysis
DLR	Deutsches Zentrum für Luft- und Raumfahrt e. V. (German Aerospace Centre)
GTO	Geostationary Transfer Orbit
IDS	Input Data Sheet
LCA	Life Cycle Assessment
LCSA	Life Cycle Sustainability Assessment
LEO	Low Earth Orbit
LH	Latin Hypercube
LM	Long March
LSS	Large Space Structure
LV	Launch Vehicle
MEO	Medium Earth Orbit
MC	Monte Carlo
RLV	Reusable Launch Vehicle
S3D	Space Sustainability and Sustainable Development
SART	Systemanalyse Raumtransport (Space Launcher Systems Analysis)
SI	Structural Index
SMC	Sequential Monte Carlo
SSO	Sun-Synchronous Orbit
STSM	Space Transportation System Mass
TOSCA	Trajectory Optimisation and Simulation of Conventional and Advanced Spacecraft
UQ	Uncertainty Quantification

Symbols

Symbol	Definition	Unit
a	Semi-major axis	[m]
A	Area	[m ²]
C_b	Ballistic coefficient	[-]

Symbol	Definition	Unit
C_d	Drag coefficient	[-]
C_M	Moment coefficient	[-]
Δv	Velocity change	[m s ⁻¹]
F	Thrust	[N]
γ	Flight path angle	[°]
g_0	Gravity	[m s ⁻²]
i	Inclination	[°]
I_{sp}	Specific impulse	[s]
$\dot{m}$	Mass flow rate	[kg s ⁻¹]
M	Mass	[kg]
μ	Standard gravitational parameter	[m ³ s ⁻²]
p	Pressure	[Pa]
s	Distance	[m]
t	Time	[s]
T/W	Thrust-to-Weight ratio	[-]
V	Velocity	[m s ⁻¹]
w	Exhaust velocity	[m s ⁻¹]

Introduction

The number of global rocket launches has been growing in the past few years [1]. Consequently, the propellant mass used during these launches has increased on a large scale, as seen in Figure 1.1. This rise has intensified interest in the environmental footprint of spaceflight. Rocket launches are unique, as their exhaust of gases and particles is directly injected into the middle and upper atmosphere. No other transport has such a characteristic. As a result, they have a huge potential to be a major contributor to the climate impact of space transport activities. However, there are still many uncertainties regarding the exhaust's chemical composition, plume behaviour and the formation of particles such as soot [2].

Current and past research highlights several concerning atmospheric effects of rocket launches. Their emissions have been shown to cause stratospheric warming, affect the atmosphere's chemistry, disrupt the ozone layer's regeneration, and increase radiative forcing [3–8].

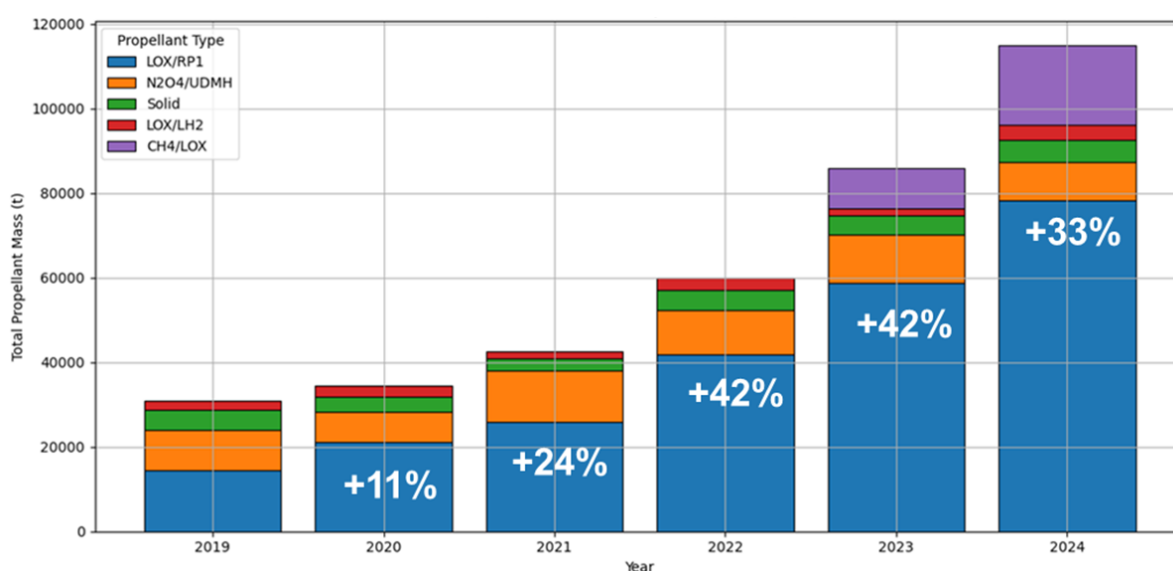


Figure 1.1: Total propellant mass used by rockets per year for the five most used propellants.

Despite these findings, the literature consistently stresses that current knowledge remains limited. In particular, there is a strong need for a more detailed qualification and quantification of rocket emissions during their launches, in terms of species composition and their distribution across the atmosphere.

A central element in assessing the impact of rockets is their emission profile. This profile includes the emissions' distribution, mass, and chemical composition. It depends on several launch vehicle (LV) characteristics, including the engine cycle, propellant choice or ascent trajectory [3, 5, 9]. Since different fuels and design choices produce distinct combustion products, an accurate LV remodelling is required to quantify species such as carbon dioxide, water vapour, and soot at varying altitudes.

To address the significant knowledge gap regarding the impact of LV emissions in the atmosphere, the German Aerospace Center (DLR) is currently working on the Space Sustainability and Sustainable Development (S3D) project. This initiative aims to advance the assessment of sustainability in the space sector by developing a tailored Life Cycle Sustainability Assessment (LCSA) framework. Unlike the traditional Life Cycle Assessment (LCA), which focuses more on the environmental effect, LCSA adds the social and economic dimensions, broadening its perspective. Within S3D, a global launch emission inventory is being built. It consists of detailed LV models with the corresponding flown ascent and recovery trajectories for all launches conducted in 2024. This inventory enables the calculation of the exhaust species and distributions through the atmosphere. Thus, once this database is complete, the obtained emission data can be used for advanced atmospheric modelling, enhancing the understanding of spaceflight's impact.

Early S3D results already indicate substantial differences in emissions between hydrogen and methane-fuelled systems. They also highlight how reusable LV designs may increase total exhaust mass due to recovery manoeuvres. However, these outcomes come from a conceptual LV with different configurations and propellants, engine types and reuse methods [10]. To obtain the preliminary results, many trajectories with various architectures of this vehicle were simulated (see Figure 1.2). Therefore, even though they are a good reference point for the real results, they do not exactly represent them.

The global emissions exhaust inventory's flagship is the precise and realistic modelling of the individual launch systems. To address the inaccuracies from the predecessor exhaust inventories, the real LVs and their launch trajectories are needed. This way, one can extract their exhaust at the different altitudes and locations of the atmosphere where they occur. The actual rockets need to be remodelled to obtain their launch trajectories and combustion processes.

However, there are two main problems with this remodelling process: the time it takes and the uncertainties. To get a realistic model for an LV, one needs to build a mass model and an aerodynamic model first, based on the LV's technical data. Afterwards, the trajectory has to be manually optimised. This requires much time since it is a very complex task. On the other hand, extensive research is needed to acquire all the data necessary from publicly available information. The issue lies in the contradicting data frequently found, as often multiple sources suggest a different value for a certain

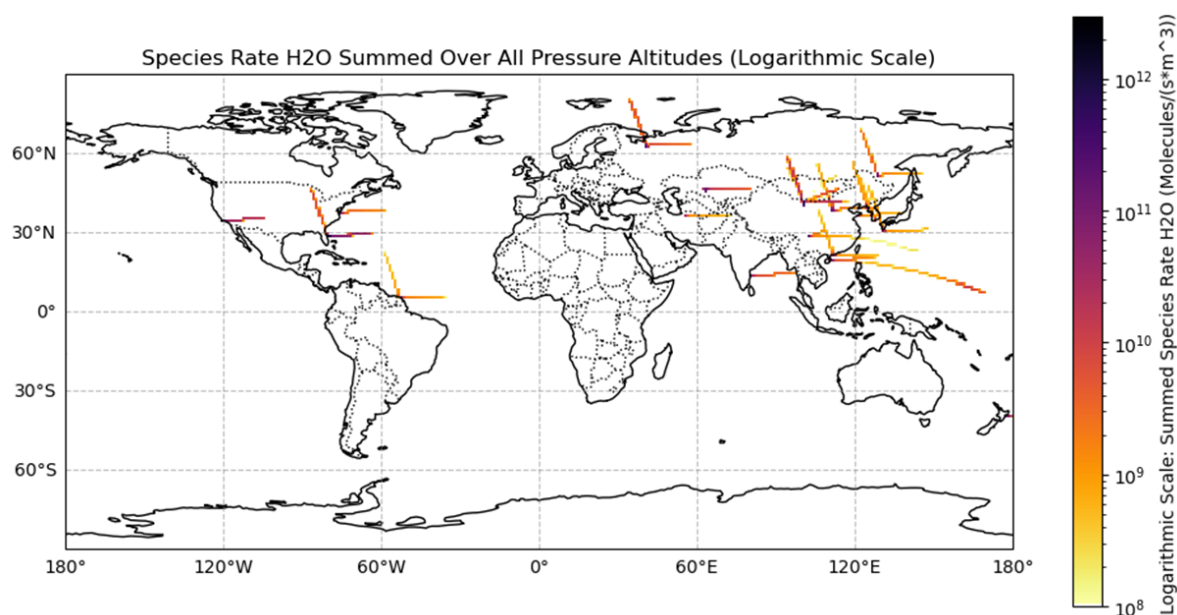


Figure 1.2: Example of a test result for S3D: H2O emissions for different trajectories using a default launch vehicle [11].

parameter. Furthermore, for other parameters there are no sources at all. Both situations induce a high degree of uncertainty, making it hard to define a value to be used in the remodelling process.

To solve these problems, this thesis will introduce a semi-automated framework capable of handling incomplete or uncertain input data while guaranteeing efficiency and reliability in the remodelling process. This approach will provide probabilistic estimates of unknown parameters to reduce the dependence on manual work and the uncertainty barriers by integrating computational statistical methods. This will improve the scalability and flexibility of this process, allowing for a larger number of rockets to be remodelled and included in DLR's database.

1.1. Outline

Figure 1.3 shows the flowchart for the remodelling framework. It begins with an input data sheet (IDS), which stores the data from the research required for each LV. It will be flexible, adaptable, and organised in a user-friendly manner. The user can enter a specific value or a range between the estimated minimum and maximum values for each input parameter necessary for the remodelling process. Additionally, a payload range must be provided for each rocket's mission being analysed, usually based on information supplied by the LV's manufacturer. The mission(s) and the payload range(s) will be used to validate the launcher. Subsequently, statistical sampling methods will be applied to these data. Parameters without a specific value will undergo a sampling process to generate multiple combinations of possible LV configurations. Afterwards, their mass and aerodynamic models will be created, and their optimised trajectories and related payload capacities will be automatically evaluated using the Complex Model (CM). The CM uses a genetic optimiser wrapped

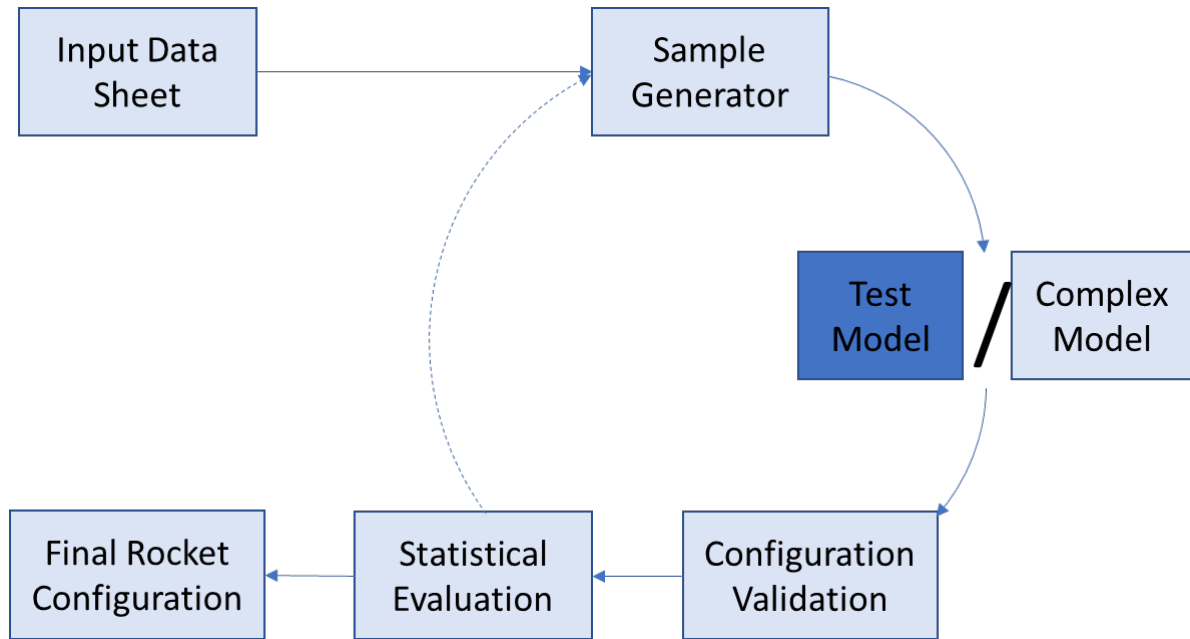


Figure 1.3: Flowchart of the major steps of the remodelling framework. The Test Model has a different colour because it is not part of the final loop, only of the testing phase, although the flowchart is the same for both methods.

around a trajectory optimiser to find the maximum payload for that LV configuration. Validation involves checking whether the payload result for each LV configuration falls within the pre-defined range. Finally, the user will select one or multiple configurations that meet the requirements.

Furthermore, it is important to explain that the final statistical method chosen will be the result of a comparison done as part of this work. Its goal is to determine the most adequate statistical sampling method for this application. Since it requires an intensive testing phase with hundreds of thousands of samples, a faster model was developed. This was necessary because the CM is very time-consuming, given its optimisation algorithm. Thus, this comparison between methods uses a simpler model, denominated as the Test Model (TM). The TM uses equations based on the LV input parameters to quickly estimate the payload to a certain orbit. Due to its simplified nature, it is not as accurate and overarching as the CM. During this testing phase, three LVs were used as test cases (see Section 3.2.6). They had different numbers of stages, which is directly related to the number of unknowns and, hence, complexity. Multiple scenarios were created for each of them, which allowed several results and conclusions to be drawn. These test cases were also later used to validate the CM.

1.2. Company Background

The thesis documented in this report was performed at DLR. DLR is the German national research centre for aeronautics, space, energy, transport, security, and digitalisation. It was founded (with the current name) in 1969. However, its predecessor organisations are more than 115 years old [12]. During this time, it has

become one of the most prominent research centres for aerospace. DLR's mission mostly addresses sustainable innovation in order to have a positive impact on society without disregarding the environment and our future. They also participate in multiple missions in many sectors, collaborating with partners such as the European Space Agency (ESA) or the National Aeronautics and Space Administration (NASA) [13, 14].

This thesis was done in collaboration with the DLR's Space Launcher Systems Analysis (SART) department, which dates back over 25 years. It is part of the Institute of Space Systems, in Bremen (see Figure 1.4), which researches innovative design methods, space technologies, and space systems. The SART department narrows the scope to analysing and designing future space launch systems. In particular, it looks beyond the classic LVs, investigating all kinds of transporters for orbit transfer, atmospheric re-entry or hypersonic flight [15]. It mostly focuses on concept vehicles in the early phases of development, assessing their performance, cost, and environmental impacts.



Figure 1.4: DLR facilities in Bremen (retrieved from [16]).

2

Literature Review and Project Plan

This chapter presents the literature study's results to assess the current state of research relevant to this project. The need for this work has already been established in Chapter 1. Therefore, the focus here is on determining whether similar approaches have been developed, either directly for launcher remodelling or in related fields such as vehicle modelling more broadly. Particular attention is given to two core areas: the remodelling process itself and statistical methods for parameter estimation and model validation under uncertainty. By examining existing work in these domains, this review aims to identify knowledge gaps, discover transferable methodologies, and determine to what extent current solutions address the challenges outlined. The findings of this chapter will serve as the foundation for formulating the final research objective and research questions.

2.1. Remodelling Process

This section focuses on existing approaches to launcher remodelling. The goal is to identify the different methods employed by other researchers. In many cases, launcher remodelling is closely related to trajectory optimisation. This occurs because reconstructing a realistic LV model often requires accurately replicating the flight trajectory it would take to deliver a payload to orbit. In the DLR context, trajectory optimisation is an integral part of the remodelling process (see Section 3.1.3). It provides a way of validating whether a given configuration can achieve the target orbit with a payload capacity close to its technical specification. For this reason, part of this literature review will also examine trajectory optimisation methodologies.

The optimisation of rocket trajectories (ascent or descent) is not a new advance in science. Many people have tested various methods to achieve the ideal result. However, most examples focus only on the trajectory optimisation part alone or integrate a bit of rocket design without considering every factor at play. Others are only tailored for specific types of rockets, and some are used for optimising current launchers with all the necessary data/information available. At the time when this literature review was conducted, no overarching, multi-disciplinary, and flexible optimisation and remodelling tool was found.

As previously mentioned, the optimisation of a rocket's launch trajectory has been done in different ways and by multiple people. A few examples of work revolve mostly around the optimisation of the descent part of the trajectory, taking into account the throttling changes and the complex and shifting aerodynamic forces [17, 18]. Other research works address full trajectory optimisation [19, 20].

Another case uses an optimisation algorithm that does not require any initial guess and can simultaneously improve trajectory and thrust profile [21]. This last work is designed for liquid rockets in particular, so it's not as global as it is desired for this thesis. Comparably, there is also the case where the research focuses on solid propellant rockets [22]. The proposed algorithm is not only able to optimise the selected LV's launch trajectory, but it also refines the grain geometry for maximum payload capacity.

In [23], the authors present a software tool to optimise both the ascent trajectory and the mass distribution among stages of a multi-stage rocket. The created model incorporates structural design, aerodynamics, control and propulsion systems and stage sizing. The ultimate goal was to maximise the payload-to-lift-off ratio while complying with every safety constraint. The authors of [23] developed an optimisation process that includes variables such as propellant and structural mass or attitude angles. It also involves an optimal stage-sizing part. The propellant and stage masses are computed using parameters such as the mass fraction or the effective exhaust velocity. Simultaneously, a non-linear optimisation is solved, helped by the discretisation of the flight time into multiple intervals as well as some of the parameters. Overall, this procedure shares some resemblances with the one used by DLR (see Section 3.1.4).

Another example is defined in the work developed by Ünal, E. in [24]. The goals of this research project are closely aligned with those in the LV design, modelling, and trajectory optimisation framework presented in this thesis. It provides an approach to the integration of mass modelling, aerodynamic modelling, and trajectory optimisation similar to what was done in [23] and what will be used in this work. The multiple-stage sizing algorithm is identical and relies on the same type of inputs as the previous research work mentioned. However, the final formulas are different.

Concerning the aerodynamic model, there is a simplification of the forces acting on the LV. Only lift and drag are included, so side forces and aerodynamic moments are ignored. This is different from the model used by DLR, which can affect the final results. Still, the way Ünal, E. calculates the aerodynamic coefficients does not differ much from the one used in this thesis and by DLR. In a summarised way, the external shape and volume of the LV define these values. The thrust model is based on rocketry equations and relationships, as is done with the model used in this research project (see Section 3.2.3). Furthermore, the work explores a particle swarm optimisation strategy. It is used to find the optimal LV trajectory and performance based on specified mission constraints. Finally, the whole algorithm is validated with a real case example. Overall, the structure and methodology of Ünal's work and what is done in this thesis are identical.

A similar work was developed by Civek, E. in [25]. A simplification is also made

around the aerodynamic forces (only drag is considered). The optimisation is done for thrust profile, vehicle mass, and trajectory planning, all at the same time. This is one of the key contributions of this work to the literature. It maximises the payload that a certain LV can carry while minimising the gross lift-off mass. The work is also interesting because it compares the results with known configurations and performances. Moreover, it can be applied for the preliminary mission design of a current rocket or to develop a concept design for an LV with certain requirements. However, some aspects to be improved are mentioned, such as controllability or integration of more external factors (wind or other aerodynamic forces, for instance). In addition, further enhancements are desired when it comes to solid rocket motors. All these suggestions are either already included in DLR's tools or will be applied to this thesis.

DLR has also developed a previous model capable of optimising LV trajectories [26]. This work has the particularity of opting for an object-oriented modelling approach. That is the same technique used in this thesis.

There is one more work where trajectory optimisation considering both physical and structural constraints was performed with stage mass sizing [27]. As part of this research project, the authors also improved the LV's configuration to reduce its gross lift-off mass. They applied this to a couple of real examples. The authors of [27] point out that the ultimate goal should be to combine rocket design with achieving the optimal trajectory. That is done in this thesis, aided by DLR's software toolbox.

In [28], Morgado, F. M. P. et al. developed a procedure for coupled rocket design and trajectory optimisation. It minimises the LV's weight while achieving the optimal ascent trajectory. It is a viable tool to design a rocket in a preliminary phase, given a set of mission requirements. However, the results are not close enough to the precision desired for this thesis. This tool seems to be better for designing and not remodelling. Nonetheless, its rationale is relevant and is related to what will be developed in this research project. Finally, an interesting approach proposes a way of sizing several rockets belonging to the same family, that is, sharing commonalities, while optimising their trajectories [29]. This approach eases the design of a set of similar rockets and lowers their development and production costs. However, once again, it is more suited to designing more than simply remodelling existing configurations.

In conclusion, a review of the existing literature on LV remodelling reveals a strong focus on trajectory optimisation and vehicle design, with particular emphasis on stage sizing, aerodynamic modelling, and propulsion system integration. While some studies address uncertainties within optimal control and trajectory planning, these efforts are largely confined to variations in predefined design parameters. They don't deal with incomplete or missing input data. No work was identified that offers the full set of capabilities sought in this research project: the ability to perform complete LV remodelling while handling unknown input parameters.

2.2. Statistical Methods to Handle Uncertainties

In LV remodelling, handling uncertainties in the input parameters is an important challenge, as incomplete or inconsistent data can significantly impact the accuracy

and reliability of the output results. Fortunately, many statistical methods have been developed to enable robust modelling and optimisation even when information is lacking. Examples such as Bayesian computation, Monte Carlo (MC) simulations, or uncertainty quantification frameworks allow systematic incorporation of data variability into computational models, improving their predictive capability. This section explores relevant research work that has addressed uncertainties.

One method that has proven to be successful when dealing with uncertainties in the input parameters is Approximate Bayesian Computation (ABC). It is a powerful methodology for parameter estimation in complex models, especially when traditional likelihood-based methods are infeasible due to a lack of knowledge or computationally expensive likelihood functions [30].

This has been used with a positive impact in many cases. For instance, one work investigating galactic cosmic rays was completed using ABC to estimate values for unknown parameters. They were important to calculate the final value of the desired variable. They allowed the modelling of the parameters that lacked information, similar to the issues faced during this thesis, thus allowing the research to be conducted [31]. Moreover, some authors demonstrate an ABC approach for flow parameter estimation using laser absorption spectroscopy, a technique commonly used in combustion diagnostics. Their method combines experimental spectroscopy data with ABC-based inference to estimate key flow parameters, such as temperature, pressure, and species concentrations, without requiring an explicit likelihood function. This work suggests that ABC can be successfully applied to experimental and computational models where traditional estimation techniques struggle [32].

Furthermore, some authors developed a probabilistic framework using ABC to estimate genome rearrangement rates, allowing them to infer key evolutionary parameters without requiring an explicit likelihood function [33]. Their method relies on generating simulated data, comparing it with observed genome arrangements, and iteratively refining model parameters using ABC. Although their research is in molecular evolution, the core principle of using ABC to estimate unknown parameters can be applied to this thesis. In another example, ABC was applied to a different field than rocket remodelling [34]. The focus was on structural integrity assessment, and it required estimating unknown parameters in pipe-like structures using ultrasonic-guided wave signals. Again, despite being applied to a different purpose, the used framework can be adapted to accurately predict some uncertain input parameters in the work developed throughout this thesis.

Another developed work compares two Bayesian methods: Bayesian Model Updating and ABC [35]. The study highlights the strengths of ABC when dealing with complex, high-dimensional models. A similar challenge will be dealt with in this thesis, given the vast amount of parameters that can be unknown.

One variation of the traditional ABC was developed to reduce computational cost [36]. This research presents an approach around the traditional ABC to improve parameter inference in models where the likelihood is hard to deal with. The method enhances the efficiency of ABC by lowering the variance and thus the errors, making it suitable for complex dynamic models with uncertain parameters.

Also, an improved version has been developed [37]. This study introduces an enhanced ABC Sequential Monte Carlo (ABC-SMC) algorithm aimed at improving efficiency in model selection and accuracy in parameter estimation. The proposed methodology addresses challenges associated with the more traditional ABC, such as low efficiency in model selection and estimation inaccuracies, making it applicable to complex systems where input parameters are uncertain.

Overall, these studies illustrate the versatility of ABC in handling parameter uncertainties across various fields. This method can enhance the robustness of the LV remodelling process by effectively addressing the lack of complete input parameter information.

The last investigated methods to deal with the uncertainties in the input parameters for the rocket remodelling process are MC algorithms. It is a more "brute force" way since it relies on a large amount of random samples to generate the probability distribution of outcomes.

A study employed this strategy to assess performance variations in liquid rocket engines with gas-generator cycles [38]. By accounting for uncertainties in parameters like the turbine and pump efficiencies or the nozzle geometry, this research provides a method of generating output results based on multiple random combinations of the input values. This is useful when there is a lack of information about the input parameters, as will happen in this thesis.

In another field, MC was used to estimate uncertainty in rising sewer models, where input parameters, such as fluid properties, boundary conditions, and structural variations, can be hard to measure directly [39]. Although the work is not related to rockets, the methodology can be applied to this thesis, as it is an efficient way to handle incomplete input data. Moreover, another study presents a probabilistic mid-air collision risk model to integrate uncrewed aircraft into unsegregated airspace [40]. Here, an MC method is used to reconstruct a probable traffic scenario for the aircraft. Input parameters like traffic density are unknown. Hence, implementing this strategy helps attenuate this issue. One final example of MC implemented to solve uncertainties within input parameters in a subject other than rockets is in [41]. The author uses MC simulations to evaluate the robustness of guidance, navigation and control algorithms across diverse scenarios. Parameters like wind field, sensor errors or magnetic fields vary according to the random values generated. This way, the reliability of the results is enhanced. The lack of knowledge about these parameters is thus bypassed.

In another case, MC was used to predict the internal ballistic performance of a solid rocket motor under the influence of numerous random factors [42]. These results were then compared to a limit deviation design method. They turned out to be better, which strengthens MC's position as a strong valuable way to mitigate the lack of information in input parameters.

A modified version of MC was used in [43]. The author applied Latin Hypercube (LH) sampling to analyse kinetic and inertial soil-foundation-structure interactions. The idea was to overcome the uncertainties from the structural and geotechnical properties.

The LH method was implemented to reduce the computational cost compared to the traditional MC sampling. Therefore, it might be useful for this research project, given the limitation in computational power. In addition, in [44], the authors use LH sampling to generate samples while handling uncertainties. Zhang et al. go even further, by adding an optimisation after the sampling is performed [45], similar to what will be done in this thesis. Thus, despite being applied to different research areas, these examples demonstrate how MC and its variations can tackle uncertain data.

From all the examples above, the MC method and its variants can be inferred to be a good way to deal with the lack of input parameters. Its versatility allows it to be used in multiple fields and problems, as the one dealt with in this thesis.

2.3. Research Gap

Based on the literature review, there have been significant advancements in works related to rocket remodelling or dealing with a lack of information/uncertainties in input parameters. However, to the best of the author's knowledge, nothing simultaneously addressing both topics has been proposed in literature yet.

On the one hand, prior works related to rocket remodelling have only dealt with known parameters when reconstructing and designing these vehicles. Usually, these studies assume complete datasets, missing out on the uncertainty part that is often found in real-case scenarios. On the other hand, extensive research has been conducted on dealing with uncertainties and unknown parameters in various fields. ABC, MC, and LH methods have proven to be useful in some way when it comes to this. However, no literature has implemented any of these when remodelling rockets. This thesis aims to bridge this gap.

This research will provide a novel approach for remodelling LVs, even in the absence of complete technical specifications, by integrating methods to handle unknown parameters in the remodelling framework. It will combine physical modelling, trajectory optimisation, and advanced statistical methods to handle uncertainties. This will enable the generation of mass, aerodynamic, and trajectory models from incomplete datasets. By covering the research gap and addressing the need for such a tool, this thesis intends to facilitate the creation of a more complete exhaust inventory. Once a database with the recently launched rockets' models is built, this exhaust inventory can be computed, and further conclusions can be drawn through the S3D project.

A detailed explanation of the steps undertaken in this thesis is available in Appendix A. There, it is possible to visualise a Gantt chart showing the duration and sequence of each task, as well as a description of each work package.

2.4. Research Goals

This section presents the research goals. It will help the reader understand the thesis's focus and relevance and the questions it addresses. In the first place, the research objective will be presented, followed by the research question and sub-questions.

2.4.1. Research Objective

This research's objective is:

“To build a semi-automated rocket remodelling framework that integrates data management, model generation and validation while handling incomplete or uncertain input parameters, creating the mass and aerodynamic models along with an optimised trajectory for each launch vehicle, and defining the possible ranges for the unknown technical characteristics with computational statistics.”

To better understand the research objective, it is better to divide it into several sub-objectives:

- To establish a data management methodology by organising and processing available technical input data while identifying missing or uncertain parameters.
- To automate aerodynamic, mass, and trajectory models generation by implementing and adapting existing DLR software tools to the research's needs.
- To implement a validation process that determines the possible ranges of technical characteristics by using computational statistics methods to cover their full range.
- To develop a method to select the most valid/realistic models based on the validation process and the range of the technical characteristics.

2.4.2. Research Question

Now that the research goals are defined, the research question will be elaborated.

“How can the remodelling process for launch vehicles be automated to create mass and aerodynamic models and optimise their trajectories, while handling incomplete or uncertain input parameters?”

Additionally, given the complexity and scope of the problem at hand, the research question was split into several sub-questions, similar to what was done with the research objective. The resultant sub-questions are:

1. What are the key challenges in managing and processing incomplete or uncertain input parameters for rocket remodelling?
 - (a) What data structure or input data sheet format can be used to handle varying levels of data availability?
 - (b) What is the minimum set of input parameters required to realistically remodel a launch vehicle?
2. How can mass, aerodynamic, and trajectory models be constructed with a shortage of input parameters?
 - (a) How can mass and aerodynamic models be developed when some technical characteristics of the launch vehicle are unknown?
 - (b) What influence does each uncertain input parameter have on the final result?

3. Which statistical and computational method(s) is(are) best suited to define the possible ranges of unknown parameters?
4. How can the proposed rocket framework have the same accuracy for launch vehicles with varying data quality and complexity?

2.5. Requirements

This section will list and briefly explain the requirements for this project. They are grouped into key areas to ensure a structured and comprehensive project development.

1. Functional Requirements

- (a) The framework shall support two rocket modelling paths:
 - i. A test model based on analytical rocketry equations.
 - ii. A complex model using DLR's trajectory optimisation tools.
- (b) The input data sheet shall dynamically assess which parameters are still required based on the user's input.
- (c) The input data sheet shall use the already written inputs to calculate others.
- (d) The input data sheet shall notify the user when some combinations of parameters are inconsistent.
- (e) The input data sheet shall contain all the parameters required by the remodelling tools.
- (f) The input data sheet shall be flexible and capable of dealing with any type of expendable rocket.
- (g) The input data sheet shall handle both exact values and ranges in case of an unknown parameter.
- (h) The framework shall store all relevant results for future reference or additional processing.
- (i) The architecture shall support future extensions, including new models or functionalities.

2. Performance Requirements

- (a) The test model shall run in under 20s.
- (b) The complex model shall run in under 48h.

3. Interface Requirements

- (a) The input data sheet shall have a user-friendly interface.
- (b) The input data shall be readable by Python code.
- (c) User's interference shall be minimised while the algorithm runs.

4. Visualization and Output

- (a) The user shall visualise intermediate and final results for both the test and complex models.
 - i. The user shall be able to access all valid configurations and the corresponding parameters.
 - ii. The algorithm shall update or write a new Excel sheet with completed or revised data fields, especially for previously unknown parameters.

5. Validation and Selection Criteria

- (a) The framework shall validate all model outputs to guarantee they correspond to physically and technically feasible launchers.
- (b) The user shall be able to choose criteria to select the final model (e.g., lowest mass, highest thrust).
- (c) The system shall include a sensitivity analysis module to assess how input uncertainties influence output performance and model feasibility.

3

Methods

This chapter presents the background and overall picture of the thesis by combining both theoretical principles and their practical implementation. On the one hand, it introduces the underlying theory required to build the framework. On the other hand, it describes how these principles are applied in practice through the proposed workflow. The chapter is therefore structured to move from theory to application. Firstly, all the auxiliary theoretical concepts are explained. Then, the methodology for integrating them into a semi-automated launcher remodelling process is outlined. Finally, the expected results of the workflow are defined, together with the verification and validation strategy that will ensure the reliability of the proposed approach.

3.1. Theory

The following section lays the theoretical foundations for the framework developed in this thesis. It presents the auxiliary methods, equations, and tools used. The goal is to provide the necessary context to understand the principles behind the developed LV remodelling framework that handles uncertain data.

The section covers both the remodelling aspects, such as rocketry equations, and the statistical approaches used for uncertainty quantification and parameter estimation. It also describes the two models used to analyse LV configurations. Finally, it details the software tools that form the backbone of the workflow. They ensure seamless integration between remodelling components and statistical methods.

3.1.1. Rocketry Equations

As shown in Section 1.1, the remodelling process starts with the input data sheet (IDS). As a preliminary step in the methodology, a set of rocketry equations will be implemented within it. These equations allow the computation of key input parameters, such as the masses or the performance variables of the engines, based on some available subset of input variables. Their purpose is twofold: first, to identify which known parameters are sufficient to infer others through rocketry equations; second, to assist in estimating unknown values before starting the remodelling process.

The implementation in the IDS is based on core equations, mostly from rocket theory found in literature [46]. These include, but are not limited to, the following:

$$F = m \cdot w \quad (3.1)$$

which relates thrust (F) with the mass flow rate (m) and the exhaust velocity (w);

$$I_{sp} = \frac{w}{g_0} = \frac{F}{m \cdot g_0} \quad (3.2)$$

this equation shows the relationship between the specific impulse (I_{sp}) and the exhaust velocity (and thus the thrust and mass flow rate due to Equation 3.1;

$$F = m \cdot w + (p_e - p_a) \cdot A_e \quad (3.3)$$

which shows the correlation of thrust with linear momentum through the nozzle exit ($m \cdot w$) and a pressure-related term ($(p_e - p_a) \cdot A_e$), where p_e is the pressure of the gas flow at the nozzle exit, p_a is the constant ambient pressure (atmospheric pressure), and A_e is the exit area of the nozzle; this equation is used at sea-level whereas Equation 3.1 is a simplification for vacuum conditions;

$$F_{sea-level} = F_{vacuum} - p_a \cdot A_e \quad (3.4)$$

this expression is used to estimate the engine diameter based on the thrust values, or the other way around; it results from rewriting Equation 3.3 for both sea-level and vacuum conditions and then combining the expressions;

$$SI = \frac{M_{dry}}{M_{dry} + M_{propellant}} \quad (3.5)$$

an equation that is used to correlate the dry and propellant masses through the structural index (SI) [47];

$$M_{total} = M_{dry} + M_{propellant} = M_{dry} + M_{propellant_{ascent}} + M_{propellant_{descent}} + M_{propellant_{reserve}} + M_{propellant_{residual}} \quad (3.6)$$

this last expression displays the relationship between the different masses considered in this remodelling process. Only the most relevant equations are listed here. The other are easily traced back to the variables already mentioned here.

This equation-driven structure present in the IDS also plays a pivotal role in the statistical estimation of unknown parameters. Once a sufficiently large subset of inputs is provided, the embedded equations identify which values are missing and which can

be estimated. For cases where key parameters are unavailable from public sources, the IDS will accept lower and upper boundaries. Additionally, in case the user has provided ranges for other inputs that can be combined to calculate another one, they will be used to calculate not a single value, but a minimum and maximum values. These will define the two extremes of the range used for sample generation. More details about how the IDS behaves are available in Section 3.2.1. Thus, the analytical foundations laid by these equations serve not only to complete the IDS but also to start the uncertainty-handling strategy of the overall methodology.

3.1.2. Statistical Methods to Handle Uncertain Parameters

This research assesses three main statistical techniques to address the challenge of incomplete or uncertain input parameters. The central aim of these methods is to generate and assess samples for each unknown variable, thereby enabling a systematic exploration of the full unknown parameter space. This way, one can achieve the desired valid ranges for each initial unknown variable. Monte Carlo simulations, Latin Hypercube sampling, and Approximate Bayesian Computation are considered, each offering a different way to capture and manage uncertainty.

In brief, Monte Carlo simulations are commonly used to generate multiple random samples that can create an output probability distribution. Latin Hypercube sampling is a method similar to MC, but it aims to spread the sample points more evenly across all possible values. The last procedure is Approximate Bayesian Computation. It can be used to estimate unknown parameters by comparing simulation outputs with observed data.

These statistical methods are useful tools for alleviating the problems that arise from the lack of input parameters. They will be applied to make sure that the remodelled LVs remain consistent and as close as possible to the real configurations, despite data availability limitations. Chapter 2 provided some context on prior applications of these procedures.

Monte Carlo

Monte Carlo methods are a class of statistical algorithms that rely on consecutive random sampling to obtain numerical results [48]. These techniques are commonly used to solve deterministic problems that are too complex for analytical solutions. Some examples include high-dimensional integration, optimisation, and uncertainty quantification. The core idea is to model the problem mathematically and then generate random samples from the relevant probability distributions. Afterwards, these samples are used to estimate quantities of interest. As the number of samples increases, the approximation generally converges to the true value. Therefore, the accuracy improves as more iterations are performed. MC methods are especially valued for their ability to handle problems involving randomness, uncertainty, or a large number of variables. These are all types of situations where traditional deterministic approaches may be infeasible [49, 50].

MC simulations are usually highly flexible. However, they may require a large number of samples to achieve convergence, especially when dealing with high-dimensional problems.

Latin Hypercube

In this research project, two MC-based methods will be tested: random sampling and Latin Hypercube (LH) sampling. The first one refers to the idea of randomly generating samples. On the other hand, LH is an advanced form of stratified sampling. It improves the efficiency of MC simulations by ensuring that the entire input space is sampled more uniformly. Introduced by McKay et al. [51], LH divides the probability distribution of each input variable into equally probable intervals and draws one sample from each. This way, it guarantees that the space is populated more uniformly, avoiding clustering and redundancy. Unlike purely random sampling, LH guarantees that each portion of the input distribution is represented (see Figure 3.1 for better visualisation). Thus, it is more suitable for high-dimensional problems and it reduces the number of simulations needed to obtain accurate statistical estimates (as already described in the literature review in Chapter 2).

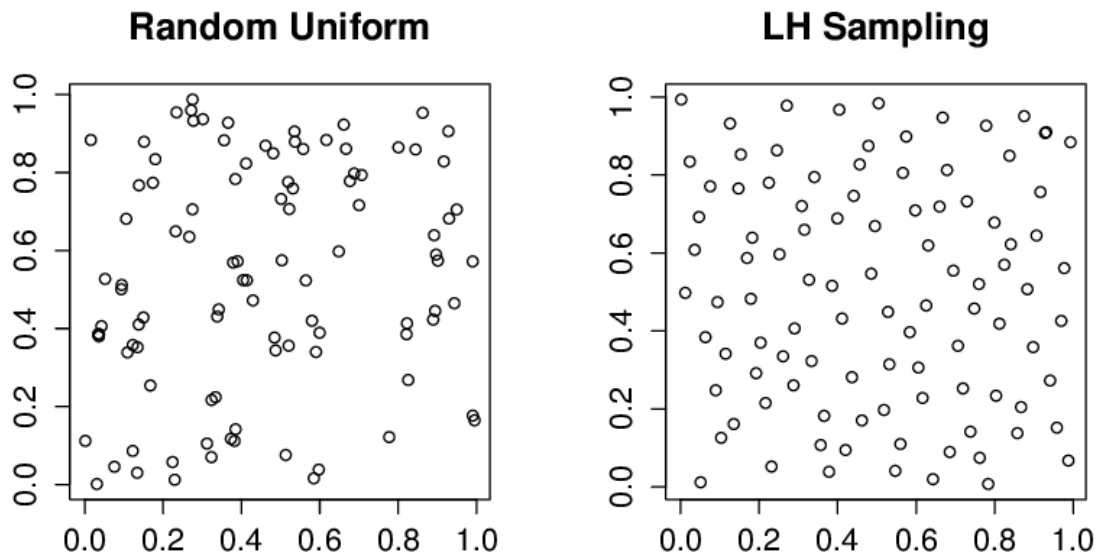


Figure 3.1: Monte Carlo-based sampling methods visualisation (retrieved from [52])

LH sampling is widely used for sensitivity analysis, design of experiments, and uncertainty propagation. Its advantages are its computational efficiency, improved space coverage, and compatibility with linear and non-linear models [53]. This makes it appealing to the LV remodelling framework presented in this thesis, where multiple input parameters may be uncertain, and a balance between fidelity and computational cost is required.

Approximate Bayesian Computation

Approximate Bayesian Computation (ABC) is a type of likelihood-free inference method. In statistics, inference refers to drawing conclusions about a complete system based on a sample taken from it. It derives from the Bayes theorem [54]:

$$p(\theta | y) = \frac{p(y | \theta)\pi(\theta)}{p(y)}, \quad (3.7)$$

where θ denotes the parameters of interest; $f(y|\theta)$ is the likelihood: the probability of observing the data y given a parameter θ ; $\pi(\theta)$ is the prior: the knowledge/assumptions one has before observing the new data; and $p(\theta | y)$ is the posterior distribution, i.e. the updated knowledge about the parameters after incorporating the new data.

However, it is often the case that the likelihood function cannot be derived or is too computationally expensive to calculate. This is where ABC provides a solution, avoiding its explicit calculation, hence being designated as a likelihood-free inference method. ABC relies on the ability to simulate data from a selected model, given a set of input parameters. The simulated data are then compared to observed or reference data using summary statistics and a predefined distance metric. If the distance between the simulated and observed data is sufficiently close, usually below a certain tolerance, the corresponding parameter values are accepted as samples from the approximate posterior distribution [55, 56].

Bayesian inference relies on updating prior beliefs about parameters θ given observed data through the posterior distribution [57]:

$$\pi(\theta | y_{obs}) = \frac{f(y_{obs} | \theta)\pi(\theta)}{\int f(y_{obs} | \theta)\pi(\theta)d\theta} \quad (3.8)$$

where, as before, $f(y_{obs}|\theta)$ is the likelihood and $\pi(\theta)$ the prior.

This prior distribution may be based on our previous understanding of likely values for θ . If one knows that a parameter is more likely to be in a certain range of values, its distribution can be defined to represent this situation. On the other hand, one might know nothing about the distribution of θ . In that case, its prior should be uninformative or spread widely over the allowable range for θ . Overall, it should represent the previous belief one has in the parameter's probability of a random value generation. In addition, the likelihood function is used to update the prior belief based on the new observations made. It represents the probability of observing the given data under a set of parameters.

This method is particularly beneficial when it is computationally expensive or infeasible to compute this likelihood function [58, 59]. To overcome this problem, ABC is based on the following procedure (generic version):

1. It generates a parameter (or set of parameters if it is a multi-dimensional problem) from the prior distribution: θ_i . This sample is highly influenced by the type of distribution chosen.
2. Using the generated sample, a new result (y_i) is computed through the implemented model.
3. The distance between the generated result and the observed data is calculated:
 $d = \|y_{obs} - y_i\|$

4. If the distance is smaller than a pre-defined threshold value (ϵ), the sample is accepted as valid: $d < \epsilon$. If not, the sample is rejected by the algorithm.
5. The previous steps are repeated until enough valid samples are produced (chosen by the user).
6. Lastly, the prior distributions are updated based on the samples that produced a valid result.

The accepted samples (θ_i) approximate the posterior $\pi(\theta \mid y_{obs})$ with an accuracy that improves as ϵ tends closer to 0.

Approximate Bayesian Computation - Sequential Monte Carlo

A particular variant of the ABC method was selected for this thesis: the Sequential Monte Carlo method (SMC). An already implemented library was available for Python at the time of this choice [60]. Since the main scope of this work was not to develop new statistical methods, but to implement them, this allowed for considerable time saving. ABC-SMC builds upon ABC fundamentals but adds SMC techniques, which enhance the efficiency of the parameter space's exploration and the convergence to the target posterior distribution.

In ABC-SMC, a population of parameter samples is propagated through a sequence of intermediate distributions, each defined by a progressively stricter tolerance threshold (ϵ). The process begins by sampling parameters from the prior distribution. Afterwards, one has to simulate data from the model. The simulated data is then compared to the observed data using a distance metric, and only those parameter samples that produce simulated data close enough to the observed one are retained [61–63]. The tolerance definition involves a trade-off between computational efficiency and accuracy of the posterior distributions. While the former increases with higher ϵ , the latter is improved as the tolerance decreases [64]. Defining this threshold depends on the specific model, statistical choices and distance function, and it may require intuition or experimentation.

Unlike the simple ABC rejection method, ABC-SMC introduces a sequence of decreasing tolerances ($\epsilon_1 > \epsilon_2 > \dots > \epsilon_n$). These threshold values gradually guide the population of particles from the prior towards the posterior distribution. At each stage, the surviving particles are perturbed (for example, using a kernel) and reweighed according to their prior probability and distance results (a more detailed explanation can be found in [65]). Then, the process is repeated with a tighter tolerance until the lowest chosen tolerance value is achieved. This sequential approach allows ABC-SMC to focus computational effort on regions of high posterior probability efficiently. Consequently, it reduces the risk of getting trapped in low-probability areas and improves the overall sampling efficiency.

This method has multiple possible applications, such as updating the uncertainty of the parameters over time, sequential addition of new information to the model, or Bayesian calibration [66]. In the context of this rocket remodelling project, ABC offers a powerful framework for estimating unknown parameters based on partially available or uncertain input data. By comparing the outputs of simulated LV configurations to known performance metrics (e.g., payload capacity), it becomes possible to infer

plausible parameter ranges. The biggest advantage is that the method doesn't require a complete rocket specification or an explicit likelihood function. Hence, it is well aligned with the needs of LV remodelling while handling uncertain data, as explored in this thesis.

3.1.3. Calculation Models

As outlined in Chapter 1, once the IDS is fully developed, the remodelling process can proceed through two paths: the Test Model or the Complex Model, according to Req. 1a. The main rationale behind this choice is to reduce the computational time. A faster model was needed since the statistical methods testing phase requires calculating and assessing millions of samples, and the complex model is too computationally expensive. This section will describe the theory behind both models.

In essence, the models aim to evaluate the performance of the LVs by calculating parameters, such as the payload, which can be used for verification and validation (VV). Within the framework, these models are crucial. Without them, it would be impossible to bridge the gap between raw input data and the LV configurations that will be used for the exhaust inventory that motivates this research.

Test Model

The Test Model plays a vital role in the overall project. It serves as a tool for quick analysis. It was developed to support fast statistical evaluation and parameter exploration, due to Req. 2a. Because of its low computational cost, it is particularly well-suited for applications such as uncertainty quantification, parameter sweeps, or statistical estimation of missing inputs.

The TM relies exclusively on simple mathematical equations, avoiding complex numerical optimisations. In this framework, the required velocity change (Δv) for the launch vehicle to reach its target orbit is estimated using classical rocketry equations. Several simplifying assumptions are introduced to streamline the model, which may affect accuracy but allow fast calculations. From these calculations, key performance indicators such as the maximum payload capacity can be derived. This enables this model's VV as described in Section 3.4. Additionally, the TM serves as a testing platform for the statistical methods. Due to its high computational speed, it can evaluate multiple computational statistics methods and identify the one that most effectively handles incomplete input parameters.

This section describes the TM's structure and main functionalities and assumptions in more detail. Starting with the structure, the idea was to develop a model that could calculate the required Δv for a LV to reach its target orbit without complex calculations. In order to do so, the author first had to find the main components that are part of the Δv budget equation. Once gathered, one could think of how to estimate each using exclusively the rocket's input parameters, plus the target orbit and the launch site's characteristics. After some research, the expression to calculate this budget was found [67]: Equation 3.9.

$$\Delta v_{margin} = \Delta v_{rocket} + \Delta v_{Earth} - (\Delta v_{potential} + \Delta v_{orbit} + l_{grav} + l_{drag} + l_{thrust}) \quad (3.9)$$

where Δv_{rocket} is the velocity change that the rocket can achieve; Δv_{Earth} is the velocity contribution due to the Earth's rotation; $\Delta v_{potential}$ is the orbit's potential energy term; Δv_{orbit} refers to the orbital velocity of the target orbit; and the final three parts correspond to the gravitational losses, drag losses and thrust losses, respectively. Δv_{margin} corresponds to the difference between the terms that positively contribute to the velocity change that LV can produce and the terms that negatively contribute to it.

At its core, the TM is trying to determine whether a given LV configuration can deliver a payload to a specified orbit. This requires balancing the velocity the rocket can generate against the velocity demands imposed by the target orbit and the various physical losses that occur during ascent. In other words, the model evaluates if the propulsion system provides sufficient Δv to overcome gravity, drag, and thrust inefficiencies while still reaching the required orbital velocity.

Complex Model

Contrary to the TM, the Complex Model depends on DLR's software to test the performance of a LV. In particular, it uses a trajectory optimisation tool (Section 3.1.4) with mass and aerodynamic models. DLR's software programs are comprehensive and account for, among others, gravitational and aerodynamic losses, the complexities of Earth's shape and atmosphere, or the feasibility of a given trajectory. This last one refers to visualising the trajectory and the values of certain parameters, such as dynamic pressure or heat loads, and judging if they are realistic, even if the rocket was mathematically capable of reaching the target orbit. Therefore, its results are far closer to reality than those from the TM. However, this comes at the cost of significantly higher computational efforts. Therefore, the statistical sampling methods are first tested in the TM before being implemented in the CM. Its detailed description and implementation are laid out in Section 3.2.4.

3.1.4. Tools

This section introduces the computational tools employed in this thesis. Most of the tools are part of the DLR software, complemented by Python-based implementations developed by the author. Python acts as the central interface, enabling the automation of workflows, the transfer of data between modules, and the integration of statistical methods with the remodelling framework. Together, these tools form the technical foundation on which the methodology of this thesis is built.

STSM - Space Transportation System Mass

Space Transportation System Mass (STSM) is a computational tool designed for mass estimation and analysis of single and multi-stage space transportation systems. It is particularly useful in the preliminary design phase, providing essential data for the early-stage evaluation process.

The tool uses empirical formulas and methods from different sources [68–70] to estimate the mass of individual vehicle components and their overall distribution.

Internally, it iteratively modifies the mass of individual components to guarantee a consistent global mass value, particularly when certain elements are defined as a percentage of the vehicle's total mass. Additionally, STSM allows the inclusion of design margins, which are crucial in the conceptual design phase.

The user inputs a limited set of parameters related to vehicle subsystems, such as thrust level, propellant mass, engine mass or fairing mass. Then STSM calculates the mass of various subsystems for each stage, using one of two methods. The first one is when the user specifies the mass of a certain component. The other option is to provide auxiliary parameters that the program uses to calculate the actual mass of the component through embedded equations. One example is the fairing mass, which is often hard to find. The user can define the maximum dynamic pressure faced by the LV and the fairing's surface area. Then, STSM computes the mass. Beyond mass estimation, STSM can compute the centre of gravity variation along the vehicle's trajectory. This is possible for the complete vehicle and for a single stage.

CAC - Calculation of Aerodynamic Coefficients

Calculation of Aerodynamic Coefficients (CAC) is a computational tool used to estimate the aerodynamic properties of space transportation systems, mostly during the preliminary design phase. It provides rapid calculations of aerodynamic coefficients, including lift (C_L), drag (C_D), and moment (C_M) coefficients, in the subsonic, transonic, and supersonic regimes in ascent, re-entry, and cruise flight conditions.

This tool is based on semi-empirical methods similar to those used in Missile DATCOM [71], offering fast computational performance while relying on approximate aerodynamic models. Instead of generating a computational mesh, CAC decomposes the vehicle's geometry into its primary components, such as bodies, wings, flaps, etc. Then, it assesses their aerodynamic coefficients separately based on their exterior geometry/shape. The general aerodynamic characteristics are then obtained by superimposing these individual contributions [72].

Despite its simplified approach, CAC delivers results that align well with more complex computational fluid dynamics (CFD) simulations and wind tunnel experiments [73]. It can also compute the centre of pressure as a function of the angle of attack, Mach number, and control surface deflections.

In this study, CAC will provide the aerodynamic coefficient data necessary for trajectory simulations and stability analyses. For the tool to work accurately, a simplified representation of the complete geometry of the vehicle is needed. In turn, this requires some data about the LV, such as stages' lengths and diameters, or fairing dimensions.

TOSCA - Trajectory optimisation and Simulation of Conventional and Advanced Spacecraft

The Trajectory Optimisation and Simulation of Conventional and Advanced Spacecraft (TOSCA) tool is a numerical tool used to simulate and optimise the ascent and descent trajectories of spacecraft. It integrates the point-mass dynamics equations in three dimensions, making it well-suited for trajectory analysis [72]. However, it does not

account for six-degree-of-freedom (6-DOF) motion, as rotational dynamics are not included. Nonetheless, it includes some trajectory control values that simulate them, such as angles of attack or bank angles.

Despite this limitation, TOSCA offers extensive flexibility by incorporating various trajectory control mechanisms, such as angle of attack, bank angle, and thrust vector control.

TOSCA is used primarily to determine the optimal trajectory to maximise the insertion of the payload mass into a given orbit. It can also be used only to simulate a trajectory, without optimising it, based on the user's inputs. These inputs have to be provided in both cases. They include some of the launcher's characteristics (propulsion, aerodynamics, and mass properties), the launch site coordinates, and the target orbit characteristics (e.g. perigee and apogee altitude). Once an optimal solution has been found, an output file is generated. It contains multiple plots showing important parameters that allow an overarching trajectory visualisation, as shown in Figure 3.2.

The optimisation process considers initially defined parameters, such as the pitch rate after vertical ascent, the angles of attack during powered flight, or the burn duration of each stage. It is often the case that these parameters need different values after a first manual optimisation attempt does not converge. This is one of the reasons why this process is so time-consuming.

Moreover, TOSCA can handle different orbit insertion modes, depending on the desired boundaries and constraints. Additionally, in cases where the upper stage can be re-ignited, the tool can support transfer orbit calculations, followed by a secondary burn to achieve circularisation.

Together with STSM and CAC, TOSCA is one of the tools encompassed in what is designated by SART toolbox.

CMA-ES optimiser

This thesis uses a genetic algorithm wrapped around TOSCA to perform the trajectory/payload optimisation in the CM. The algorithm is the Covariance Matrix Adaptation Evolution Strategy (CMA-ES). CMA-ES iteratively adapts the sampling distribution of candidate solutions based on their performance. It is well-suited for problems with complex, high-dimensional search spaces such as trajectory optimisation. For more details about the theory behind it, see Hansen [74]. The genetic algorithm serves as a global optimiser, iteratively searching for input configurations that maximise payload-to-orbit performance. At the same time, they must satisfy physical and operational constraints.

The optimisation process begins by defining search bounds for TOSCA's initial condition parameters, such as azimuth and pitch rate; the payload also needs defined search bounds. Additionally, the launch site coordinates have to be specified; the stage-specific control points, including angle of attack and thrust control profiles, are also specified. For each stage, the user can decide which of these parameters are to be optimised (usually, one wants to optimise the angles of attack for the upper stages). Additionally, target orbit tolerances (e.g., perigee altitude, inclination) and

flight constraints (e.g., maximum dynamic pressure and acceleration) are imposed to ensure the feasibility of candidate trajectories. They will dictate which trajectories the optimiser should accept or reject.

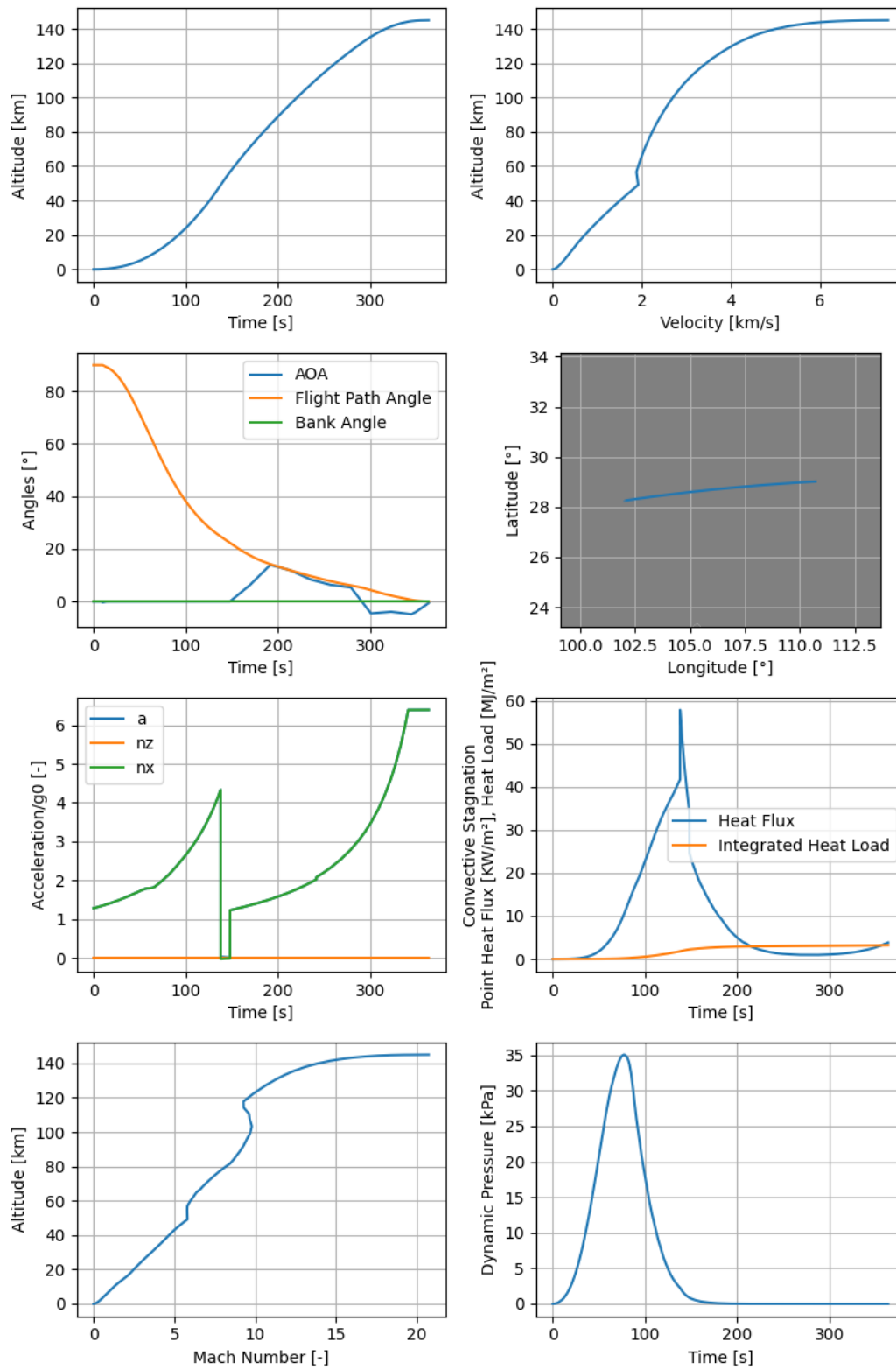


Figure 3.2: TOSCA output visualisation example - Long March 2C Low Earth Orbit (LEO) mission.

During optimisation, the genetic algorithm generates a population of candidate solutions by sampling within the predefined bounds. Each sample is evaluated by simulating the corresponding trajectory in TOSCA with a certain payload mass. If the resulting trajectory satisfies all constraints, the configuration is considered valid, and the optimiser attempts to increase the payload for that configuration. If the trajectory violates one or more constraints, the solution is discarded. Moreover, the feedback is used to steer future sampling toward more promising regions of the search space.

A solution is deemed acceptable if it reaches the target orbit perigee with a valid inclination, and any apogee above the pre-defined one is allowed. This flexibility makes the optimiser faster than if the apogee altitude were a constraint, since it enables more trajectories to be explored, increasing the potential to find a more efficient one. In addition, this does not compromise its accuracy, since the results are still very close to the original target orbit. If the apogee were much higher than the desired one, the payload would be lower, as the orbit would require a higher Δv . Hence, this would not be the optimal case. The optimisation process continues until the convergence criteria are satisfied or a maximum number of generations is reached.

CLAVA

DLR has developed the Concurrent Launch Vehicle Analysis (CLAVA) tool to support the design and analysis of launch vehicles [75]. It acts as a centralised data structure and single source of truth for all relevant rocket data. By concentrating all required information in one place, CLAVA eliminates the risk of inconsistencies between input files, formats, and software components, as long as the data it stores is accurate. This approach improves the robustness of the overall remodelling framework. It ensures that all simulations and models, whether related to aerodynamics, mass, or trajectory, are driven by the same set of validated inputs.

Beyond consistency, CLAVA enables automation and integration across various stages of the launcher analysis workflow. Once the configuration is defined, it can be accessed and reused seamlessly by multiple software tools, such as the ones in the SART toolbox used in this project (e.g. TOSCA). This reduces the need for manual intervention and reformatting. Thus, it increases the overall efficiency by minimising human errors.

CLAVA's structure reflects the hierarchical nature of a LV. At the top level, it models the rocket as a whole. Each rocket includes a main body composed of one or more stages, as well as additional components such as payload fairings. Moreover, a rocket can have other bodies, for instance, to represent the boosters. Each stage, in turn, has a certain number of specified engines and is characterised by its structure parameters. The stage can then be linked to a particular engine containing the propulsion parameters. This modular and extensible structure makes CLAVA well-suited for supporting both simple and highly complex LVs. Such a feature is what makes this tool so useful for this thesis and the general DLR's rocket analysis process.

In this project, CLAVA will be used as the central repository for each rocket's parameters read from the interactive IDS. These inputs will be integrated into a valid CLAVA model once the user provides the necessary parameters, either as fixed values or ranges. It will then be combined with an Uncertainty Quantification (UQ) model.

When the sampling is finished, multiple CLAVA files will be generated, one for each set of parameters. Furthermore, without additional data preparation, an adapter will use this information to create the SART toolbox files, such as the STSM input file. CLAVA will be the "source" of all data, and it will store it in an organised format that is easily read.

UQ

In the context of launcher remodelling, managing uncertainties is a critical step to ensure realistic and reliable predictions. In particular, given the variable public data availability for LVs, this plays a vital role in this thesis. The DLR has developed the UQ framework [76] adopted in this thesis. It systematically captures and propagates input uncertainties through the launcher remodelling process.

The adopted framework follows a modular and flexible architecture (similar to CLAVA), allowing it to be applied consistently across both the test and complex remodelling approaches. It begins by defining the uncertainties in the input parameters. These uncertainties are typically caused by incomplete information, ranges of reported values in literature, or engineering estimates. In the pre-processing step, these uncertain parameters are represented using probability distributions or bounded intervals, which are then propagated through the rocket performance models. Each variable is saved, along with its lower and upper bounds.

For the propagation itself, statistical methods are used. These define how the samples are generated. Afterwards, they are used to calculate the outputs. In this work, the outputs will be the payloads achieved for each mission.

By integrating this UQ approach into the remodelling workflow, the developed methodology not only delivers accurate estimates for rocket performance parameters but also provides ranges and samples for the uncertain inputs, offering more realistic outputs.

Adapters

A set of adapter scripts was used to easily transfer data from the Python/CLAVA/UQ framework to the SART toolbox framework. These adapters act as intermediaries, handling the formatting, transfer, and parsing of data between Python and the SART software environment. This approach allows the output of the statistical sampling process, such as parameter sets generated, to be directly used as an input for trajectory simulations. Due to these adapters, the whole remodelling framework is smooth and modular, and it functions efficiently across different tools.

3.2. Practical Implementation

This section presents how the proposed launcher remodelling framework that handles uncertain data is implemented. Its structure reflects its modular nature. Firstly, the storage of input data and the determination of unknown parameter ranges are described. Then, the integration with external software tools and adapter functions is detailed. Finally, the two modelling paths are introduced, showing how each one is applied depending on computational requirements. By bridging theory and execution,

this section demonstrates how the framework can be effectively applied to remodel LVs while handling incomplete or uncertain data.

3.2.1. Data Collection

This section will address the process from collecting the data necessary for the remodelling process to having it ready to be used for sample generation. The first task of the whole remodelling framework is to gather all the information about the rocket. Therefore, creating an effective interface to store all the data is crucial to guarantee a solid starting point for a successful framework.

Input Data Sheet

This interface has to meet some requirements, as shown in Section 2.5. So, first of all, it is relevant to break them down.

These requirements are the major drivers for the interface format choice. The first one (Req. 1b) means that the IDS has to recognise the parameters that already have values. Then, using pre-defined equations (see Section 3.1.1 for more details), it should indicate which other parameters are redundant to fill in, by, for instance, greying those fields out. This facilitates the research process and saves the user some time, since it avoids looking for values that can be immediately obtained. Req. 1c is related to the previous one, as the IDS not only has to identify the parameters that it can compute, but also calculate them automatically. Furthermore, Req. 1d is important to point out inconsistencies in the input values, avoiding errors later in the remodelling process. Finally, Req. 1g is necessary to allow uncertain parameters to be dealt with. The interface was developed to distinguish the known parameters from the unknown ones.

When selecting the IDS format, two main options were considered: Excel and Python. Excel offers numerous capabilities and user-friendliness, providing a clear and intuitive way to visualise and edit input parameters. On the other hand, Python would naturally integrate with the rest of the framework since the SART software tools have interfaces in this language. Besides, the data would eventually need to be transferred to a Python environment. However, usability and accessibility for future users of the framework was also considered. A purely Python-based script, while efficient, would likely be harder to interpret for future users, unfamiliar with this framework. In contrast, an Excel-based solution provides a more straightforward and readable interface (Req. 3a), making it easier to use. For these reasons, Excel was selected as the primary platform for the IDS, with Python handling the subsequent processing and integration steps.

The final decision was to develop an IDS with Excel and assess whether all requirements were satisfied. Before that, the relevant parameters to be included were identified. This selection process involved analysing the CAC, STSM, and TOSCA tools and extracting parameters corresponding to rocket data typically available in public sources. These primarily consisted of geometric characteristics (e.g., dimensions and masses) and engine performance data (e.g., thrust and I_{sp}). Their complete list is available in Appendix B.

Besides the requirements already mentioned, this format choice complied with

Req. 1e and Req. 1f. Specifically, the IDS was designed to incorporate all necessary parameters while remaining adaptable to different types of expendable launch vehicles (see Section Excel Input Data Sheet for further details). Furthermore, extra parameters can be added, which complies with Req. 1i. Finally, Excel files can be easily read by Python code, fulfilling Req. 3b.

Excel Input Data Sheet

After selecting the parameters needed as input for the framework, the next step is to organise the Excel file. It is preferable to split the information into several Excel tabs, given the amount of data required. This way, the IDS is cleaner and more organised. Hence, the following tabs were created; only the main Excel tab that handles the uncertain parameters has a picture in this section (Figure 3.3). The remaining ones are shown in Section B.2:

- *Rocket*: This tab is used only for the LV identification. It contains the rocket's name and ID (used for CLAVA). It also includes a cell for the IDS version. This is important because the whole software that reads the Excel file might need changes if there are modifications to the IDS. Thus, this cell is read as a checkpoint to guarantee that the software that will read the IDS is up to date with its version.
- *General*: This tab contains some general rocket parameters, not specific to a particular stage. For instance, it encompasses the number of stages, the number of boosters, and some information about the fairing and booster nose (dimensions and mass). Additionally, it contains a few constants, such as gravitational acceleration, atmospheric pressure and the Excel tolerance. This last value is not used for any SART software tool but is a comparison factor. The Excel data sheet knows which other parameters are redundant to fill in, and it calculates them automatically. However, if the user finds a source for those parameters, they can be written in the file and compared to the Excel-computed value. The Excel value has infinite decimal digits, which will never match a user's input. To address this, the tolerance allows the user to define the acceptable margin of error. The IDS verifies whether the difference between the values falls below the user-defined tolerance. This mechanism provides flexibility by letting the user adjust the strictness of the comparison depending on the desired precision.
- *Stage Template*: This tab contains the information about a stage in particular (Figure 3.3). The template is the same for every stage. One tab is generated for each stage, according to the information provided in the "General" tab. Likewise, if the rocket that is being remodelled has boosters, an extra tab is created for them. Each of these tabs has the same structure and parameters to be filled in. Some examples are the stage's length and maximum diameter, the number of engines and the propellant type. Moreover, the user defines if there is a thrust profile, and writes the engine performance parameters (e.g., I_{sp}) and the mass information (e.g., total propellant mass).

General							Sources (PLEASE ADD THE CORRECT REFERENCE NUMBER!)	Comments/Notes
Parameter	Value	Lower Boundary	Upper Boundary	Calculated Value	Calculated Lower Boundary	Calculated Upper Boundary		
Length (m)								
Diameter (m)								
Structural Index								
Surface Quality Factor								
Propulsion							Sources (PLEASE ADD THE CORRECT REFERENCE NUMBER!)	Comments/Notes
Parameter	Value	Lower Boundary	Upper Boundary	Calculated Value	Calculated Lower Boundary	Calculated Upper Boundary		
Number of Engines								
Propellant Type (TRUE=Liquid, FALSE=Solid)	TRUE							
Thrust Profile (TRUE=Yes/FALSE=No)	FALSE							
Engine Name								
Single Engine Mass (kg)								
Single Engine Diameter (m)								
Mix Ratio								
Max Mass Flow Rate per Engine (kg/s)								
Vacuum								
Vacuum Max Isp (s)								
Vacuum Max Exhaust Velocity (m/s)								
Vacuum Total Max Thrust (kN)								
Sea Level								
Sea Level Max Isp (s)								
Sea Level Max Exhaust Velocity (m/s)								
Sea Level Total Max Thrust (kN)								
Structure							Sources (PLEASE ADD THE CORRECT REFERENCE NUMBER!)	Comments/Notes
Parameter	Value	Lower Boundary	Upper Boundary	Calculated Value	Calculated Lower Boundary	Calculated Upper Boundary		
Total Mass (kg)								
Dry Mass (w/o payload) (kg)								
Ascent Propellant (kg)								
Descent Propellant (kg)								
Reserve Propellant (kg)				0				
Residual Propellant (kg)								
Total Propellant Mass (kg)								
Propulsion Mass (kg)								
Structure Mass (kg)								
Reserve Propellant Percentage (%)								
Residual Propellant Percentage (%)								

Figure 3.3: Excel stage template tab: contains the main technical data about a rocket stage needed for the remodelling process.

- **Stage Thrust Profile Template:** This tab is used for LVs with a thrust profile for a certain stage. After selecting that option in the "Stage Template" tab, the user now defines the thrust/mass flow profile in this tab. Afterwards, it will be used by the trajectory optimisation tool to model the thrust/mass flow of that stage.
- **Validation Cases:** This is the last tab. It contains information about the orbits and missions flown by the rocket. This information will be used as validation for the remodelling process. Each column represents a different mission. The user has to write the orbit type (e.g., LEO), the launch site's name and coordinates, the target orbit details (apoapsis, periapsis and inclination) and the minimum and maximum payloads. These last two entries represent, respectively, the minimum value that the rocket has to be able to carry to the specified target orbit, and the maximum payload value. This is important for the algorithm to know the range of payloads in which to search for valid configurations. The maximum threshold is important to avoid unrealistic LV configurations with an overestimated performance.

This structure was defined to comply with this thesis's requirements and goal: to define the range of unknown parameters that yields a valid rocket model. Therefore, it was paramount to make sure the IDS could handle these uncertain parameters.

Looking at the "Stage Template" Excel tab, six columns are provided for entering parameter values. In the first three – "Value", "Lower Boundary" and "Upper Boundary" – the user is allowed to write. The first one is used when the parameter's exact value is known. If so, the other two should be left empty. If there are ambiguous values or only a range is known, the user should write the lower and upper limits in the corresponding columns, leaving the first one blank.

The three columns on the right – "Calculated Value", "Calculated Lower Boundary", and "Calculated Upper Boundary" – are outputs. The first one shows the parameter's

value if it can be estimated based on other inputs. The other two show the computed parameter range, considering other values and ranges from different input parameters. This way, as the user fills in the IDS, some parameters might be immediately calculated, saving some research time. Whenever the user is confident in the values that originated the “Calculated Value”, they can write that result in the “Value” column, which might trigger other embedded equations, thus helping with the data gathering process.

Additionally, as previously mentioned, this IDS is highly flexible and automated. It adapts to any kind of expendable rocket and can calculate some of the parameters based on others. However, this feature is “limited” to the formulas chosen by the developer. Nonetheless, because of the chosen format, new formulas can be easily added if necessary.

This calculation process starts with the IDS checking whether all the cells needed for an equation have a value inside. If this condition is met, it computes the desired parameter’s value based on them. Moreover, if more than one equation can be used, the algorithm first checks whether every cell required for all equations has a value. If so, it compares the results of all equations through the absolute error and the Excel tolerance. If they are within the defined tolerance value, one of the formulas is used to get the desired parameter. Otherwise, there is an error message. If not all values for all formulas are available, it applies the equation for which the parameters are obtainable, if that is the case. This versatility makes it capable of dealing with some uncertainty scenarios. It can compute an unknown value through different formulas according to what is available. The only downside is that the author has to write the formulas to be tested *a priori*. Still, a thorough research was conducted to find the maximum possible relationships between the input parameters. The major ones are listed in Section 3.1.1.

In the end, once the calculation process is over, the IDS can be updated with the new valid ranges obtained, complying with Req. 4(a)ii.

3.2.2. CLAVA Conversion

The next step is to transfer all the information to a CLAVA model. The CLAVA model stores the Excel data and is used to bridge the IDS and the SART Toolbox software.

As described in Section 3.1.4, CLAVA has a lot of flexibility. Its structure allows for any type of expendable rocket to be analysed. This is important to facilitate the data transfer from Excel. This part can be done in a methodical, highly adaptable way. All this process is done regardless of the number of stages or engines, since the same methodology is applied to define each.

The missions are also created cyclically since they always have the same structure and required parameters. So, there is no limit on the number or type of missions that can be used. The overarching data conversion process can be seen in Figure 3.4. It starts with loading the Excel file into the Python environment. Thereafter, all the necessary parameters are read to create both the rocket “object” and the missions it flies.

Figure 3.5 demonstrates the flow of commands the code executes. The first thing to do is create the engines. Each engine is assigned to one or more stages. Then, the fairing is created, with its mass and dimensions/geometry. The next step is to assess if there are boosters and build their engines and stages in the file. Afterwards, the bodies are established. There are usually the "main" body and the booster bodies. Finally, all bodies are aggregated under a larger object, creating the rocket. This is the method used to build the rocket itself. However, CLAVA can also save the LV's missions. This is the next step.

The values used in the creation of the rocket "object" are gathered following a certain logic (see Figure 3.6 for more details). In summary, the code tries to read a single value for a certain parameter in the IDS. If it cannot find one in the two columns for specific values, it looks for a range for the parameter in the columns for boundaries. If this is the case, the code tags this parameter as "unknown" and thus appends it to the "uncertain parameters" list, which includes a connection to the UQ model. This is how the whole framework knows to which parameters the sampling method needs to be applied. Since the CLAVA file needs a single value, not a range, for all parameters, the mid-value of the range is assigned to the CLAVA object. Later, when the sampling process generates a new CLAVA file for each sample, this value will be replaced by the sampled one.

The illustrative flowchart (Figure 3.5) shows that the rocket's missions are set up separately. The data required for this are read from the "Validation Cases" tab of the IDS. Firstly, the algorithm gets the launch site coordinates and creates a launch site in the CLAVA model that can be reused for other missions. Subsequently, it takes the orbit parameters to generate an orbit. This orbit is then combined with a launch site to form a mission. Lastly, trajectories are formed by joining the rocket with the missions. Once all this is completed, the CLAVA file is created. It consists of a *.xml* file with all the data structured as explained in Section 3.1.4. It is very easily accessed for further processing. Moreover, the UQ file is created to complement the CLAVA one. It contains the uncertain parameters, their lower and upper boundaries, the number of samples and the sampling method. It is also a *.xml* file.

3.2.3. Test Model

The TM was implemented as a faster tool to estimate LV performance with a lower computational effort. While its theoretical foundations were outlined in Section 3.1.3, this section details its practical implementation.

Figure 3.7 shows the workflow of the TM. The first step is to retrieve the necessary

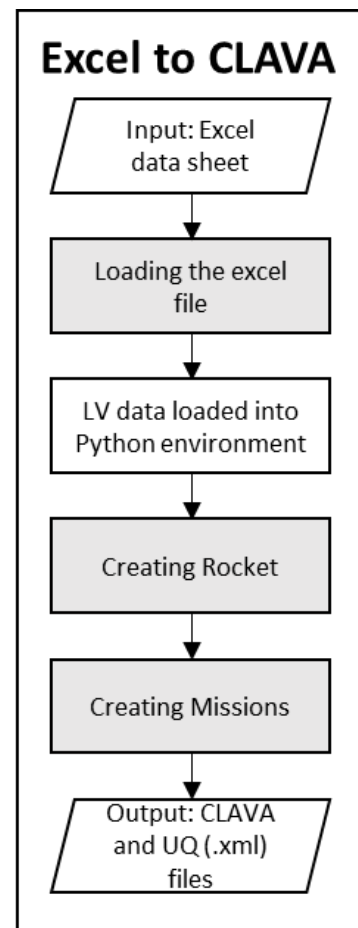


Figure 3.4: Structure of the conversion from Excel to CLAVA.

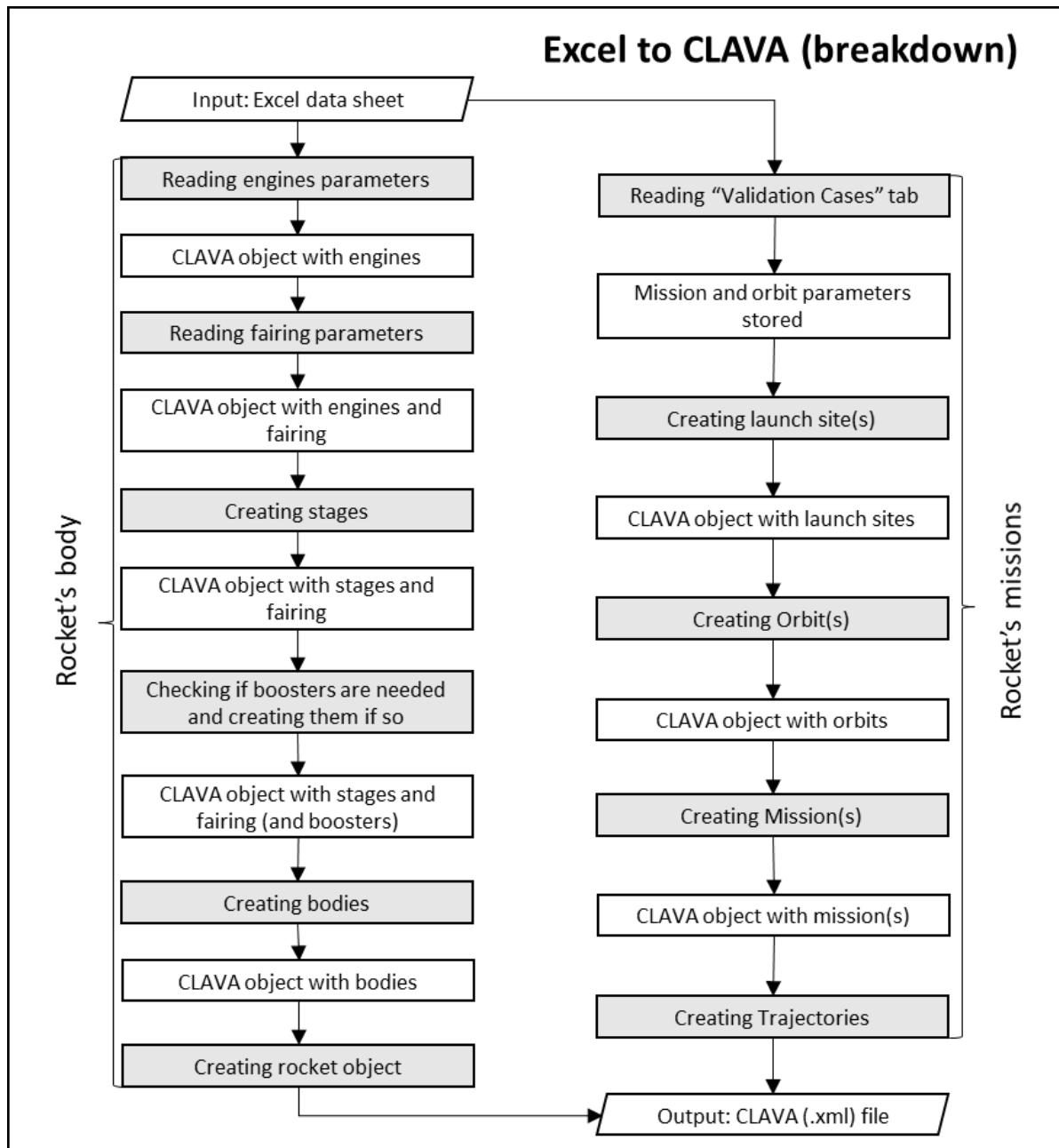


Figure 3.5: Excel to CLAVA conversion detailed in the two branches combined to generate the file: the rocket object and the trajectories.

information from the input CLAVA file. The parameters read are the launch coordinates, the target orbit's apoapsis, periapsis, and inclination, and the minimum target payload. These will be useful in calculating some elements of the Δv budget (Equation 3.9).

Afterwards, the user defines the losses. A suggested default value will be predefined, but it is always possible to alter it. They will also be different depending on the mission type (e.g. LEO, SSO, etc.). An elaborate explanation for this is available in the Section Losses.

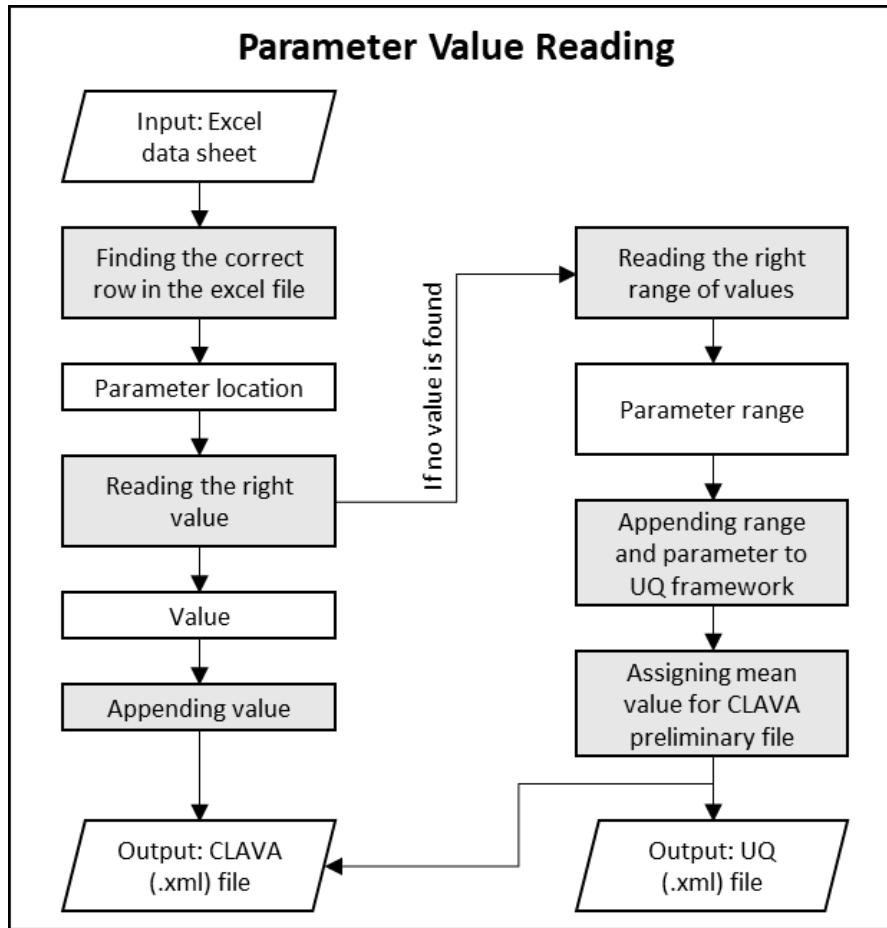


Figure 3.6: Structure of the procedure to read values from the IDS and assign them to the CLAVA and UQ files.

Then, the orbital Δv (Δv_{orbit}), the Earth's rotation contribution Δv (Δv_{Earth}) and the potential energy Δv ($\Delta v_{potential}$) are calculated. At this point, the components independent of the final payload have already been defined. The remaining ones depend on the payload, thus they cannot be estimated yet.

Finally, a root-finding algorithm is implemented to compute the maximum possible payload that the rocket configuration can carry to the target orbit. The result is written in an updated UQ file, which is the output of this workflow. Now that the structure of the TM is defined, each component will be explained in more detail.

Orbital Velocity

This section describes the calculation of the most significant component, in absolute value, of the Δv budget that does not depend on the LV: the orbital velocity. This corresponds to the velocity that the LV needs to be in the target orbit. It was assumed that the launch would be directed to the target orbit's perigee, since it is the easiest point to achieve in terms of required velocity increment.

The orbital velocity can be calculated through the *vis-viva* equation [77]: Equation 3.10.

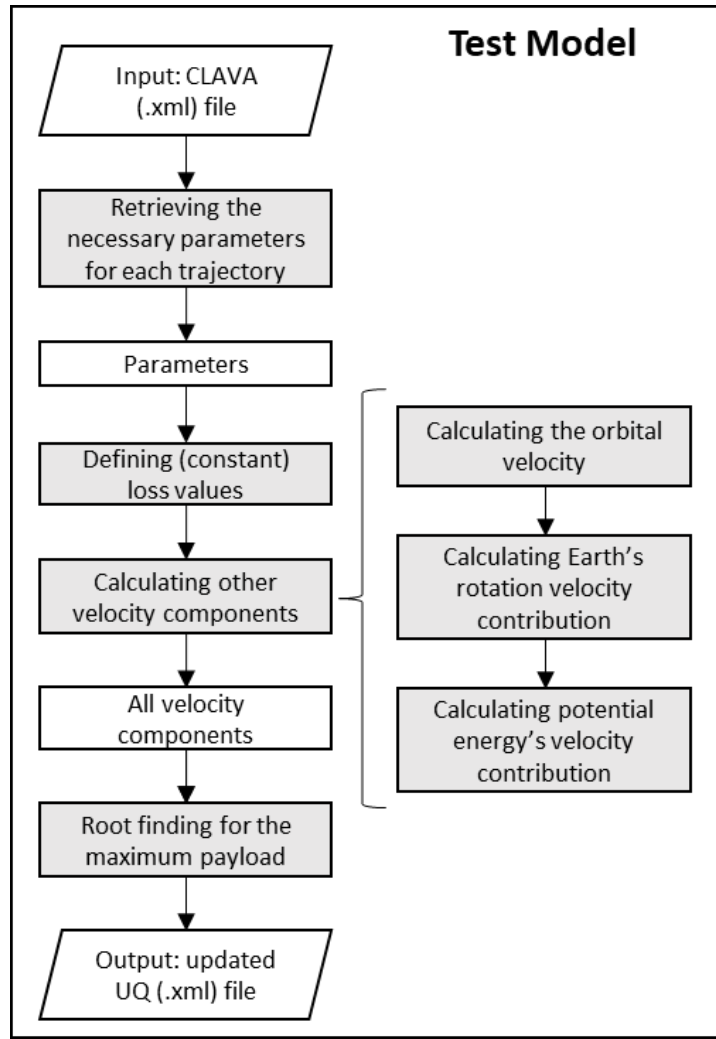


Figure 3.7: Flowchart that demonstrates how the Test Model's algorithm is implemented.

$$\Delta v_{orbit} = \sqrt{\mu \left(\frac{2}{r} - \frac{1}{a} \right)} \quad (3.10)$$

where r is the current LV's altitude, including the Earth's radius, and a is the orbit's semi-major axis. In this particular case, r is the perigee's altitude.

Rocket's Δv

This section explains how to compute the Δv that the LV can produce. As mentioned in Section 3.1.3, the TM relies on simple calculations involving the input parameters. One of the most important equations in rocket science is Tsiolkovsky's equation [46]:

$$\Delta v_{rocket} = I_{sp} \cdot g_0 \cdot \ln \left(\frac{M_0}{M_f} \right) \quad (3.11)$$

where M_0 is the mass at the beginning of the impulsive manoeuvre and M_f is the mass at its end.

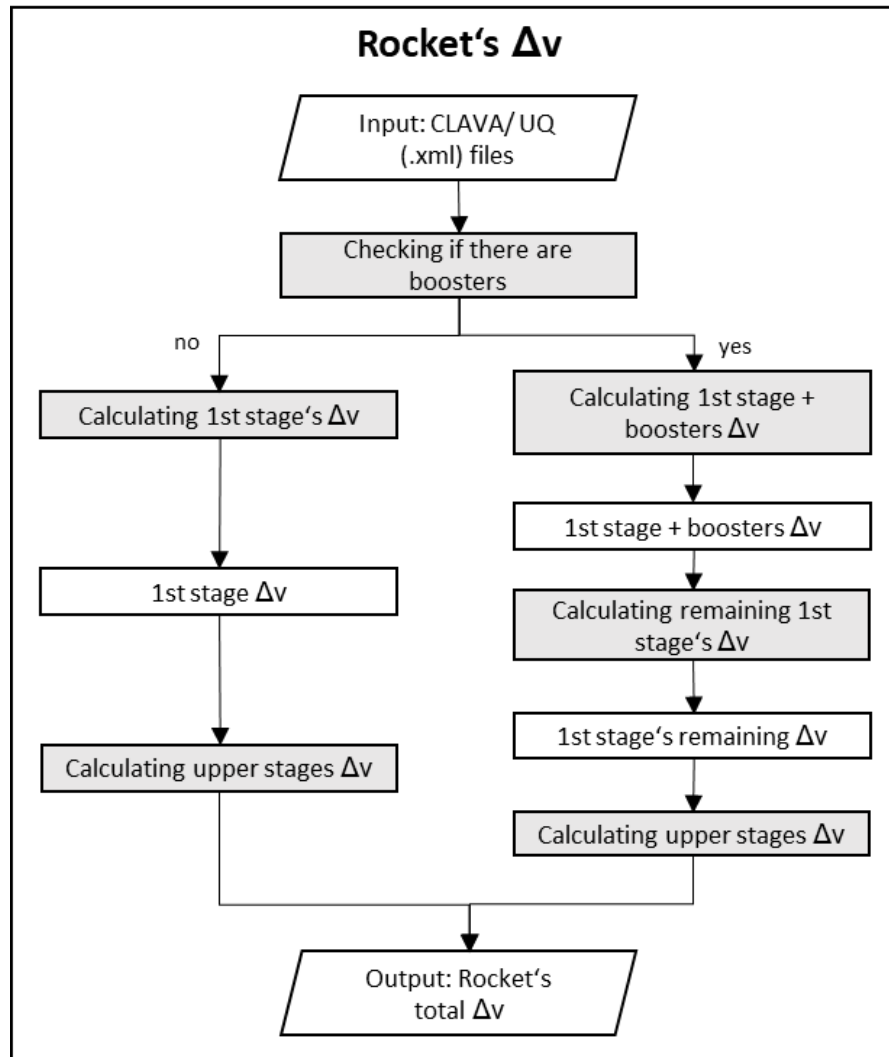


Figure 3.8: Calculation algorithm to obtain the Δv that the rocket can produce.

This equation computes the Δv that a rocket can produce using some of its parameters. This is the idea behind the TM. However, it needs to be broken down into the different stages, which slightly increases its complexity.

Figure 3.8 shows the flowchart of actions to calculate the total velocity change that the LV can achieve. Equation 3.11 is applied to calculate the contribution of each stage. The difference in the two paths of the rocket's Δv flowchart occurs when the LV has boosters. If this is the case, then an average I_{sp} is computed for the first burning phase. Initially, the average I_{sp} is calculated for the boosters and the first stage separately. It consists of a normal average between the sea-level and vacuum I_{sp} values. Subsequently, a weighted average is defined, based on the mass flow rate, to achieve the overall average I_{sp} of the impulsive manoeuvre in which both stages fire simultaneously. This way, one accounts for the larger influence that the boosters have in this initial burning phase, since they are usually two or more, against the single core stage. Equation 3.12 contains the formula used.

$$I_{sp_{avg}} = \frac{I_{sp_{avg_{booster}}} \cdot m_{booster} + I_{sp_{avg_{stage_1}}} \cdot m_{stage_1}}{m_{booster} + m_{stage_1}} \quad (3.12)$$

where m corresponds to the stage's total mass flow rate (so the sum of all engines' mass flow rate), whether it is the boosters or the core one.

After the initial phase, the remaining propellant available for the first stage is used to calculate the next Δv . At this point, the vacuum I_{sp} is assumed to be the current one, since the LV is already at a very high altitude. Once this stage runs out of fuel, before moving to the following one, the fairing is assumed to be jettisoned. This means that, besides the dry mass and the residual and reserve propellant mass, one has to subtract the fairing mass from the final mass of the previous stage (this part is done both for LVs with and without boosters).

Regarding the calculation of the different masses needed, Figure 3.9 shows the commanding flowchart for this part of the code. Usually, the final mass after an impulsive manoeuvre is equal to the initial one minus the propellant mass used. Afterwards, the stage's dry mass is subtracted to compute the initial mass for the following impulsive manoeuvre. This loop continues for all stages.

Losses

In real launch scenarios, the ideal Δv required to reach the target orbit, as predicted by Tsiolkovsky's rocket equation (Equation 3.11), is not enough. One must account for additional terms, hereinafter referred to as losses, which increase the necessary Δv to get to orbit. These losses highly affect the actual performance requirements of an LV.

In the TM, three main types of losses are considered: drag (aerodynamic) loss, gravitational loss, and thrust loss. They are the same ones that TOSCA accounts for in its optimisations. To integrate them into the TM, they had to be simplified, as they are difficult to compute without an actual trajectory. Still, these components are essential to ensure the results remain within realistic bounds.

Initially, research was conducted to find correlations between the three types of losses and specific rocket parameters. The idea was to analyse already remodelled trajectories and retrieve the losses from TOSCA's optimisations. Then, the author tried to find a correlation between several of the rockets' parameters and the different losses. A meaningful correlation

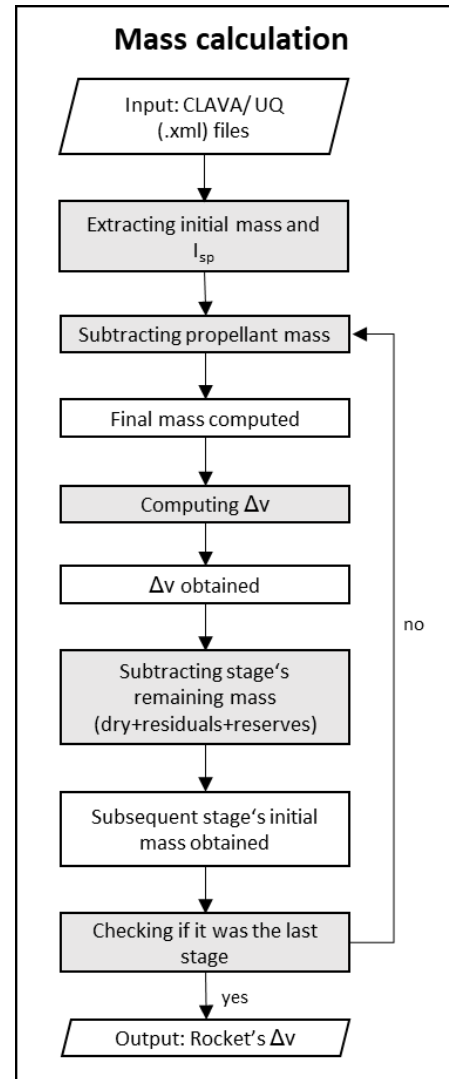


Figure 3.9: Mass calculation procedure.

was identified for drag losses, while the other two showed less conclusive trends. The losses for each rocket used in this research are in Appendix C. In the following sections, each type of loss will be explained in more detail.

Gravitational Loss

Among the different types of losses, the gravitational loss is the most relevant one. It represents the portion of a rocket's available energy that is expended simply to counteract gravity's pull. It occurs when the LV accelerates against gravity, such as during a launch from Earth's surface.

When a rocket fires its engines during the ascent's initial phase, a significant fraction of its thrust must be directed upward to oppose gravity. As a result, not all of the rocket's propulsive effort contributes to increasing its velocity. Some energy is continually "lost" to maintain its position against gravity.

Mathematically, gravity loss can be defined as:

$$\Delta V_g = \int_0^{t_b} \sin \gamma g_0 dt \quad (3.13)$$

where t_b is the burn time, γ is the flight path angle and g_0 is the gravitational acceleration. Based on Equation 3.13, the more vertically a rocket is firing its engines, the larger the gravitational loss. However, no trajectory is available in this test model. This means it is impossible to know the flight path angle over time. Thus, another way to estimate gravity losses had to be found.

As previously mentioned, the author tried to find a correlation between the gravitational losses and some of the rockets' parameters. For instance, T/W or burn time combined in different ways were tested. The higher the former and the lower the latter, the lower the gravitational losses, since the rocket accelerates faster and in a shorter time. However, no good correlations were found (see Section C.2 of Appendix C).

Therefore, always keeping in mind that this model's main goal is to test the statistical models, a fixed value was defined. After carefully analysing the collected data, the range for the gravitational losses was quite wide. Therefore, it was decided to adjust this value according to the type of mission (e.g. LEO, SSO, etc.). For instance, LEO missions generally have lower gravitational losses than an SSO. However, the ranges were still large. Hence, for each kind of mission, the minimum value of the gravitational loss among all studied trajectories was taken. This way, there was no risk of discarding realistic trajectories due to a "false lower payload configuration". In the end, the gravitational losses for each kind of mission will have a default value of:

- LEO: 1300 m/s
- SSO: 1300 m/s
- LSS: 2000 m/s
- MEO: 1350 m/s
- GTO: 1450 m/s

However, it is important to note that the user can change these values as needed. They are only to be used as a reference.

Drag Loss

Another type of loss is the drag loss. It is a consequence of the aerodynamic drag that a rocket experiences as it accelerates through the Earth's atmosphere. This loss is particularly impactful in the early phases of flight, because the atmospheric density is higher at lower altitudes [78], thus creating further aerodynamic resistance.

For the drag losses, some parameters were tested, such as the cross-sectional area, the length, the drag coefficient, the flight time or combinations of all these, among others. However, the (combination of) parameter(s) that yielded the best correlation was the ballistic coefficient. It is defined as:

$$C_b = \frac{M}{C_d A} \quad (3.14)$$

where C_b is the ballistic coefficient, M is the rocket's initial mass, C_d is the initial drag coefficient, and A is the cross-sectional area of the rocket. The maximum drag coefficient was also tested for the ballistic coefficient's estimation. However, it did not show such a good correlation. The final results are shown in Figure 3.10

The coefficient of determination (R^2) was around 0.77. Although this is not a great result, it shows a clear correlation between the ballistic coefficient and the drag losses. The higher the former, the lower the latter. The larger deviations could be explained by more uncommon orbits, such as to space stations (LSS), or by possible mistakes in the LV models used. The final expression used to compute the drag loss was

$$\text{Drag Loss} = -0.0013C_b + 202.87 \text{ m/s}$$

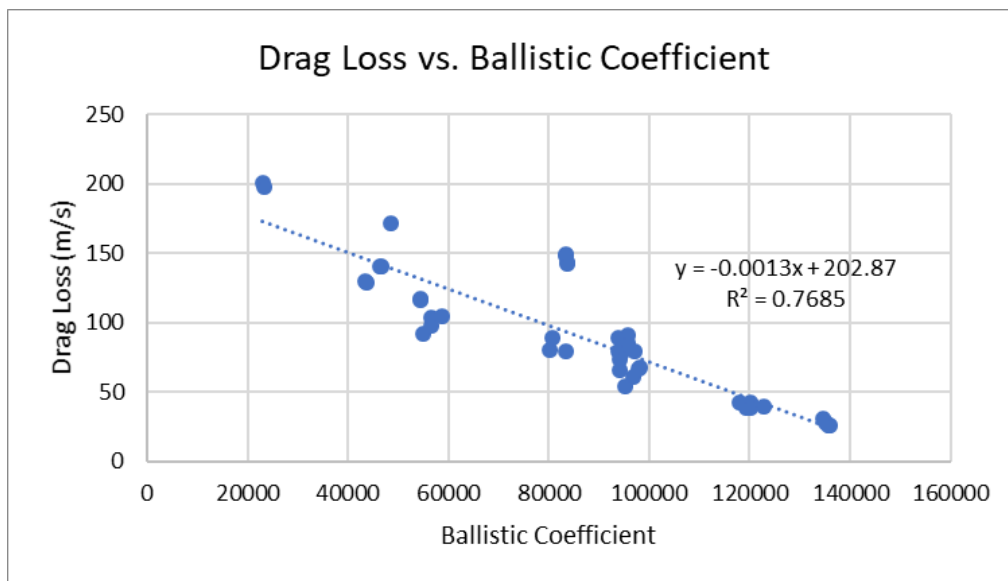


Figure 3.10: Drag Loss vs Ballistic Coefficient correlation.

The initial drag coefficient was computed by combining the nose drag coefficient with a constant factor. Since the velocity is approximately null in the first instant of the LV's launch, another formula had to be selected (see Section C.3 in Appendix C) instead of the traditionally used one [79].

Thrust Loss

The last category of losses to consider during a rocket's launch is thrust loss. It refers to the reduction in the effective propulsive force delivered by a rocket engine. It is primarily caused by misalignments between the engine's thrust vector and the LV's intended direction of motion. This is how these losses are defined in TOSCA, since other factors, such as nozzle and combustion inefficiencies, can also impact real case scenarios.

Similar to the two previous types of losses, the author tried to find a correlation between some of the input parameters and the thrust loss. T/W , burn time, or I_{sp} , among others, were tested in several ways. For instance, a longer burn time leads to a higher thrust loss because the LV spends more time with its thrust vector potentially misaligned relative to its velocity or desired trajectory. However, no satisfactory results were achieved (see Section C.4 for more details). Thus, a fixed value was used, since it doesn't vary as much as the gravitational loss, and it is one order of magnitude smaller than it. Additionally, it can always be changed by the user. The default value selected was slightly below the average value for all models assessed in this study: 100 m/s, for the same reason as the gravitational loss.

At this point, all the losses have been defined. The following sections address and explain the remaining components for the Δv budget.

Earth's Rotation

Earth's rotation is another important factor that influences the overall Δv budget. It provides an extra boost for LVs launched in the direction of our planet's revolution. As a result, rockets launched eastward require a smaller self-generated velocity increment to achieve orbital speed.

The magnitude of this contribution depends on the target orbit's inclination and the launch site's latitude. Equatorial launches benefit the most because of the higher tangential speed at lower latitudes. The formula to precisely compute this value is (a more detailed explanation is found in [67]):

$$\Delta V_{rot} \cong V_E^e \cos(i) \quad (3.15)$$

where i is the target orbit's inclination and V_E^e corresponds to Earth's rotation speed at the equator, which is given by:

$$V_E^e = \frac{2\pi \cdot R_E}{23.934472 \text{ h} \cdot 3600 \text{ s/h}} \quad (3.16)$$

where R_E is Earth's radius (6378 km). This yields $V_E^e = 465.091 \text{ m/s}$.

Potential Energy

Finally, the last part of the Δv budget is the potential energy's contribution. This effect is particularly important when dealing with higher orbits. Without it, launching to a higher orbit requires a smaller velocity increment compared to lower orbits (e.g. SSO vs LEO). This is misleading if one only considers the orbital velocity, since higher orbits have lower orbital velocities. The potential energy contribution solves this issue. It is calculated by comparing the energy state at the launch altitude with that at the perigee of the target orbit. According to [67], this can be calculated by:

$$\Delta V_{\text{poten}} = \sqrt{\frac{2\mu}{R_E + h_{\text{launch}}} - \frac{\mu}{R_E + h}} - \sqrt{\frac{\mu}{R_E + h}} \quad (3.17)$$

where h_{launch} is the launch site's altitude and h is the perigee's altitude.

Root finding

The root finding algorithm is a crucial part of the TM. All the aforementioned Δv budget components are combined to obtain the maximum payload through it.

There are multiple root-finding algorithms. They have different characteristics, such as reliability and speed. In this work, the adopted methodology was defined by R. P. Brent in [80]. Considering that this project's main goal was not this small optimisation and given all the priority tasks, little time was spent investigating all the root-finding methods. However, some research was done. According to some sources [81], Brent's Method combines "the sureness of bisection with the speed of a higher-order method". This means that it is fast and reliable [82]. Furthermore, the problem in question uses a few simple equations. Thus, the computational power needed is considerably low. Therefore, no further improvement was considered necessary after testing this method and its speed and results.

The idea behind this algorithm is to optimise the payload, given the velocity budget for that LV and mission. This is done based on Equation 3.9 for the Δv margin. This relationship and all its components are based on the description provided by Prof. Arif Karabeyoglu [67]. It encompasses all the elements described throughout this section. Some depend on the payload value, such as the rocket's Δv and the drag loss. This is why they are computed inside the root finding loop, as seen in Figure 3.11. At every iteration of this algorithm, new values are calculated for both. They are then used to update the margin function, based on the newly generated payload.

Figure 3.11 shows the structure inside this method. The idea is to start with a range for possible payloads, for which the margin function results will be symmetric. So, if a is the minimum and b is the maximum: $f(a) \cdot f(b) < 0$. The values used in this model were 0 kg and 1000000 kg, respectively, for a and b . This way, it was sure that the results would comply with the symmetry requirement, as every rocket can carry a null payload to a target orbit it has flown to (hypothetical scenario just used as a lower boundary). Besides, no rocket can take 1000000 kg of payload to any orbit, as of the date of this document's writing. Afterwards, a payload value between this value is generated, and thus, the rocket's Δv and the drag loss. Then the margin function is computed, and the updated payload value will be higher or lower based on whether

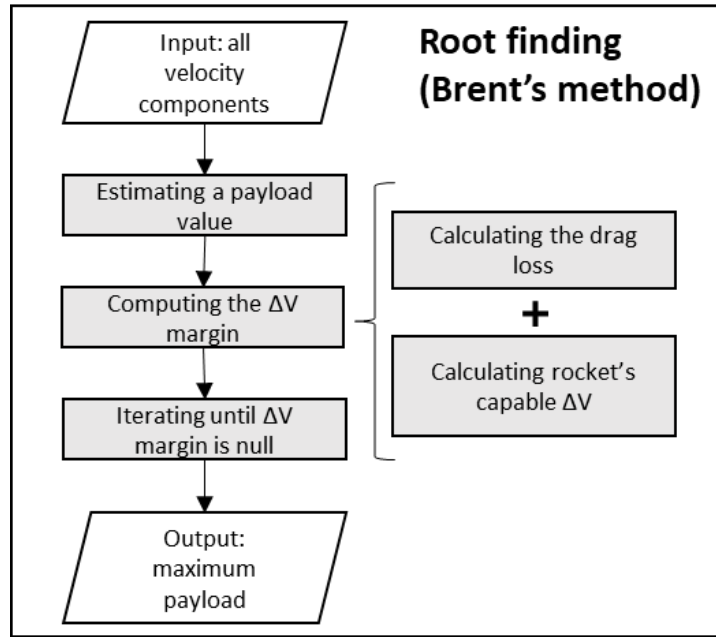


Figure 3.11: Brent's method structure applied in this thesis. It calculates the maximum payload based on all velocity budget components.

its result is positive or negative. The iterations run until a defined tolerance value is achieved (10^{-6} was used, so approximately 0). That final payload is the maximum that the tested rocket's configuration can take to the target orbit.

Validation

This section describes the steps used to validate the TM. The approach involved comparing the maximum payload given by the model to the result from TOSCA's optimisation. After verifying some rockets, the author noticed that there was an error margin for the calculated Δv , and thus the payload. The TM's velocity gain results were within 5% of the optimised ones. Although this might seem rather small, the order of magnitude for the Δv is around 10^4 m/s. So, the error is still significant enough in absolute value to affect the final payload.

To circumvent this issue and test whether the logic behind the payload calculation was correctly implemented, another version was created. The difference was that the total required Δv was given by the user, instead of only defining the losses. This way, one could use the exact velocity gain needed by the manually optimised trajectory and check if the TM's payload results were still accurate. After testing a few rockets, the results were still different. Three main reasons for this were identified:

- I_{sp} averaging: assuming the exact average between vacuum and sea-level I_{sp} is not correct. After analysing the results from the optimised trajectories and extracting the average I_{sp} , it was different from the average one from the TM, which alters the output of Equation 3.11.
- Fairing jettisoning: assuming that the fairing is jettisoned immediately after the first stage is dropped is incorrect. After examining the optimised trajectories' results, one can see that this usually occurs at a different point in time, usually

slightly later. Thus, Equation 3.11 will have a different result, because the masses are affected.

- **Simplifications:** TOSCA's algorithm contains complex calculations that are dynamically changed at every optimisation point. That makes it very robust and reliable, but the complex mechanics of an LV ascent are hard to simplify accurately. The TM completely disregards the LV's ascent trajectory, a vital part of these calculations. For instance, the gravitational and the drag losses depend on it [67]. The formula in Equation 3.9 was the most comprehensive one found in the literature. Still, it cannot replace more sophisticated and accurate trajectory optimisation tools, like TOSCA, due to all the simplifications it has.

Again, it is relevant to emphasise that the TM's main goal is to investigate the statistical sampling methods. Therefore, despite probably being possible to mitigate these error causes, it would require additional time. However, it was decided not to invest more time in this task and rather focus on the main goal. The assumptions and simplifications done in this work were always kept in mind to get the best possible result, while being fast and computationally cheap. Ideally, this model could be used as a pre-filtering step before the CM. This way, one could immediately reduce the number of samples needed to be run with the more expensive model. However, given the achieved error margin, this is not feasible.

In order to validate the TM despite these problems, a new method was implemented. The author adapted an existing LV by assuming the same I_{sp} at sea-level and vacuum conditions, and by assuming an almost null fairing mass. This way, the interferences caused by two of these problems were eliminated. Then, an optimised trajectory was computed for the chosen rocket. The same strategy was implemented here. One retrieved the total Δv required and applied that to the TM. The payload results are shown in Table 3.1.

The obtained error was below 0.5%, which validates the logic behind the maximum payload calculation. To make this test model even more robust, the aforementioned problems should be addressed.

Table 3.1: Validation results comparing TOSCA and Test Model outputs.

Results	TOSCA [kg]	Test Model [kg]	Error [%]
LM2C	2816	2809	0.25
LM4C	4300	4318	0.42

Having validated the TM, assessing the performance of the statistical methods became possible. Those results are shown in Chapter 4.

3.2.4. Complex Model

The CM represents the core of this rocket remodelling framework, enabling the most realistic representation of LV trajectories and performance. Unlike the TM, the CM directly integrates with DLR's SART Toolbox software to create mass, aerodynamic, and trajectory models. At its heart lies TOSCA, which optimises feasible trajectories

consistent with real-world launch conditions, wrapped by a CMA-ES optimiser. Its implementation considers having the least interference from the user as possible, as stated in Req. 3c.

In particular, the CM employs the CMA-ES as a genetic optimiser around TOSCA. The whole process is shown in Figure 3.12. It starts by defining:

- Setup parameters for TOSCA: some of them are specific to a certain stage, such as thrust control points or the angle of attack profile. Others are general, such as initial pitch rate, initial launch azimuth, and initial payload. A range is defined for each. They bound the space that the CMA-ES optimiser will explore.
- CMA specific parameters: the population size factor, the stagnation tolerance, the function value tolerance, the history tolerance, and the number of parallel processes. More details about each can be found in the Python documentation for this library [83].
- Constraints: maximum dynamic pressure, maximum acceleration, and target orbit's perigee and inclination tolerances.

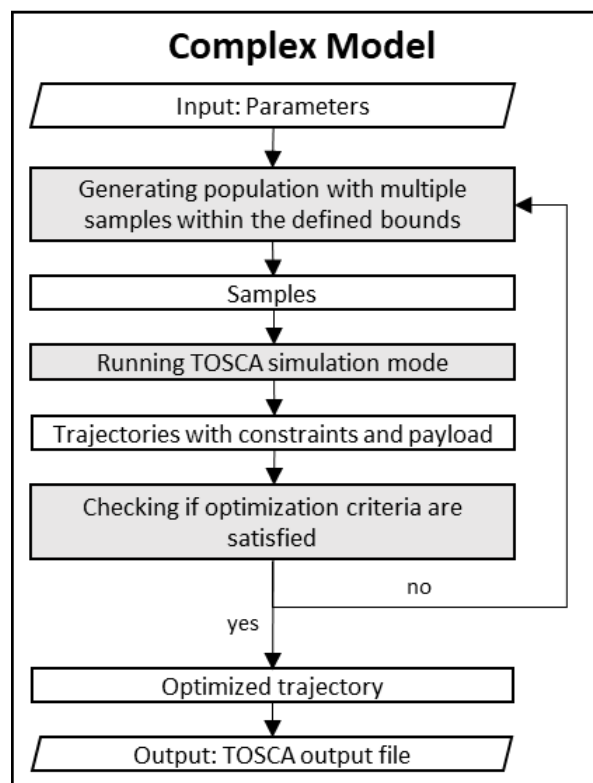


Figure 3.12: Workflow of the CMA-ES optimiser wrapped around TOSCA as part of the Complex Model.

Once everything is defined, the CMA optimiser generates a population of samples within the ranges for the TOSCA setup parameters. Then, it runs TOSCA in simulation mode for each sample and chooses the one with the highest payload from the ones that respect the constraints. These constraints can be read from the TOSCA output. Based on that best sample, it generates others and explores the sampling space until

finding the absolute minimum for the cost function. So, TOSCA does not perform the optimisation; the CMA-ES optimiser does.

This combined CMA-ES–TOSCA workflow allows the CM to converge towards launch trajectories that maximise payload capacity while respecting all defined constraints. By doing so, the CM serves as the validation backbone of the framework. More details regarding the CM's results are available in Chapter 4.

The integration of the CM in the overall framework is similar to the TM one. All the steps of the remodelling procedure from the IDS until the creation of the CLAVA and UQ files are identical. However, the SART Toolbox files are generated before the optimisation begins, in order to be able to run TOSCA. Then, the statistical methods generate samples to be evaluated. Afterwards, the CMA optimiser starts the loop shown in Figure 3.12 for each of these samples generated. In the end, the valid samples are saved for further analysis. This integration is shown in Figure 3.13.

Finally, the CM implementation was made to comply with Req. 2b. A balance between the number of samples and the accuracy was achieved. This means that the results shown in Chapter 4 used as many samples as possible to obtain better results while complying with the time requirement.

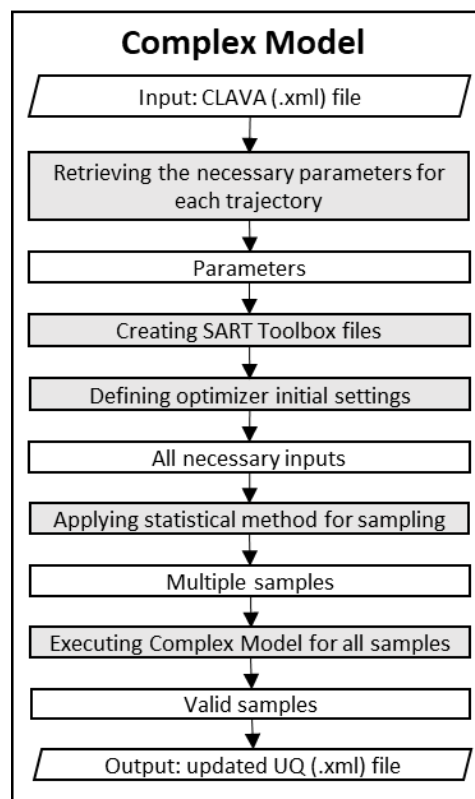


Figure 3.13: Flowchart of the integration process of the Complex Model in the overall framework.

Final Sample Selection

Besides the CM integration in the whole framework, explaining how the final samples are selected is important. After the sampling procedure, each sample is tested. Then,

there will be a set of valid samples, regardless of the chosen statistical method. However, a single configuration is desired for DLR and the overall purpose of this LV remodelling framework. Hence, it is important to have a good strategy for this step of the process.

The list of valid samples is always available, even for the TM, as needed per Req. 4a. This means the user has the possibility of filtering, ordering or rearranging the results in any possible way. For instance, in case the goal is to use the sample with the lowest achieved payload, the highest propellant mass, or the highest I_{sp} , this is all possible, as long as these last two parameters are part of the unknown ones (as needed per Req. 5b). These options are part of a more targeted sample selection approach. On the other hand, a broader sample selection method was used during the testing phase and as a general procedure.

The method implemented here is based on the mean distance between each sample and all the others. The mean pairwise Euclidean distances between samples in the normalised parameter space are computed. Then, a percentage of the samples with the lowest mean distance are selected as the final ones, from which the user can retrieve the unknown parameters values. Samples with the smallest mean distances lie in the densest regions of the space, i.e., the most representative of the population. Using squared Euclidean distances would have artificially inflated the influence of outliers, making the chosen option more appropriate.

3.2.5. Statistical Methods to Handle Uncertain Parameters

To apply the modelling approaches described in the previous sections, the statistical methods were integrated into the framework to address the cases where the input parameters were incomplete or uncertain. Their role is to generate possible values for unknown inputs and assess the feasibility of the resulting launcher configurations. In practice, the statistical methods connect the IDS and the models. Once the known parameters are defined, they provide the generated samples for the unknown ones. These are then passed to the TM or CM for evaluation.

Monte Carlo-based Methods

The first type of methods integrated were the MC-based ones: normal MC and LH. Their shared principle is the generation of large numbers of input samples that cover the entire space of uncertain parameters. Each sample corresponds to a possible LV configuration, which is then passed to the simulation model for validation. The differences lie in how the samples are distributed, as explained in Section 3.1.2.

A summary of the sampling and analysis process for MC-based methods is given. This can be better visualised with the aid of Figure 3.14. After filling in the IDS and once the CLAVA and UQ files are created, the Python code reads the data. Based on user-defined settings, N samples are generated at each step. This results in N different LV configurations. Each of them is run through the simulation model to be evaluated. Afterwards, the resulting maximum payload will be computed, and the validation criteria will be verified. In case they are fulfilled, that sample/configuration is valid. Otherwise, it is rejected. In the end, one wants to focus on the accepted samples.

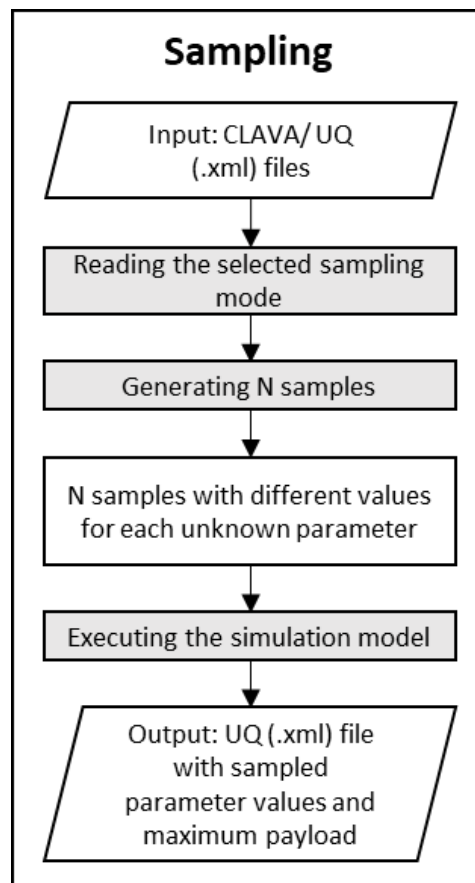


Figure 3.14: Integration of the sampling process for Monte Carlo-based methods in the overall framework.

Approximate Bayesian Computation

ABC was implemented as a smarter approach compared to the sampling methods described above. Instead of simply generating samples across the entire parameter space, ABC adopts a filtering process that evaluates how close the simulated outcomes are to the observed values.

Figure 3.15 illustrates the implementation of ABC in the overall framework. The starting point are the CLAVA and UQ files generated after reading the IDS. In this case, the UQ file is only used to identify the uncertain parameters, since its sampling functionality will not be used. The algorithm then determines the missing parameters and locates their positions in the CLAVA file. Once this mapping is complete, the sampling process begins: the unknown values are replaced with sampled ones, resulting in a complete LV configuration that can be passed to the simulation model. Recalling Section 3.1.2, the fundamental idea of ABC is to compare simulated outputs (produced by an approximate model) to observed or reference results.

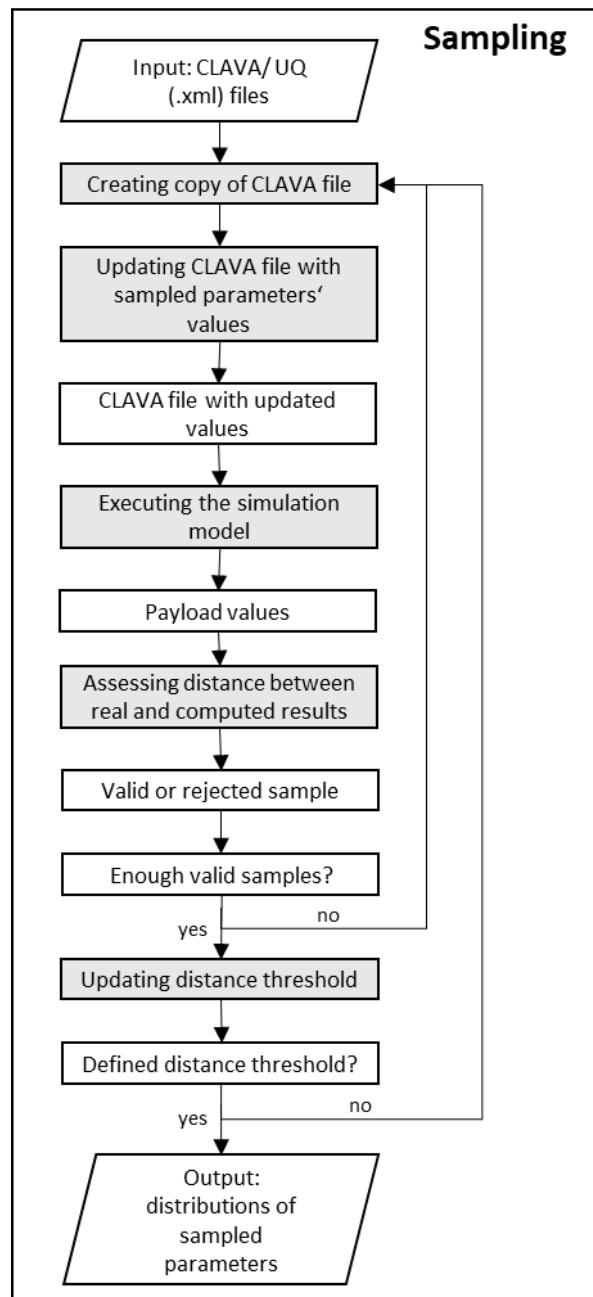


Figure 3.15: Integration of the sampling process for the Approximate Bayesian Computation method in the overall framework.

After the LV configuration is run through the model, the corresponding payloads for the rocket's missions are obtained. The next step is to calculate the distance between these simulated results and the observed ones. If the results fall within the predefined acceptance bounds, a very small distance value (e.g. 0.1) is assigned. Otherwise, the distance to the closest boundary is calculated and used as the ABC distance metric. This value determines whether the configuration is accepted or rejected, depending on the current tolerance threshold.

Once this number is reached, the threshold is updated, as prescribed by the ABC-SMC

procedure (see Section 3.1.2). If not enough valid samples are obtained, additional configurations are generated until the quota is satisfied. With each new generation, the threshold gradually decreases, improving the quality of the approximation. The process continues until the final user-defined tolerance is reached. At this stage, the algorithm outputs the probability distributions of the previously uncertain parameters, which represent the ranges of values most consistent with realistic LV behaviour across all tested cases.

At the start of the sampling procedure, the user defines the number of valid samples required per generation. Once this number is reached, the threshold is updated (see Section 3.1.2). If not enough valid samples are obtained, additional configurations are generated until the quota is satisfied. With each new generation, the threshold gradually decreases until the final user-defined tolerance value is reached. At this stage, the sampling process ends, and one obtains the probability distributions for each sampled parameter. This will most likely indicate the range of values that appear realistic for each tested parameter.

3.2.6. Test Cases

This section describes the test cases used throughout this thesis. They were used to test the statistical methods, and to validate the CM, in the end. The idea was to have three LVs with different levels of complexity, in order to assess whether the CM would only work with simpler rockets. In this case, complexity translates into the number of stages. Since each stage requires additional optimisation points, more complex rockets take longer to converge. Therefore, the following LVs from the Long March (LM) family were selected:

1. Long March 2C (Figure 3.16a): it is the oldest LM rocket still active. It is a medium-lift LV that has two stages, and it uses hydrazine as fuel and has a lift-off mass of 233 t [84]. It is used for multiple types of missions and its engines are the baseline for the improved versions on board of the subsequent LVs [85];
2. Long March 4C (Figure 3.16b): it is one of the most recent hydrazine fuelled Long March rockets to be developed [86]. It is a medium-lift LV with three stages, all fuelled by hydrazine, and it has a lift-off mass of around 249 t. It is most commonly used for LEO and SSO missions;
3. Long March 3B/E (Figure 3.16c): it is an extended version of the heaviest variant of the Long March 3 family, the LM 3B. It has larger boosters and a larger first stage. It is a medium-lift LV with three stages and four strap-on boosters. All stages use hydrazine as fuel, except the third one, which uses liquid hydrogen. It has a lift-off mass of around 456 t and it is mostly used for GTO missions [87];



Figure 3.16: Images of the three rockets used as test cases throughout this thesis.

Additionally, different numbers of unknown parameters were tested for each LV. This was very important to understand if the results are only accurate for a limited number of unknowns, or if this factor doesn't have much influence. For the LM 2C, a larger number of unknown parameters were tested. For the others, there are fewer test cases because the results had a similar behaviour regardless of having 3, 5, 7, or 10 unknown parameters (see Section 4.3). Table 3.2 summarizes all test cases.

It is important to note that the unknown parameters for each test case were chosen in a way that all were tested at least once. Moreover, the configurations with more unknowns are identical to the ones with fewer unknown parameters, but with additional ones. The input ranges also remain the same when adding more unknowns. For instance, test case LM2C 2U is the same as LM2C 1U but with the structure mass of stage 2 as an extra unknown. The only exception is test case LM3BE 5U because it includes the maximum mass flow rate per engine. This parameter was not used in the TM, so it is excluded from the other two test cases from this LV. However, during the CM's testing phase, one test case had to include it to be sure that this parameter could also be used as one of the unknown ones. Lastly, the number of missions translates to more payload values to be estimated, providing a more comprehensive validation.

Table 3.2: Summary of test cases and corresponding unknown parameters.

Test case	Labels	Parameters treated as unknown
LM2C (1–2 missions)	1U	vacuum max. I_{sp} of stage 2.
	2U	vacuum max. I_{sp} of stage 2; structure mass of stage 2.
	3U	vacuum max. I_{sp} of stage 2; structure mass of stage 2; sea-level max. I_{sp} of stage 1.
	5U	vacuum max. I_{sp} of stage 2; structure mass of stage 2; sea-level max. I_{sp} of stage 1; sea-level max. I_{sp} of stage 2; structure mass of stage 1.
	7U	same as 5U, plus engine mass of stage 1; engine mass of stage 2.
	10U	same as 7U, plus ascent propellant of stages 1 and 2; reserve propellant of stage 2.
LM4C (3 missions)	2U	vacuum max. I_{sp} of stage 2; ascent propellant of stage 3.
	3U	same as 2U, plus structure mass of stage 1.
	5U	same as 3U, plus ascent propellant of stage 2; vacuum max. I_{sp} of stage 3.
	10U	same as 5U, plus engine mass of stage 1; reserve propellant of stage 1; sea-level max. I_{sp} of stage 1; structure mass of stage 2; vacuum max. I_{sp} of stage 1.
LM3BE (1 mission)	3U	vacuum max. I_{sp} of boosters; sea-level max. I_{sp} of stage 1; ascent propellant of stage 3.
	5U	same as 3U, plus maximum mass flow rate per engine of stage 1; vacuum max. I_{sp} of stage 2.
	10U	same as 3U, plus ascent propellant of boosters; ascent propellant of stage 1; sea-level max. I_{sp} of boosters; structure masses of stages 2 and 3; vacuum max. I_{sp} of stages 2 and 3.

3.3. Expected Results

This section outlines the different types of results that are expected to be generated throughout the course of this thesis. Since the framework integrates multiple stages, from data handling to model generation, there are multiple types of outcomes. The expected results, therefore, include both intermediate outputs, such as parameter estimates and validation metrics, and final products, such as reconstructed LV models and optimised trajectories.

The first intermediate result is the IDS. This is an Excel file containing all the input parameters, either as specific values or ranges. Afterwards, the CLAVA and UQ files are generated, containing the Excel data in a more structured way, ready to be used with the SART Toolbox. From here, the algorithm generates samples. Each one will have its own CAC, STSM, and TOSCA files, which the CM uses for the optimisation.

Table 3.3: Overview of data types, collection methods, and processing details.

Type of Data	File Format	Collection Method	Purpose	Storage	Access
LV Characteristics	Excel file	Research	Model generation	DLR server	DLR
LV Characteristics	CLAVA file (.xml)	DLR Software	Model generation	DLR server	DLR
Uncertain LV Characteristics	UQ file (.xml)	DLR Software	Model generation	DLR server	DLR
Mass	STSM file	DLR software	Traj. optimisation	DLR server	DLR
Aerodynamic	CAC file	DLR software	Traj. optimisation	DLR server	DLR
Trajectory	TOSCA file	DLR software	Validation	DLR server	DLR

The final results will be the updated Excel file with the valid parameter ranges, as well as the SART Toolbox files for the chosen configurations.

The data collected and the results obtained are summarised in Table 3.3. It shows the different types of files and data resulting from each type of model (e.g. mass). Besides, it displays where the information is stored and who can access it.

3.4. Verification and Validation

In this section, the verification and validation (VV) method for this thesis work will be addressed.

Ensuring the reliability of the semi-automated rocket remodelling framework requires a solid VV methodology. This study adopts multiple validation techniques, combining manual verification methods previously used during an internship with additional computational and observational approaches. The aim is to assess if the generated mass, aerodynamic, and trajectory models are realistic, as needed per Req. 5a, even when handling uncertainties in input parameters.

The primary validation method relies on payload capacity estimation. It serves as a good indicator of the accuracy of the remodelled LV. A payload distant from the one given by the LV manufacturer indicates that something in the model is wrong. Either the trajectory is sub-optimal, thus reducing the achieved payload, or something is over/underestimated in the rocket parameters. In any case, the results would not be realistic, thus it is not desired.

The VV process follows the same methodology applied in the previous manual remodelling procedure during the internship that preceded this master's thesis. After finishing the development of the mass and aerodynamic models and optimising the trajectory, the computed payload capacity is compared to the manufacturer's claimed payload for the given LV. If the difference between the estimated and actual payload remains within $\pm 10\%$, the model is considered sufficiently accurate. However, if the deviation exceeds this margin, refinements are needed. For instance, some strategies that can be implemented might involve improving unknown parameters estimation, particularly for cases in which there is a lack of data. This iterative refinement process guarantees that the remodelling process produces trustworthy results that accurately

reflect the real LVs.

Additionally, another strategy can be used for some rockets where publicly available telemetry data is accessible. For example, usually SpaceX's launches are recorded and show telemetry on the streaming video [88]. This other validation step might include comparing key flight parameters, such as altitude, velocity, and acceleration, with the results of the optimised trajectory. Significant disparities between the computed and the provided telemetry data indicate that further refinements in the initial conditions/assumptions or the mass or aerodynamic models may be necessary. This approach strengthens the exactness of the remodelling framework by combining it with real flight performance data from the LV's operating company.

However, it is often hard to find the telemetry data for rocket launches. Therefore, another strategy imposes itself in scenarios where LVs lack publicly available telemetry data. By analysing the rocket's launch video, its initial acceleration can be estimated. This process implies knowing the height of the launch tower that supports the LV before its take-off. The next step is to measure how long the lowest part of the LV takes to be above the highest point of the launch tower. The acceleration can be approximately obtained with that information and using:

$$s = s_0 + v_0 t + \frac{1}{2} a t^2 \quad (3.18)$$

where s is the distance (launch tower height) s_0 is the initial altitude (null), v_0 is the initial velocity (null), a is the average acceleration and t is time. This is used to obtain the acceleration. However, this is only the "visible" acceleration, since part of the total value is used to overcome gravity. So, in reality, the LV's acceleration is $a_{real} = a + g_0$.

With the acceleration and the initial LV mass, one can estimate the initial thrust:

$$F = GLOM \cdot a_{real} \quad (3.19)$$

where GLOM is the Gross Lift-Off Mass. Then, by comparing this to the LV's thrust found in the research, one can check if that value is realistic. While this method is less precise than direct telemetry comparisons, it still provides an additional measure to verify whether the remodelled rocket aligns with the observed launch behaviour.

To sum up, the validation process follows an iterative refinement strategy, ensuring that the developed framework produces consistent and realistic results, given any circumstances. Integrating multiple validation strategies in this work will ensure the proposed methodology is reliable. To complement it, a sensitivity analysis will also be performed (results are available in Section 4.1).

4

Results and Discussion

This chapter presents and analyses the outcomes of the methodology developed throughout this thesis. The results are structured to illustrate how the proposed framework performs across its different components, from the IDS to the implementation of statistical methods and the application of the test and complex models. Each set of results is critically discussed, highlighting their significance, limitations, and implications for this launcher remodelling framework.

The discussion aims to connect the results to the research objectives and questions defined in Chapter 1. In particular, emphasis is placed on evaluating whether the framework can reliably handle incomplete datasets, generate feasible LV configurations, and balance computational cost with modelling accuracy.

This chapter will have three main sections presenting results: a first one, a sensitivity analysis section, which identifies the parameters with the most influence on the payload result; a second one that compares the different statistical methods; and a final one that shows the results with the CM for the test cases, validating the proposed framework. To end the chapter, an overall discussion about all the results is done, together with recommendations for future work. All the results presented were stored, as needed per Req. 1h.

4.1. Sensitivity Analysis

To complement the statistical exploration of uncertain input parameters, a sensitivity analysis is performed, complying with Req. 5c. It allows for the assessment of each parameter's relative influence on the remodelling framework's final results. While the statistical methods provide ranges of feasible values, sensitivity analysis helps identify which inputs drive the most significant variations in payload capacity. This information is critical for prioritising data collection and understanding the framework's robustness when confronted with incomplete or uncertain datasets.

The analysis not only highlights the most influential parameters but also reveals their interactions. By quantifying how sensitive the model outputs are to variations in specific inputs, the framework can provide users with clearer insights into which

parameters must be known with higher accuracy and which can be approximated with less impact. Ultimately, this step strengthens the overall methodology by ensuring that uncertainty is strategically understood.

The Sensitivity Analysis Library in Python was used [89, 90]. It allows the user to assess each component's individual impact and their combined effect on the result.

The strategy was to go from the least complex LV, i.e., the one with the fewest stages, to the most complex one. The idea was to use the test cases with ten unknown input parameters to understand the impact of the largest number of parameters possible. However, running such large simulations is impossible since the computational power available is limited. For this reason, only one test case was run with the CM, and it has five unknown parameters.

4.1.1. Test Model Results

The results for test case LM2C 5U are presented in Figure 4.1. The parameter with the greatest influence on the payload result is the structure mass of the second stage. The second parameter with the highest impact is the same stage's vacuum maximum I_{sp} . Looking at Equation 3.11, these two parameters directly influence the Δv produced.

The structure mass for this test case had a considerably wide range compared to its real value. It had an upper limit of more than 30% of its lower value. Such variability strongly impacts the rocket equation's mass ratio, reducing the effective Δv that the stage can provide. Moreover, I_{sp} values are usually higher for upper stages, which means that the same relative influence on the mass ratio will have an absolute impact higher for these stages compared to the first one. These reasons explain the smaller impact of the structure mass of the first stage.

The influence of the vacuum maximum I_{sp} is related to its role in measuring an engine's efficiency. A higher I_{sp} translates into more Δv for the same propellant mass, and thus improved performance. However, its relative effect is less pronounced than that of the structural mass, since a typical variation in I_{sp} is proportionally smaller compared to the potential variability in the mass ratio.

The remaining input parameters are significantly less relevant. Overall, if a parameter has a large relative influence in Equation 3.11, it will have a great impact on the result. This means that parameters like the engine's mass, or the reserve or residual propellant mass are mostly insignificant, because their relative importance in the overall vehicle mass is low.

Additionally, the second-order indices are insignificant for this test case (see Figure D.1 in Appendix D). This indicates that pairwise interactions do not contribute significantly to payload variance. As described in Equation 3.11, Δv , hence payload, is driven mostly by independent contributions, whose behaviour is not influenced by the remaining ones. For instance, a higher I_{sp} always increases the payload, regardless of the masses. Thus, the payload variance is explained by first-order effects, meaning that the contributions of the parameters add up rather than interact in a significant way.

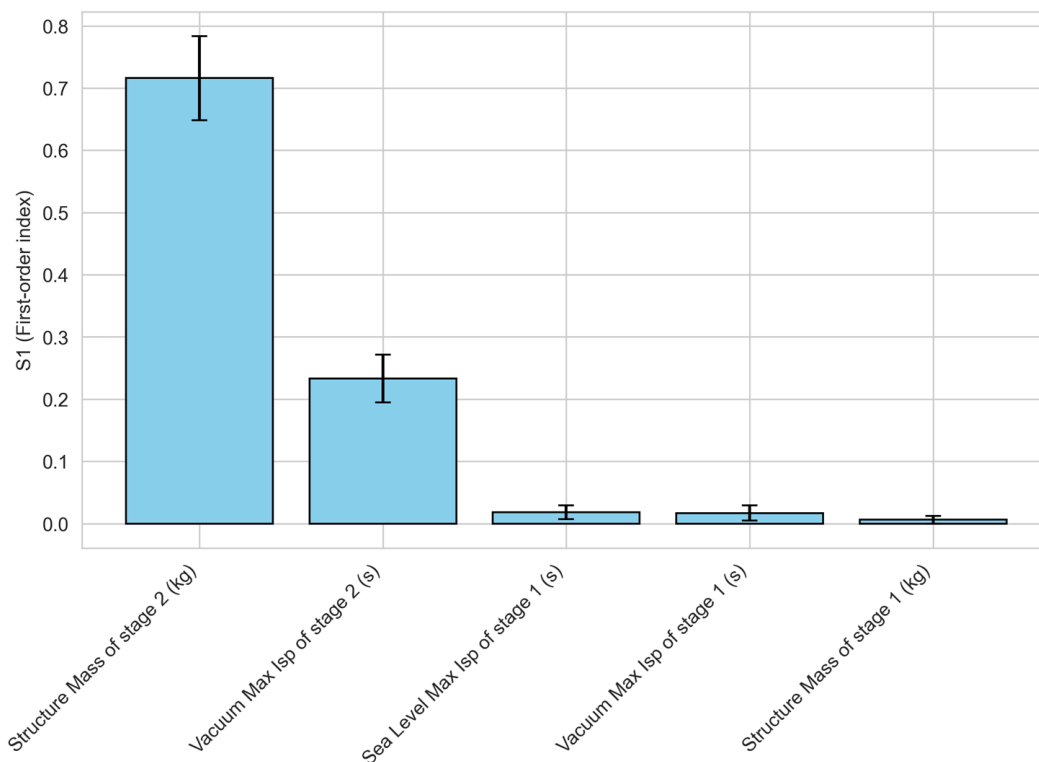


Figure 4.1: Sensitivity analysis first order results including confidence intervals for test case LM2C 5U using the Test Model.

Also, the estimated indices exhibit relatively wide confidence intervals. This could be reduced with a larger sample size, but the computational resources were limited. In addition, having more samples would likely only reduce the interval widths without altering the conclusion, as the upper confidence limits remain small.

Since the goal was to analyse as many parameters as possible, the next step was to run test cases with ten unknowns using the TM. The results are shown in Figure 4.2 and Figure 4.3. Both plots provide a more overarching picture of the influence of the input parameters on the results. There are now more parameters that have a relevant impact.

In the plot showing the results for test case LM4C 10U the vacuum maximum I_{sp} of several stages is very important. On the other hand, for test case LM3BE 10U, the structure mass has a very significant impact, similar to what happened for test case LM2C 5U. It is hard to point out the parameters with the largest influence. As discussed for test case LM2C 5U, having a large relative variation in one parameter might increase its influence. On the other hand, having a small significance when considering the whole LV, for instance, its overall mass, decreases the parameter's impact. This is visible when comparing upper stages with first or booster stages. Due to their greater relative influence on the stage mass, the upper stages have more influential mass-related parameters than the heavier ones.

For both test cases, the second-order indices were also insignificant and had large confidence intervals, which suggests the need for larger sample counts.

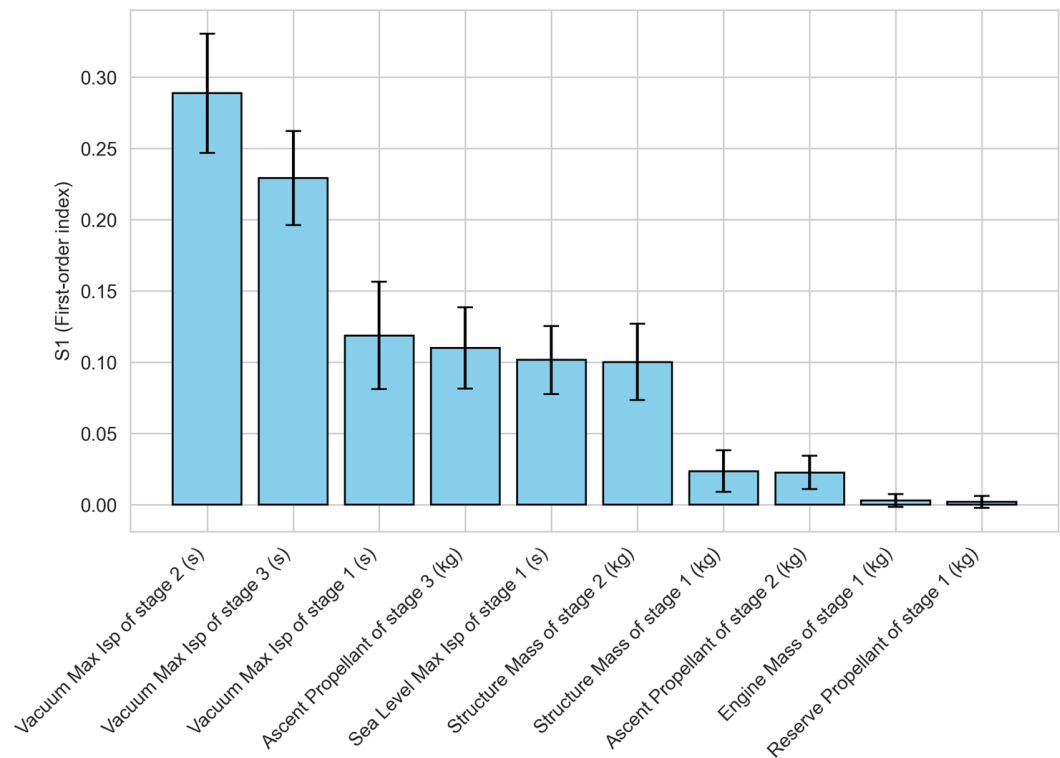


Figure 4.2: Sensitivity analysis first order results including confidence intervals for test case LM4C 10U using the Test Model.

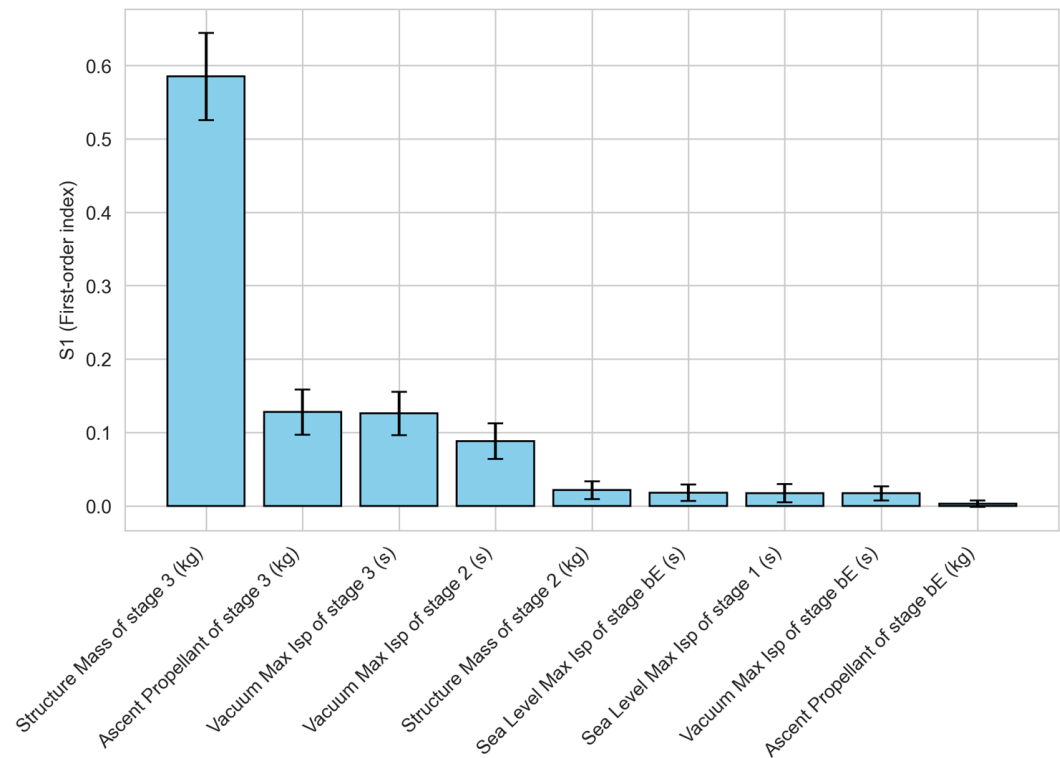


Figure 4.3: Sensitivity analysis first order results including confidence intervals for test case LM3BE 10U using the Test Model.

4.1.2. Complex Model Results

It is interesting to compare the TM results to the ones achieved using the CM. It is expected to find similar parameters with the highest and lowest impacts. At the same time, the results might not be identical because the models used are different. Still, if the results are very different from each other, it means that there is probably something not being covered in one of the models. The CM results are available in Figure 4.4.

As predicted, the results are identical in terms of the most influential parameters, despite the exact first-order indices being different. Before getting into more detail, it is clear that the confidence intervals are relatively wide. Once again, it was not possible to increase the sample size due to limited computational resources. For that reason, it is harder to interpret the exact first-order indices results.

In any case, the second stage's structure mass is the most influential parameter, despite having a lower effect than in the TM. However, when looking at the maximum bound for its confidence interval, it might be close to the TM result. The same reasoning applies to the second stage vacuum maximum I_{sp} .

Overall, the CM results are more distributed and have wider confidence intervals. This occurs because although both models are based on the same physics, the TM relies almost exclusively on Equation 3.11, while the CM is significantly more complex. For instance, it considers an atmospheric model and the trajectory itself while analysing the whole ascent instant by instant, not in a general way as the TM. Thus, it can offset variations in some input parameters by adjusting the trajectory. This reduces the visible direct influence of individual parameters.

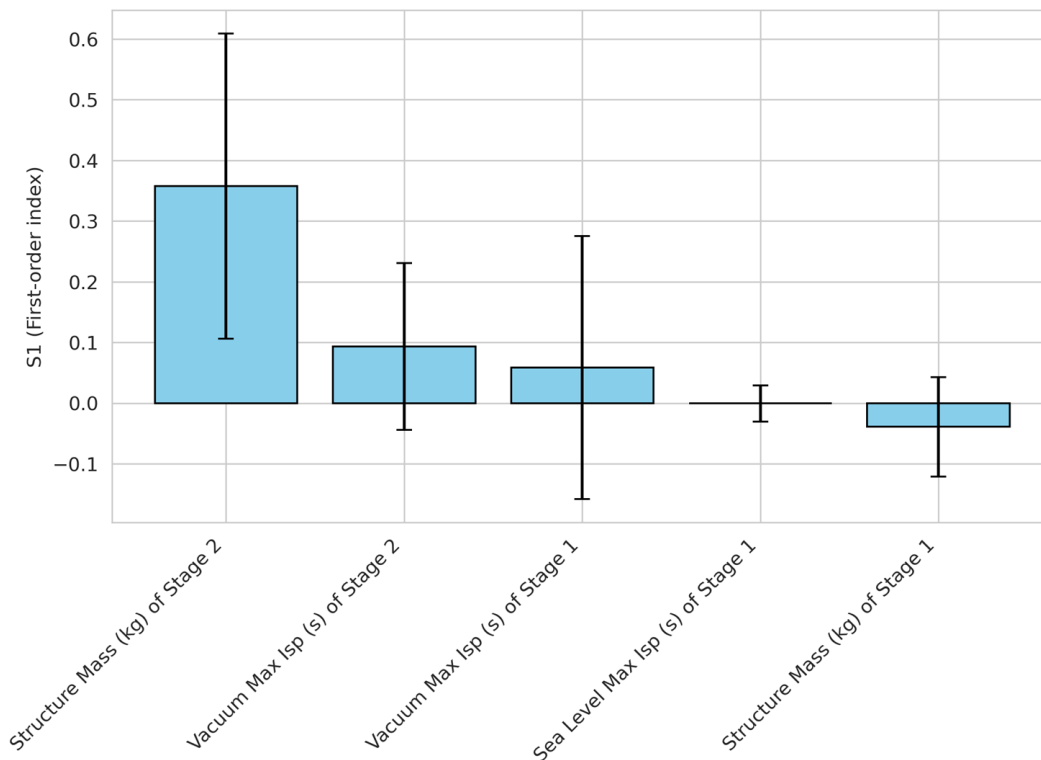


Figure 4.4: Sensitivity analysis first order results including confidence intervals for test case LM2C 5U using the Complex Model.

The remaining parameters have a less significant impact, as occurred for the TM. Similarly, the second-order indices are low (see Figure D.2 in Appendix D), demonstrating how each parameter's influence on the payload calculation is independent. Again, an increase in the sample count could bring narrower confidence intervals, which would result in a more accurate assessment.

4.2. Statistical Methods Performance

This section presents the results of the analysis conducted to the statistical sampling methods. Its goal is to identify the most suitable method to address the research question posed in this thesis. Three main algorithms are evaluated: random MC sampling, LH sampling and ABC. The objective is to determine which method integrates better with the CM. To enable this evaluation with minimal computational cost, the TM will be employed as a fast environment for comparing each statistical method's performance and practicality. The primary selection criteria for a suitable sampling method are accuracy and speed. In this context, accuracy refers more to how well-covered the sampling space is rather than the payload results themselves, since these are not influenced by the sampling method chosen.

4.2.1. Monte Carlo and Variants

MC sampling and its variants were the first statistical methods tested within the framework. Their simplicity and general applicability made it a natural starting point for exploring how uncertain input parameters affect the remodelling outcomes. Parameter values were sampled from predefined distributions in this approach. Then, they were propagated through the TM to evaluate the resulting LV performance. By running several simulations, it was possible to identify feasible configurations and observe how varying the inputs influenced the payload capacity. This section presents the key findings from the MC-based methods. They will serve as a baseline for comparison against more advanced methods such as ABC.

Monte Carlo-based Methods Comparison

As presented in Section 3.1.2, two main MC methods were tested in this thesis. Their choice comes from Chapter 2, and their implementation was simplified since the UQ framework described in Section 3.1.4 already integrates some sampling capabilities. Additionally, while MC-based methods differ in their sampling techniques (ranging from purely random to stratified or quasi-MC approaches [91]), they all fundamentally rely on random or pseudo-random (something that appears to be random but is deterministic) sampling to approximate solutions.

The first step in investigating the MC-based methods is to select one that covers the space more uniformly. It is crucial to guarantee that the whole uncertain range of possible values for each parameter is tested. This way, one avoids missing out on any possible valid LV configurations. Simultaneously, clustering is also undesirable, since it might overpopulate some areas, which could alter the final parameter's probability distribution.

Taking into account the criteria mentioned above, a test scenario was created. The idea was to capture the distribution of the generated samples. This way, one could

compare the different sampling methods regarding how well they cover the whole unknown range. In order to verify this, the same analysis was run with the same number of samples for each MC-based procedure. The test case used was the LM3BE 10U. All the parameters had a uniform probability distribution.

Figure 4.5 shows the results of the previously described procedure. Three different MC-based sampling methods were tested with 100 samples. Sobol sampling was included as a comparison here, since it was used for sensitivity analysis (Section 4.1). In the lower plots, each dot represents a generated sample corresponding to a combination of two parameters: the vacuum maximum I_{sp} of the second stage and the ascent propellant of the first stage. They are among the ten unknown parameters in this test case. The upper plots display the distribution of samples across their sampling space for a single parameter: the vacuum maximum I_{sp} of the second stage.

Starting with the random sampling (Figure 4.5a), one can see a significant gap in the top right corner. As expected, random sampling does not cover the sampling space uniformly. That is supported by the distribution plot, which shows a non-uniform shape in its upper region. Similarly, Sobol sampling had a non-flat distribution. This method was not further used for sampling purposes, as the results didn't differ too much from LH in terms of time consumed, while they were worse regarding coverage of the sampling space. Lastly, no clear gaps are visible when looking at the LH sampling results (Figure 4.5b). On top of that, the distribution plot is perfectly flat between the sampled parameter's boundaries. That demonstrates how effective this method is in the coverage of the sampling space.

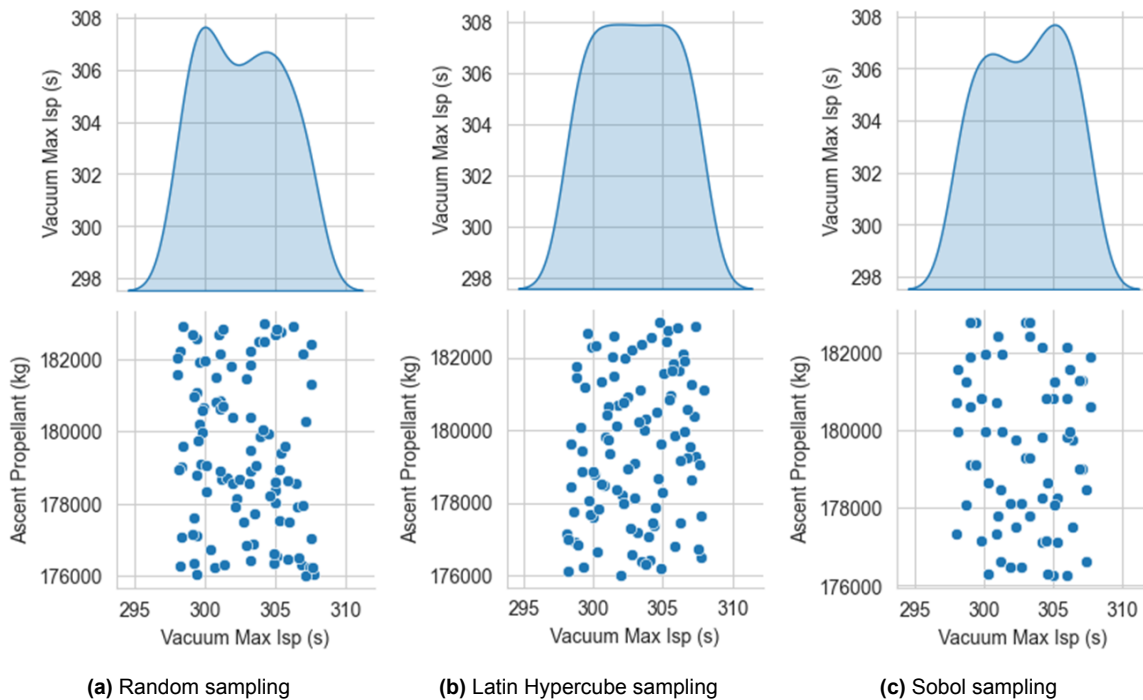


Figure 4.5: Comparison between the results achieved by using different Monte Carlo-based sampling methods with the Test Model - 100 total samples.

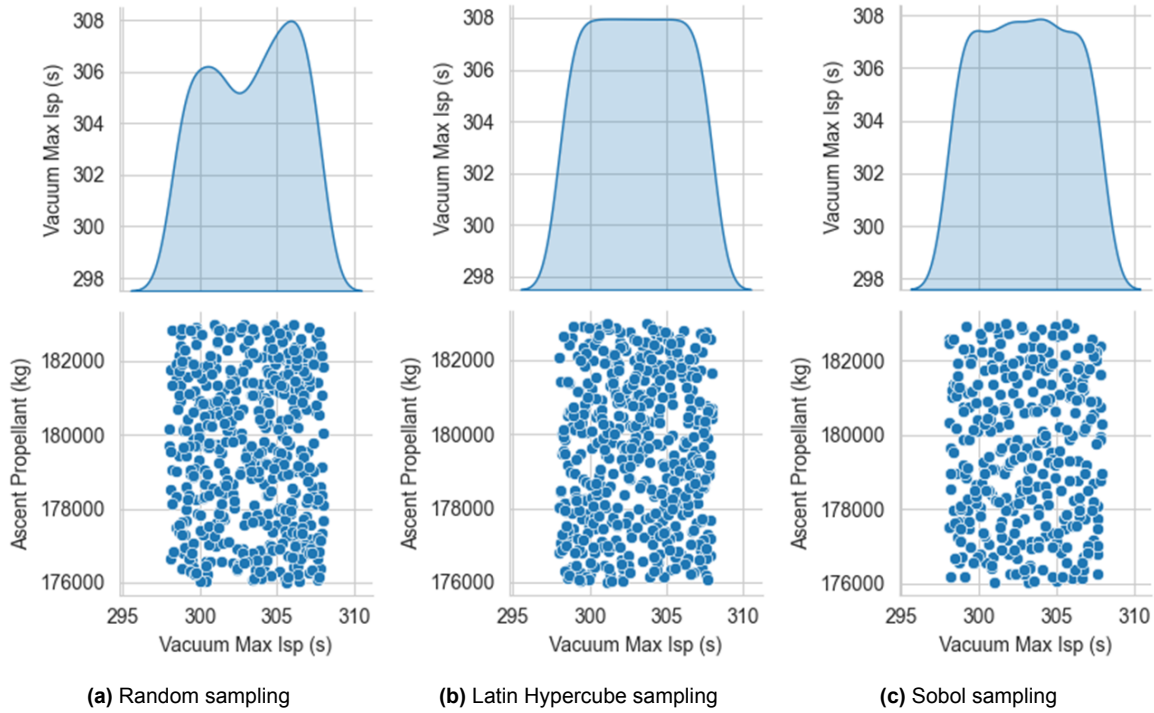


Figure 4.6: Comparison between the results achieved by using different Monte Carlo-based sampling methods with the Test Model - 500 total samples.

However, it is important to note that these problems may have arisen due to the low sample count. Increasing the number of samples will likely lead to a better covered sampling range. The same procedure was repeated to verify this, but with 500 samples. Figure 4.6 shows those results.

Overall, the sampling space has fewer gaps than for 100 samples. However, that is mostly because the number of samples was increased. Random sampling has a distribution plot with peaks and valleys again, which is undesirable. Sobol shows an improvement compared to the lower sample count. Still, LH remains the sampling algorithm with the best coverage of the sampling space. For this reason, it was chosen as the MC-based method to be further tested and later compared to ABC.

4.2.2. Latin Hypercube Performance

After discarding the purely random sampling method due to its inefficiency, the next step is understanding how fast and reliable LH is. As discussed at the beginning of Section 4.2, the criteria for a suitable sampling method are coverage quality and speed. The former has already been discussed, so the next step is to analyse the latter.

To verify how fast the statistical methods are, it is essential to identify how many samples are needed before the optimisation process converges. A tailored test procedure was conducted to get the required number of samples for LH. The same simulation was run multiple times for each test case with increasing values of N , representing the total number of samples. The results were then recorded and plotted using summary statistics, such as minimum, maximum, median, etc. The goal was to visualise how these metrics change by adding more samples. That way, one could

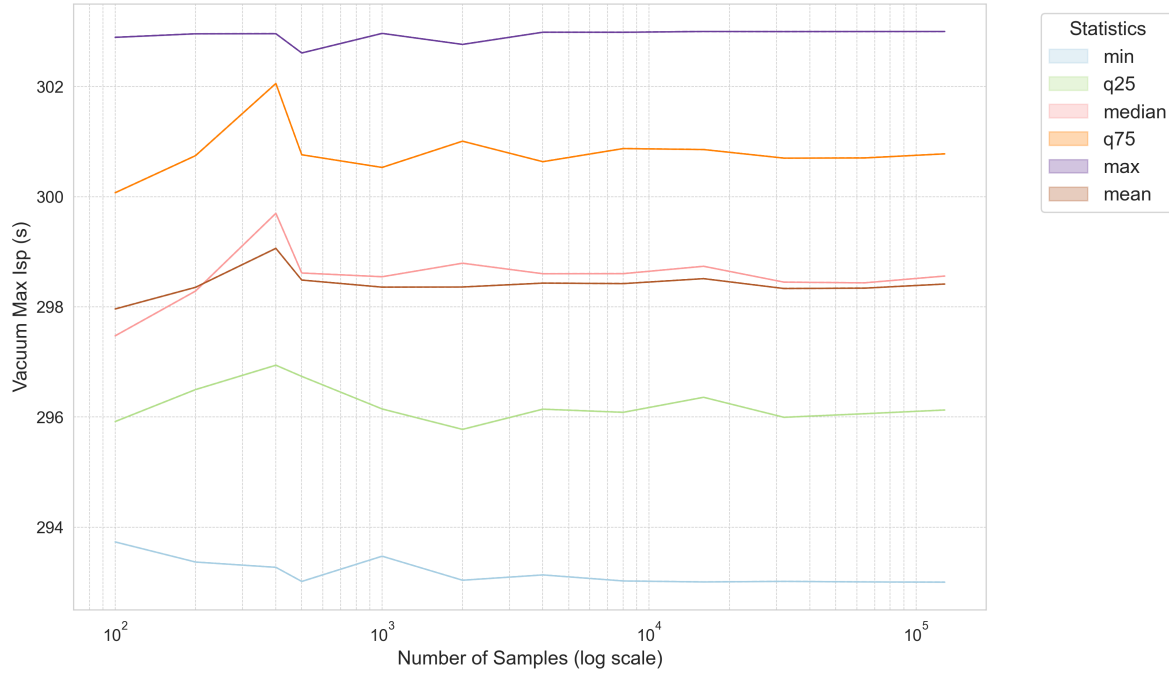


Figure 4.7: Convergence behaviour of Latin Hypercube sampling for the vacuum maximum I_{sp} of stage 2 in test case LM2C 2U. The curves show the evolution of three summary statistics (minimum, median, and maximum) of the valid results as a function of the number of samples (logarithmic scale).

identify a convergence or stagnation point beyond which the outcomes remained stable. That point would provide a sufficient sample size for practical use in the remodelling framework. The output of these simulations was stored in tables identical to Table D.1 of Appendix D.

Figure 4.7 shows the results for the vacuum maximum I_{sp} of the tested LV's second stage. At low sample counts, randomness and undersampling can lead to significant outcome variability. By splitting the region into larger gaps, some valid samples in between them are likely missed. Then, when the sample count increases, the gaps between the samples are smaller, which better covers the sampling space and encompasses the previously ignored ones. However, improvements are less visible beyond a certain point. By then, there are already enough samples to fully cover the sampling space and produce results that do not change when more samples are added.

In concrete, most statistical summary parameters converge around 32000 samples. However, some oscillations still occur around the highest sample counts. Besides, there could be some randomness associated with the results. To better understand whether this result would also be verified in other parameters, one needs an extra plot. Thus, the equivalent plot for the structure mass of the second stage is shown in Figure 4.8.

For this parameter, every summary statistic also converges at around 32000 samples. There are slight oscillations afterwards, but the lines are practically horizontal from that point onward. Given this new information, one could infer that 32000 samples would

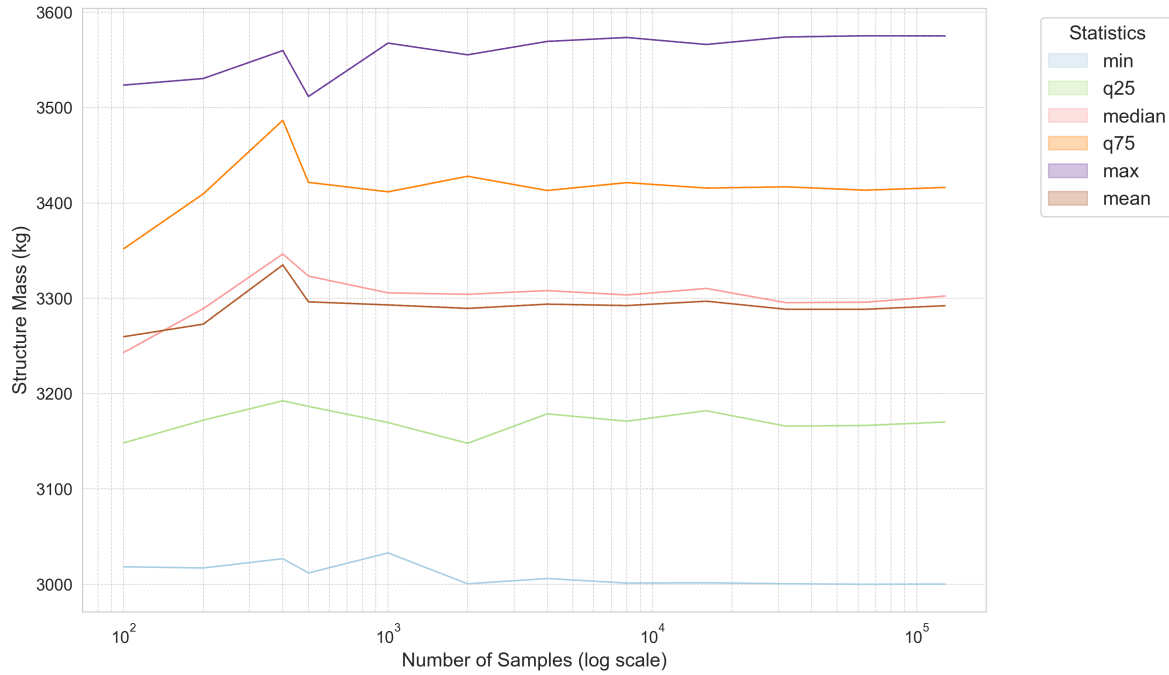


Figure 4.8: Convergence behaviour of Latin Hypercube sampling for the structure mass of stage 2 in test case LM2C 2U. The curves show the evolution of three summary statistics (minimum, median, and maximum) of the valid results as a function of the number of samples (logarithmic scale).

be the minimum value of samples needed to apply LH sampling with accurate results. However, it is possible that the parameters' sampling process had a peculiar/random behaviour in this simulation. Doing a single run for each sample count does not guarantee that the results will always be similar to the ones achieved so far.

To circumvent the uncertainties observed in the previous analysis, the author decided to repeat the simulation ten times under identical conditions. Each run used the same LV and sample sizes. This way, it allowed the effect of sampling variability to be assessed. Rather than plotting a single line per statistical metric, this approach resulted in a band around each summary statistic, such as the minimum, maximum, etc.

For every combination of parameter and sample size, one set of summary statistics was recorded per run. After completing all ten runs, there were ten values for each metric at each sample size. For each run and each unknown parameter, there was a table similar to Table D.1 of Appendix D. These tables were used to make the plots analogous to Figure 4.9 that appear throughout this chapter. To better visualise the spread and convergence behaviour, the minimum and maximum values from these ten runs were used to define the upper and lower bounds for each statistic. These bounds were then plotted as convergence bands across the increasing number of samples. When these bands (coloured area in between the dashed lines) begin to flatten or exhibit no significant change, it indicates that result stability has been reached. Thus, further increasing the number of samples is unlikely to affect the results. This point can be interpreted as the effective minimum required sample size for robust performance under the given conditions. To aid in observing the results, the mean is also shown

as a solid line.

These results can be seen in Figure 4.9 and in Figure 4.10. On the first plot, the results are similar to the plot in which only a single run was shown (Figure 4.7). As before, all statistical summaries converge around 32000 samples. On the second plot, a similar stagnation point is found. Additionally, the randomness associated with lower sample counts is visible. For instance, the maximum value oscillates before stabilising around its final value.

By running the same simulation multiple times, the randomness factor is mitigated. There is still a possibility that ten times is not enough. However, it is the maximum possible/available given the computational constraints, even for the TM. With the CM, simulating with 128000 samples once would have taken more than one month, so it would not be feasible.

The results still show that 32000 samples would probably be an enough amount of samples. This is more than desired, so the next step was testing the ABC method and determining if this number could be reduced. It is also important to add that the number of samples needed for the final validation cases with the CM might be different from the number needed for the TM. Therefore, the main goal in the following sections is to assess which statistical method is better, rather than the exact number of samples.

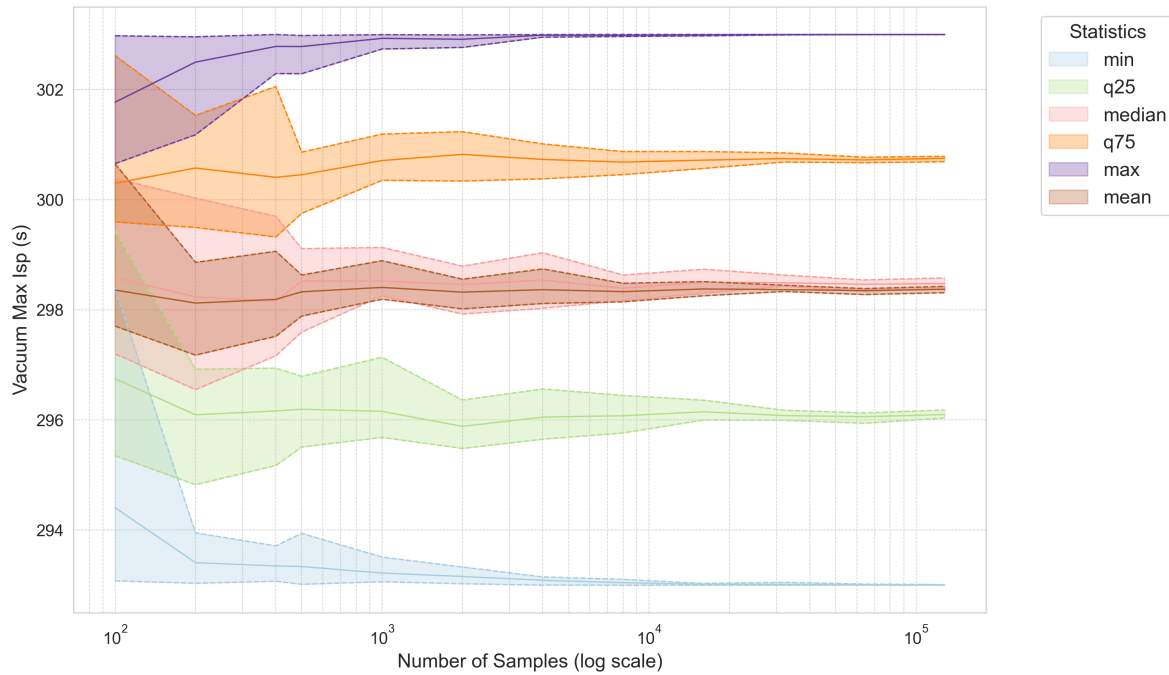


Figure 4.9: Convergence behaviour of Latin Hypercube sampling for the vacuum maximum I_{sp} of stage 2 in test case LM2C 2U. The curves show the evolution of three summary statistics (minimum, median, and maximum) of the valid results as a function of the number of samples (logarithmic scale). Each simulation was repeated ten times to account for variability. The coloured regions represent the convergence bands, defined by the minimum and maximum values across the ten runs for each statistic.

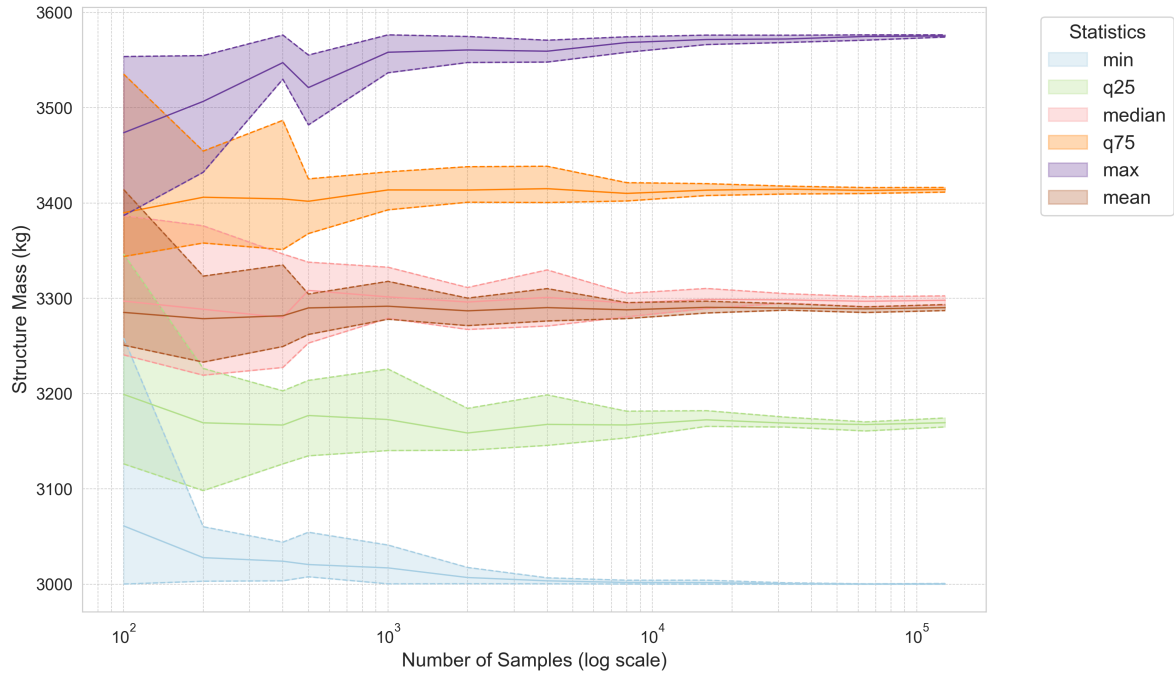


Figure 4.10: Convergence behaviour of Latin Hypercube sampling for the structure mass of stage 2 in test case LM2C 2U. The curves show the evolution of three summary statistics (minimum, median, and maximum) of the valid results as a function of the number of samples (logarithmic scale). Each simulation was repeated ten times to account for variability. The coloured regions represent the convergence bands, defined by the minimum and maximum values across the ten runs for each statistic.

4.2.3. Approximate Bayesian Computation Performance

This section presents the results of the testing procedure for the ABC method, with a focus on its convergence behaviour. The testing process was very similar to the one used for LH sampling. Starting from the same reference IDS, the ABC sampling was executed ten times with multiple sample sizes. This repetition was implemented to ensure that the results are not biased by randomness in a single run. The results for the maximum vacuum I_{sp} for the second stage of LM2C are shown in Figure 4.11.

Before analysing the plot, it is important to clarify that all reported ABC results correspond to the total number of generated samples, rather than only the valid ones. These totals were estimated by averaging the number of samples across repeated runs for each specified sample count, as slight variations occurred between runs. Since the user can only define the number of valid samples per generation, one cannot estimate how many total samples will be generated. Therefore, not all ABC results will start in the same minimum number of samples.

That being said, the observed behaviour for ABC is similar to the one observed for the MC sampling in Figure 4.9. The bands are wide for lower sample counts and tend to narrow down as the number of samples increases. A more detailed comparison will be made in Section 4.3, with more test cases, combined plots for both methods, and a thorough discussion about the findings.

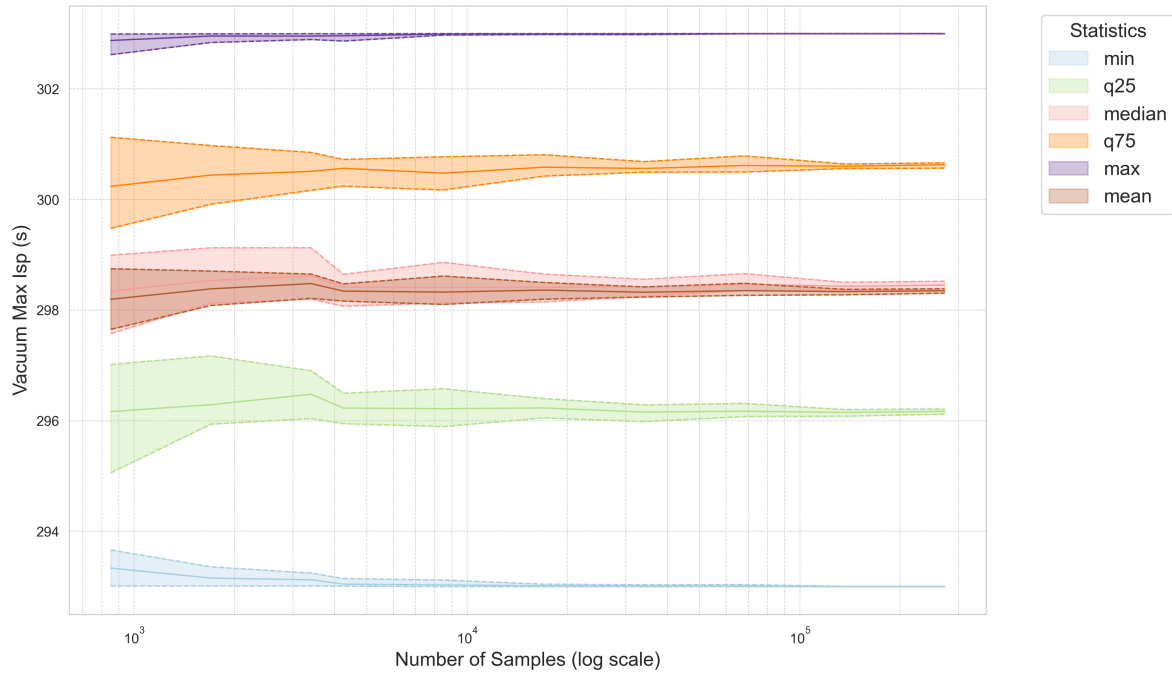


Figure 4.11: Convergence behaviour of Approximate Bayesian Computation (ABC) for the maximum vacuum I_{sp} of stage 2 in test case LM2C 2U. The curves show the evolution of three summary statistics (minimum, median, and maximum) of the valid results as a function of the number of samples (logarithmic scale). Each simulation was repeated ten times to account for variability. The coloured regions represent the convergence bands, defined by the minimum and maximum values across the ten runs for each statistic.

4.3. Comparison Between Latin Hypercube and Approximate Bayesian Computation

This section presents the comparison plots and results between the two sampling methods tested: LH and ABC. It consists mostly of plots similar to before, but instead of having one method separate from the other, the results are grouped into single plots for easier visualisation. Due to the overlap of both sampling methods, the plots were split into two. One contains the "Core Statistics", such as the maximum (max), minimum (min), and mean summary statistics. The other contains the "Extended Statistics", which include the median and the 25% and 75% quartiles (q25 and q75, respectively). This way, interpreting the results is easier.

The main aspects to be considered throughout this comparative analysis are:

- the sampling method that converges faster;
- the influence that the number of unknowns has on the results;
- the influence of the LV's complexity on the results;

Due to the factors mentioned above, the overall strategy for this testing phase is based on LV complexity. Initially, the rocket with fewer stages will be tested, with a test case which has relatively few unknown parameters. Afterwards, the idea is to compare this scenario to the test cases with more unknowns. Finally, the analysis is extended to

the more complex rockets, again going from few to many unknown parameters. This plan enables the assessment of the three points listed above by progressively building from the simplest test case to the most complex one.

The first plot shown (Figure 4.12) is for the maximum vacuum I_{sp} for the second stage of test case LM2C 2U. For this particular parameter, there are not many discrepancies between LH sampling and ABC. All summary statistics represented there have an identical width for the same number of samples.

Another fundamental detail is whether the maximum and minimum bands converge to the same parameter value for both methods. If they don't, it suggests that one of them is not behaving correctly, because they should achieve consistent results with a sufficiently high number of samples. Ensuring the sampling methods adequately cover the entire sampling space and find solutions near the boundaries is crucial. This prevents the exclusion of potentially feasible LV configurations, which increases the overall framework's robustness. In this case, both methods stabilise around the same values for the minimum and maximum metrics, which correspond to the sampling boundaries.

To be able to better judge the results, it is important to look at the other parameter as well. Figure 4.13 shows the results for the structure mass of this LV's second stage.

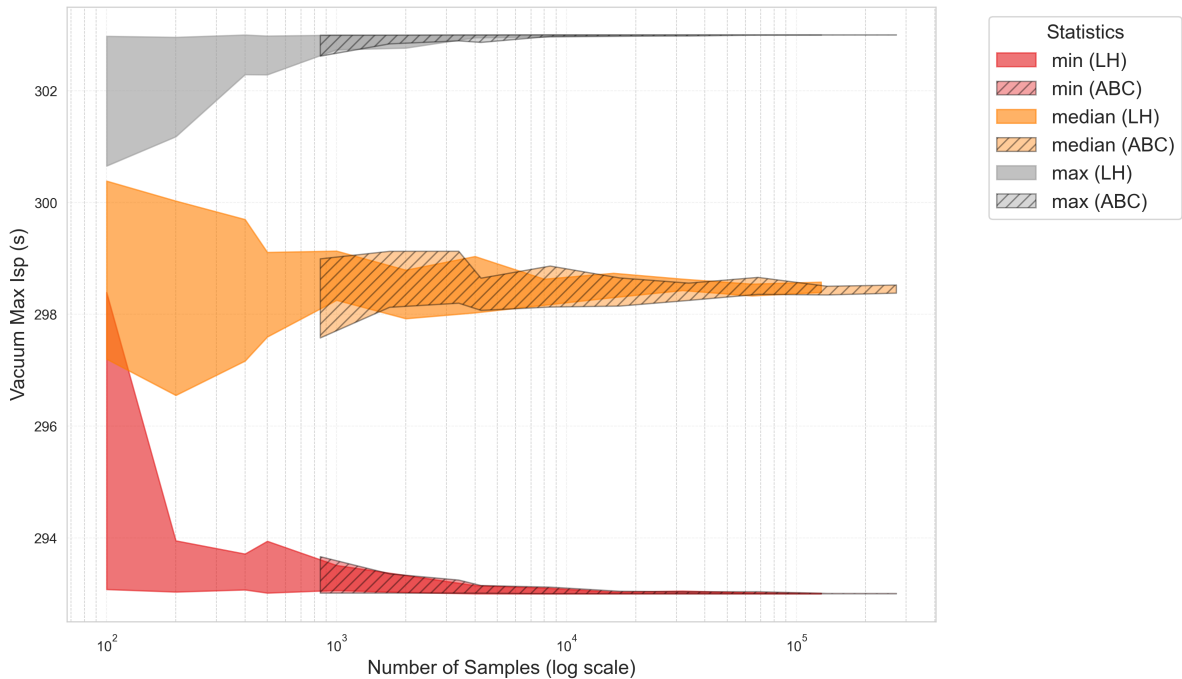


Figure 4.12: Convergence comparison of Latin Hypercube sampling and Approximate Bayesian Computation for the maximum vacuum I_{sp} of stage 2 in test case LM2C 2U. The curves show the evolution of three summary statistics (minimum, median, and maximum) of the valid results as a function of the number of samples (logarithmic scale). Each simulation was repeated ten times to account for variability. The coloured regions represent the convergence bands, defined by the minimum and maximum values across the ten runs for each statistic.

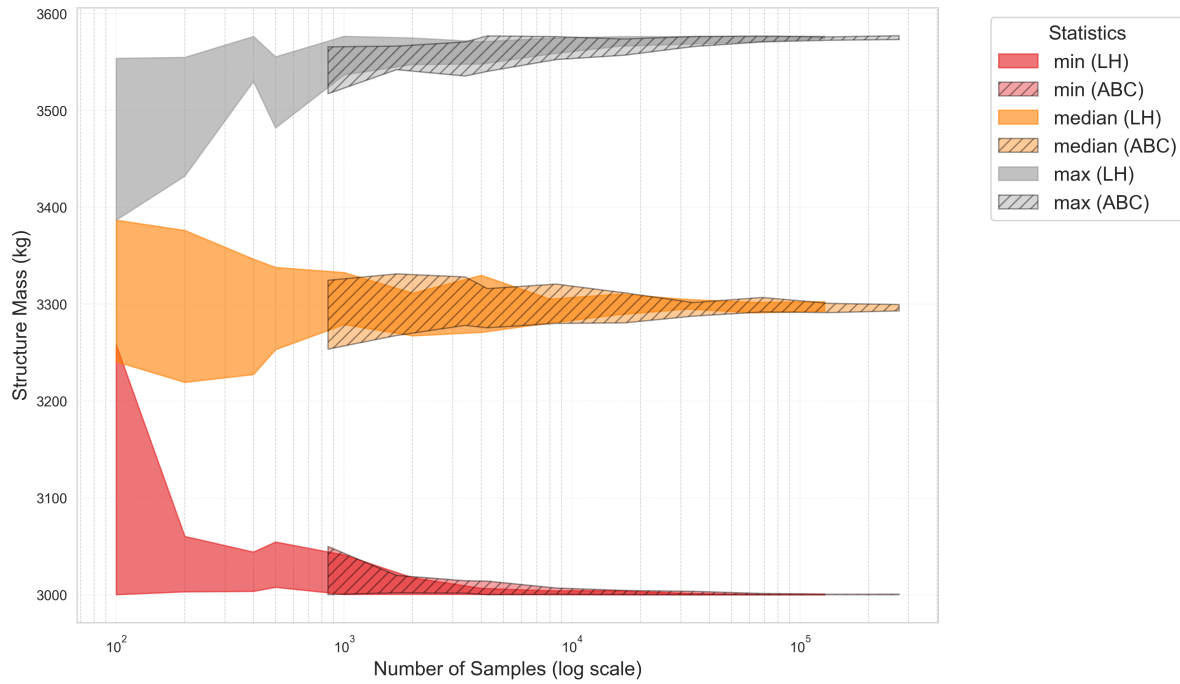


Figure 4.13: Convergence comparison of Latin Hypercube sampling and Approximate Bayesian Computation for the structure mass of stage 2 in test case LM2C 2U. The curves show the evolution of three summary statistics (minimum, median, and maximum) of the valid results as a function of the number of samples (logarithmic scale). Each simulation was repeated ten times to account for variability. The coloured regions represent the convergence bands, defined by the minimum and maximum values across the ten runs for each statistic.

Once again, LH sampling is better than ABC since all its summary statistics flattened out slightly earlier. Additionally, they are usually narrower or equally wide for the same number of samples. This means that, as occurred with the maximum vacuum I_{sp} of stage 2, LH's results surpass ABC's results in both speed and stability. However, this was only one example of a rocket with two unknown parameters. To make an accurate assessment of these methods' performance, one has to analyse more test cases.

One of the main features to evaluate is the impact of the number of uncertain parameters. The results shown so far were for a test case with two unknowns, so it is interesting to compare them to a single unknown parameter case. Thus, the results of test case LM2C 1U will be presented next.

The first observation from Figure 4.14 concerns the amplitude of the results. For test case LM2C 2U the maximum vacuum I_{sp} was between 293 s and 303 s for valid samples. On the other hand, when this was the only uncertain parameter, the valid results were confined to between 297 s and 299.25 s. The difference is caused by introducing a second unknown parameter. Their combined effect made it possible to have a wider range of valid values for the I_{sp} . In this case, a lower or higher structure mass of the second stage allowed the same stage's maximum vacuum I_{sp} to be lower or higher, respectively, compared to when this mass was fixed. This clearly illustrates that simply by adding a second unknown, the final ranges for both uncertainties can be very wide.

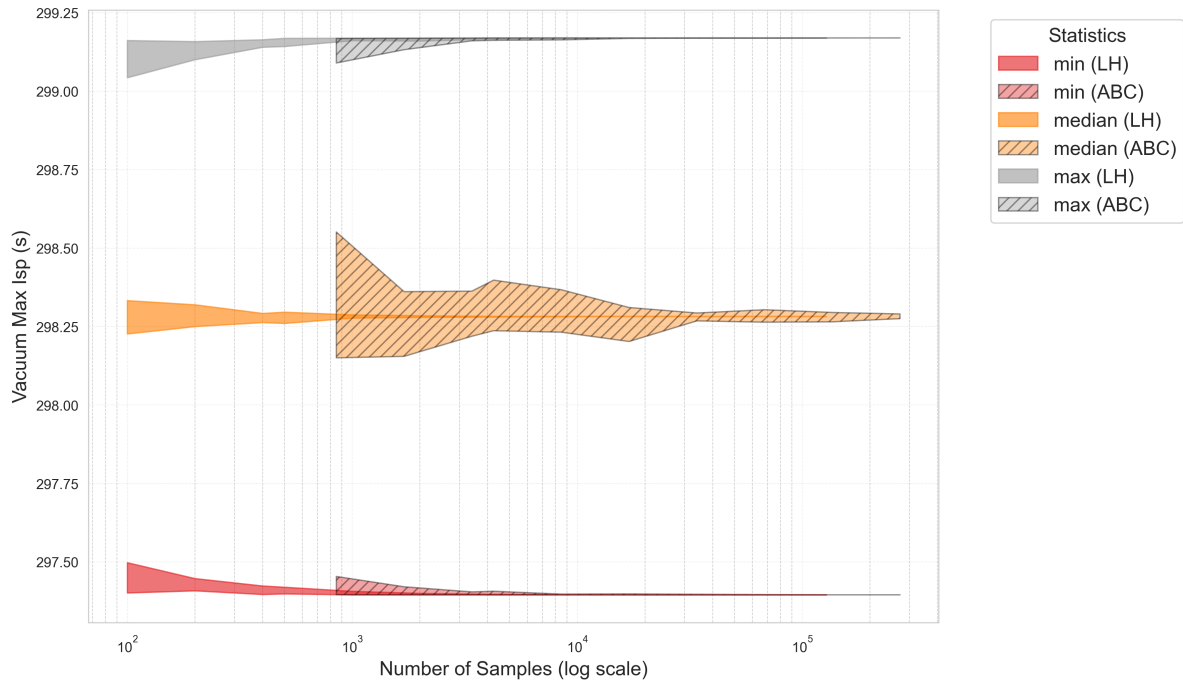


Figure 4.14: Convergence comparison of Latin Hypercube sampling and Approximate Bayesian Computation for the maximum vacuum I_{sp} of stage 2 in test case LM2C 1U. The curves show the evolution of three summary statistics (minimum, median, and maximum) of the valid results as a function of the number of samples (logarithmic scale). Each simulation was repeated ten times to account for variability. The coloured regions represent the convergence bands, defined by the minimum and maximum values across the ten runs for each statistic.

Furthermore, LH sampling completely outperforms ABC in this test case. This outcome was expected, since the easiest way to explore the sampling space for a single unknown parameter is to split it and check the boundaries for the valid results. This is what LH does. In contrast, ABC is a more complex method, which might not be so helpful in this situation. Its iterative process of generating populations with decreasing tolerance values takes longer to converge, as it requires more samples in total. Thus, so far, LH sampling is performing better than ABC.

At this stage, one knows that two uncertain input parameters can yield very different results compared to a single one. The next logical step is to assess what happens when there are additional unknown parameters. After analysing those results, the author noticed that there are not many differences between 3, 5, 7 or 10 unknowns. Therefore, some of the results for test case LM2C 10U of the Long March 2C rocket will be presented, since this is the example with more uncertainties.

Figure 4.15 shows the resulting valid range for the maximum vacuum I_{sp} of the second stage, for the test case with ten unknown input parameters. The first thing to notice is how the minimum and maximum values are more distant compared to test case LM2C 1U. In fact, both summary statistics stabilise at the boundaries defined for this parameter, as happened with test case LM2C 2U. The median value is also similar to the scenario with two unknown parameters. This behaviour indicates one of two things: having more than one unknown parameter always has a similar impact, regardless

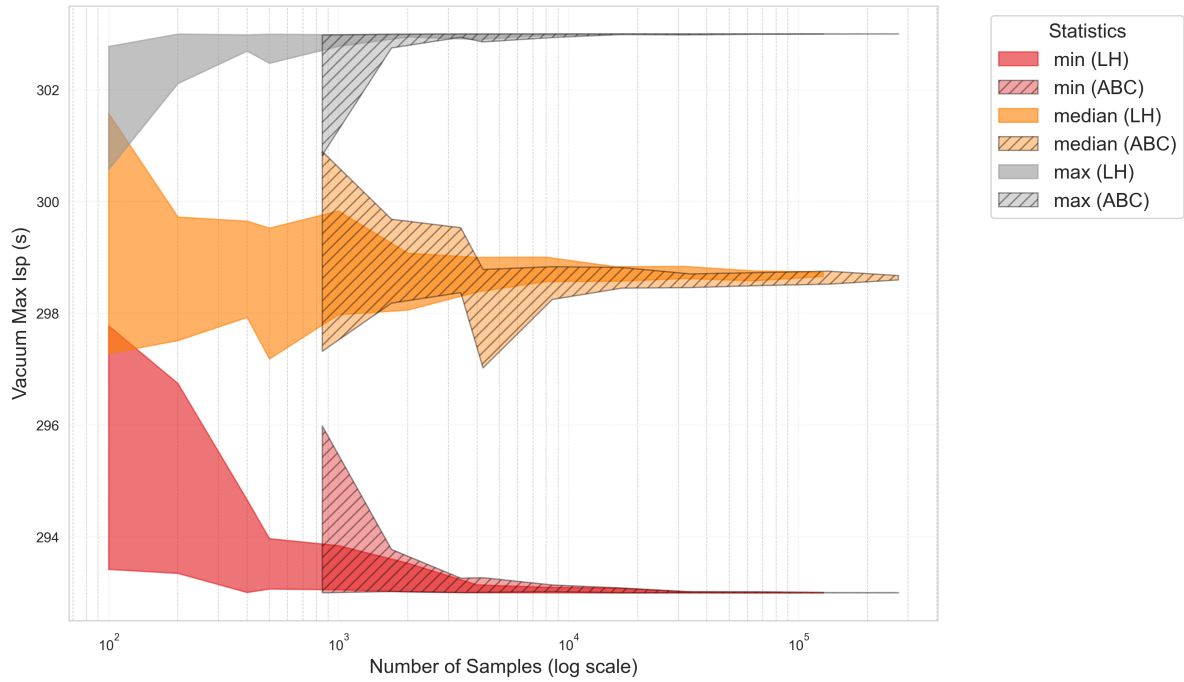


Figure 4.15: Convergence comparison of Latin Hypercube sampling and Approximate Bayesian Computation for the maximum vacuum I_{sp} of stage 2 in test case LM2C 10U. The curves show the evolution of three summary statistics (minimum, median, and maximum) of the valid results as a function of the number of samples (logarithmic scale). Each simulation was repeated ten times to account for variability. The coloured regions represent the convergence bands, defined by the minimum and maximum values across the ten runs for each statistic.

of how many extra parameters; or one of these two parameters has such a large influence that adding other uncertainties doesn't significantly affect the results. In any case, further test cases have to be assessed before one can be sure if it is only the second option, as it was proven in Section 4.1.

Additionally, in this case, LH sampling also converges faster than the ABC method, although the difference is not too substantial. Moreover, the convergence value does not perfectly coincide for the median. However, this might be explained by a slight difference in the results distribution for both methods. Figure 4.16 contains the final distributions for the vacuum I_{sp} of the second stage for a single run of the ten made. It compares the results of LH sampling and ABC.

The plot shows that the distributions for each method do not perfectly coincide, even though they show the same tendency. The sampling strategies have different algorithms, that is, they generate samples in different ways, as explained in Section 3.1.2. Thus, the resulting sets of samples do not completely overlap, leading to slight discrepancies in the outcomes. This also explains why the median values do not converge to the same value in Figure 4.15. The green surface, which corresponds to LH, extends further into the region of higher values. Hence, the median is marginally shifted upwards. The ABC results are more concentrated around the central region. The results presented here correspond to a single run, but at the highest sample counts the variance is small, making this run representative of the overall behaviour

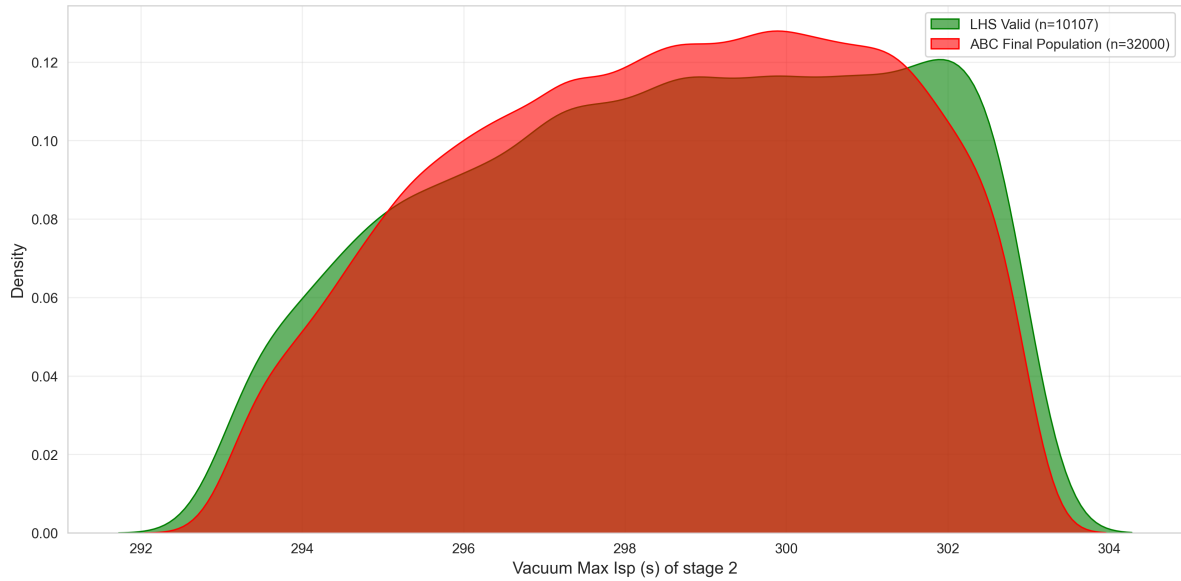


Figure 4.16: Comparison of the final distributions for one run of Latin Hypercube sampling and Approximate Bayesian Computation for the maximum vacuum I_{sp} of stage 2 for test case LM2C 10U.

across multiple repetitions.

To better understand the difference that arises when dealing with two versus ten unknown input parameters, it is useful to analyse the results for the structure mass of the second stage. Figure 4.17 shows the comparison between the results of LH and ABC for this parameter.

The first notable feature is that the maximum metric only stabilises at a very high sample count. This suggests that the actual number of unknown input parameters can affect the results. Besides, the maximum value is lower than in test case LM2C 2U, which means that the combination of the other parameters restricts the realistic range of this one. In LM2C 2U, valid samples relied on the real LM 2C values for the remaining parameters. With more unknowns, as in LM2C 10U, those parameters deviate from their real values, and combinations that were previously valid may no longer be so. This further suggests that the structure mass of the second stage is particularly influential in determining the overall results. Otherwise, the full tested range would remain valid. This has been demonstrated in Section 4.1.

Finally, the last note about Figure 4.17 concerns the median's behaviour. Unlike Figure 4.16, here both methods converge to virtually the same value. This observation is supported by the distributions plot in Figure 4.18. It shows nearly identical shapes with pronounced peaks around the same value. This close alignment explains why the medians of the LH and ABC results match better than before.

Now that a 2-stage LV has been analysed, one already has a preliminary understanding of how the number of unknown parameters influences the results. Besides, LH has a better convergence speed than ABC. The next logical step is to increase the LV's complexity and check if this tendency remains. Therefore, the next results concern the LM 4C.

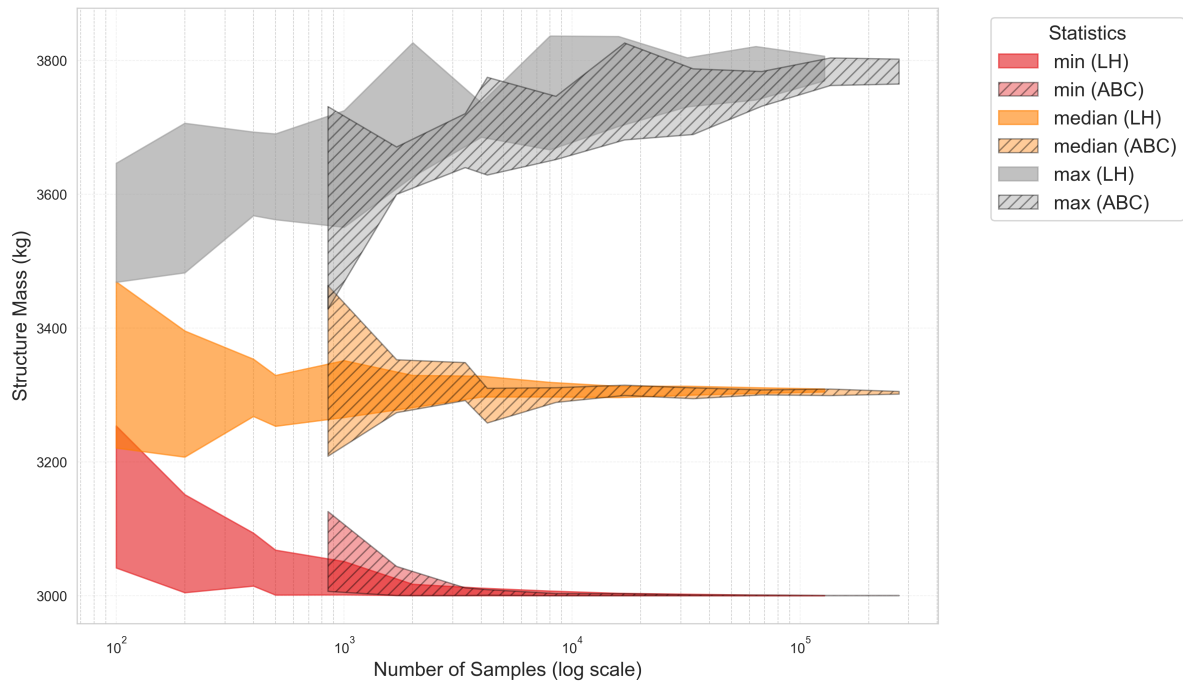


Figure 4.17: Convergence comparison of Latin Hypercube sampling and Approximate Bayesian Computation for the structure mass of stage 2 in test case LM2C 10U. The curves show the evolution of three summary statistics (minimum, median, and maximum) of the valid results as a function of the number of samples (logarithmic scale). Each simulation was repeated ten times to account for variability. The coloured regions represent the convergence bands, defined by the minimum and maximum values across the ten runs for each statistic.

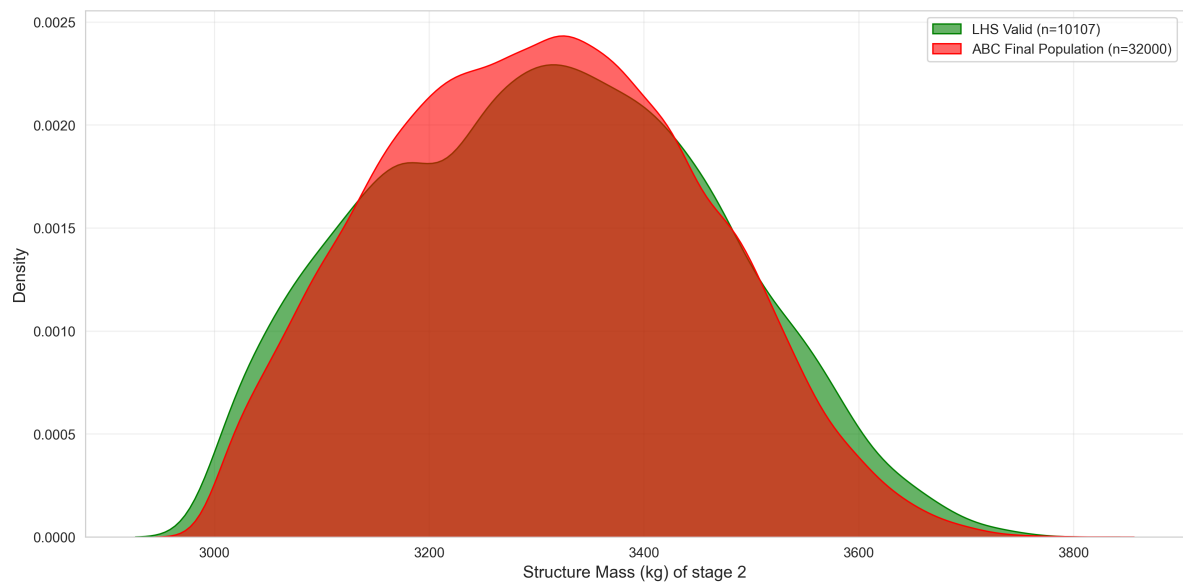


Figure 4.18: Comparison of the final distributions for one run of Latin Hypercube sampling and Approximate Bayesian Computation for the structure mass of stage 2 for test case LM2C 10U.

According to Figure 4.19, LH converges faster than ABC, and the median is slightly different again. Figure 4.20 shows the distributions for a single run for both methods.

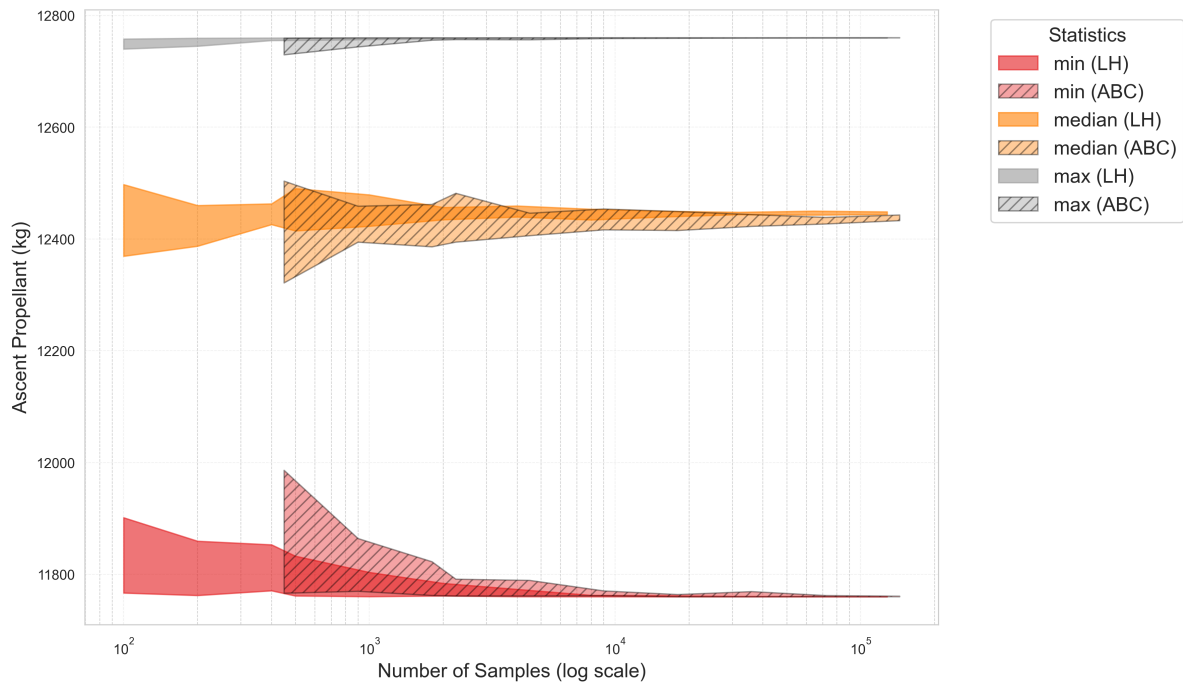


Figure 4.19: Convergence comparison of Latin Hypercube sampling and Approximate Bayesian Computation for the ascent propellant mass of stage 3 in test case LM4C 3U. The curves show the evolution of three summary statistics (minimum, median, and maximum) of the valid results as a function of the number of samples (logarithmic scale). Each simulation was repeated ten times to account for variability. The coloured regions represent the convergence bands, defined by the minimum and maximum values across the ten runs for each statistic.

Once again, they follow the same trend without perfectly coinciding. The LH distribution has its peak at a higher value than ABC, thus explaining the difference seen in the convergence bands plot.

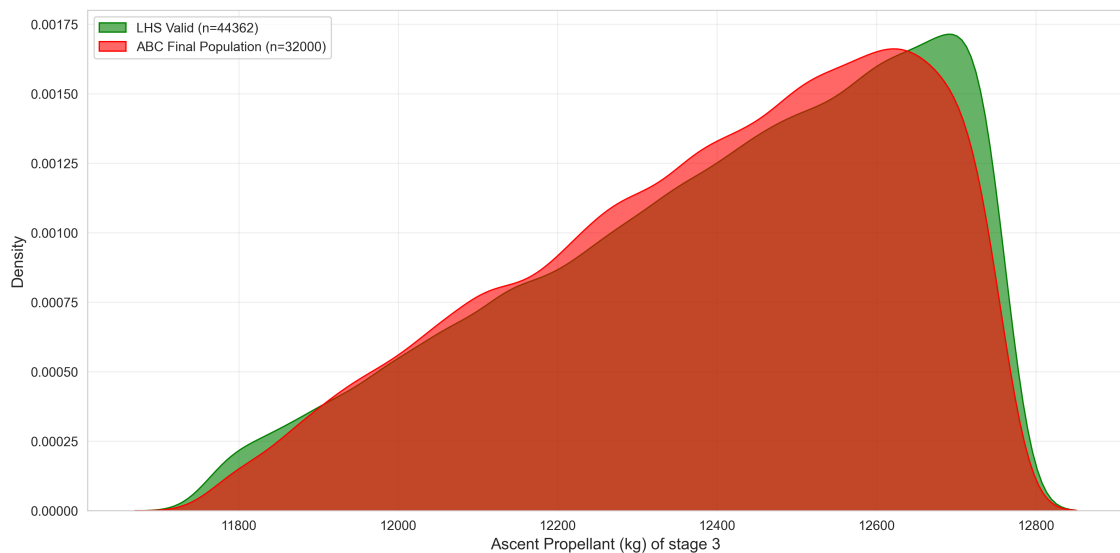


Figure 4.20: Comparison of the final distributions for one run of Latin Hypercube sampling and Approximate Bayesian Computation for the ascent propellant mass of stage 3 for test case LM4C 3U.

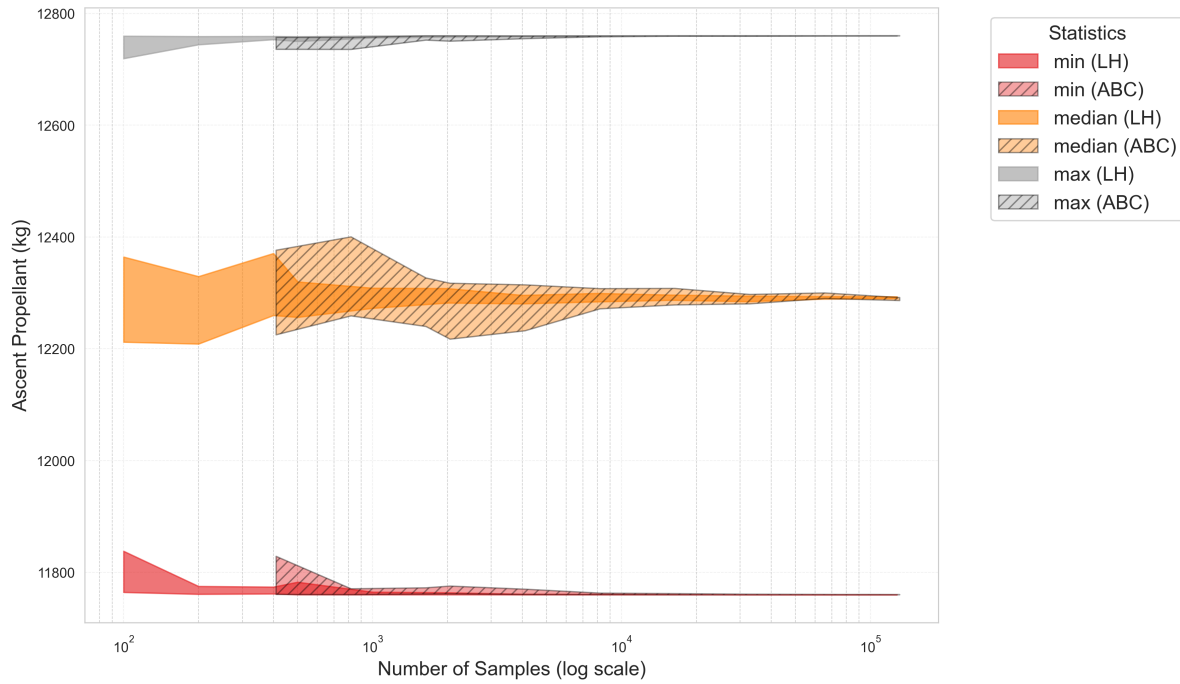


Figure 4.21: Convergence comparison of Latin Hypercube sampling and Approximate Bayesian Computation for the ascent propellant mass of stage 3 in test case LM4C 10U. The curves show the evolution of three summary statistics (minimum, median, and maximum) of the valid results as a function of the number of samples (logarithmic scale). Each simulation was repeated ten times to account for variability. The coloured regions represent the convergence bands, defined by the minimum and maximum values across the ten runs for each statistic.

Given these results, adding complexity to the LV does not seem to alter the best-performing sampling method. Still, testing the same rocket with more unknown parameters will confirm whether this is true. Therefore, test case LM4C 10U was assessed, and the results are shown in Figure 4.21. The valid range remained the same. Only one plot is shown since the remaining parameters have an identical behaviour.

The same pattern observed so far occurs again in this test case. Both methods converge towards the same values, with LH doing so faster than ABC. This happens for all summary statistics shown. Additionally, it is interesting to check how the convergence is achieved earlier for the minimum value compared to test case LM4C 3U. This is due to more unknown parameters leading to a higher range of valid values for each. This way, there will be more combinations of valid parameters, including some values rejected for the test case with fewer unknowns.

At this point, one knows that LH sampling is better both for 2-stage rockets and 3-stage rockets. Additionally, the number of unknown input parameters does not impact this result. Then the next step is to assess whether this behaviour remains for even more complex rockets, with three stages and boosters.

The LM3BE underwent an identical testing procedure as the previous LVs shown. The vacuum maximum I_{sp} of the boosters stage was chosen to be presented. Since all parameters showed a similar behaviour and relationship between both sampling

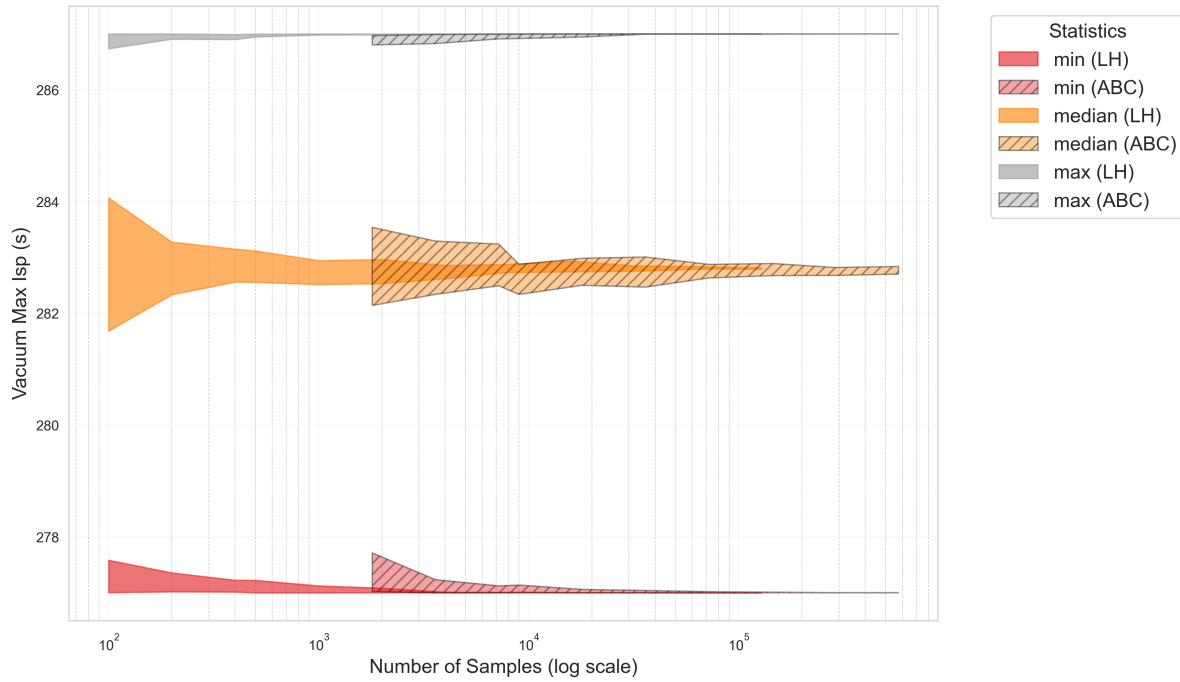


Figure 4.22: Convergence comparison of Latin Hypercube sampling and Approximate Bayesian Computation for the vacuum I_{sp} of the boosters stage in test case LM3BE 3U. The curves show the evolution of three summary statistics (minimum, median, and maximum) of the valid results as a function of the number of samples (logarithmic scale). Each simulation was repeated ten times to account for variability. The coloured regions represent the convergence bands, defined by the minimum and maximum values across the ten runs for each statistic.

methods, no other input parameter will be present in this section. Figure 4.22 and Figure 4.23 show the results for both test cases with different numbers of unknown input parameters.

In both cases, the ABC is outperformed by LH. Although the two methods converge towards the same final value for the three summary statistics, LH sampling does so faster. In addition, when comparing both plots, the final convergence value is very similar. The difference happens because the introduction of extra unknown parameters slightly changes the combinations of sampled values and, thus, the distributions of valid samples.

In conclusion, LH is a better choice for this thesis. Besides performing better, it has the advantage of allowing the post-processing to be changed. For ABC, once the simulations are complete, the method does not allow flexible adjustments to the acceptance criteria or tolerance definitions without rerunning the procedure. It only saves the valid samples for the defined validation criteria. On the other hand, LH sampling saves every sample, and then the payload ranges can be changed to validate them in different ways. Consequently, in case the simulation is first run with a wide payload range and later a more exact value is found, the post-processing can be altered in a few seconds. Finally, for LH the user can define a total number of samples based on an estimate of how long the simulation will take to run. Contrarily, for ABC the user defines the number of valid samples and one cannot know how long that will

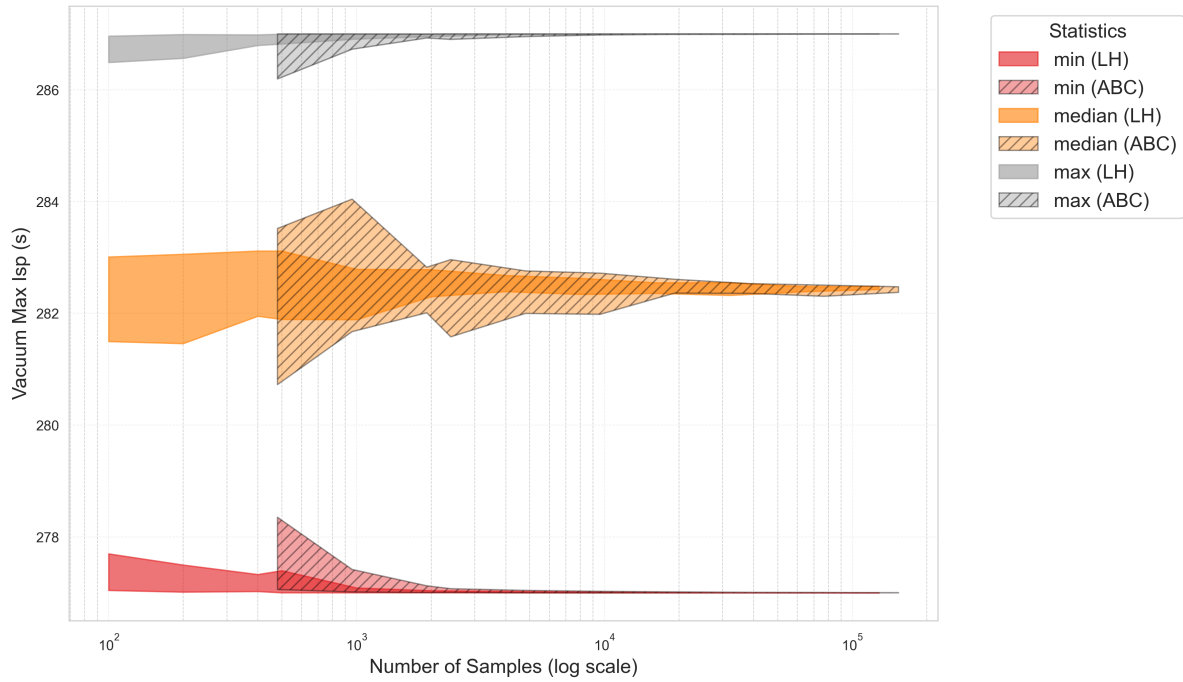


Figure 4.23: Convergence comparison of Latin Hypercube sampling and Approximate Bayesian Computation for the vacuum I_{sp} of the boosters stage in test case LM3BE 10U. The curves show the evolution of three summary statistics (minimum, median, and maximum) of the valid results as a function of the number of samples (logarithmic scale). Each simulation was repeated ten times to account for variability. The coloured regions represent the convergence bands, defined by the minimum and maximum values across the ten runs for each statistic.

take. This can ultimately lead to not complying with Req. 2b.

Now that the optimal sampling method is identified, the TM has fulfilled its role. The next step is to apply the LH sampling to the CM and obtain real results. The aim is to evaluate whether the method operates accurately and is effective across various scenarios.

4.4. Complex Model Results

Having established the most suitable statistical method for handling incomplete and uncertain input parameters, the next step is to apply the CM. This section presents the results of combining the selected statistical approach with the trajectory optimisation tool. By comparing the CM outcomes with known reference values, the reliability of the proposed methodology can be assessed under more realistic and demanding conditions.

The test cases listed in Section 3.2.6 were used again. The advantage in using them is that they have already been remodelled. Thus, by defining some input parameters as unknown and by defining a payload range that encompasses the one already achieved, one can assess whether the valid results match the complete model parameters. The results presented are the outcome of single runs and sampling procedures. Since they took one or two days to be completed, there was not enough computational power to run them ten times, as was done for the TM. The first test

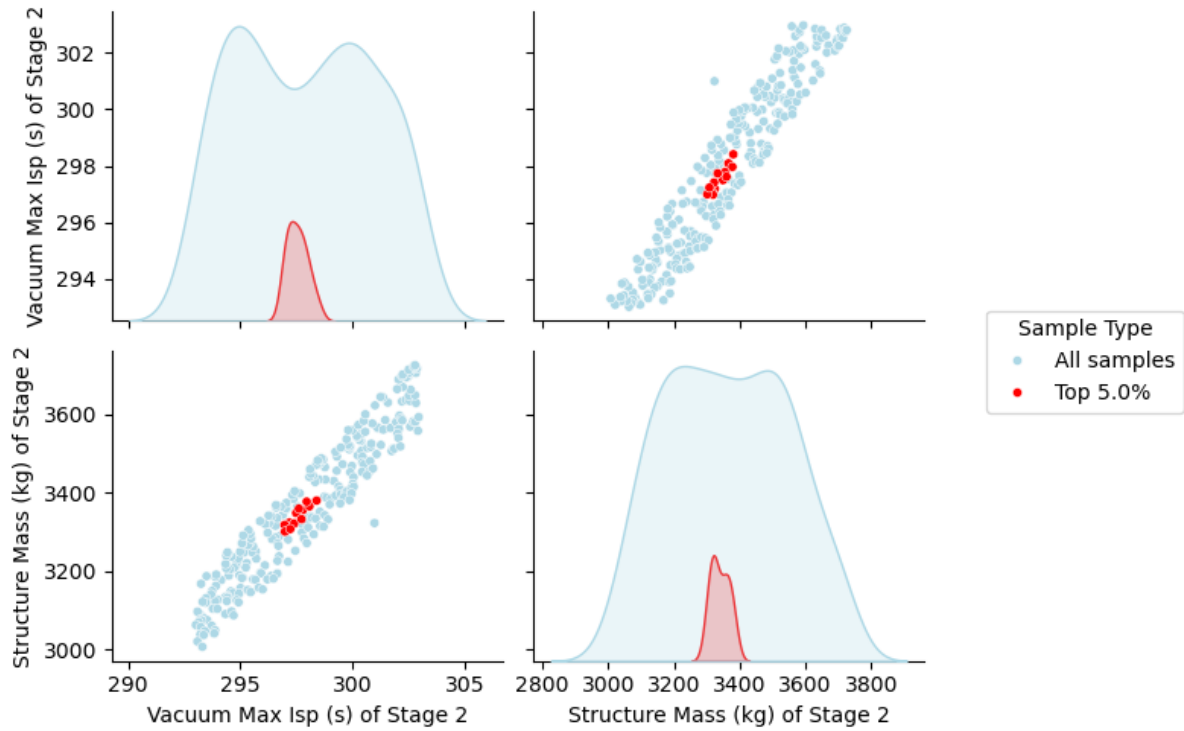


Figure 4.24: Combination of unknown parameters for all the valid samples of the Complex Model with Latin Hypercube sampling for test case LM2C 2U with a payload margin of 7%. The highlighted samples correspond to the top 5% samples with the lowest summed mean distance to the others.

case to be tested was the LM2C 2U. The results are shown in Figure 4.24.

In Figure 4.24, one can observe the unknown parameters values for each valid sample of test case LM2C 2U. In this case, there is a relative difference of 7% between the minimum and maximum payloads (payload margin) used to filter the valid samples. One can see a clear correlation between the structure mass and the maximum vacuum I_{sp} of the second stage. LH makes sure that the whole sampling space is covered, which means that the empty area corresponds to rejected samples.

The red dots represent the samples with the lowest mean Euclidean distance to all the others. Instead of a single sample, the top samples with the lowest distances were selected. The chosen percentage was 5%, but the user can change this if needed. One observes that these highlighted samples are clustered in the central region. This is expected as the use of this metric tends to disregard the outliers and the samples close to the boundaries.

For this particular LV, the payload values for the mission used in the remodelling process were found. For this reason, the final values for each unknown parameter can be compared to the real ones. However, this is not always the case. Often one cannot find a specific payload for a certain mission. Moreover, it is common to identify a value for a standard LEO or SSO mission. The problem is that when researching missions to use during the remodelling process, there are usually different LEO or SSO orbits that the rocket has flown to. Thus, it can be hard to select a fine range for the payload, ultimately leading to pre-defining a wide range. This makes the overall

framework accept wider ranges of the unknown parameters, making it hard to select the values for the final LV model. This will be discussed in more detail later in this chapter.

Going back to test case LM2C 2U, one can compare the "real" values used in the previous manual remodelling with the obtained ones (Table 4.1). They are very close. Additionally, it is important to note that the minimum payload boundary defined is approximately the same as the one obtained through the manual remodelling. This way, the validation is similar in both cases.

Table 4.1: Comparison between the obtained results and the ones which are assumed to be real and thus were used in the manual remodelling process for test case LM2C 2U.

Parameter	CM Results	"Real" Results
Vacuum Max. I_{sp} (s) of Stage 2	298	298
Structure Mass (kg) of Stage 2	3300	3288

After the functioning of LH has been proven, it was assessed if the ABC method also yields similar results to this one. So, a similar test was conducted with this sampling strategy.

The results are in Figure 4.25. After analysing the plots, it is clear that the results are very similar. Using the same initial unknown parameter ranges and the same payload boundaries, both methods have identical final parameter ranges. Even though the exact combinations of the sampled values for each parameter do not match, the results are the same. This proves that both methods work.

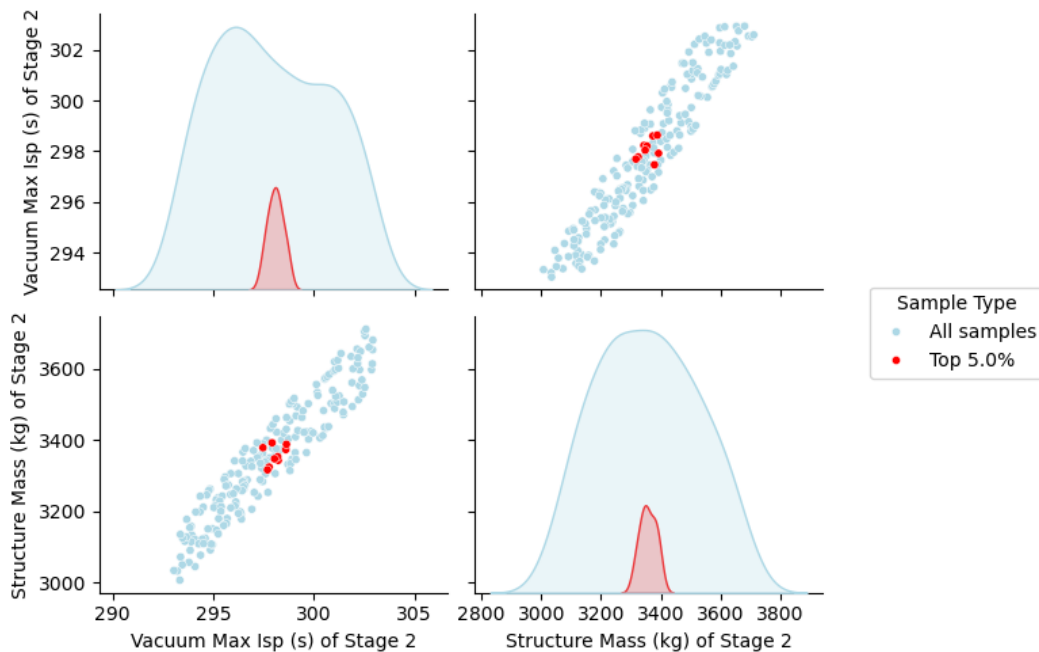


Figure 4.25: Combination of unknown parameters for all the valid samples of the Complex Model with Approximate Bayesian Computation for test case LM2C 1U with a payload margin of 7%. The highlighted samples correspond to the top 5% samples with the lowest summed mean distance to the others.

A similar reasoning to the TM was applied here. The idea was to assess, once again, whether more complex rockets or more unknown input parameters would influence the results. Given the computational resources available, testing all the test cases listed in Section 3.2.6 was not possible. Therefore, only a few of those were tested. The chosen ones were selected to try and cover as much as possible of the whole spectrum of possibilities.

The next step was to add some complexity and run the CM for test case LM4C 2U. On top of having an additional stage compared to the previous LV, this simulation was run for three simultaneous missions. This means that a sample was only valid when it obtained a payload within the pre-defined range for all missions tested. The results are available in Figure 4.26.

When analysing the results for test case LM4C 2U, there is a difference compared to the previous example. Firstly, this test case is one example for which it was hard to find a concrete payload value for the missions being assessed. The author was not confident about the most accurate payload values. Thus, instead of aiming for a narrow range, a 20% margin was used between the lower and higher value. Since the payload ranges were too wide, most of the sampled area for this test case is covered with valid samples. Consequently, almost every sample was accepted. This situation exposes one of the problems of this framework. When the payload range is not targeted enough for a certain mission, results like this can happen.

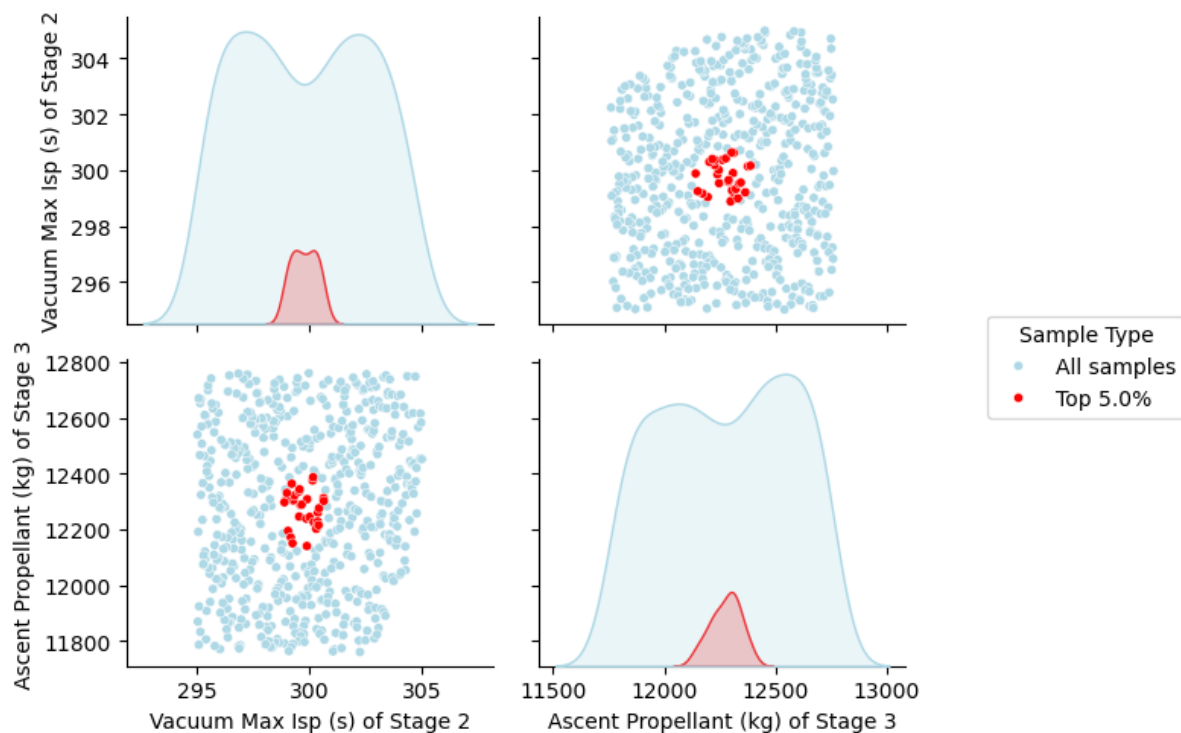


Figure 4.26: Combination of unknown parameters for all the valid samples of the Complex Model with Latin Hypercube sampling for test case LM4C 2U with a payload margin of 20%. The highlighted samples correspond to the top 5% samples with the lowest summed mean distance to the others.

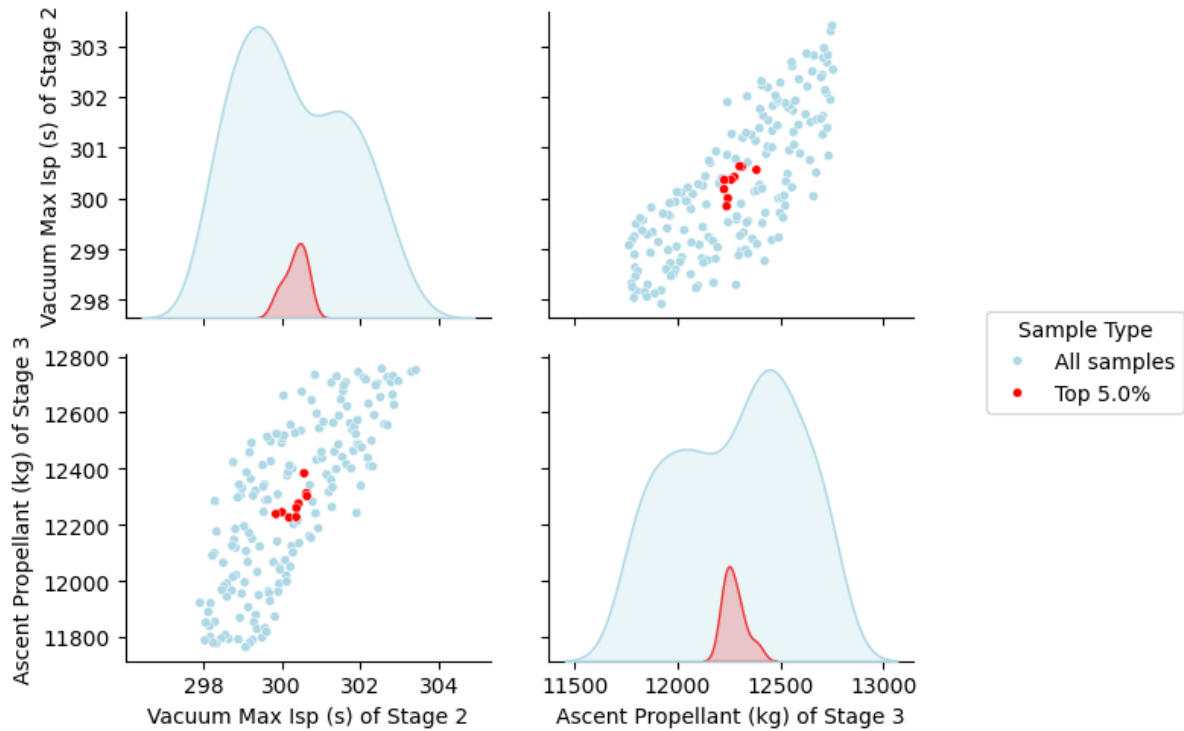


Figure 4.27: Combination of unknown parameters for all the valid samples of the Complex Model with Latin Hypercube sampling for test case LM4C 2U with a payload margin of 2%. The highlighted samples correspond to the top 5% samples with the lowest summed mean distance to the others.

However, a more selective sample validation will occur once these payload ranges are tightened. For instance, when the payload ranges are reduced to a margin of 2%, the results significantly change. Figure 4.27 shows them. The first thing to notice is how these are not scattered throughout the sampling space. There is a clear correlation between the ascent propellant and the maximum vacuum I_{sp} that yielded valid results. Given the uncertainty surrounding the payload values, the focus here was not on the final ranges for the unknown values. Rather, the author wants to highlight how the pre-defined payload range influences these results.

Figure 4.28 shows the results for the same test case but with a different payload range. By reducing the lower payload boundary by just 60 kg, the results have changed. The top valid samples in terms of minimum mean distance have shifted to lower values for the vacuum maximum I_{sp} , despite remaining identical for the ascent propellant mass. This demonstrates the influence that the pre-defined payload limits have in filtering the valid samples. A slight change can have a great impact on the results. Therefore, it is crucial to have an accurate payload value for the mission being analysed to achieve realistic results for the unknown LV parameters.

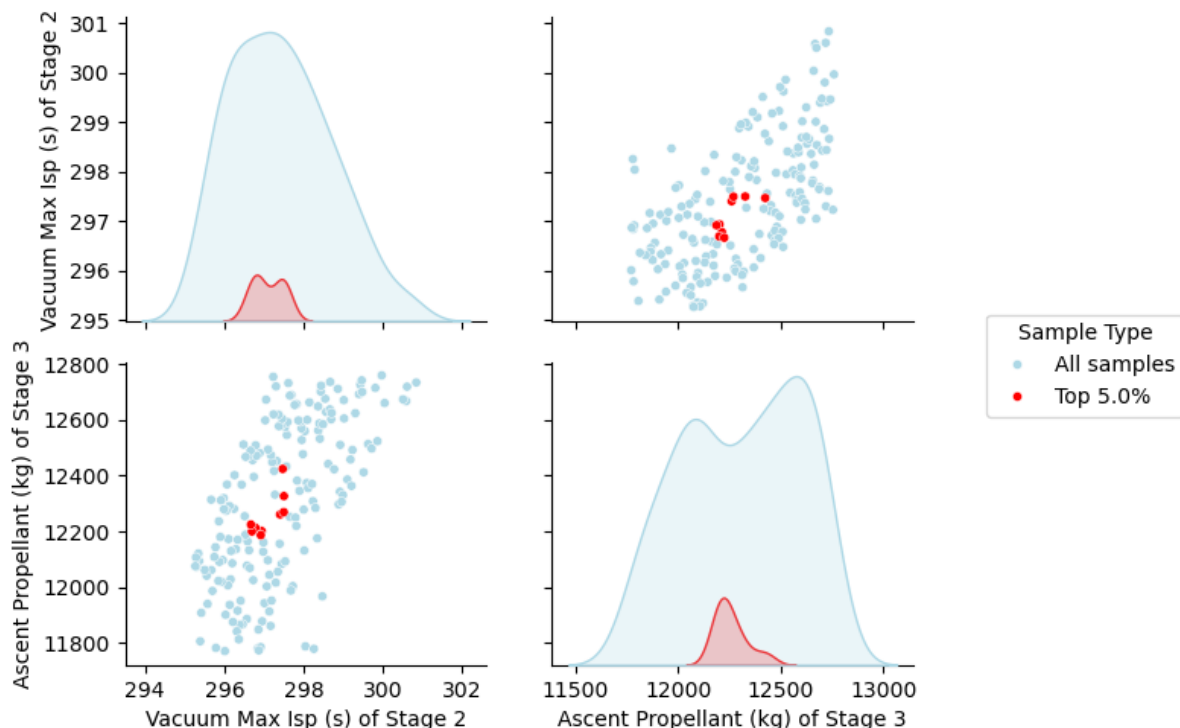


Figure 4.28: Combination of unknown parameters for all the valid samples of the Complex Model with Latin Hypercube sampling for test case LM4C 2U with a payload margin of 2% and a lower minimum payload value. The highlighted samples correspond to the top 5% samples with the lowest summed mean distance to the others.

To continue evaluating the methodology and framework, an increase in the LV's complexity was introduced. This is a critical test because adding a booster stage to a three-stage rocket results in the highest number of stages among existing LVs: four. Among the vehicles analysed by the author that consume the most propellant, none was found to have more than four stages. Therefore, the next logical step was to assess the LM3B/E.

Besides adding an extra stage, this test case dealt with five unknown input parameters. The reasoning was to explore the CM capabilities in handling more unknown parameters. Since two unknown parameters work, it was decided to verify whether this was also true with five. Figure 4.29 shows the results for this test case with a payload margin of 7%.

Similarly to what happened with the first variant of test case LM4C 2U, there are valid samples dispersed through the entire sampling region. Again, one of the reasons is that the pre-defined payload boundaries are too wide, hence allowing most or even all samples to be considered valid.

Furthermore, when there are more unknown input parameters, there is a higher chance of having parameter combinations that compensate each other and yield a valid payload result for that mission. Some parameters have a positive influence on the payload, such as the maximum I_{sp} . That is, the higher their value, the higher the payload will be, considering that the other parameters remain the same. On the

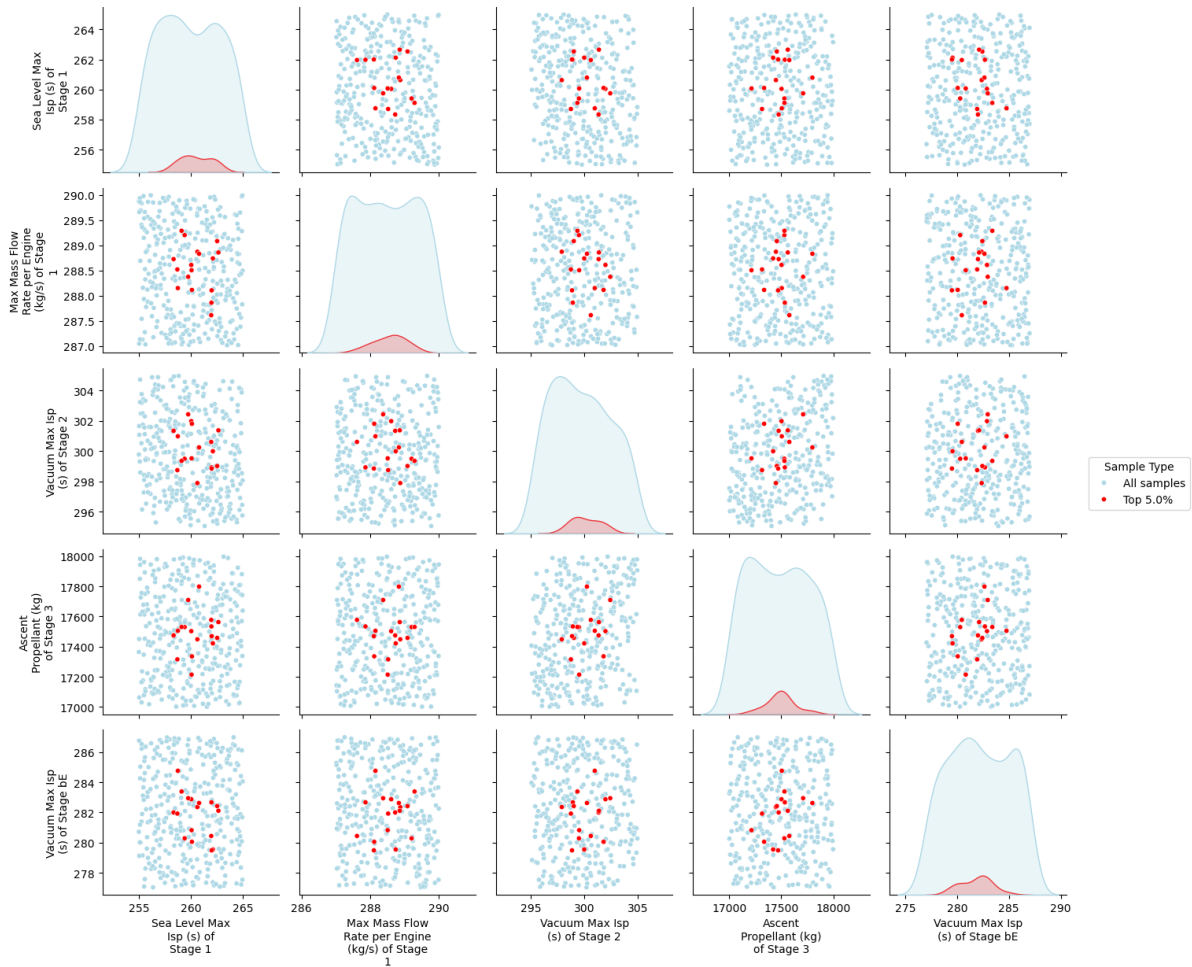


Figure 4.29: Combination of unknown parameters for all the valid samples of the Complex Model with Latin Hypercube sampling for test case LM3BE 5U with a payload margin of 7%. The highlighted samples correspond to the top 5% samples with the lowest summed mean distance to the others.

other hand, some parameters have a negative influence, as, for instance, the structure mass. This means that when multiple variables are being simultaneously sampled, their combined effect might be identical to a LV configuration with a completely different set of sampled unknown parameters.

This particular example has multiple parameters, including several maximum I_{sp} values for different stages. This leads to numerous combinations, some using higher values of one parameter, others using lower. Their combined effect results in a broad range of valid samples for all parameters. For instance, it might be that one sample has a vacuum maximum I_{sp} higher than the real value for the second stage and lower for the boosters stage. Another sample has the opposite. Both might yield valid payloads. When this effect is scaled for all parameters, the results are scattered around the sampling space. Looking at the plots in Figure 4.29, by fixing the value of a certain parameter, one can see how there are valid samples for the entire range of the others. This can be applied to all the unknown parameters. This situation is another downside of the taken approach.

However, these results were obtained with a relatively wide payload range, which provides some margin for the validation step. To further investigate this framework's capability of handling unknown parameters, it is important to check what happens with tighter payload boundaries.

Figure 4.30 shows the results when the maximum payload is reduced to only 2% more than the minimum payload. There is a significant difference, particularly in the vacuum maximum I_{sp} of the second stage. Even though the other parameters remain mostly unchanged, the valid region of this one is different. Now the valid samples are confined to part of the sampling space if the outliers are ignored. One thing to infer from this is that the vacuum maximum I_{sp} of the second stage is a very influential parameter for the payload result, similar to test case LM3BE 10U discussed in Section 4.1. Most valid samples have values in the lower half of this parameter's range. Regarding the remaining ones, either they do not have much influence, except for the ascent propellant of the third stage, which indicates a subtle correlation with the vacuum maximum I_{sp} of the second stage; or their combined contributions balance out.

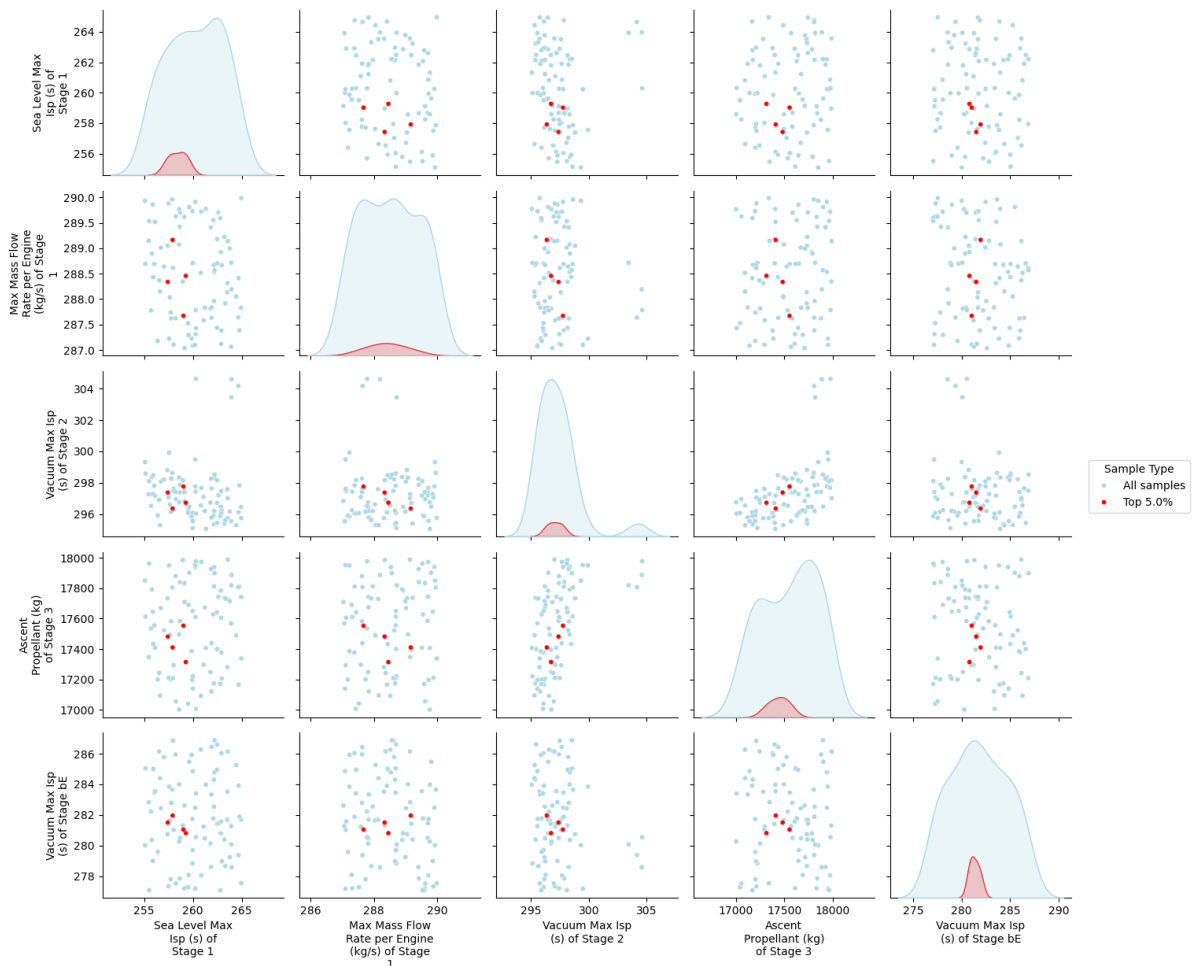


Figure 4.30: Combination of unknown parameters for all the valid samples of the Complex Model with Latin Hypercube sampling for test case LM3BE 5U with a payload margin of 2%. The highlighted samples correspond to the top 5% samples with the lowest summed mean distance to the others.

The method selected for identifying the final configurations also proves to be useful. In this test case, the outliers are ignored, since they are too distant from all the other samples. That is the advantage of using the mean distance strategy. Finally, since the payload for this particular mission was known, one can compare the "real" values to the obtained ones, using the average of the top valid results. This comparison is in Table 4.2. These results demonstrate how the CM is accurate when the payload ranges defined are narrow. Its results are all within 2% of the "real" ones. Additionally, it proves that the framework is working for five unknowns, although the final ranges for most remain broad when not focusing on the top samples.

It should be emphasised, however, that this approach strongly depends on the accuracy of the payload range. If incorrect payload values were used, artificially tight payload boundaries would lead to misleading and inaccurate results (see Figure D.3 in Appendix D).

Table 4.2: Comparison between the obtained results and the ones which are assumed to be real and thus were used in the manual remodelling process for test case LM3BE 5U.

Parameter	CM Results	"Real" Results
Vacuum Max. I_{sp} (s) of Stage 1	260	260.7
Max. Mass Flow Rate per Engine (kg/s) of Stage 1	288.5	289.6
Vacuum Max. I_{sp} (s) of Stage 2	299.8	298
Ascent Propellant (kg) of Stage 3	17483	17829
Vacuum Max. I_{sp} (s) of Boosters Stage	282.1	280

At this point, the CM is working for the most complex type of expendable LV, up to five unknowns. The final test was to increase this value and check whether the results are still accurate, to assess if there is a certain limit for the number of input unknown parameters. Consequently, test case LM2C 10U was tested and the results are in Figure 4.31. Only some of the total plots are shown, but they are representative of the general behaviour and trends found in the remaining ones.

The payload margin used was 7%. After analysing the plots, it is clear that the results are too broad. All the unknown parameters have valid samples throughout their entire sampling ranges. Even the top samples with the lowest sum of mean Euclidean distances are scattered around the plots, instead of being clustered around a specific region. This indicates that the payload range used for validation might be too wide. Another reason might be related to having such a high number of unknown parameters. With ten of them, there is an elevated probability of generating an LV configuration capable of delivering a payload within the pre-defined validation boundaries. As explained when analysing Figure 4.29, the joint effect of each parameter's positive and negative contributions produces valid configurations with varied parameter combinations.

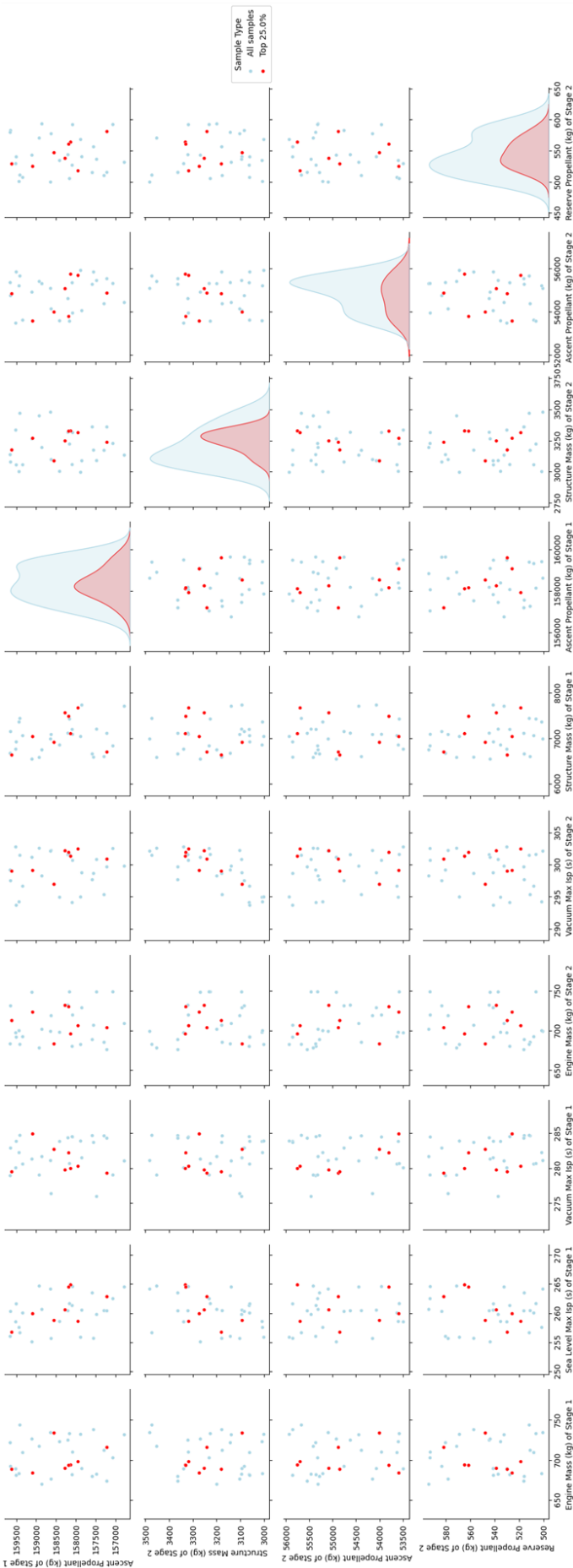


Figure 4.31: Combination of the selected unknown parameters for all the valid samples of the Complex Model with Latin Hypercube sampling for test case LM2C 10U with a payload margin of 7%. The highlighted samples correspond to the top 25% samples with the lowest summed mean distance to the others.

To further assess whether this test case with ten unknowns could be accurate, the margin between the maximum and minimum payload values was decreased. This way, the validation process would be more selective. However, even reducing this margin to 2.5% revealed to be insufficient to achieve concrete results (see Figure D.4 in Appendix D).

Based on these results, having ten initial uncertain or unknown parameters is not a good option if the user wants accurate or narrow final ranges for them. With so many uncertainties, the probability of combining all parameters to create valid LV configurations increases. Thus, the framework is not able to sufficiently narrow down the initial ranges of the unknown parameters. Given the test cases used during this work, five uncertain input parameters would be the limit.

4.4.1. New Test Case - Long March 7

This section presents a new test case that differs from the previous ones. Unlike the earlier LVs, which had already been remodelled and could therefore serve as reference points, this rocket is considered here for the first time. As such, it provides a valuable opportunity to demonstrate how the framework performs when applied to a new launcher. This makes the case particularly relevant for assessing the framework's capability to handle uncertain input data in a real case situation, where no validated baseline exists. Therefore, the results obtained not only extend the test set but also indicate how the methodology could be used in practice for new rockets.

The LV in the analysis is the LM 7. It is a two-stage rocket with boosters, which is used to launch a spacecraft to resupply the *Tiangong* Space Station [92]. For this reason, it has consistently flown to approximately the same orbit. Therefore, the payload it can carry to this orbit is well-known: around 14000 kg.

After researching this LV to determine all the input parameters, two of them were uncertain: the structure mass of the second stage and the engine mass of the second stage. Thus, a range was assigned to each one of them. Afterwards, the CM was run, and the results are in Figure 4.32.

First of all, it is important to note that the payload range for this mission was very tight because this LV's mainly flown mission is well-documented. Hence, a minimum payload of 13950 kg was used with a maximum value 1% higher. This margin is useful to accept tiny deviations, as selecting only a specific value would be hard to match precisely. By contrast, the input parameters ranges were wide relative to their absolute values. There was very little publicly available data, so it was difficult to define narrower bounds with confidence.

The results highlight how the framework deals with these uncertainties. The two unknown parameters jointly define the stage's dry mass. As shown in the plot, a large portion of the parameter space is empty. This reflects how the framework accurately filtered out infeasible samples. The valid samples are those that successfully achieved the intended payload capacity, meaning that they correspond to configurations that could realistically represent this LV. Among these, the highlighted points indicate the solutions with the lowest mean distance. They could be used for the S3D exhaust inventory.

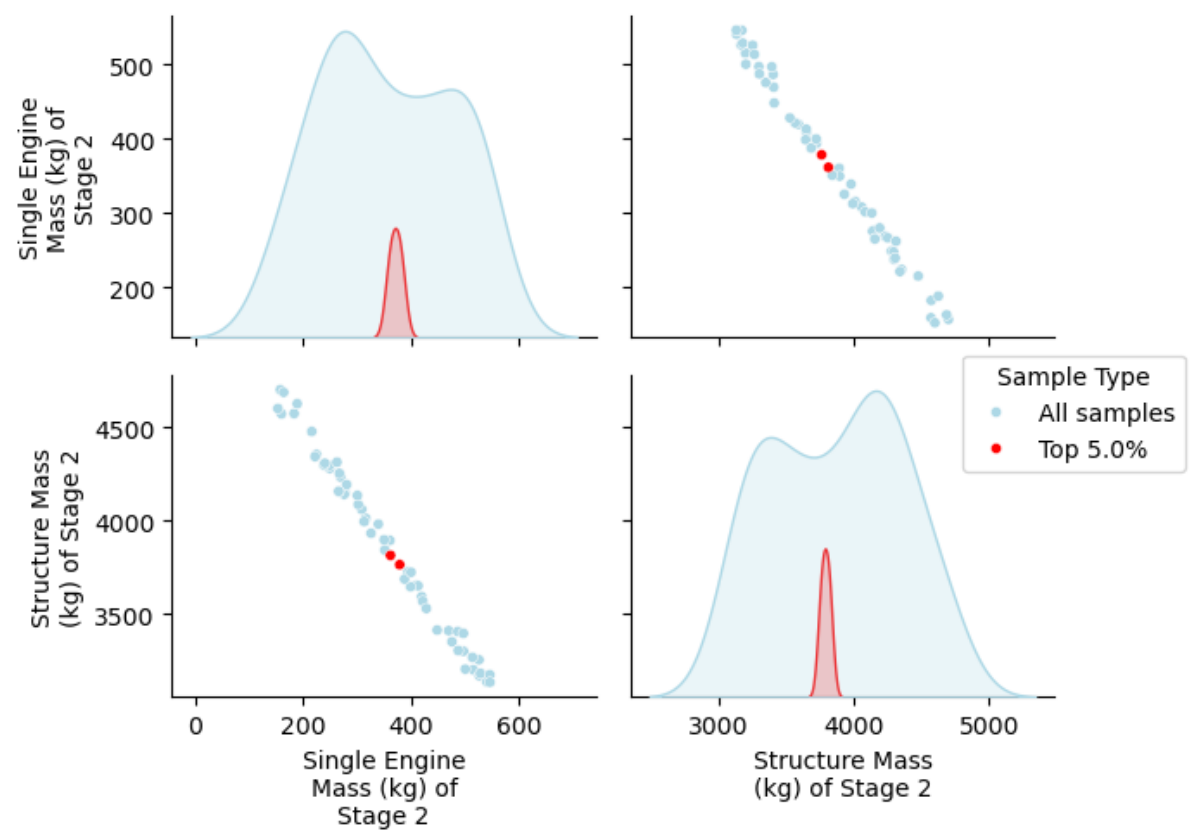


Figure 4.32: Combination of unknown parameters for all the valid samples of the Complex Model with Latin Hypercube sampling for the Long March 7 with a payload margin of 1%. The highlighted samples correspond to the top 5% samples with the lowest summed mean distance to the others.

Overall, this test case demonstrates that the framework can accurately remodel a new launcher even with limited and uncertain data. The ability to generate valid configurations under such conditions illustrates the reliability of the methodology, strengthening its potential to be extended to any expendable LV.

4.5. Concluding Discussion

This section summarises the results presented in the previous sections and discusses their implications in the context of the research objectives. While the initial part of this chapter focused on presenting findings and interpreting them within their specific scope, this section consolidates these insights and addresses the research questions. By doing so, it reinforces the coherence of this thesis and describes the contribution of each step to the overarching aim of the work.

Before answering the research questions, it is important to check whether all the requirements were fulfilled. Table 4.3 summarises this analysis and shows that all requirements are met, with the corresponding sections in which they are validated.

Table 4.3: Requirements with the corresponding sections where their VV is mentioned.

ID	Requirement Summary	Section Reference
1 (a)	Framework with two modelling paths.	Section 3.1.3
1 (b)	IDS identifies missing parameters.	Section 3.2.1
1 (c)	IDS computes dependent parameters.	Section 3.2.1
1 (d)	IDS finds inconsistent parameter combinations.	Section 3.2.1
1 (e)	IDS has all parameters required by the tools.	Section 3.2.1
1 (f)	IDS is flexible for all expendable LVs.	Section 3.2.1
1 (g)	IDS handles exact values and parameter ranges.	Section 3.2.1
1 (h)	Framework stores all relevant results for reuse.	Chapter 4
1 (i)	Architecture supports future extensions.	Section 3.2.1
2 (a)	TM runtime below 20 seconds.	Section 3.1.3
2 (b)	CM runtime below 48 hours.	Section 3.2.4
3 (a)	IDS provides a user-friendly interface.	Section 3.2.1
3 (b)	Input data readable by Python code.	Section 3.2.1
3 (c)	User interaction minimised during execution.	Section 3.2.4
4 (a)	User visualises results for both models.	Section 3.2.4
4 (a) i	User can access all valid configurations.	Section 3.2.4
4 (a) ii	Algorithm updates IDS with final data.	Section 3.2.1
5 (a)	Framework validates outputs for feasibility.	Section 3.3
5 (b)	User can choose selection criteria.	Section 3.2.4
5 (c)	System includes sensitivity analysis.	Section 4.1

Then, the main research question and sub-questions will be answered in turn. For each, the corresponding findings are outlined, and their significance is critically examined. In addition, the strengths and limitations of the proposed framework are considered.

Starting with the first research question, regarding the challenges of having uncertain input data, the results of this thesis confirmed that incomplete or uncertain data introduce some problems. The first is the variability of data availability across LVs. While some rockets are well documented, for many, one can only find parts of the technical data. Additionally, there are cases for which multiple sources indicate different values for the same input parameter. In any case, these scenarios require a flexible framework to accommodate different levels of detail and to operate with ranges of unknown values. The IDS developed in this thesis directly addresses this issue by allowing the user to input a specific parameter or a range to be explored when an exact value cannot be found. Thus, it can deal with different levels of data availability. Besides, due to the way it was built, the IDS can be used for all expendable LVs.

A second challenge lies in the propagation of the uncertainties. Small deviations in stage mass or engine performance can significantly influence payload estimates, as discussed in Section 4.1, or trajectory feasibility. The remodelling results risk becoming unreliable without a mechanism to quantify and manage this uncertainty.

Here, the integration of statistical sampling methods was essential, allowing ranges of unknown parameters to be methodologically explored and validated.

Finally, the results highlight the computational burden of managing incomplete data. Running thousands of simulations is expensive, even more when using complex trajectory optimisation. For instance, each plot like Figure 4.14 required around two or three days of simulations with the current computational resources. For each test case run with the CM, it took around two days to be completed, using only one or two thousand samples. This required a detailed plan for a strategy to approach the testing phase, as testing every possibility was not possible. This justified the choice of having test cases with increasing complexity. The goal was to cover as many scenarios as possible with as few computational resources as possible. Still, some more testing is recommended to improve this framework's efficiency and reliability, but that will be discussed in Section 4.6. Still, all the research done regarding statistical methods was useful. It allowed the author to find the most efficient strategy to explore the pre-defined ranges for the uncertain parameters within the available computational power.

All these challenges together impose certain limitations on the framework. One cannot have an LV with ten unknown input parameters because the results are not sufficiently accurate. Looking at Figure D.4, the whole uncertain range was filtered as valid for most unknowns. This means that the framework was not successful in identifying a realistic LV configuration, not even for a narrow payload range, as desired. The validation strategy using payload boundaries has proven reliable for most test cases when these boundaries are tight. However, beyond a certain number of unknown input parameters, their combined effect generates valid configurations across most of the sampling space. Due to limited computational resources, it was not possible to test every possible number of unknown input parameters. Nonetheless, test cases up to five unknown parameters yielded narrow results when the payload range was below 3%.

Considering this, five is the maximum number of uncertainties needed to achieve a final valid range for each parameter that differs from the entire sampling space. However, the results have shown that the accuracy of this framework heavily depends on how broad the pre-defined payload ranges are. Some cases with only two uncertain input parameters produce results scattered throughout the entire sampling region. Although precise outcomes were not obtained for more than five unknown input parameters, the payload range must still be kept narrow for LVs with more available data.

Regarding the second research question, about creating the various models without accurate data, the findings demonstrate that even with incomplete information, reliable LV models can be built. As seen with the LM 2C and the LM 3B/E, the obtained results are within 2% of their real values when the payload range is known and below 3%. Most of them were even within 1%. Compared to the traditional uncertainty associated with LV remodelling, which is around 5% to 10%, it is a great improvement. Still, only expendable liquid rockets were analysed, as it was not possible to test other types of LVs (more details about this in Section 4.6). Thus, the remodelling framework

developed in this thesis can make mass and aerodynamic models from uncertain input parameters, by pinpointing the initial uncertain range towards the realistic value.

Sensitivity analyses revealed that not all parameters contribute equally to the outcome. A subset of parameters, such as structural mass or vacuum maximum I_{sp} , showed a much higher influence on payload capability than others. Another conclusion was that if a parameter has a relatively large uncertainty range, its influence is more noticeable on the final result. The same occurs for parameters which are relatively important when considering the terms of Equation 3.11. As a result, the more impactful parameters will filter themselves out, as they will cause a higher variation in the payload. This means that the user should particularly focus on finding close boundaries for the less sensitive parameters. These will likely have a wide valid range, even for tighter payload boundaries.

Finally, the integration with the CM preserves the feasibility of the trajectories. The framework ensures that mass, aerodynamic, and trajectory models remain realistic, even when based on incomplete input data.

Concerning the third research question, about the best statistical sampling method for this remodelling framework, the comparative testing procedure showed clear differences in their performance. LH sampling proved to be the most efficient one. In the end, both methods work and yield similar results. However, ABC converges more slowly, requiring more samples to achieve the same accuracy. It is also not possible to predict the total number of samples required for the ABC to converge, as one defines the number of valid samples. Thus, the simulation might run for longer than the established limit of 48h (Req. 2b). Finally, as explained in Section 4.3, LH enables the user to change the post-processing by, for instance, redefining the payload range. This is particularly useful to remodel newer LVs, with less publicly available information. As they are more frequently launched, more data can be gathered, and a tighter payload range can be defined. Therefore, even if it was initially wide, it can be narrowed down later to yield more realistic results. For these reasons, LH sampling is the best statistical method for any type of LV and for any number of uncertainties. Thus, it is the best-suited method to define the possible ranges of unknown parameters.

The last research question, regarding overall accuracy, has already been partially answered. The whole framework, starting with the IDS, was developed to be as flexible as possible. The main principle behind it was to have a final product able to handle varying data availability and complexity levels. The test cases confirm that the results can be accurate even for LVs with poor data availability.

Together, the research questions demonstrate that the proposed framework successfully addresses the central goal of automating the LV remodelling process and handling incomplete or uncertain data. By structuring the workflow around a flexible IDS, dedicated statistical methods, and an overarching rocket trajectory optimiser, the framework ensures reliability and accuracy even when some input parameters are missing or uncertain. Thus, it provides a novel and practical solution to the main research question. Still, a few limitations remain that should be acknowledged.

Firstly, estimating the payload before the remodelling process is difficult. In cases

where only generic payload capacities are available (e.g., “to LEO”), it is unclear whether this refers to a lower-altitude or higher-altitude orbit. Such ambiguities can significantly affect the outcomes. As seen in Section 4.4, wide payload ranges negatively impact the accuracy of the results. Finding the exact payload to a certain mission is challenging, and it can complicate the remodelling process, particularly if there are around five or more unknown input parameters.

Moreover, the computational effort must be considered. The number of samples chosen strongly influences the runtime. In addition, if multiple missions are selected for the same LV, the simulation time is scaled up by a factor equal to the number of missions. With the available computational resources, 96 cores, the nominal runtime for a single rocket is between one and two days. With an everyday laptop with four cores, this should take more than 20 times longer than the current duration. Using a supercomputer, for instance, with 96 cores twice as fast as the available ones, could improve this run-time by half. To sum up, with the current computational power available, complex LVs (with four stages) must be limited to one mission.

Finally, this framework’s implementation is still not overarching enough. Currently, it only works for expendable LVs with liquid propellant stages without a thrust profile. Developing this remodelling tool was quite complex, so the starting point was defined for this kind of LV. However, the whole framework is ready to handle other rockets, as long as the necessary changes are made.

4.6. Future Work

Based on the limitations identified above, there is room to further enhance the performance and applicability of this framework.

Firstly, improvements to the TM should be considered. While it fulfilled its role as a quick environment for testing statistical methods, its physical accuracy could be refined. In particular, modelling the losses (gravitational, drag, and thrust) requires deeper investigation to find better correlations with the input parameters. Similarly, the averaging of the boosters I_{sp} and the instant of the fairing jettisoning could be refined to better capture their effects on the Δv budget. These enhancements would increase the TM’s reliability. They might even strengthen its role as a potential pre-filtering tool for the CM. If parts of the initial unknown ranges could be removed with a very small computational effort by using the TM with a larger tolerance, the CM could be used for a narrower region, thus requiring fewer samples and time.

Secondly, this remodelling framework should be expanded for any expendable LV, as a first step, and subsequently to reusable LVs. Even though it is limited to liquid propellant rockets, this is an important first step. It allowed for the framework’s validation. The next step is adding capabilities to deal with different LVs. This has been taken into account from the beginning, nonetheless. The IDS already has fields for solid rocket motors and thrust or mass flow rate profiles. Also, CLAVA and the UQ files need a new structure within the engine object to handle thrust profiles. Finally, TOSCA and STSM are prepared to deal with any kind of LV, so the only thing to do is to make the adapter read this extra information from CLAVA and pass it to the SART Toolbox files. The last step to do is to run test cases to validate this improved

framework. Regarding reusability, the framework does not have anything planned yet, as this was too complex to be considered during such an initial phase of this work. Lastly, still regarding the SART Toolbox, an engine model could be created for each LV. This details the engine's functioning and helps further understand its behaviour. It was not included in this thesis, but it is another functionality that could be seamlessly added.

Thirdly, the limited computational resources led to an optimised strategy during the testing phase. The author tried to test as many scenarios as possible, but this implied some trade-off. For instance, not every number of unknown parameters was tested. This means that five might not be the real limit for an accurate result. In addition, this number may vary depending on the parameters used as uncertain. With five parameters with an insignificant impact, maybe the valid ranges for each unknown will be as wide as their initial ones. Since not all combinations of unknown input parameters were tested, this problem could not be solved. Furthermore, the SA could not be performed as thoroughly as desired. Results with higher sample counts and more test cases would be interesting to see with the CM. This is recommended to be made in the future, as it might help reduce the confidence intervals and improve the assessment of each parameter's influence. Overall, some more testing should be performed to sharpen this framework's accuracy and extract its maximum performance.

Lastly, the framework could benefit from better strategies for handling payload uncertainties. Currently, relying on generic payload values (for example, 'to LEO') introduces ambiguity that can jeopardise the results. Investigating ways to estimate or constrain these payload ranges more rigorously would improve the reliability of the results. Perhaps one could implement a reinforcement learning model trained on known vehicles or missions. In addition, the same model could be applied to estimate ranges for unknown parameters based on similar LVs. It would use a database with existing models to learn similarities between rockets with the same number of stages, propellants, payload capabilities, etc. Then, once a new LV was desired, this algorithm could suggest values or ranges for each unknown parameter based on LVs with similar characteristics. This could accelerate the remodelling process while yielding even more realistic results, as it would narrow down the uncertain ranges.

Despite these constraints, the framework was able to provide meaningful answers to all the research questions. This demonstrates that the central problem of automating LV remodelling while handling incomplete or uncertain input data can be addressed effectively, even if certain practical challenges remain. Thus, this framework has the baseline to start creating new rocket models, to improve the existing S3D results that only consider the LVs already remodelled, as in Figure 4.33.

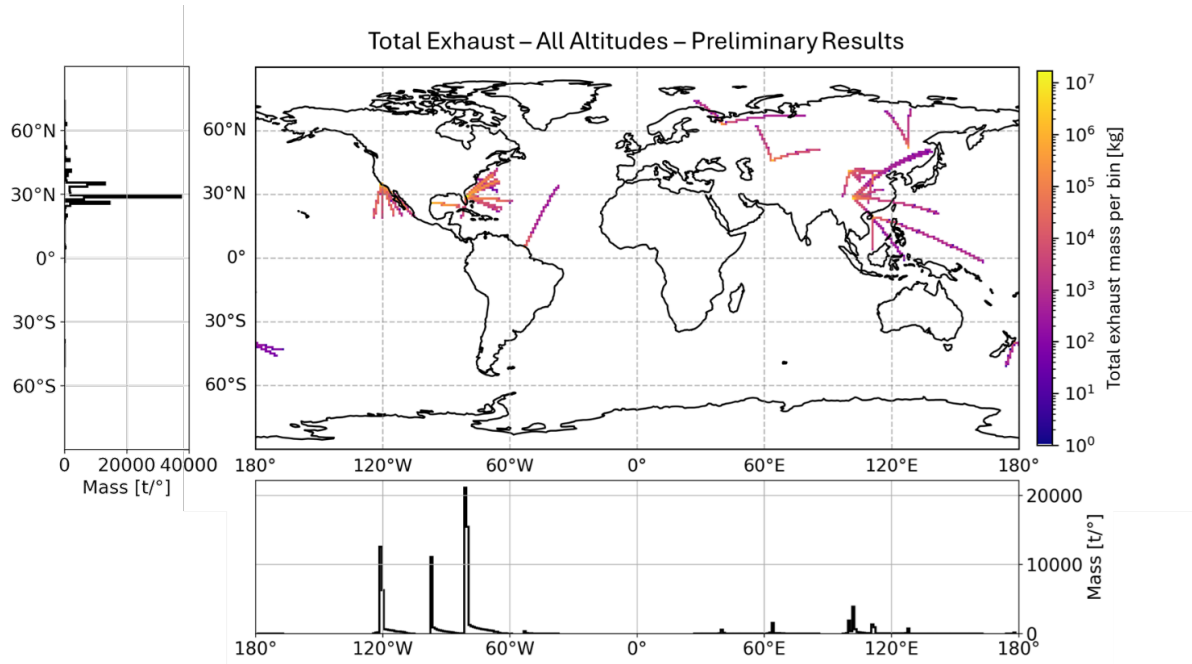


Figure 4.33: Preliminary results of rocket emissions in 2024 by latitude and longitude using remodelled launch trajectories (retrieved from [93]).

5

Conclusion

This thesis presents the development of a semi-automated framework for launcher remodelling that handles uncertain data. The framework integrates a flexible input data sheet, statistical estimation methods, and trajectory modelling tools to automate the creation of mass and aerodynamic models while validating optimised trajectories. Through this work, it was demonstrated that the remodelling process can be significantly accelerated while still achieving reliable results, even when the available data is limited or inconsistent.

Its main contribution is to bridge two previously separated research domains: rocket remodelling and statistical estimation of unknown input parameters. Existing remodelling approaches use complete technical datasets, while statistical methods such as Monte Carlo, Latin Hypercube, and Approximate Bayesian Computation have mainly been applied in unrelated fields. By combining these two domains, this thesis addressed a clear research gap and delivered a methodology capable of handling realistic scenarios, where technical parameters are often missing or contradictory.

The framework was successfully tested in multiple case studies. That showed that applying statistical methods to estimate unknown parameters results in feasible and validated launch vehicle configurations. Consequently, the thesis provided a useful tool for launcher analysis and new insights into the role of statistical sampling in the aerospace sector.

The framework offers several significant benefits. Its modular design ensures flexibility, allowing it to handle rockets with different numbers of stages and varying levels of input data availability. This is considered from the interactive input data sheet up to the trajectory optimisation tool. Also, once validated against a narrow payload range, the Complex Model provides accurate and realistic results, within 2% of the real ones. The post-processing stage is easily adaptable, giving users control over selection criteria and validation thresholds. In addition, the workflow reduces the dependence on manual work while keeping the process easily understood by other users apart from the author. Overall, the framework is a practical tool that can be adapted, extended, and used to build more realistic launcher models.

Nevertheless, the framework has some limitations. Currently, it is restricted to expendable launch vehicles with liquid propulsion, excluding hybrid, solid, and reusable systems, which nowadays make up a significant portion of launch activity. Moreover, it demands a lot of computational power that is not easily accessible. Even when available, it takes around two days of simulations to produce a valid launch vehicle model. The number of unknown parameters it can handle at once is also limited to five. For more than that, the accuracy decreases. Additionally, the framework requires a relatively narrow payload range, below 3%, to validate configurations. If only generic payload data (e.g., “to LEO”) is available, the results become less reliable. Lastly, while it is useful for testing statistical methods, the Test Model relies on simplifying assumptions for losses, fairing jettison, and propulsion averages, which limit its predictive power as a pre-filtering tool. All of these issues are areas where future refinement is possible.

Beyond the technical scope, this thesis’s relevance extends to broader questions of sustainability in spaceflight. As highlighted in the introduction, the environmental impact of rocket exhaust is still poorly quantified. Yet, it represents a potentially critical driver of atmospheric change [3–8]. Projects such as the S3D depend on accurate launch emission inventories to assess emissions at different altitudes. To create them, it is vital to have launcher models as realistic as possible. By making the remodelling process faster, more reliable, and less dependent on manual work, this thesis directly contributes to the development of these comprehensive inventories.

In summary, although challenges remain, this work demonstrates that it is possible to automate the remodelling of launch vehicles and design it to handle uncertainty. Therefore, it improves the capability to assess the environmental footprint of spaceflight. The launcher models resulting from this framework will be more realistic than the current ones, enabling a better evaluation of their exhaust and its consequences. Once a complete global launch emission inventory is built, some conclusions will arise such as, for example, the propellant that causes the least environmental impact. Thus, as launch activity continues to grow, this framework will play an essential role in enabling more sustainable space transportation activities.

References

- [1] J. C. McDowell. *BGCAT: General Catalog of Artificial Space Objects*. URL: <https://www.planet4589.org/space/gcat/web/launch/ldes.html> (visited on 12/06/2024).
- [2] Jascha Wilken et al. *DLR Initiative S3D: Advancing Space Sustainability and Sustainable Development*. Mar. 2025. DOI: <https://doi.org/10.5194/egusphere-egu25-9526>.
- [3] Christopher M Maloney et al. “The Climate and Ozone Impacts of Black Carbon Emissions From Global Rocket Launches”. In: *Journal of Geophysical Research: Atmospheres* 127.12 (June 2022), e2021JD036373. DOI: 10.1029/2021jd036373.
- [4] Christopher M. Maloney et al. “Investigating the Potential Atmospheric Accumulation and Radiative Impact of the Coming Increase in Satellite Reentry Frequency”. In: *Journal of Geophysical Research: Atmospheres* 130.6 (2025). DOI: <https://doi.org/10.1029/2024JD042442>. eprint: <https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2024JD042442>. URL: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2024JD042442>.
- [5] R. G. Ryan et al. “Impact of Rocket Launch and Space Debris Air Pollutant Emissions on Stratospheric Ozone and Global Climate”. In: *Earth’s Future*. June 2022. URL: <https://doi.org/10.1029/2021EF002612>.
- [6] J.A. Dallas et al. “The environmental impact of emissions from space launches: A comprehensive review”. In: *Journal of Cleaner Production* 255 (2020), p. 120209. ISSN: 0959-6526. DOI: <https://doi.org/10.1016/j.jclepro.2020.120209>. URL: <https://www.sciencedirect.com/science/article/pii/S0959652620302560>.
- [7] Laura E. Revell et al. “Near-future rocket launches could slow ozone recovery”. In: *npj Climate and Atmospheric Science* 8.1 (2025). DOI: 10.1038/s41612-025-01098-6.
- [8] Tyler F. M. Brown, Michele T. Bannister, and Laura E. Revell. “Envisioning a sustainable future for space launches: a review of current research and policy”. In: *Journal of the Royal Society of New Zealand* 54.3 (2023), pp. 273–289. DOI: 10.1080/03036758.2022.2152467.
- [9] Moritz Herberhold, Jascha Wilken, and Martin Sippel. “Impact of Engine Cycles, Propellants and Recovery Methods on the Launch Exhaust Profile of Reusable Booster Stages”. In.

- [10] Moritz Herberhold. *Overview of DLR Initiative S3D: Space Sustainability and Sustainable Development Clean Space Days 2024*. URL: https://indico.esa.int/event/516/contributions/9936/attachments/6264/10976/CSD2024_Herberhold_OverviewDLRInitiativeS3D.pdf.
- [11] Moritz Herberhold. *Private communication*. Personal communication, March 2025. 2025.
- [12] *History of DLR*. URL: <https://www.dlr.de/en/dlr/about-us/history-of-dlr> (visited on 01/29/2025).
- [13] DLR. URL: <https://www.dlr.de/en/dlr/about-us>.
- [14] DLR. 2025. URL: <https://www.dlr.de/en/latest/news/2025/dlr-and-nasa-continue-joint-space-radiation-research-with-dlr-radiation-detector>.
- [15] DLR. *Space Launcher Systems Analysis Department*. 2020. URL: <https://www.dlr.de/en/irs/about-us/departments/space-launcher-systems-analysis>.
- [16] DLR. *DLR Bremen, main building*. 2022. URL: <https://www.dlr.de/en/irs/images/dlr-bremen-main-building-1>.
- [17] Xinfu Liu. “Fuel-Optimal Rocket Landing with Aerodynamic Controls”. In: *Journal of Guidance, Control, and Dynamics* 42.1 (Jan. 2019), pp. 65–77. DOI: 10.2514/1.g003537.
- [18] Yin DIAO et al. “Energy management and trajectory optimization technology for rocket landing”. In: *Acta Aeronautica et Astronautica Sinica* 45.15 (2024), p. 229634. DOI: 10.7527/S1000-6893.2024.29634. URL: <https://www.sciope.n.com/article/10.7527/S1000-6893.2024.29634>.
- [19] Alexander T. Miller and Anil V. Rao. *End-to-End Ascent-Entry Mission Performance Optimization Using Gaussian Quadrature Collocation*. 2021. arXiv: 2104.12296 [math.OC]. URL: <https://arxiv.org/abs/2104.12296>.
- [20] Sholto O. Forbes-Spyratos et al. “Trajectory Optimization of a Partially Reusable Rocket–Scramjet–Rocket Launch System Including Fly-Back”. In: *Journal of Spacecraft and Rockets* 60.3 (May 2023), pp. 779–796. DOI: 10.2514/1.a35535.
- [21] Vishnu Suresh Nair and Aravind Vaidyanathan. “Ascent trajectory design and optimization of a two-stage throttleable liquid rocket”. In: *Advances in Space Research* 69.12 (2022), pp. 4358–4375. ISSN: 0273-1177. DOI: <https://doi.org/10.1016/j.asr.2022.03.023>. URL: <https://www.sciencedirect.com/science/article/pii/S0273117722002095>.
- [22] Lorenzo Federici et al. *Integrated Optimization of Ascent Trajectory and SRM Design of Multistage Launch Vehicles*. 2019. arXiv: 1910.03268 [math.OC]. URL: <https://arxiv.org/abs/1910.03268>.
- [23] Jaeyoul Ko et al. “Simultaneous Optimization of Launch Vehicle Stage and Trajectory Considering Flight-Requirement Constraints”. In: *International Journal of Aeronautical and Space Sciences* 25.4 (May 2024), pp. 1563–1573. ISSN: 2093-2480. DOI: 10.1007/s42405-024-00737-1. URL: <http://dx.doi.org/10.1007/s42405-024-00737-1>.

- [24] Emre Ünal. "LAUNCH VEHICLE DESIGN, MODELING AND TRAJECTORY OPTIMIZATION". MA thesis. Istanbul technical University, 2021.
- [25] Ezgi Civek. "Multistage Launch Vehicle Design with Thrust Profile and Trajectory Optimization". PhD thesis. Middle East Technical University, 2014.
- [26] Lâle Evrim Briese, Klaus Schnepfer, and Paul Acquatella B. "Advanced modeling and trajectory optimization framework for reusable launch vehicles". In: *2018 IEEE Aerospace Conference*. 2018, pp. 1–18. DOI: 10.1109/AERO.2018.8396704.
- [27] Pedro Orgeira-Crespo et al. "Optimization of the Conceptual Design of a Multistage Rocket Launcher". In: *Aerospace* 9.6 (2022). ISSN: 2226-4310. DOI: 10.3390/aerospace9060286. URL: <https://www.mdpi.com/2226-4310/9/6/286>.
- [28] F. M. P. Morgado, A. C. Marta, and P. J. S. Gil. "Multistage rocket preliminary design and trajectory optimization using a multidisciplinary approach". In: *Structural and Multidisciplinary Optimization* 65.7 (June 2022). DOI: 10.1007/s00158-022-03285-y.
- [29] Byeong-Un Jo and Koki Ho. "Simultaneous Sizing of a Rocket Family with Embedded Trajectory Optimization". In: *Journal of Spacecraft and Rockets* 61.1 (2024), pp. 248–262. DOI: 10.2514/1.a35781.
- [30] Jarno Lintusaari et al. "Fundamentals and Recent Developments in Approximate Bayesian Computation". In: *Systematic Biology* 66.1 (Sept. 2016), e66–e82. ISSN: 1063-5157. DOI: 10.1093/sysbio/syw077. eprint: <https://academic.oup.com/sysbio/article-pdf/66/1/e66/24194739/syw077.pdf>. URL: <https://doi.org/10.1093/sysbio/syw077>.
- [31] Anna Wawrzynczak and Piotr Kopka. "Approximate Bayesian Computation for Estimating Parameters of Data-Consistent Forbush Decrease Model". In: *Entropy* 20.8 (2018), p. 622. DOI: 10.3390/e20080622.
- [32] Jason D. Christopher et al. "Flow parameter estimation using laser absorption spectroscopy and approximate Bayesian computation". In: *Experiments in Fluids* 62.2 (2021). DOI: 10.1007/s00348-020-03122-2.
- [33] Asher Moshe et al. "An Approximate Bayesian Computation Approach for Modeling Genome Rearrangements". In: *Molecular Biology and Evolution* 39.11 (Oct. 2022), msac231. ISSN: 1537-1719. DOI: 10.1093/molbev/msac231. eprint: <https://academic.oup.com/mbe/article-pdf/39/11/msac231/47262578/msac231.pdf>. URL: <https://doi.org/10.1093/molbev/msac231>.
- [34] Zijie Zeng et al. "Guided wave-based characterisation of cracks in pipes utilising approximate Bayesian computation". In: *Thin-Walled Structures* 192 (2023), p. 111138. ISSN: 0263-8231. DOI: <https://doi.org/10.1016/j.tws.2023.111138>. URL: <https://www.sciencedirect.com/science/article/pii/S026382312300616X>.

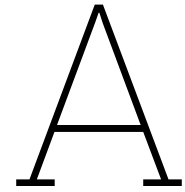
- [35] Silvia Monchetti et al. "Approximate Bayesian Computation for structural identification of ancient tie-rods using noisy modal data". In: *Probabilistic Engineering Mechanics* 77 (2024), p. 103674. ISSN: 0266-8920. DOI: <https://doi.org/10.1016/j.probengmech.2024.103674>. URL: <https://www.sciencedirect.com/science/article/pii/S0266892024000961>.
- [36] Richard G Everitt. *Ensemble Kalman inversion approximate Bayesian computation*. 2024. arXiv: 2407.18721 [stat.ME]. URL: <https://arxiv.org/abs/2407.18721>.
- [37] Yue Deng et al. "Model Selection and Parameter Estimation for an Improved Approximate Bayesian Computation Sequential Monte Carlo Algorithm". In: *Discrete Dynamics in Nature and Society* 2022 (2022), pp. 1–14. DOI: 10.1155/2022/8969903.
- [38] Sinyoung Park, Eunkwang Lee, and Minwoo Lee. "Monte-Carlo Simulation for Analyzing the Performance Variation of a Liquid Rocket Engine Using Gas-Generator Cycle". In: *International Journal of Aeronautical and Space Sciences* (June 2024). DOI: 10.1007/s42405-024-00760-2.
- [39] Ahmed Khalil et al. "Uncertainty analysis of rising sewer models with respect to input parameters and model structure using Monte Carlo simulations and computational fluid dynamics". In: *Water Science and Technology* 83.10 (Apr. 2021), pp. 2486–2503. ISSN: 0273-1223. DOI: 10.2166/wst.2021.139. eprint: <https://iwaponline.com/wst/article-pdf/83/10/2486/891884/wst083102486.pdf>. URL: <https://doi.org/10.2166/wst.2021.139>.
- [40] Aliaksei Pilko, Mario Ferraro, and James Scanlan. "Quantifying Specific Operation Airborne Collision Risk through Monte Carlo Simulation". In: *Aerospace* 10.7 (2023). ISSN: 2226-4310. DOI: 10.3390/aerospace10070593. URL: <https://www.mdpi.com/2226-4310/10/7/593>.
- [41] Eduardo Gallo. *Stochastic High Fidelity Autonomous Fixed Wing Aircraft Flight Simulator*. 2023. arXiv: 2305.02016 [cs.R0]. URL: <https://arxiv.org/abs/2305.02016>.
- [42] Qiang Cai et al. "Research on the Internal Ballistic Performance Prediction of Solid Rocket Motor Based on Monte Carlo". In: *2021 12th International Conference on Mechanical and Aerospace Engineering (ICMAE)*. 2021, pp. 193–198. DOI: 10.1109/ICMAE52228.2021.9522461.
- [43] Anna Karatzetzou. "Uncertainty and Latin Hypercube Sampling in Geotechnical Earthquake Engineering". In: *Geotechnics* 4.4 (2024), pp. 1007–1025. ISSN: 2673-7094. DOI: 10.3390/geotechnics4040051. URL: <https://www.mdpi.com/2673-7094/4/4/51>.
- [44] Cihang Lu and Zeyun Wu. "Uncertainty Quantification of the 1-D SFR Thermal Stratification Model via the Latin Hypercube Sampling Monte Carlo Method". In: *Nuclear Technology* 208.1 (2022), pp. 37–48. DOI: 10.1080/00295450.2021.1874779. eprint: <https://doi.org/10.1080/00295450.2021.1874779>. URL: <https://doi.org/10.1080/00295450.2021.1874779>.

- [45] Wencan Zhang et al. “Multi-physics coupling model parameter identification of lithium-ion battery based on data driven method and genetic algorithm”. In: *Energy* 314 (2025), p. 134120. ISSN: 0360-5442. DOI: <https://doi.org/10.1016/j.energy.2024.134120>. URL: <https://www.sciencedirect.com/science/article/pii/S0360544224038982>.
- [46] Barry Zandbergen. “Thermal rocket propulsion”. In: *Delft University of Technology* 2 (2023).
- [47] Marcelo Dasilva. en-US. 2023. URL: <https://www1.grc.nasa.gov/beginners-guide-to-aeronautics/mass-ratios/>.
- [48] F James. “Monte Carlo theory and practice”. In: *Reports on Progress in Physics* 43.9 (Sept. 1980), p. 1145. DOI: 10.1088/0034-4885/43/9/002. URL: <https://dx.doi.org/10.1088/0034-4885/43/9/002>.
- [49] Nicholas Metropolis and S. Ulam. “The Monte Carlo Method”. In: *Journal of the American Statistical Association* 44.247 (1949). PMID: 18139350, pp. 335–341. DOI: 10.1080/01621459.1949.10483310. eprint: <https://www.tandfonline.com/doi/pdf/10.1080/01621459.1949.10483310>. URL: <https://www.tandfonline.com/doi/abs/10.1080/01621459.1949.10483310>.
- [50] Russel E. Caflisch. “Monte Carlo and quasi-Monte Carlo methods”. In: *Acta Numerica* 7 (1998), pp. 1–49. DOI: 10.1017/S0962492900002804.
- [51] M. D. McKay, R. J. Beckman, and W. J. Conover. “A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code”. In: *Technometrics* 21.2 (1979), pp. 239–245. ISSN: 00401706. URL: <http://www.jstor.org/stable/1268522> (visited on 06/21/2025).
- [52] Michael Rustell. “Knowledge Extraction and the Development of a Decision Support System for the Conceptual Design of Liquefied Natural Gas Terminals under Risk and Uncertainty”. PhD thesis. University of Surrey, Jan. 2016.
- [53] Bertrand Iooss and Paul Lemaître. *A review on global sensitivity analysis methods*. 2014. arXiv: 1404.2405 [math.ST]. URL: <https://arxiv.org/abs/1404.2405>.
- [54] Mikael Sunnaker et al. “Approximate Bayesian Computation”. In: *PLOS Computational Biology* 9.1 (Jan. 2013), pp. 1–10. DOI: 10.1371/journal.pcbi.1002803. URL: <https://doi.org/10.1371/journal.pcbi.1002803>.
- [55] Brandon M. Turner and Trisha Van Zandt. “A tutorial on approximate Bayesian computation”. In: *Journal of Mathematical Psychology* 56.2 (2012), pp. 69–85. DOI: 10.1016/j.jmp.2012.02.005.
- [56] Maxime Lenormand, Franck Jabot, and Guillaume Deffuant. “Adaptive approximate Bayesian computation for complex models”. In: *Computational Statistics* 28.6 (July 2013), pp. 2777–2796. ISSN: 1613-9658. DOI: 10.1007/s00180-013-0428-3. URL: <http://dx.doi.org/10.1007/s00180-013-0428-3>.
- [57] Scott A Sisson, Yanan Fan, and Mark Beaumont. *Handbook of Approximate Bayesian Computation*. CRC Press, Sept. 2018. ISBN: 9781351643467.

- [58] Mark A. Beaumont. "Approximate Bayesian Computation". In: *Annual Review of Statistics and Its Application* 6.1 (Mar. 2019), pp. 379–403. DOI: <https://doi.org/10.1146/annurev-statistics-030718-105212>.
- [59] Jean-Michel Marin et al. "Approximate Bayesian computational methods". In: *Statistics and Computing* 22.6 (Oct. 2011), pp. 1167–1180. DOI: <https://doi.org/10.1007/s11222-011-9288-2>. URL: <https://link.springer.com/article/10.1007/s11222-011-9288-2>.
- [60] Emmanuel Klinger and Yannik Schaelte. *pyABC - distributed, likelihood-free inference — pyABC documentation*. 2025. URL: <https://pyabc.readthedocs.io/en/latest/index.html>.
- [61] Mark A. Beaumont. "Approximate Bayesian Computation in Evolution and Ecology". In: *Annual Review of Ecology, Evolution, and Systematics* 41. Volume 41, 2010 (2010), pp. 379–406. ISSN: 1545-2069. DOI: <https://doi.org/10.1146/annurev-ecolsys-102209-144621>. URL: <https://www.annualreviews.org/content/journals/10.1146/annurev-ecolsys-102209-144621>.
- [62] Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. "An adaptive sequential Monte Carlo method for approximate Bayesian computation". In: *Statistics and Computing* 22.5 (Aug. 2011), pp. 1009–1020. DOI: <https://doi.org/10.1007/s11222-011-9271-y>.
- [63] Paul Marjoram. "Approximation Bayesian Computation." In: *OA genetics* 1 3 (2013), p. 853. URL: <https://api.semanticscholar.org/CorpusID:760253>.
- [64] Khanh N. Dinh et al. *Approximate Bayesian Computation sequential Monte Carlo via random forests*. 2024. arXiv: 2406.15865 [stat.CO]. URL: <https://arxiv.org/abs/2406.15865>.
- [65] Jah Shamas et al. "Calibrating the Discrete Boundary Conditions of a Dynamic Simulation: A Combinatorial Approximate Bayesian Computation Sequential Monte Carlo (ABC-SMC) Approach". In: *Sensors* 24.15 (2024). ISSN: 1424-8220. DOI: 10.3390/s24154883. URL: <https://www.mdpi.com/1424-8220/24/15/4883>.
- [66] Matthias Speich, Carsten F. Dormann, and Florian Hartig. "Sequential Monte-Carlo algorithms for Bayesian model calibration – A review and method comparison". In: *Ecological Modelling* 455 (2021), p. 109608. ISSN: 0304-3800. DOI: <https://doi.org/10.1016/j.ecolmodel.2021.109608>. URL: <https://www.sciencedirect.com/science/article/pii/S0304380021001708>.
- [67] Arif Karabeyoglu. *Lecture 7 Launch Trajectories AA 284a Advanced Rocket Propulsion*. Stanford University, 2019. URL: https://web.stanford.edu/~cantwell/AA284A_Course_Material/Karabeyoglu%20AA%20284A%20Lectures/AA284a_Lecture7.pdf.
- [68] C.R. Glatt. *WAATS: A Computer Program for Weights Analysis of Advanced Transportation Systems*. NASA Contractor Report. NASA, 1974. URL: <https://books.google.de/books?id=Xhx0jgEACAAJ>.

- [69] Gary Harloff and Brian Berkowitz. "Hypersonic Aerospace Sizing Analysis for the Preliminary Design of Aerospace Vehicles". In: *Journal of Aircraft* vol. 27 (Feb. 1990), p. 97. DOI: 10.2514/3.45903.
- [70] G. N. Hirsch. *EDIN0613P weight estimating program*. NASA Contractor Report. NASA, 1976. URL: <https://ntrs.nasa.gov/citations/19790009715>.
- [71] Ralph Carmichael. *Description of USAF Digital Datcom*. Oct. 2022. URL: <https://www.pdas.com/datcomDescription.html>.
- [72] Tommaso Mauriello. "Multidisciplinary Design Analysis & Optimization of the SpaceLiner Passenger Stage". PDF auf Anfrage bei RY-SRT erhältlich! MA thesis. Politecnico di Milano, Feb. 2024. URL: <https://elib.dlr.de/204869/>.
- [73] M. Sippel. "Concurrent Launcher Engineering at DLR". In: *Concurrent Engineering Workshop*. Ed. by ESTEC. LIDO-Berichtsjahr=2004, 2004, pp. 1–8. URL: <https://elib.dlr.de/1720/>.
- [74] Nikolaus Hansen. *The CMA Evolution Strategy: A Tutorial*. 2023. arXiv: 1604.00772 [cs.LG]. URL: <https://arxiv.org/abs/1604.00772>.
- [75] Philipp M. Fischer et al. "Enabling a Conceptual Data Model and Workflow Integration Environment for Concurrent Launch Vehicle Analysis". In: *Proceedings of the International Astronautical Congress, IAC*. Oct. 2018. URL: <https://elib.dlr.de/124541/>.
- [76] Aaron D. Koch, Jascha Wilken, and Marko Alder. "Uncertainty quantification data model for the probabilistic design of the thermal protection system of a reusable launch vehicle stage". In: *CEAS Space Journal* (Feb. 2025). URL: <https://elib.dlr.de/213481/>.
- [77] Karel Wakker. *Fundamentals of Astrodynamics*. Institutional Repository Library Delft University of Technology, Jan. 2015. ISBN: 978-94-6186-419-2.
- [78] Manuel Martinez-Sanchez. *Lecture 32: Orbital Mechanics: Review, Staging*. URL: https://ocw.mit.edu/courses/16-512-rocket-propulsion-fall-2005/e49a0b602e0e2a7828ba86be8a8fba07_lecture_32.pdf.
- [79] Nancy Hall. *Drag Coefficient*. July 2022. URL: <https://www1.grc.nasa.gov/beginners-guide-to-aeronautics/drag-coefficient/>.
- [80] Richard Brent. "Algorithms For Minimization Without Derivatives". In: *Englewood Cliffs, Prentice Hall* 19 (Jan. 2002). DOI: 10.2307/2005713.
- [81] William H Press. *Numerical recipes in FORTRAN : the art of scientific computing*. Cambridge England ; New York, Ny, Usa: Cambridge University Press, 1992. ISBN: 9780521430647.
- [82] Luo Yuanzheng, Li Shufa, and Huang Jiang. "A New Auto-Balancing Algorithm Based on Brent's Method for Impedance Measurements". In: *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*. ACAI '19. Sanya, China: Association for Computing Machinery, 2020, pp. 73–77. ISBN: 9781450372619. DOI: 10.1145/3377713.3377724. URL: <https://doi.org/10.1145/3377713.3377724>.

- [83] Youhei Akimoto and Petr Baudis. *cma*. July 2025. URL: <https://pypi.org/project/cma/>.
- [84] China Great Wall Industry Corporation. 2015. URL: <https://www.cgwic.com/Launchservice/LM2C.html>.
- [85] Norbert Brügge. *China's YF-21 and YF-24 rocket engines*. 2025. URL: https://b14643.eu/Spacerockets/Specials/China_YF-21-24_engines/index.htm.
- [86] Andrew Jones published. *China's Long March rocket family: History and photos*. Apr. 2022. URL: <https://www.space.com/china-long-march-rockets-family>.
- [87] China Great Wall Industry Corporation. 2015. URL: <https://www.cgwic.com/Launchservice/LM3B.html>.
- [88] *Watch SpaceX Falcon 9 Rocket Launch (With 51 Starlink Satellites)*. URL: https://www.youtube.com/watch?v=TxJBS3hQn_4 (visited on 03/03/2025).
- [89] Takuya Iwanaga, William Usher, and Jonathan Herman. "Toward SALib 2.0: Advancing the accessibility and interpretability of global sensitivity analyses". In: *Socio-Environmental Systems Modelling 4* (May 2022), p. 18155. DOI: 10.18174/sesmo.18155. URL: <https://sesmo.org/article/view/18155>.
- [90] Jon Herman and Will Usher. "SALib: An open-source Python library for Sensitivity Analysis". In: *The Journal of Open Source Software 2.9* (Jan. 2017). DOI: 10.21105/joss.00097. URL: <https://doi.org/10.21105/joss.00097>.
- [91] Fred J. Hickernell, Nathan Kirk, and Aleksei G. Sorokin. *Quasi-Monte Carlo Methods: What, Why, and How?* 2025. arXiv: 2502.03644 [math.NA]. URL: <https://arxiv.org/abs/2502.03644>.
- [92] Jianyu Lei et al. "Research and Development of the Tianzhou Cargo Spacecraft". In: *Space: Science & Technology 3* (2023), p. 0006. DOI: 10.34133/space.0006. eprint: <https://spj.science.org/doi/pdf/10.34133/space.0006>. URL: <https://spj.science.org/doi/abs/10.34133/space.0006>.
- [93] Moritz Herberhold et al. *DLR Global Launch Emission Inventory 2024: Overview and Initial Results*. Oct. 2025.
- [94] Sighard F Hoerner. *Fluid-dynamic Drag*. Hoerner, Sighard F, 1965. URL: <https://archive.org/details/FluidDynamicDragHoerner1965>.
- [95] Aerodynamics Handbook Staff of The Johns Hopkins University Applied Physics Laboratory. *Handbook of Supersonic Aerodynamics*. Bureau of Naval Weapons, 1959. URL: <https://apps.dtic.mil/sti/tr/pdf/ADA279187.pdf>.



Planning

In this appendix, the work to be done is divided into work packages. Then, they are described and their interconnections are explained. The final project planning can be seen in the Gantt chart (Figure A.1).

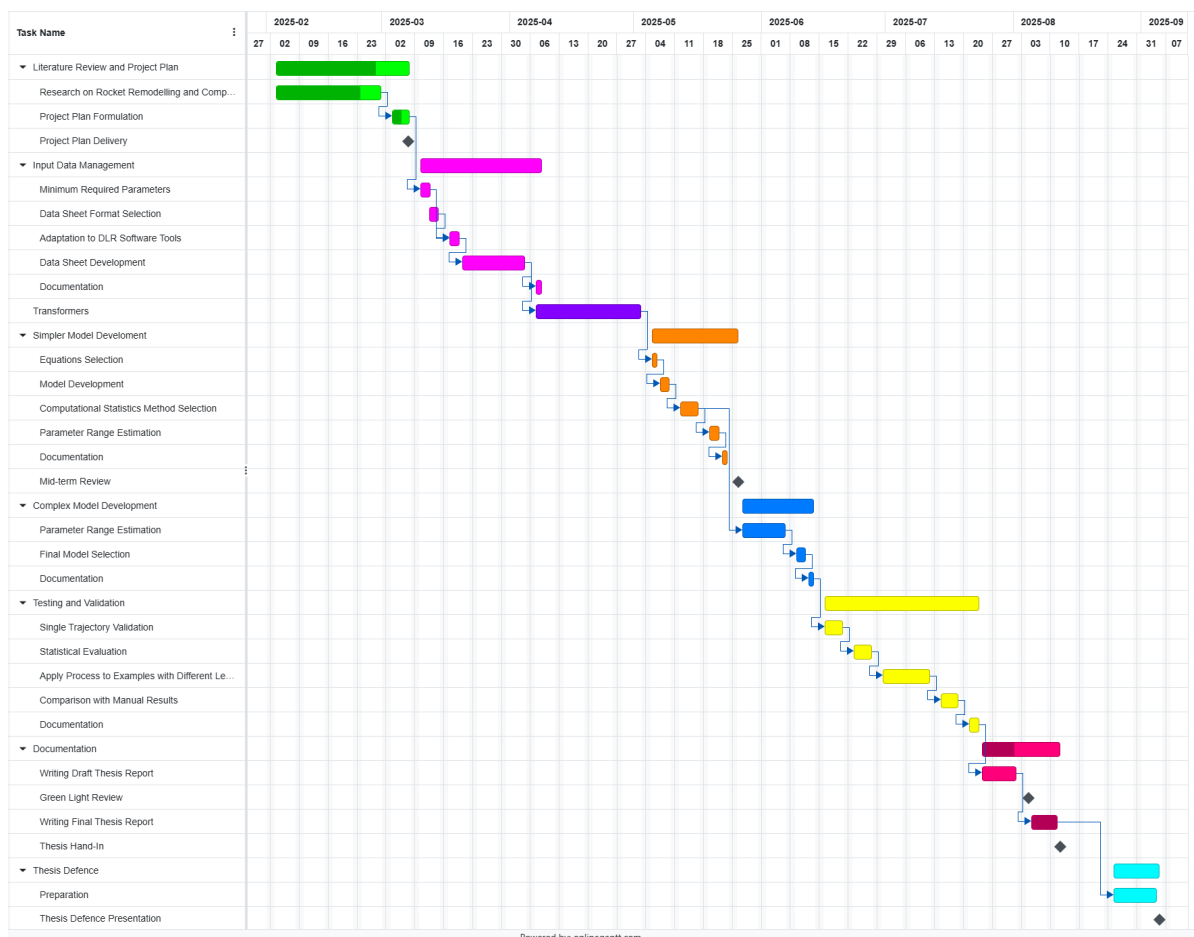


Figure A.1: Gantt chart with all work packages and their duration and organisation in the overall planning structure.

Firstly, the work packages are listed and explained:

1. Literature Review and Project Plan (5 weeks): This task has the Project Plan as a deliverable.
 - (a) Research on Rocket Remodelling and Computational Statistics Literature: this task consists of finding the relevant previous literature related to both rocket remodelling and statistical methods used to solve identical problems with a lack of input parameters.
 - (b) Project Plan Formulation: compiling all the findings and writing the project plan.
2. Input Data Management (4 weeks): by the end of this work package, the input data sheet should work with the required capabilities.
 - (a) Adaptation to DLR Software Tools: this task is important for the author to get acquainted with the new software tools (different from the previous internship) that will be used for this thesis.
 - (b) Data Sheet Format Selection: this task consists of finding suitable formats to be able to handle the missing data and choosing the best one for this problem.
 - (c) Minimum Required Parameters: before starting to develop the data sheet, it is required to identify the different combinations of the minimum amount of input parameters needed both for the remodelling process and for the validation cases.
 - (d) Data Sheet Development: this task involves the creation and evolution of the data sheet, making sure that it is infused with the data handling capabilities required.
 - (e) Documentation: Writing the documentation regarding this work package.
3. Transformers (4 weeks): this task involves developing the "connectors" between the input data sheet and the SART Toolbox programs; they will transform the data (from the TBD input data sheet format) into a format readable by those tools. By the end of this task, all combinations of the SART Toolbox files (TOSCA, STSM CAC, etc...) should be ready to be executed. The code should now be able to receive the input parameters and create all the combinations of SART Toolbox files to be run.
4. Test Model Development (4 weeks): by the end of this task, the test remodelling method should be working and capable of dealing with uncertainties. Furthermore, the statistical methods for generating data samples and evaluating the unknown parameter ranges should be tested and functioning. The code should be able to generate simplified results at this stage. In addition, the mid-term review will take place near the end of this task.
 - (a) Equations Selection: this task involves researching the equations needed to calculate the Δv necessary, as well as any other parameters that might be used for VV.

- (b) Model Development: after selecting the equations needed for the remodelling process, they have to be implemented in the code with the necessary interconnections.
 - (c) Computational Statistics Method Selection: this task consists of thoroughly analysing the computational statistics methods presented in the literature review and then selecting the most appropriate one(s) for this particular task.
 - (d) Parameter Range Estimation: after carefully opting for a certain statistical method, the next step is obtaining the range for each unknown parameter to make the remodelling process realistic, followed by a sensitivity analysis.
 - (e) Documentation: Writing the documentation regarding this work package.
5. Complex Model Development (3 weeks): by the end of this task, the more complex remodelling method should be working and capable of dealing with uncertainties and providing the combinations of working models for each rocket.
- (a) Parameter Range Estimation: after selecting and testing the best-suited statistical methods, they are applied again, to estimate the range for each unknown parameter, but now using DLR's software tools, followed by a sensitivity analysis.
 - (b) Final Model Selection: this task involves developing the algorithm capable of selecting the best or most exact models for each type (mass, trajectory, etc.), among the numerous ones that passed the validation step.
 - (c) Documentation: Writing the documentation regarding this work package.
6. Testing and Validation (5 weeks): by the end of this task, the whole algorithm should be finalised and contain no issues.
- (a) Single Trajectory Validation: this is a step to verify if the obtained trajectory is feasible and realistic for every combination of sampled input parameters for each LV.
 - (b) Statistical Evaluation: this task involves gathering the range of each input parameter that produced a realistic rocket model and then deciding whether the loop process occurs or the results are already definitive.
 - (c) Apply Process to Examples with Different Levels of Data Availability: this task consists of testing the methodology with examples containing different levels of data availability (from full, to some, to little) and iteratively refining the process as errors arise.
 - (d) Comparison with Manual Results: this task involves comparing the proposed methodology with manual results, for instance, the ones developed in the internship that led to this thesis.
 - (e) Documentation: Writing the documentation regarding this work package.
7. Documentation (3 weeks): by the end of this task, the MSc Thesis report should be completed; moreover, the green light review is planned to be close to the end of this task, and the thesis will be handed in after this work package is completed.

- (a) Writing Draft Thesis Report: this task consists of finalising the writing of the final report's first draft and iteratively improving it.
 - (b) Writing Final Thesis Report: this task involves making small adjustments to the report after all the feedback is given.
8. Thesis Defence (2 weeks): by the end of this task, the MSc Thesis will be completed; it culminates in the completion of the master's degree.
- (a) Preparation: this task consists of preparing the slides and content for the final presentation.

However, even though the work is split into the previously listed packages, the code's implementation order will likely differ. Figure A.2 shows a flowchart with the detailed planned algorithm steps.

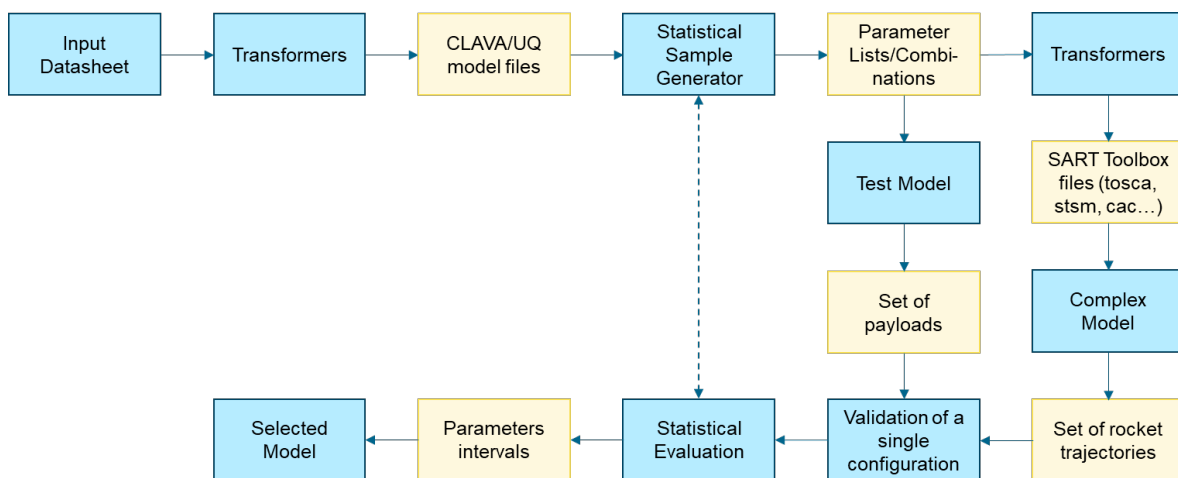


Figure A.2: Detailed algorithm flowchart containing the main components of the developed framework and their interconnections.

The predicted order of execution of the algorithm is now explained. The first step is to fill in the input data sheet to identify the missing parameters. Then, the statistical method will estimate a viable range for these unknowns, thus creating several lists of parameters with all the possible combinations to be tested. They will then be converted into a format that the SART Toolbox can read, and then the TOSCA, STSM and CAC files will be created, one for each combination. Then, the user either runs the test or the complex model, but, no matter which option is chosen, the outcome will be several payloads, again, one for each parameter combination. Finally, a validation method will filter the feasible trajectories and estimate the ranges of unknown parameters that yield realistic rocket models. Finally, if necessary, further iterations on the statistical sample generator will be performed to narrow down the unknown parameters' scope. If the user intends, the last step will be to choose the model/trajectory that fulfils a specific performance criterion.

B

Input Data Sheet

B.1. Required Parameters List

This section presents the list of parameters about a LV needed for the remodelling process to be automated. Firstly, the fairing/booster's nose parameters are listed:

- Number of stages/boosters (if applicable);
- Stage/booster length;
- Stage/booster diameter;
- Fairing nose radius (r);
- Fairing nose length;
- Fairing radius;
- Fairing cylindrical length;
- Fairing connection length;
- Fairing last radius;
- Fairing mass;
- Booster nose radius (r);
- Booster nose length;
- Booster nose base radius;
- Booster nose extra cylindrical length;
- Booster nose mass;

Figure B.1 contains two sketches to help better understand each parameter listed above.

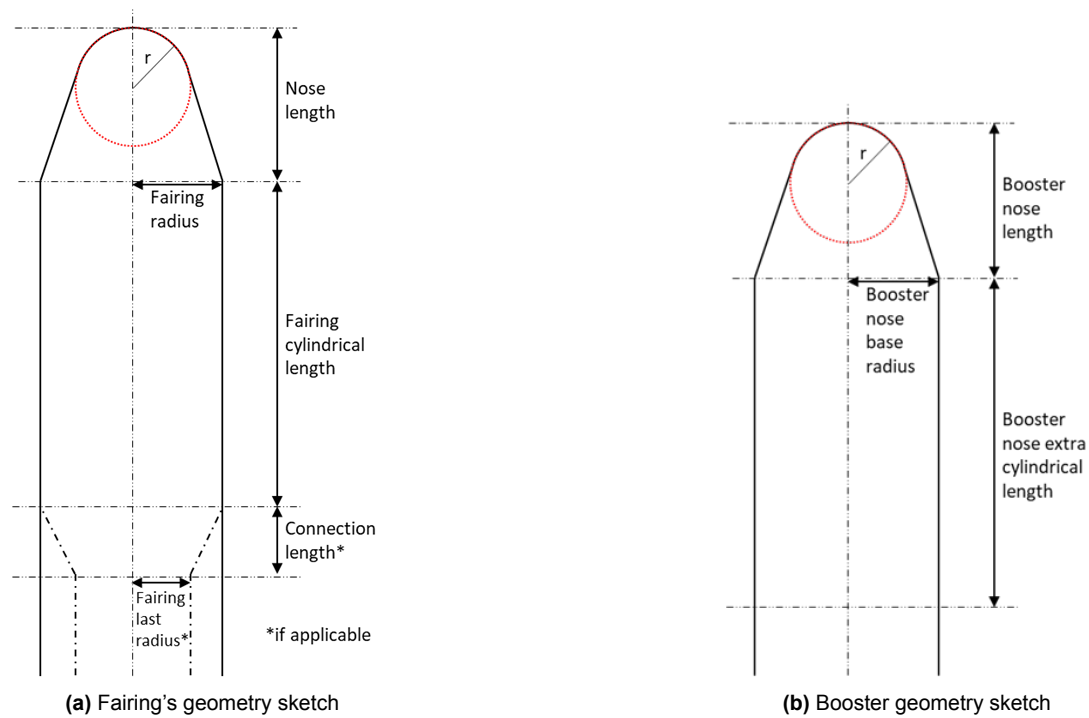


Figure B.1: Component sketches of the rocket: fairing and booster

Then, the list of each individual stage/booster stage characteristics (geometrical, propulsive, structural):

- Length;
- Diameter;
- Surface quality factor;
- Number of engines;
- Single engine mass;
- Engine's exhaust diameter;
- Maximum mass flow rate per engine;
- Maximum Isp (vacuum and sea-level);
- Maximum thrust (vacuum and sea-level);
- Total mass;
- Dry mass;
- Ascent propellant mass;
- Descent propellant mass (if applicable and for the future development of this framework);
- Reserve propellant mass;
- Residual propellant mass;
- Total propellant mass;

- Propulsion mass;
- Structure mass;

Finally, the mission(s) details are presented:

- Launch site coordinates (latitude, longitude and altitude);
- Target orbit apoapsis;
- Target orbit periapsis;
- Target orbit inclination;

There are some extra parameters/information fields in the Excel sheet, but they are not essential nor needed for the current state of this LV remodelling framework.

B.2. Excel Tabs Pictures

This section contains the pictures of the remaining tabs of the IDS not shown in the main text.

General		Sources (PLEASE ADD THE CORRECT REFERENCE)	Comments/Notes			List of References	Links/Citations
Parameters	Value						
Number of Stages	0					1	
Boosters	FALSE					2	
Number of Boosters	2					3	
GLOM (kg)	0					4	
						5	
						6	
						7	
Fairing		Sources (PLEASE ADD THE CORRECT REFERENCE)	Comments/Notes			List of References	Links/Citations
Parameters	Value						
Nose Radius (m)						8	
Nose Length (m)						9	
Fairing Radius (m)						10	
Fairing Cylindrical Length (m)							
Fairing Mass (kg)							
Connection Length (m) *							
Fairing Last Radius (m) *							
*only if applicable							
Constants							
gravitational acceleration (m^2)	9.80665						
tolerance	0.001						
atm pressure (kPa)	101.325						
Booster Nose		Sources (PLEASE ADD THE CORRECT REFERENCE)	Comments/Notes				
Parameters	Value						
Booster Nose Radius (m)							
Booster Nose Length (m)							
Booster Nose Base Radius (m)							
Booster Nose Extra Cylindrical Length (m)							
Booster Nose Mass (kg)							

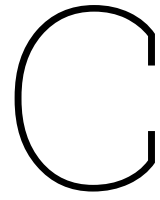
Figure B.2: IDS general tab

Parameter	Name/Value
Rocket Name	
Rocket ID	
Version	1

Figure B.3: IDS identifier tab

Figure B.4: IDS thrust profile template tab

Figure B.5: IDS validation cases tab



Test Model

C.1. Losses Results

The losses obtained from the trajectory optimisations performed for 16 rockets over more than 40 target orbits in total are presented in this appendix. The author remodelled the LM rockets during his internship, while the others were done by different colleagues.

Table C.1: Gravitational, Drag, and Thrust Losses for Multiple Rockets and Orbits

Rocket	Orbit	Gravitational Loss (m/s)	Drag Loss (m/s)	Thrust Loss (m/s)
LM 2C	LEO	1411.697	78.785	34.296
LM 2C	SSO	1421.914	61.056	8.404
LM 2D	LEO	1308.375	67.844	44.171
LM 2D	SSO	1310.163	66.718	34.562
LM 2F	LSS	2265.440	103.023	191.036
LM 3B/E	GTO	1462.985	117.311	190.975
LM 3B/E	MEO	1373.974	115.837	150.585
LM 3B/E	SSO	2244.109	91.780	203.774
LM 4B	LEO	1365.865	91.282	185.262
LM 4B	LEO	1346.446	84.857	165.201
LM 4B	SSO	1753.991	53.999	38.689
LM 4C	LEO	1640.504	72.900	181.095
LM 4C	LEO	1386.362	77.366	141.221
LM 4C	SSO	1372.449	79.115	132.646
LM 4C	SSO	1326.246	89.366	204.933
LM 4C	SSO	1478.559	65.657	49.215
LM 5	GTO	1788.172	104.375	104.059
LM 5B	LSS	1995.636	97.550	91.881
Ariane 5	GTO	1460.881	88.726	520.093
Ariane 5	TLI	1659.249	79.774	295.651
Ariane 6	GTO	1582.436	171.123	171.908
Vega C	SSO	1237.784	79.433	3.583
Falcon Heavy	LEO	1590.956	38.818	84.843

Continued on next page

Table C.1 – Continued from previous page

Rocket	Orbit	Gravitational Loss (m/s)	Drag Loss (m/s)	Thrust Loss (m/s)
Falcon Heavy	LEO	1576.795	42.039	79.211
Falcon Heavy	LEO	1915.287	39.446	22.925
Falcon Heavy	LEO	1943.230	38.001	25.711
Falcon Heavy	LEO	1924.681	39.203	25.442
Falcon Heavy	LEO	1559.064	42.738	124.604
Falcon 9	LEO	1502.359	29.583	79.830
Falcon 9	LEO	1986.406	27.272	59.818
Falcon 9	LEO	2004.453	25.319	192.777
Falcon 9	LEO	1900.653	26.021	136.119
Falcon 9	LEO	1888.845	30.589	88.544
Electron	LEO	1494.826	200.125	11.859
Electron	LEO	1512.113	197.539	103.441
Firefly Alpha	LEO	1596.082	140.173	22.209
Firefly Alpha	SSO	1583.931	140.036	27.414
Terran 1	LEO	1765.345	129.415	38.734
Terran 1	SSO	1811.382	128.248	45.315
Vega-C	SSO	1372.132	148.164	4.694
Vega-C	LEO	1456.218	142.657	20.097
Average	—	≈ 1600	—	≈ 100

C.2. Gravitational Loss Estimation

In this section, a couple of plots are presented to demonstrate the lack of a useful correlation between the gravitational loss and the LV's parameters that could influence it.

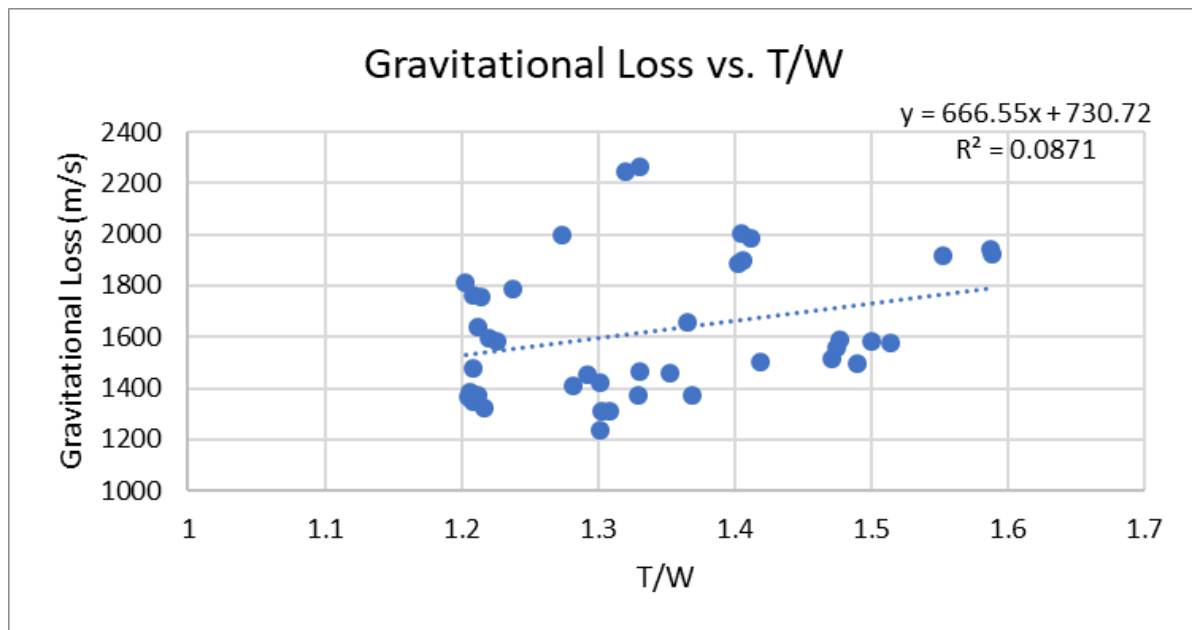


Figure C.1: Gravitational Losses correlation with T/W

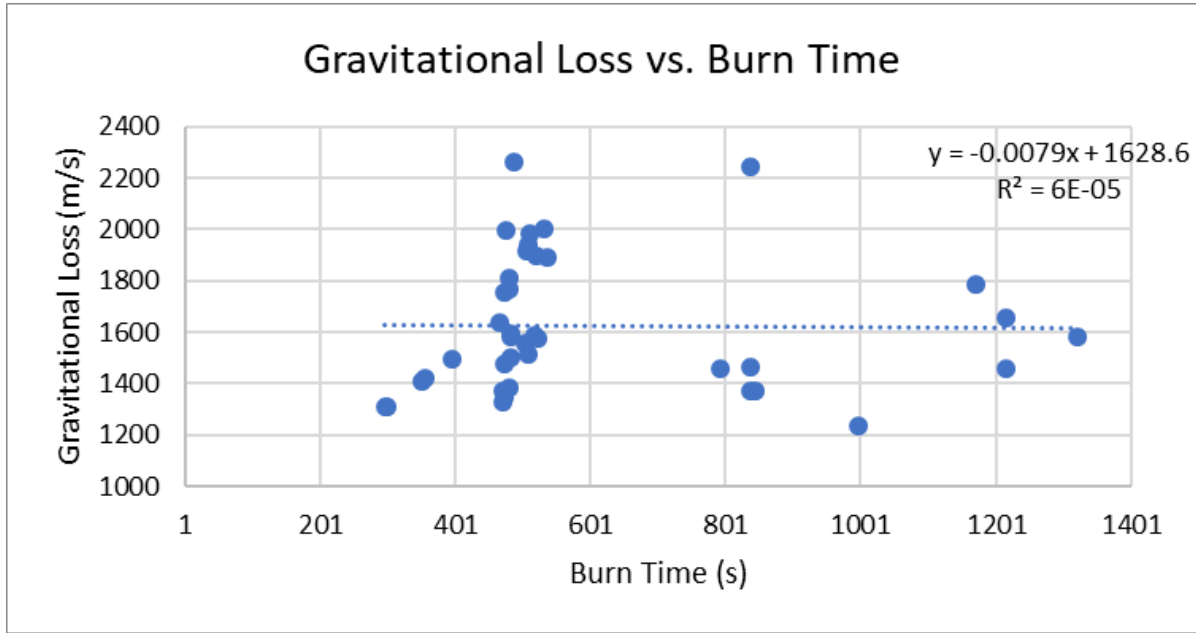


Figure C.2: Gravitational Losses correlation with Burn Time

Figure C.3 and Figure C.4 show two of these examples. In both scenarios, the coefficient of determination is very low, hence confirming the lack of a clear correlation. A few more possibilities and combinations of parameters were tested (e.g. burn time per stage, assuming an average flight path angle, adding powers to the parameters, etc.). Still, the results were not the desired ones.

C.3. Drag Loss Estimation

After some research, the best way to estimate the drag coefficient without using any velocity term was by including a term accounting for the impact of the nose's shape/geometry. Some authors studied the correlation between length-to-diameter ratio and the drag coefficient ([94], [95]).

After analysing the available literature, the conclusion was that the total drag coefficient had a fixed term ($C_{d,base}$) and a varying term based on its nose shape ($C_{d,nose}$), as shown in Equation C.1.

$$C_d = C_{d,nose} + C_{d,base} \quad (C.1)$$

Then, the first step was to find a formula that would accurately estimate the nose drag. Based on the work developed by Hoerner, F. ([94]), an approximated expression was computed for the nose variable drag coefficient part:

$$C_{d,nose} = \frac{0.05}{(L/D)^{0.5}} \quad (C.2)$$

where L corresponds to the length of the nose's ogival section, whilst D corresponds to the nose's base diameter (diameter of the cylindrical part of the nose). It is important

to note that this approximate expression only works for a finesse ratio (L/D) below 2. However, in this study and in general, no rocket was found with such a high value for this parameter. Additionally, this expression is rather simple to achieve. On the other hand, incorporating the whole spectrum of possible values for the finesse ratio results in a far more complex equation. Thus, the simplified version was used.

The following step was to add the fixed term to this. To compute it, the adopted strategy was to subtract the variable term from the initial drag coefficient resulting from the trajectory optimisation. By rearranging Equation C.1, the $C_{d,base}$ was obtained for each case. Afterwards, since this cannot change, the average was determined. In the end, the final expression was:

$$C_d = 0.2204 + \frac{0.05}{(L/D)^{0.5}}$$

After comparing the real results with the ones obtained through the expression above, an average value error of 9% was achieved. Considering that this is only the test model and that the final impact of such deviations is not high, no further improvement was done.

C.4. Thrust Loss Estimation

In this section, a couple of plots are presented to demonstrate the lack of a meaningful correlation between the thrust loss and the LV's parameters that could influence it.

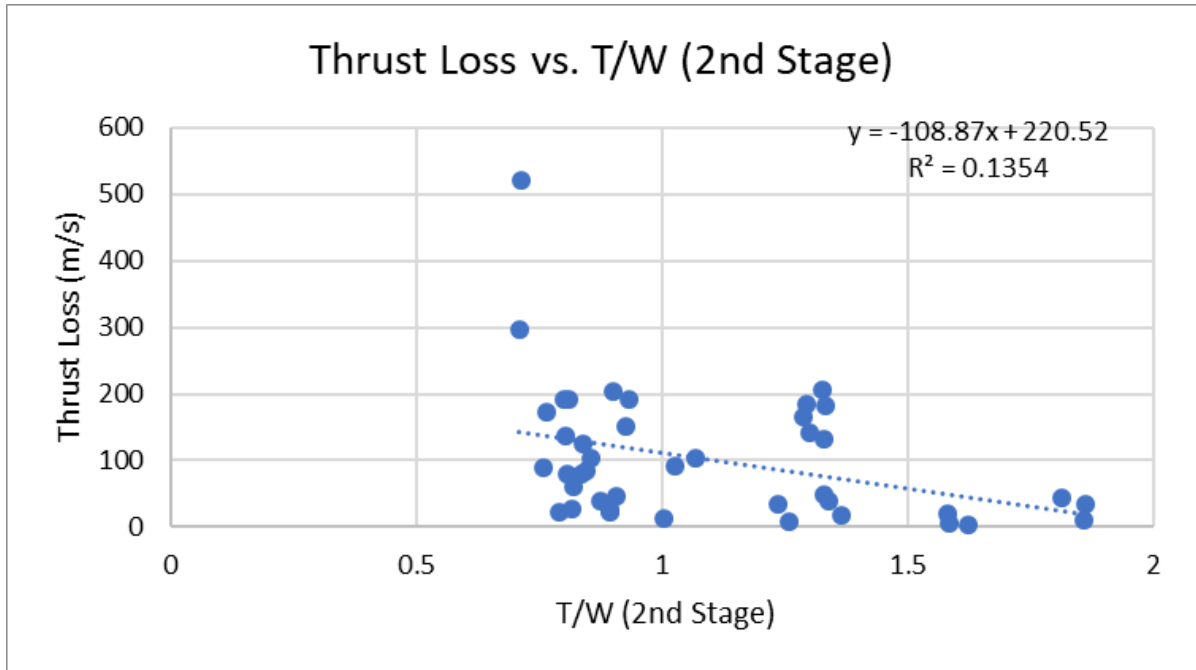


Figure C.3: Thrust Losses correlation with T/W (2nd Stage)

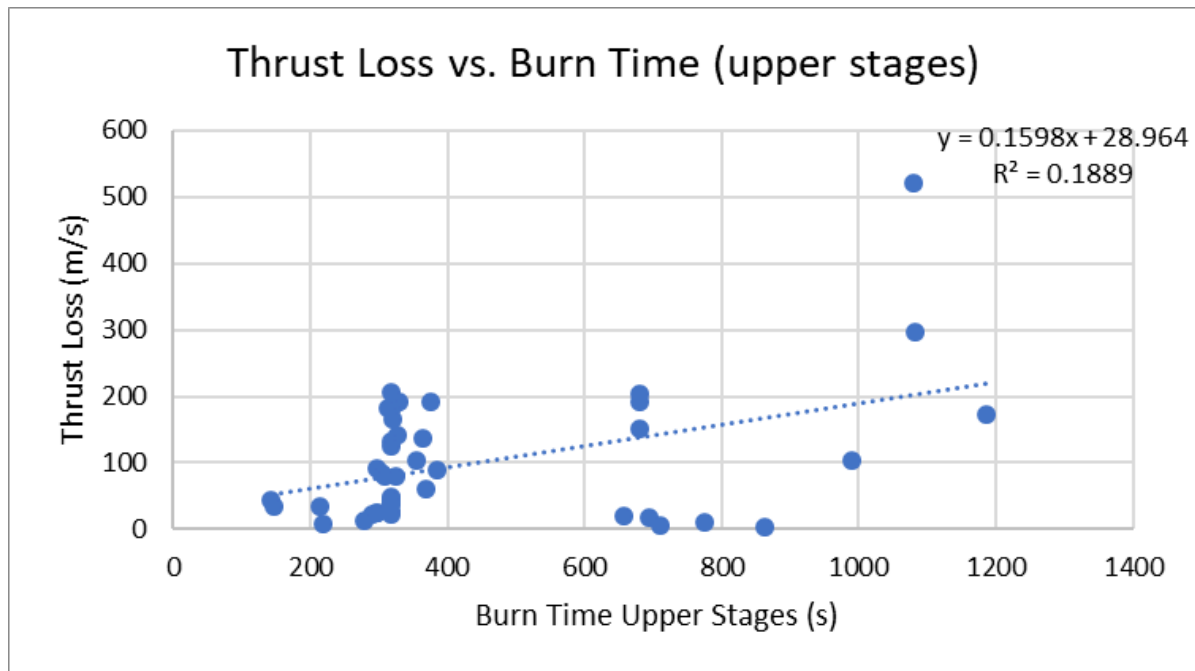


Figure C.4: Gravitational Losses correlation with Burn Time (upper stages)

The two plots above illustrate the bad correlations found for the thrust loss. The coefficient of determination is low, hence this deduction. A few more possibilities and combinations of parameters were tested (e.g. burn time divided by I_{sp} times T/W , with or without exponents, etc.). Still, the results were not the desired ones.

D

Results and Discussion

This appendix contains results from that help understand the content of Chapter 4.

D.1. Sensitivity Analysis

This section has the plots for the second-order indices results. As discussed in Section 4.1, a higher sample count is needed to better evaluate these results.

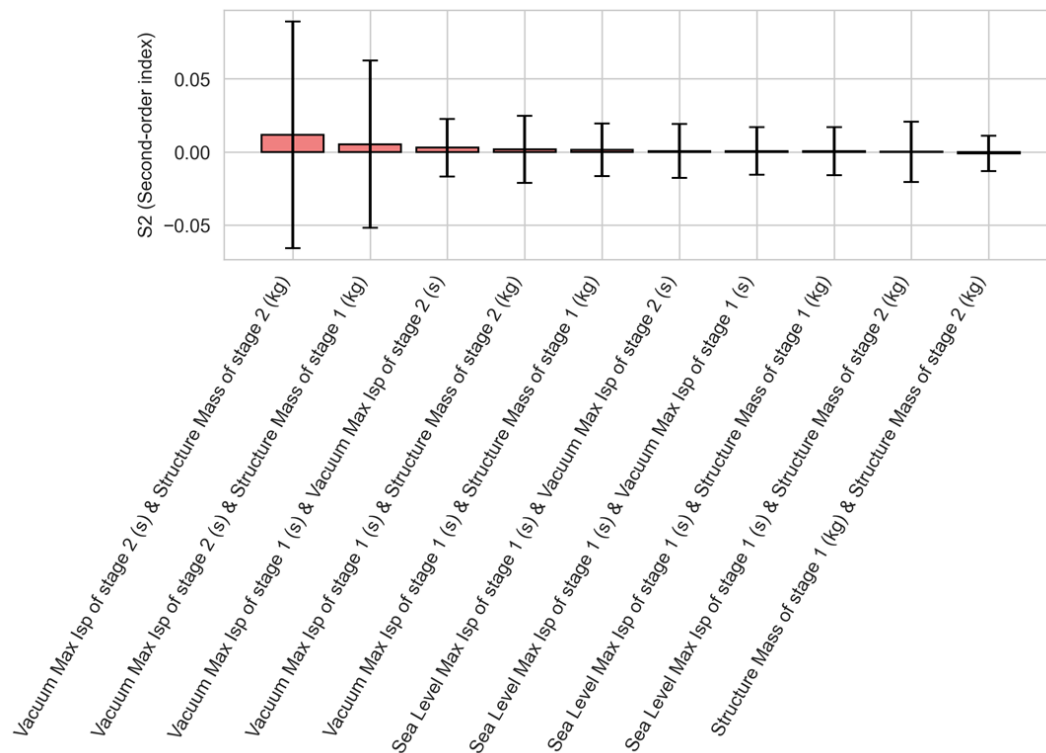


Figure D.1: Sensitivity analysis second order results including confidence intervals for test case LM2C 5U using the Test Model.

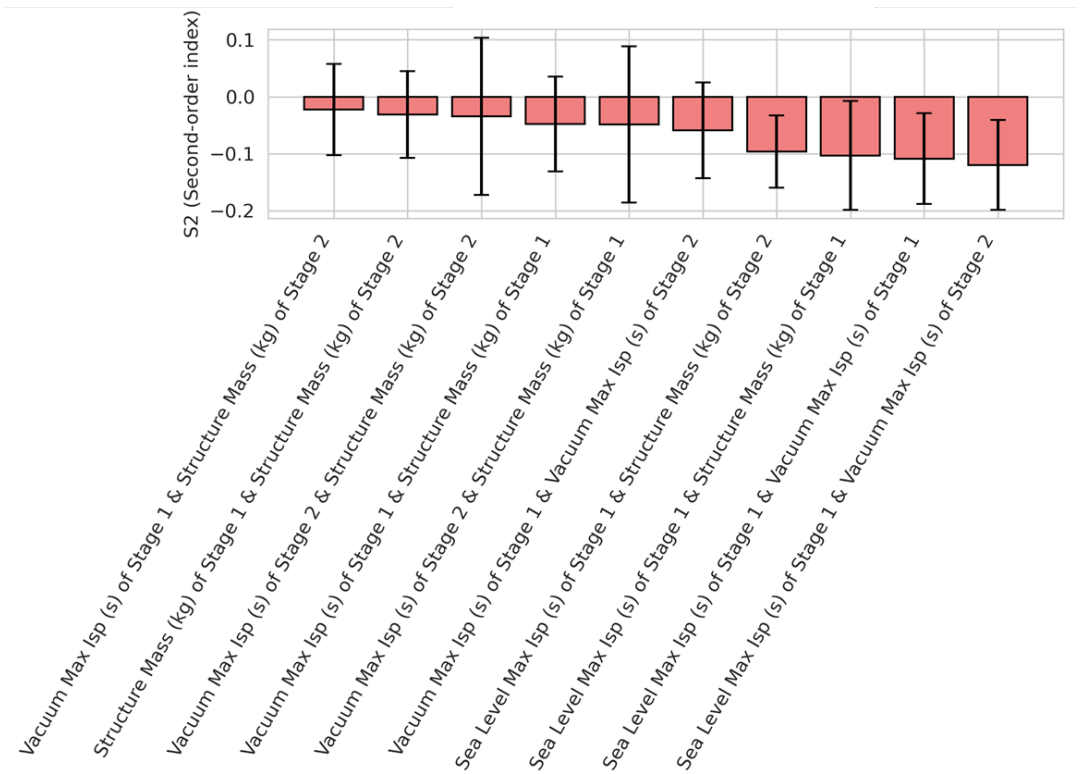


Figure D.2: Sensitivity analysis second order results including confidence intervals for test case LM2C 5U using the Complex Model.

D.2. Statistical Methods Performance

This section contains one example of a table with the results from the intensive testing phase involving the TM. Each study corresponds to a different sample count.

Table D.1: Summary statistics of the structure mass of stage 2 for test case LM2C 2U.

Study	Min	Q25	Median	Q75	Max	Mean	Std
study_1	3018.41	3148.35	3243.18	3352.03	3523.70	3259.75	156.47
study_2	3017.15	3172.18	3289.18	3409.59	3530.60	3272.95	152.56
study_3	3026.91	3192.55	3346.52	3486.79	3559.95	3335.04	170.53
study_4	3011.89	3186.67	3323.34	3421.68	3511.73	3296.32	141.64
study_5	3032.95	3169.71	3305.75	3411.75	3567.75	3293.09	145.00
study_6	3000.57	3147.96	3304.24	3428.06	3555.46	3289.45	154.23
study_7	3006.06	3178.82	3308.11	3413.12	3569.53	3293.89	146.50
study_8	3001.30	3171.12	3303.57	3421.39	3573.65	3292.43	148.79
study_9	3001.62	3182.12	3310.39	3415.67	3566.31	3296.99	145.59
study_10	3000.61	3165.91	3295.47	3416.97	3574.15	3288.54	148.11
study_11	3000.05	3166.66	3295.95	3413.45	3575.43	3288.53	148.05
study_12	3000.23	3170.25	3302.41	3416.33	3575.31	3292.28	147.96

D.3. Complex Model Results

This section contains some extra plots with results from the CM.

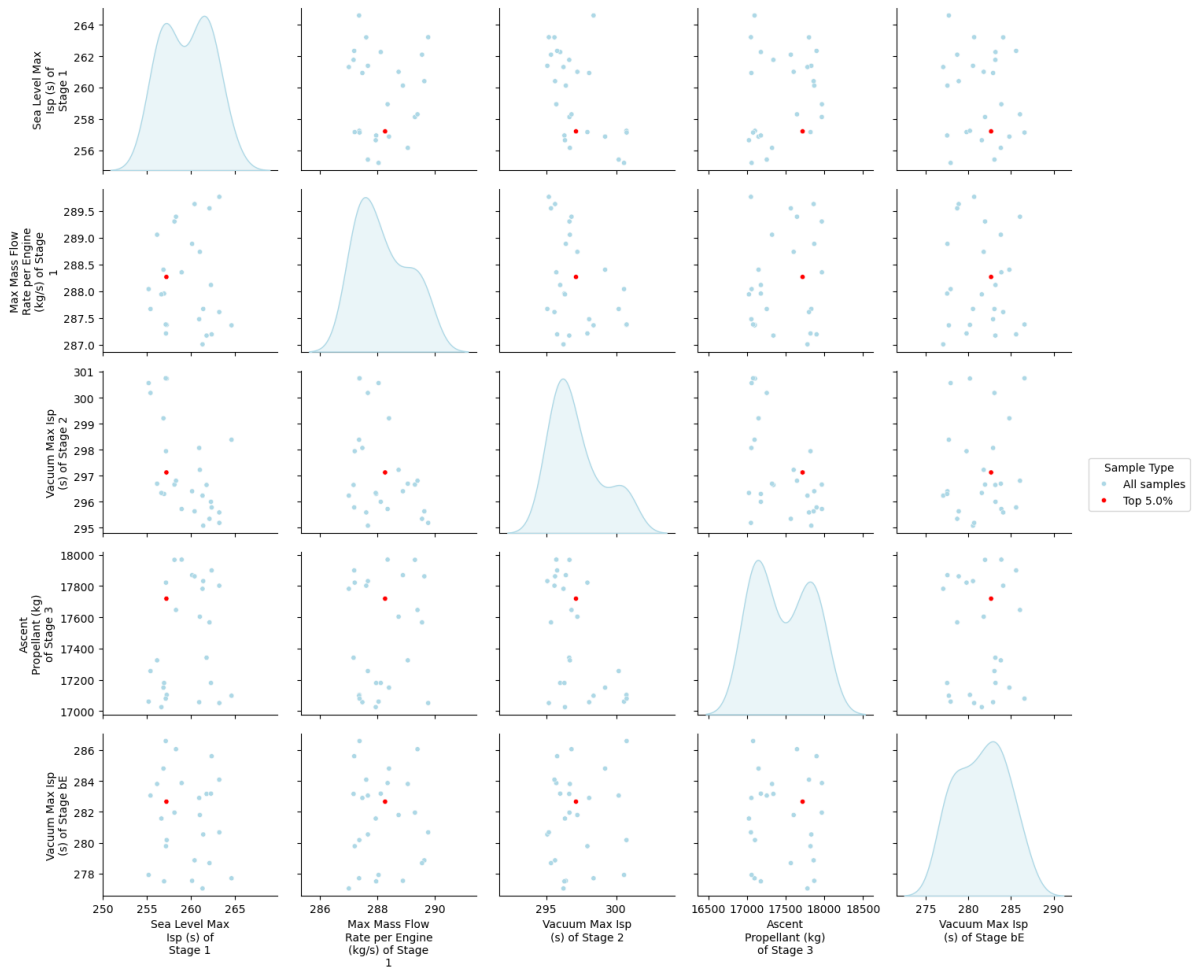


Figure D.3: Combination of unknown parameters for all the valid samples of the CM with LH sampling for test case LM3BE 5U with a payload margin of 7% intentionally deviated from the right values for comparison. The highlighted samples correspond to the top 5% samples with the lowest summed mean distance to the others.

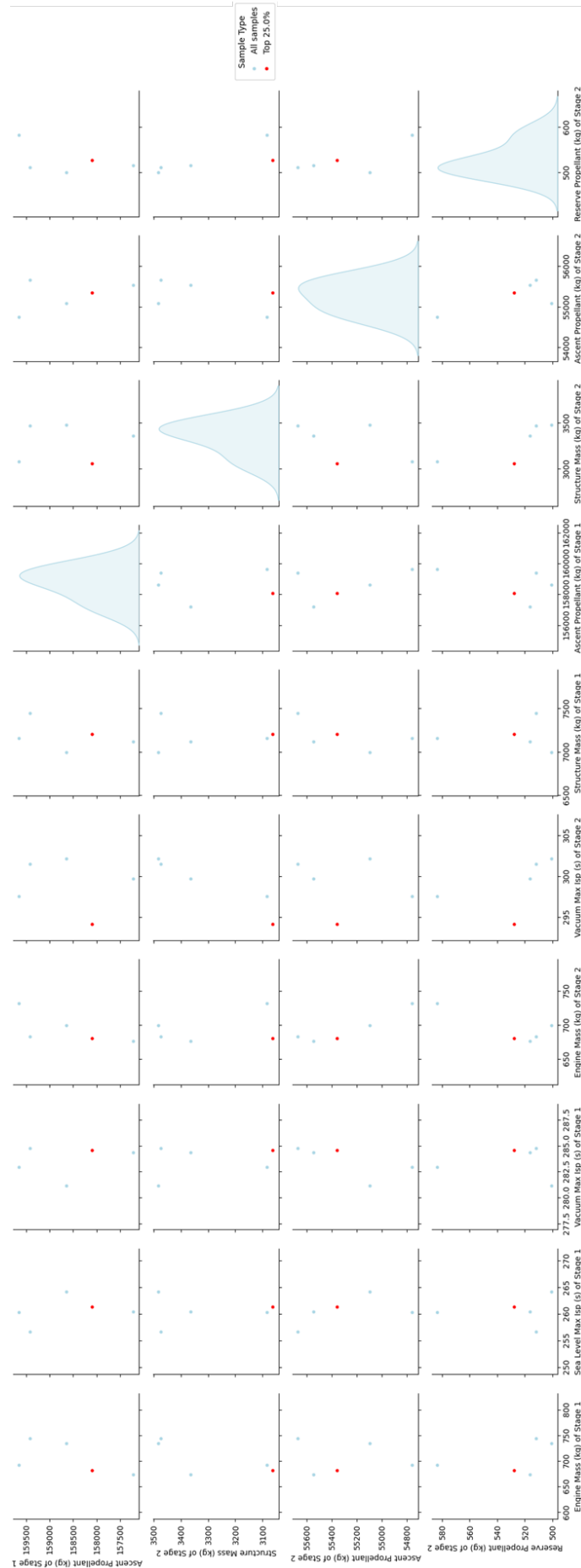


Figure D.4: Combination of the selected unknown parameters for all the valid samples of the CM with LH sampling for test case LM2C 10U with a payload margin of 2.5%. The highlighted samples correspond to the top 25% samples with the lowest summed mean distance to the others.