POLITECNICO DI TORINO

Corso di Laurea Magistrale

in Ingegneria Aerospaziale

Tesi di Laurea Magistrale

Multifidelity Methods for the Design of a Re-Entry Vehicle

Relatori

Dr. Ing. Laura Mainini

Prof. Paolo Maggiore

Candidato

Francesco Di Fiore

Anno Accademico 2019/2020

Abstract

The design and optimization of space systems presents many challenges associated with the variety of physical domains involved and their coupling. A practical example is the case of space vehicles designed to re-enter the atmosphere upon completion of their mission. For these systems, aerodynamics and thermodynamics phenomena are strongly coupled and relate to structural dynamics and vibrations, chemical non equilibrium phenomena that characterize the atmosphere, specific re-entry trajectory, and geometrical shape of the body. Blunt bodies are common geometric configurations used in planetary re-entry. These geometries permit to obtain high aerodynamic resistance to decelerate the vehicle from orbital speeds along with contained aerodynamic lift for trajectory control. The large radius-of-curvature allows to reduce the heat flux determined by the high temperature effects behind the shock wave. The design and optimization of these bodies would largely benefit from accurate analyses of the re-entry flow field through high-fidelity representations of the aerodynamic and aerothermodynamic phenomena. However, those high-fidelity representations are usually in the form of computer models for the numerical solutions of Partial Differential Equations (e.g. Navier-Stokes equations, heat equations, etc.) which require significant computational effort and are commonly excluded from preliminary multidisciplinary design and trade-off analysis.

This work addresses the integration of high-fidelity computer-based simulations for the multidisciplinary design of space systems conceived for controlled re-entry in the atmosphere. In particular, we will explore and discuss the use of multifidelity methods to obtain efficient aerothermodynamic models of the re-entering systems. Multifidelity approaches allow to accelerate the exploration and evaluation of design alternatives through the use of different representations of a physical system/process, each characterized by a different level of fidelity and associated computational expense. By efficiently combining less-expensive information from low-fidelity models with a principled selection of few expensive simulations, multifidelity methods allow to incorporate high-fidelity costly information for multidisciplinary design analysis and optimization. Modern multifidelity methods leverage active learning schemes to optimize the selection of the sample points while searching for the design optimum.

This thesis discusses multifidelity methods and compares different implementations to assist the design of aerospace systems. In particular, active learning frameworks based on multifidelity expected improvement (MFEI) are implemented and assessed for the optimization of different benchmark analytical functions. Additionally, an original MFEI algorithm is proposed and implemented for the specific case of the design optimization of the Orion-like atmospheric re-entry vehicle. The results show that the proposed MFEI algorithm gives better optimization results (lower minimum) than single fidelity active learning based on low-fidelity simulations only. The outcome suggests that the MFEI algorithm

effectively enriches the information content with the high-fidelity data. Moreover, the computational cost associated with 100 iterations of our multifidelity active learning strategy is sensitively lower than the computational burden of 6 iterations of a single fidelity framework invoking the high-fidelity model.

Ringraziamenti

Desidero ringraziare la mia relatrice Dr. Ing. Laura Mainini per il suo continuo supporto e guida in questa tesi. Laura è stata una risorsa indispensabile nel fornire il giusto bilancio tra la libertà di perseguire le tematiche di ricerca che mi appassionano e la supervisione per assicurare l'elevata qualità del risultato. La sua meticolosa revisione del lavoro ha assicurato che tutto fosse corretto e che i miei punti venissero comunicati chiaramente. Ho sinceramente apprezzato tutto il tempo e l'impegno che mi ha dedicato negli scorsi sette mesi, con una consistente presenza gioviale e con sempre pronta una parola di incoraggiamento, inclusi suggerimenti per future ricerche da intraprendere.

Desidero inoltre ringraziare il mio co-relatore Prof. Paolo Maggiore per l'innumerevole tempo speso a supporto del lavoro condotto, assicurando che la tesi potesse essere svolta nel migliore dei modi. La porta dell'ufficio di Paolo è stata sempre aperta nel corso degli ultimi due anni per qualsiasi mia necessità e gliene sono grato. Vorrei anche sottolineare l'inestimabile umanità ed umiltà con cui Paolo si è sempre posto nei miei riguardi, cosa che ho apprezzato immensamente.

Ci sono poi diversi amici del PoliTO che vorrei ringraziare per il loro supporto e cameratismo. Ringrazio Gennaro Di Leone per essere stato una splendida persona con cui condividere cinque anni di studio assieme. Ringrazio Pietro De Paola per le innumerevoli risate nelle giornate passate assieme a preparare le sessioni d'esame. Sono grato a Fabio Ferrari ed Andrea Forestieri per le stimolanti discussioni e per i progetti di successo svolti assieme. Ringrazio Riccardo Apa per il suo continuo supporto e per essere stato un fantastico compagno di allenamento. Infine devo un grazie a Ludovico Livrari e Giovanni Frasca perché senza di voi le interminabili ore di lezione non sarebbero state così divertenti.

Devo inoltre un ringraziamento agli amici di sempre per il loro sostegno a distanza nel corso di questi anni. Ringrazio Andrea Iacovone per esserci stato sempre nel corso di questi anni, pronto a dare un consiglio o una parola di conforto. Un grazie a Martina Del Grosso per le risate fatte assieme, hanno alleviato la tensione dei momenti difficili durante il percorso. Ringrazio Francesco Crivellari per la sincera amicizia che ci lega da molti anni. Da sempre abbiamo condiviso momenti indimenticabili e ti sarò sempre grato per gli splendidi anni passati insieme. Non meno importante, devo ringraziarti per tutte le scuse trovate per prenderci un caffè al posto di studiare. Ci sono poi molte altre persone che sono state dei buoni amici e sostenitori. Ringrazio Simone Di Virgilio, Mattia D'Orazio, Luca Marangione, Donato Giaccio, Matteo Ciaschetti, Emanuele Ciccotosto, Francesco Ricci e Marco Donatangelo.

Ringrazio Antonello e Tiziana, siete stati importanti in questo percorso con il vostro supporto e affetto che da sempre dimostrate nei miei confronti. Anche a voi devo questo traguardo. Grazie a Ludovica che con la sua simpatia e affettuosità mi ha sempre sostenuto e mi ha insegnato a non smettere mai di inseguire i propri sogni. Vi voglio bene.

Ringrazio la mia famiglia perché mi è sempre stata accanto e non mi ha fatto mai mancare il suo sostegno e il suo aiuto durante tutti questi anni. Senza di loro non sarei mai diventato quello che sono e non avrei potuto coronare i miei molteplici sogni. Ringrazio mio nonno Antonio e mia nonna Rachele per l'affetto e per essere sempre stati orgogliosi di me. Ringrazio mio fratello Stefano che con la sua ironia ha dimostrato di essere sempre al mio fianco. Siamo tanto uguali quanto diversi, e non avrei potuto sperare in compagnia migliore per crescere, per ridere, piangere ed essere felici. Ringrazio mio padre Dino per i sacrifici e per i preziosi insegnamenti dati in questi anni. Senza il tuo aiuto questo traguardo non sarebbe mai stato raggiunto.

Alla mia fidanzata Martina. Sei la persona che più di tutte è stata capace di capirmi e sostenermi nei momenti difficili. Sei stata la mia forza, tenacia e coraggio. Il successo conseguito non sarebbe mai stato raggiunto senza di te. Sei il mio tutto.

Dedico questo traguardo a mio padre che mi ha insegnato ad essere forte nel rialzarmi dopo ogni caduta, ad avere il coraggio di vincere ogni sfida e a diventar in grado di fare della mia vita un capolavoro.

Contents

1. Introduction

The recent developments in space exploration have led to growing interest in new space vehicles. Specifically, atmospheric re-entry vehicles represent a significant class of them. Modern space missions are increasingly supported by vehicles able to perform complex assignments (e.g. transfer astronauts to the international space station, lunar exploration missions and future landing on Mars) and return safely to the earth's surface [1]. The concept of this vehicles must take into account the multidisciplinary context and coupling of the physical domains that characterize atmospheric re-entry. Given the high speed of entry into the atmosphere, the flight regime is hypersonic for most of the trajectory travelled. As a consequence, important physical phenomena must be considered in the design process [2].

Basically, planetary entry vehicle are high-drag devices. The deceleration allows to drastically reduce the speed during the trajectory [3]. However, the vehicle structure and payload limit the maximum allowed deceleration. For unmanned re-entry probes, the structural deceleration limit can be quite high. Considering manned re-entry capsules, the maximum deceleration tolerated by the human body is of about 12 g for only a few minutes of time. Not only the maximum deceleration is a concern for aerospace designers. A reduced deceleration can lead to a bounce off the atmosphere with serious problems associated. Also, some lift is required in order to respond to uncertainties in atmospheric density, enable targeting to specific points on the earth surface and moderate the maximum accelerations on the structure and payload.

The high Mach number, that characterize the re-entry phase, involve the occurrence of high-temperature effects behind the shock wave [4]. As a consequence, the real gas effects, that is the existence of a gas in equilibrium, non-equilibrium, frozen or all those states, are important in the hypersonic flow characterization, because they determine the heat flows on the surface of the re-entry vehicle. Surface heating is one of the most important quantities in the design of an atmospheric entry vehicle, as it drives the design of the thermal protection system (TPS). Energy may be transferred to the surface of a planetary entry vehicle by particles colliding and interacting with the surface (convective heating) and radiation from excited particles in the flow (radiative heating). The thermal protection system must be engineered to withstand the high thermal flows involved.

Summing up, the design process must delicately balance the need to decelerate the capsule in order to carry out a controlled ground impact, and the survival of the vehicle upon heating.

The most common re-entry vehicle shape geometries are blunt bodies [5]. These geometries allow to produce a significant aerodynamic drag, reducing the velocity from orbital or interplanetary to landing and a small amount of lift, provided by off-setting the centre-of-gravity of the vehicle. The large radius-of-curvature of the

blunt body nose mitigate the effects of heating on the structure allowing the payload to survive during the entry trajectory.

Given the complexity of the physical domains and the strong coupling between aerodynamic and aerothermodynamic phenomena, the design and optimization of these vehicles would largely benefit from the use of accurate representative models of the phenomena. These high-fidelity models allow to closely describe the real physic environment and the interaction of the physical domains involved. However, those high-fidelity representation are usually in the form of computer models for the numerical solution of partial differential equations which require significant computational resources and are commonly excluded from preliminary multidisciplinary design and trade-off analysis.

In the early stages of re-entry vehicle design, aerospace designers prefer to implement simplified aerothermodynamic models [6, 7, 8, 9, 10, 11]. Commonly, these representations are referred to as low-fidelity models, as they represent the phenomena in a way deviated from reality introducing approximations and omissions of physical aspects considered negligible in the early phases of the decision-making process. The advantage of using low-fidelity models is the reduced computational effort associated with it. This allows to explore more technical solutions and carry out trade-off analyses in a short time during the preliminary design phase. However, despite the possible elicitation of large amount of data, low fidelity models may not be adequate to predict the real contest of operation of the system analysed. Moreover, the simplified description of physical phenomena can lead to design errors identified only in the advanced stages of the process, resulting in an increase in costs and design time.

This work addresses the implementation of multidisciplinary optimization methodologies considering complex and coupled physical domains and their accurate representation, in order to identify design opportunities, associated constraints and reduce cost and design time in the preliminary phases of the project. In particular, the integration of high-fidelity computer-based simulations for the multidisciplinary design of space systems conceived for controlled re-entry in the atmosphere will be considered [12].

Given the computational effort required by high-fidelity models, the use of multi-fidelity methods will be discussed in order to increase the computational efficiency and information accuracy of aerodynamic and aerothermodynamic computer simulations. Multi-fidelity methods allow to speed-up the design and optimization process by effectively combining low-cost information from low-fidelity models and a reduced number of invocations of more computationally expensive high-fidelity models reducing the total computational effort [13]. Modern multi-fidelity methods are based on active learning schemes in order to optimize the sampling process while searching for the optimum [14]. In Chapter 4 an overview of multifidelity methods is given.

Fot the design and optimization of a re-entry capsule addressed in this dissertation, the objective function and the simulation modules are treated as black box functions, that is, only inputs and outputs are visible/accessible. This is very common in design and engineering of aerospace multidisciplinary systems. When dealing with black box functions, the exact evaluations of derivatives is difficult and only approximations can be computed at the cost of evaluating the frequently expensive model in multiple neighbor points. For the optimization of black box functions, empirical approaches are usually adopted [16, 21, 22] which require a wise selection of the points to sample to learn about both the shape of the objective function and the path towards its global optimum.

In this work we consider Surrogate-Based Optimization exploiting Bayesian frameworks for the optimization of black box functions. Surrogate based optimization constructs surrogate model using the available information acquired from expensive experiments and/or simulations and leverage it in the optimization process. Bayesian frameworks are a popular surrogate based optimization approach. In Bayesian optimization the prior belief about the problem is given by the initial surrogate and progressively update the information about the objective function through adaptive sampling. This is also known in literature as active learning.

In addition, this dissertation considers the opportunities to leverage multiple representation of the physical phenomena at different level of fidelity in order to reduce the computational effort associated with the optimization process. Therefore, we focus on Multifidelity Bayesian Optimization framework. Multifidelity Bayesian Optimization frameworks combine effectively different information sources, approximating the objective function at different levels of fidelity, into a single surrogate model and implements active learning strategies by adaptively sampling from different fidelity levels. In Chapter 5 we present the detailed formulation of multifidelity Bayesian optimization frameworks.

In this thesis, we discuss multi-fidelity methods and compare different implementations to assist the design of aerospace systems. In particular, an active learning multi-fidelity framework based on multi-fidelity expected improvement (MFEI) are implemented and assessed for the optimization of different benchmark analytical functions [15, 16, 17].

In order to validate the effectiveness of the search for the minimum of the objective function, the MFEI algorithm is tested by means of benchmark analytical functions [20]. Despite the little engineering utility, the benchmark functions allow to test the performance of the algorithm, as the exact evaluation of the optimum is known, and the associated computational cost is reduced. Furthermore, these experiments are useful in providing information about the robustness and reliability of the implemented algorithm. Specifically, the analytic functions considered are the Forrester function, the sinusoidal squared 1D function and the Rosenbrock function. In the case of the Forrester function and the sinusoidal squared 1D function, two levels of fidelity were considered. For the Rosenbrock function 2, 4 and 8 dimensions were implemented and the MFEI algorithm was tested for multiple

levels of fidelity including a linear mapping of the function, the quadratic function and a series of low-fidelity functions defined in the literature for the specific case. Afterwards, a comparison between the performance of the MFEI algorithm and a single fidelity expected improvement active learning algorithm was made, considering the maximum fidelity function, to verify the effectiveness of the multi-fidelity approach. The experimental setup and results are presented in Chapter 7.

An original formulation of a multi-fidelity expected improvement algorithm is then presented and implemented to assist the design of an Orion-like re-entry vehicle. The proposed algorithm allows opportunely to balance the process of exploration of the domain and exploitation towards the optimum of the objective function specifically formulated for the atmospheric re-entry problem.

Considering the above, aerodynamics and aerothermodynamics phenomena are crucial in the design process of a re-entry vehicle. High-fidelity representations of this physical phenomena are characterized by considerable computational cost associated (e.g. computational fluid dynamics (CFD) simulations). In order to speed up the optimization process, the specialized MFEI algorithm combine aerodynamic and aerothermodynamic information from two fidelity models. The high-fidelity model data are acquired by hypersonic computational fluid dynamic simulations. The low-fidelity data are obtained by experimental correlations and simplified formulations of the governing equations. Low-fidelity and high-fidelity aerothermodynamic models and physical representations of re-entry phase are given in Chapter 8.

The experiments were conducted by comparing the performance of the original MFEI algorithm for the re-entry problem and a single fidelity expected improvement algorithm, where only aerothermodynamic low-fidelity models are implemented. The outcomes are presented and discussed in Chapter 8. The results suggest that the proposed MFEI algorithm gives better optimization results than the single low-fidelity expected improvement algorithm. This achievement shows a significant design opportunity given the lower TPS mass, structural temperature and burned propellant mass. Also, this result demonstrate that the algorithm is able to combine effectively a high number of cheap information from experimental correlation and a significant lower data from CFD simulations. Moreover, the computational cost associated with 100 iterations performed by the proposed algorithm is far below the computational effort of performing 6 iterations of a single fidelity framework invoking the high-fidelity model.

In Chapter 2 is given an overview of multidisciplinary design optimization approaches. In Chapter 3 we discuss the mathematical formulation of Bayesian optimization framework. In Chapter 4 is presented a multifidelity methods overview. In Chapter 5 is given the description of multifidelity Bayesian optimization. In Chapter 6 we introduce the single fidelity expected improvement and multifidelity expected improvement algorithm. In Chapter 7 are presented the experiments on analytical benchmark functions and outcome discussion. In Chapter 8 the re-entry vehicle optimization problem is defined, the implemented physical

models are described and the original algorithm is presented. Then the results are examined and compared to the single low-fidelity expected improvement algorithm.

2. Multidisciplinary Design Optimization (MDO)

Multidisciplinary design optimization (MDO) is a field of engineering involving the design of complex systems that includes a sensible number of disciplines or subsystems. MDO leverage numerical optimization techniques and advanced computational analysis tools to solve the optimization problem in the early stages, reducing time and cost of the design cycle.

The origins of MDO can be attributed to the research of Haftka [93, 94, 95, 96] and Schmit [97, 98, 99, 100] who have extended the optimization formulations of structural problems to other disciplines.

One of the most important aspects in the MDO implementation is the organization of the discipline models, approximated formulations (if any), and the optimization algorithm to achieve an optimal design.

In the following paragraphs are presented the MDO methodology, the design structure matrix, the main MDO architectures and the approaches to the MDO resolution. The main issues with MDO analysis are also presented.

2.1. MDO Methodology

In this paragraph a summary of the methodology to deal with MDO problems is presented. Specifically, in the publication "Multidisciplinary aerospace design optimization – Survey of recent developments", R. Haftka and J. Sobieski define MDO as a set of methodologies still of reference today.

The authors identify the following MDO methodology constituent elements [74]:

1. *"Mathematical Modelling of a System".* It is necessary to define numerical models capable of effectively representing the systems involved in design optimization. Furthermore, the inputs and outputs of each model must be made explicit as well as the relationships that exist between them. Determining these interconnections is important to quantify the volume of data exchanged between the models and the associated computational cost. Given the increasing computational cost associated with representative models, a balance between cost and accuracy must be exercised. Referring to the mathematical models' formulation, the decrease of computational effort of MDO problems can be done by reducing the detail level of the same physical model, using a less refined formulation, or combining effectively models of different fidelities.

2. *"Design-Oriented Analysis".* In the preliminary design and trade-off phase, various project solutions are evaluated by means of appropriate analysis tools. Specifically, the analysis tools must have some key properties to accelerate the initial design exploration or elicit detailed information in the advanced phases. In particular, the analysis can be carried out by approximate tools in order to speed up the validation of different solutions and exclude others. Typically, this is done in the early stages of the project when certain constraints and requirements are not yet definitively quantified. The most accurate analysis tools are used only at an advanced stage of the design or on subsystems that need redesign.

3. *"Approximation Concepts".* Frequently, the acquisition of exact data from MDO analysis is prohibitive, given the associated high computational cost. In the preliminary design phases, the use of approximations allows to define a simplified multidisciplinary optimization problem from which it is possible to elicit a much greater amount of information. However, these data can lead to design errors that are costly in terms of economic resources and time consuming. Consequently, the introduction of simplifications must be consistent with the physical and operational context of the system, as well as with the degree of accuracy desired by the designers.

4. *"System Sensitivity Analysis".* The sensitivity of a system indicates how much that system is dependent on the subsystems that constitute it or the physical disciplines that model it. The importance of knowing these relationships is that it allows to guide decisions and design process effectively. Furthermore, the definition of the physical dependencies of a system represents a key point in the introduction of simplifications and in its modelling. In fact, considering the operational context it is possible to introduce approximations with greater awareness, avoiding coarse design errors.

An important problem associated with MDO approaches and methodologies is the problem synthesis. The definition of the optimization problem is not clear most of times, given the difficult characterization of the integration and input/output relationship between disciplines. This characterization increases in hardness with the complexity of the system. The need to decompose the problem therefore appears to be an important aspect in MDO problems. The Design Structure Matrix is an effective tool to address the problem synthesis, defining the system components and the input-output relationships between them. In paragraph 2.2 a brief discussion on Design Structure Matrix is given.

2.2. Design Structure Matrix

The Design structure matrix (DSM) is a powerful tool to support the design process by giving a visual representation of all the subsystems constituent the system, their associated relationships and dependence. The positives of DSM are the compact representation of a considerable number of information related to subsystems (the number of which also increases as a function of the complexity associated with the system considered), the applicability of matrix analytical techniques in order to improve system structure and identification of dependencies, in the form of feedback and nested loops between subsystems [75].

Basically, DSM is a square matrix. In the cells along the diagonal are allocated the elements of the system. The upper and lower triangle present the relationships and linkages between these elements, usually represented by arrows. In the upper triangle are presented the input-output connection between diagonal cells. In the lower triangle are highlighted the relational loops between system elements. Considering a single cell belonging to the diagonal. In the upper off diagonal along the corresponding row are defined the cell outputs that will constitute inputs for a subsequent diagonal block. The inputs are identified along the column entering the cell. In the lower off-diagonal outgoing from the element along the row are described the feedback relationships.

In Fig. 1 is presented an example of DSM for the optimization problem of a re-entry vehicle. The goal is to minimize the propellant mass burned and the thermal protection system temperature (TPS) by means of an iterative process. Along the diagonal are presented the models that describe the re-entry vehicle physics. The optimization loop block defines the fixed data and the design variables in input to the diagonal elements. Along the rows are defined the physical outputs of the models. Along the column are defined the physical inputs that the element in that column receives from other elements. Specifically, the propulsion system input are the thruster and propellant data; the outputs are the thrust vector and the propellant mass burned. The trajectory model inputs are the vehicle geometry and the initial re-entry conditions; the outcomes are the fluid and thermal data acquired during the re-entry. The aerothermodynamics block take as inputs the vehicle geometry and the output of trajectory; the outputs are computation of the heat fluxes. The thermal protection system model inputs are the heat fluxes and the system data; The output is the structural thermal reached. In the lower triangle of the matrix are presented the feedback loops. In particular, the internal loop is given by the propellant mass and the outer loop is given by the structural temperature of the TPS. To minimize these outputs, the optimizer defines new design variables at the new optimization step.

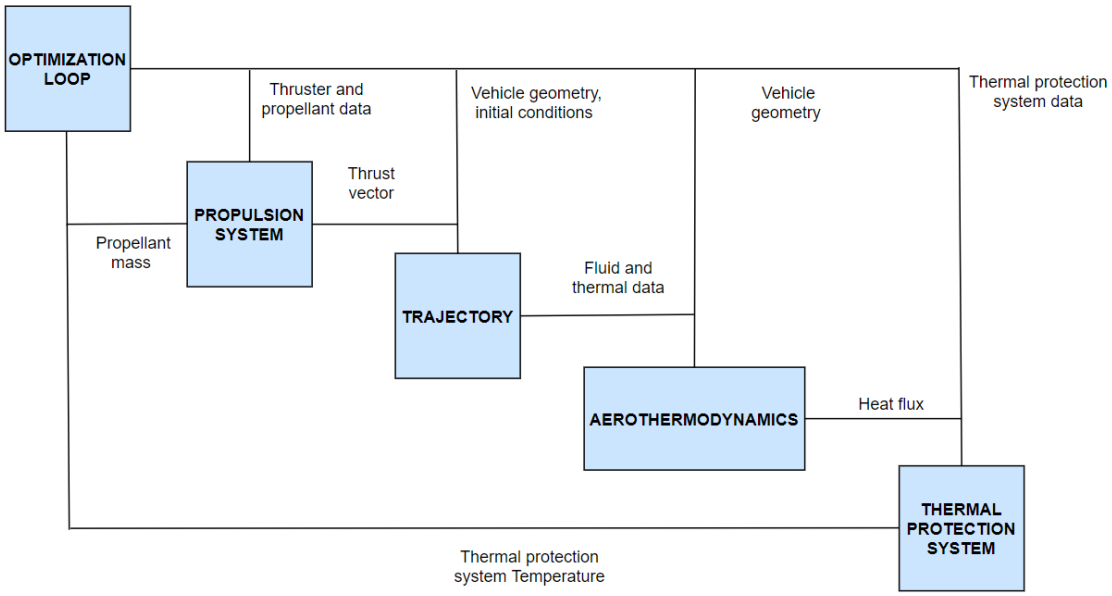

Figure 1: An example of design structure matrix for re-entry vehicle design optimization.

The DSM represent effectively the constituent relationships of the MDO problem. This allows to understand which MDO architecture is appropriate to address the multidisciplinary problem. Several MDO architectures have been proposed in the literature. Some of the most popular are discussed in paragraph 2.3.

2.3. MDO Architectures

A challenging aspect in MDO problems are the couplings between systems considered in the optimization problem. If the disciplines relationship is ignored, MDO reduces to a standard nonlinear programming problem [76], that is search the values of the design variables to minimize the objective function subject to constraints. MDO architectures provide a consistent, formal setting for managing the interdependence in the design process [77].

In order to describe the specific MDO architectures, the All-at-Once (AAO) problem statement is presented. The AAO formulate the MDO problem in the most general form. The other architectures narrated in this paragraph are derived from the following AAO framework [76]:

$$
\begin{aligned}
&\textit{minimize} && f_0(x,y) + \sum_{i=1}^{N} f_i(x_0, x_i, y_i) \\
&\textit{with respect to} && x, \hat{y}, y, \bar{y} \\
&\textit{subject to} && c_0(x,y) \geq 0 \\
& && c_i(x_0, x_i, y_i) \geq 0 && for\ i = 1, \dots, N \\
& && c_i^c = \hat{y}_i - y_i = 0 && for\ i = 1, \dots, N \\
& && \mathcal{R}_i\left(x_0, x_i, \hat{y}_{j \neq i}, \bar{y}_i, y_i\right) = 0 && for\ i = 1, \dots, N
\end{aligned}
\tag{1}
$$

Where x is the vector of design variables, y is the vector of coupling variables (outputs from a discipline analysis), $\bar{y}$ is the vector of state variables (variables used inside only one discipline analysis), $\hat{y}$ are the indipendent copies of the coupling variables distributed to other disciplines, f is the objective function, c is the vector of design contraints, c^c is the consistency constraints vector, $\mathcal{R}$ are the governing equations of a discipline analysis in residual form and N are the number of disciplines.

MDO architectures can be generally divided into monolithic architectures, where the MDO problem is solved as a single optimization problem and distributed architectures, where the optimization problem is decomposed into smaller problems. In this dissertation we focus on monolithic architectures. An exhaustive overview of distributed MDO architectures is available in the literature [76, 80, 81, 82, 83, 84, 85, 86]

The main monolithic architectures are summarized in the following points:

- *Simultaneous Analysis and Design* (SAND). This architecture simplifies the problem (1) by removing the consistency constraints introducing a single group of coupling variables to replace the separate target and response group [76]. The optimization problem is formulated as follows:

$$
\begin{aligned}
&minimize && f_0(x,y) + \sum_{i=1}^{N} f_i(x_0, x_i, y_i) \\
&with\ respect\ to && x, \hat{y}, y, \bar{y} \\
&subject\ to && c_0(x,y) \geq 0 \\
& && c_i(x_0, x_i, y_i) \geq 0 && for\ i = 1, \dots, N \\
& && \mathcal{R}_i\left(x_0, x_i, \hat{y}_{j \neq i}, \bar{y}_i, y_i\right) = 0 && for\ i = 1, \dots, N
\end{aligned}
\tag{2}
$$

- *Individual Discipline Feasible* (IDF). IDF architecture is obtained by eliminating the discipline analysis constraints. The IDF architecture is also known as distributed analysis optimization [78] and optimizer-based decomposition [77]. The problem (1) reduces to:

$$
\begin{aligned}
&minimize && f_0(x,y) + \sum_{i=1}^{N} f_i(x_0, x_i, y_i) \\
&with\ respect\ to && x, \hat{y}, y, \bar{y} \\
&subject\ to && c_0(x,y) \geq 0 \\
& && c_i(x_0, x_i, y_i) \geq 0 && for\ i = 1, \dots, N \\
& && c_i^c = \hat{y}_i - y_i = 0 && for\ i = 1, \dots, N
\end{aligned}
\tag{3}
$$

- *Multidisciplinary Feasible* (MDF). MDF architecture is obtained by removing both the discipline analysis constraints and the consistency constraints. In literature this architecture is also referred as fully integrated optimization [78] and nested analysis and design [79]. The resulting optimization problem is [76]:

$$\begin{aligned} &minimize && f_0(x,y) + \sum_{i=1}^{N} f_i(x_0, x_i, y_i) \\ &with\ respect\ to && x, \hat{y}, y, \bar{y} \\ &subject\ to && c_0(x,y) \geq 0 \\ & && c_i(x_0, x_i, y_i) \geq 0 \qquad for\ i = 1, \dots, N \end{aligned} \tag{4}$$

SAND methodology can potentially lead to the optimum quickly in comparison to other architectures because the optimizer explore region that are infeasible with respect to the analysis constraints [76]. The main issues in SAND approaches are the acquisition of state variables and the discipline modelling, leading to premature termination of the optimization problem and increase in problem size.

IDF methods allow to perform parallel computation, since the coupling between the disciplines is resolved by the variables $\hat{y}$ and consistency constraints c^c [76]. Despite the computational advantage respect to SAND approaches, the problem size still an important challenge when the discipline complexity arises.

MDF architectures present an important advantage over SAND and IDF approaches: the optimization problem formulation is small since the optimizer controls directly only the objective function, the design variables and design constraints. Moreover, the outcome of the optimization is always complying with the consistency constraints, even if the process is ended early [76]. The main negative of MDF approaches is the need to perform a full multidisciplinary design analysis (MDA) for every optimization iteration. Indeed, developing an MDA procedure can be time-consuming and computational expensive.

In approaching the MDO problem, after defining the appropriate architecture to use, it is necessary to explore the possible design alternatives with certain criteria. The techniques of exportation are also defined as design of experiments and there is a great number proposed in the literature. Design of experiments is briefly introduced in section 2.4.

2.4. Design of Experiments

Design of Experiments (DOE) is concerned with planning, conducting, analysing, and interpreting controlled test to evaluate the factors that control the value of a parameter or group of parameters. Specifically, DOE allows for multiple input design factors to be manipulated, determining their effect on a desired system response and identifying important interactions that may be missed when experimenting with one factor at a time.

In MDO problem, DOE is important because define the most suitable sampling method for the system considered in order to perform the optimization process. In this paragraph we present a brief survey of sampling methods commonly adopted in engineering optimization problems.

In statistics, sampling methods allow to select a subset of individuals from within a statistical population, referred as a statistical sample, in order to estimate properties and characteristics of the whole population.

In optimization frameworks the sampling techniques can be relevant because it allows to generate a small representative subproblem by sampling the objective domain. Such subproblem is computationally less expensive than the main problem to analyse. Therefore, the goal in optimization frameworks is to examine the subproblem and grasp as much information as possible regarding the original problem. However, the sampling technique described is suitable only in exploration, that is if the goal is to know as much as possible the objective function.

If, on the other hand, the optimization process wants to be guided towards the optimum, by sampling in the regions of the domain where it is believed (according to specific criteria) that the optimum of the objective function can be located, the sampling method must allow the balance between exploitation and exploration. In this contest, active learning frameworks (also referred as adaptive sampling) must be implemented.

The first sampling method described is *random sampling* [161, 162, 163]. In a random sampling of a given size all subsets of a sampling frame have an equal probability of being selected. Therefore, a single element of the frame has the same probability of selection as the frame constituents. Furthermore, any given pair of elements has the same probability of selection as the other pairs (and similar for triples and so on). The consequence is minimizing the bias and simplifies the analysis of the results because the variance of the overall population can be estimated by the variance of the sampled individuals. The main disadvantage of random sampling is the vulnerability to sampling errors because of the inaccurate representation of the population given the randomness of the selection.

Another significant class of sampling method are *quasi-random sampling* [164, 165, 166] approaches. In quasi-random sampling the near-random samples are generated from a multidimensional distribution. Latin hypercube sampling is an

important method belonging to this class. In statistical sampling, a square grid containing sample positions is defined a Latin square if and only if there is only one sample in each row and in each column. Considering the generalization to a multidimensional contest, a Latin hypercube is defined if each sample is the only one in each axis-aligned hyperplane containing it. Considering a function of N dimensions, the Latin Hypercube sampling scheme place M sampling points, respecting the Latin Hypercube requirements, in the M equally probable intervals in which the range of each variable is divided. The main advantage in implementing this sampling scheme is the independence of the number of samples from the increase in dimension.

The two categories of sampling methods are also referred as standard sampling approaches because the sample scheme is independent from the nature of the data available. In *active learning* frameworks [167, 168], or adaptive sampling methods, the sampling procedure is influenced from the data acquired. Considering an optimization iterative process, the determination of the sample to query is influenced from the information obtained in the previously computational steps

Latin hypercube sampling is implemented in the multifidelity Bayesian optimization algorithm presented in chapter 6 to acquire initial information about the objective function. However, in Bayesian frameworks, the iterative optimization process is based on active learning techniques. The initial information about the objective function are acquired by starting samples and during the iteration samples are selected adaptively by an acquisition function. A more detailed dissertation is given in the following paragraphs.

2.5. Optimization Algorithm for MDO

The MDO problem solution can be approached with different techniques from the optimization theories. In particular, the main methods to leverage in searching the optimum are gradient-based and gradient-free methods.

Gradient-based methods are algorithm that guide the optimum search direction through the gradient of the function at the current point. In MDO problems, adjoint equation is a popular gradient-based method. Methods based on the resolution of adjoint equations are commonly used in wing shape optimization [87], fluid flow control and uncertainty quantification. Newton-Raphson method is another common gradient-based approach. Basically, the method consists of a root-finding algorithm which outcomes better approximations to the roots of a real-valued function during the iterative process [88]. Moreover, Conjugate gradient method is an accepted approach to optimization problems. Specifically, the conjugate gradient method is used to solve unconstrained optimization problems such as energy minimization [89].

Gradient-free methods are a class of approaches that does not require the knowledge of derivatives information in the searching for the optimum. In engineering optimization problem, gradient-free methods are popular because the objective function is usually unknown, and the derivatives are unavailable or inaccurate approximations. In MDO problems, Nelder-Mead method is a popular technique [90, 91]. It is a direct search method (based on function comparison) used to minimize a multidimensional objective function using the concept of simplex, which is a special polytope of $n + 1$ vertices in n dimensions of the objective function. Another important family of derivative-free optimization method are patter search approaches. It is based on the theory of positive bases, that is attempt the optimization to find the best solution with lower error, in a multidimensional space of possibilities [92].

Moreover, the main issue in performing MDO is the associated computational effort. Usually, accurate approximation of system behaviour requires high-fidelity models capable to model effectively real phenomena. However, those representation are usually computationally expensive. Modelling with high fidelity models all the discipline that characterize a system can lead to computational costs impossible to bear especially in the preliminary design phases.

Multifidelity methods are a possible solution to the high computational cost associated with solving MDO problems. This class of approaches allows to speed up the search for the optimum by effectively combining information from models of the same discipline but with different degrees of accuracy and associated computational cost. More details are given in the paragraph 2.6.

2.6. Reduce the Computational Cost in MDO

As previously mentioned in paragraph 2.1, two elements making up the MDO methodology are the mathematical modelling of a system and the approximation concepts. Consequently, in the MDO approach it is necessary to define the system by means of a mathematical model capable of grasping its salient phenomena. However, high accuracy models are often associated with high computational costs. From this aspect derives the need to introduce model approximations in order to reduce the computational burden compared to a less accurate approximation.

Several approximation techniques are available in the literature. In this context, reference is made to the approximations proposed by Eldred et al. in [25].

According to Eldred et al. [25], the approximation of an accurate and expensive model can be made by defining a surrogate model. Surrogate models are classified in three main categories, according to the process by which they are obtained. Data-fit models are usually obtained by interpolation or regression of a set of data generated from the original model. Hierarchy surrogate models are usually obtained from the high-fidelity model by coarser discretization, relaxed tolerances, omitted physics or reduced element order. Projection-based surrogate models are obtained applying projection of the original high-dimensional system down to a small number of generalized coordinates.

In MDO problems, surrogate models can be leveraged instead of high-fidelity in order to reduce the computational burden associated. However, the reduced accuracy associated with surrogate models can lead to unsuitable optimization and design errors. A possibility to improve the data accuracy obtained by a surrogate model is the implementation of multifidelity methods.

Multifidelity methods allows to build a surrogate model combining information from multiple models, defined a fidelity and accuracy hierarchy between them. The information are obtained from the higher fidelity model and from approximated model obtained by the approaches described above. The advantage is the enrichment of the low-fidelity model by using a small and strategic number of high-fidelity data, that are expensive to obtain. As a consequence, the computational burden associated with the multifidelity-based surrogate is lower than the implementation of the high-fidelity model, but the accuracy is improved by leveraging the small amount of high-fidelity data. In Chapter 4, multifidelity methods and multifidelity optimization are presented.

3. Bayesian Optimization

Optimization problems are common in aeronautical and space engineering. Examples include the design of vehicle, systems, and structures, which require the evaluation of disciplinary models and objective functions that are frequently treated as black-box functions [110]. Basically, in black-box function the process or formulation between input and output is unknown. This is very common in the problems mentioned before because this relationship is difficult to define.

Typically, the objective function is evaluated at a given point by the algorithm till a stopping criterion is met. However, in the contest mentioned before the objective function is generally computationally expensive, so traditional methods for black-box optimization are poorly suited when a considerable number of evaluations is required.

Surrogate based optimization (SBO) synthetizes the available information into a surrogate model in order to decrease the calls of the expensive function, satisfying the necessary reduction in time, resources and associated costs [111, 112, 113, 114, 115, 116]. The efficiency of SBO can be improved in a multi-fidelity contest, where are available computationally cheaper but potentially biased approximations to the function that can be used to assist the search of optimal points [117, 118, 119, 120, 121, 122]. The use of Multi-fidelity active learning approaches for the optimization of black-box functions has been popularly studied in the Bayesian Optimization (BO) setting [123, 124, 125].

Bayesian optimization (BO) is a class of machine learning approaches for the efficient optimization of expensive black-box functions [129, 130]. The optimization problem can be formulated concisely as follows:

$$\min_{\boldsymbol{x}\in\mathcal{A}} f(\boldsymbol{x}) \tag{5}$$

Where $x \in \mathbb{R}^d$ is the input, $\mathcal{A}$ is a feasible set in which is easy to assess membership and $f(x) \in \mathbb{R}$ is the objective function.

Typically, a strong assumption is that the objective function has a known mathematical representation, is convex or is at least cheap to evaluate. However as mentioned before, in real applications evaluating the objective function is expensive or even impossible, and the derivatives and convexity properties are unknown.

Bayesian optimization is a powerful strategy for minimize the objective functions that are expensive to evaluate or where is not available a closed-form expression but where observations (often noisy) of this function at sampled values are given. It is particularly effective when these observations are costly, the access to derivatives prohibitive or when the problem considered is non-convex.

Two components are the fundaments of BO setting:

1. *Bayesian statistical model* to approximate the objective function. Typically, the statistic models are Gaussian Processes (GP). Indeed, GP are capable to model complex functions and estimate uncertainty in a probabilistic framework [126, 117, 118, 127, 128]. The efficiency of BO stems in the ability to incorporate prior belief about the problem to help direct the sampling and to trade-off exploration and exploitation of the search space.
2. *Acquisition function* to decide where to sample next, guiding the optimization process. The evaluation of the next candidate in the iteration must be a trade-off between a global exploration and a local exploitation of the surrogate.

Bayesian optimization is founded on the *Bayes' theorem*, which states that the *posterior* probability of a model M given evidence (data or observations) E is proportional to the *likelihood* of E given M multiplied by the *prior* probability of M [129]:

$$P(M|E) \propto P(E|M)P(M) \tag{6}$$

In BO the prior represents our belief about the space of possible objective functions. It is reasonable to assume that a prior knowledge about some properties of the function, such as smoothness, exists and this make some possible objective functions more plausible than others.

Let us define $\boldsymbol{x_i}$ as the *i*-th sample and $f(\boldsymbol{x_i})$ the observation of the objective function at $\boldsymbol{x_i}$. The posterior distribution can be obtained combining the accumulate observations $\mathcal{D}_{1:t} = \{x_{1:t}, f(x_{1:t})\}$ and the likelihood function $P(\mathcal{D}_{1:t}|f)$ [129]:

$$P(f|\mathcal{D}_{1:t}) \propto P(\mathcal{D}_{1:t}|f)P(f) \tag{7}$$

To sample efficiently, BO uses an acquisition function in order to define the next candidate $x_{t+1} \in \mathcal{A}$ to sample. The decision is given with a trade-off between sampling where the uncertainty of the objective function is greater (exploration) and sampling where the objective function is expected to be minimum (exploitation)

In order to describe effectively the Bayesian Optimization process is needed to define fundamental conditions.

In the following is presented the description given by Brochu et al. [129]. The authors proposed the form of the problem we are concerned with is maximization, rather than minimization, of the objective function (what has been described in the previous is valid as it is sufficient to change the sign of the objective function).

Considering a real-valued function, the maximization $x^* = argmax_x f(x)$ can be regarded as the minimization of the transformed function [129]:

$$g(x) = -f(x) \tag{8}$$

Another important assumption is that the objective function is *Lipschitz-continuous:* for all $x_1, x_2 \in \mathcal{A}$ exists some constant C (typically unknown) such that:

$$\|f(x_1) - f(x_2)\| \leq C\|x_1 - x_2\| \tag{9}$$

We can narrow the problem by defining it as of global optimization rather than local optimization. In local optimization problems, the maximization can be obtained searching a point x^* such that:

$$f(x^*) \geq f(x), \qquad \forall x \; s.t. \|x^* - x\| < \epsilon \tag{10}$$

Then any local maximum is also a global maximum if $-f(x)$ is convex.

However, as mentioned before, in our optimization problems cannot be assumed that the function is convex. So, in engineering optimization problems, the objective function is treated as a black-box function: expression of the objective or the derivatives are not available or difficult to estimate with good approximation.

Bayesian optimization uses the prior and the evidence to define a posterior distribution over the space of functions. So even when the objective function is unknown, informative priors can describe characteristics such as smoothness or the most likely locations of the maximum.

The optimization process follows the principle of *maximum expected utility* or *minimum expected risk.* The choice of the next sample requires a utility function and a way of optimizing the expectation of this utility with respect to the posterior distribution of the objective function. The utility is referred in literature as *acquisition function*.

In Tab.1 is shown the Bayesian optimization procedure. The two main components mentioned before can be found: the posterior distribution over the objective function and the acquisition function. By accumulating information $\mathcal{D}_{1:t} = \{x_{1:t}, y_{1:t}\}$, a prior distribution $P(f)$ is combined with the likelihood function $P(\mathcal{D}_{1:t}|f)$ to produce the posterior distribution $P(f|\mathcal{D}_{1:t}) \propto P(\mathcal{D}_{1:t}|f)P(f)$. In the next paragraph will be presented how Gaussian Process priors can be placed on f.

Bayesian Optimization Algorithm
1: for $t = 1,2,\ldots$ do
2: Find x_t by optimizing the acquisition function over the GP: $x_t = argmax_x u(x\|\mathcal{D}_{1:t-1})$.
3: Sample the objective function: $y_t = f(x_t) + \epsilon_t$.
4: Augment the data $\mathcal{D}_{1:t} = \{\mathcal{D}_{1:t-1}, (x_t, y_t)\}$ and update GP.
5: end for

Table 1: Bayesian optimization algorithm.

Following the discussion proposed by Brochu et al. [129], Fig.2 illustrates a typical run of Bayesian optimization on 1D problem. In this example the formulation of the optimization problem is to maximize (instead of minimizing) the objective function. The optimization process starts with two samples and at each iteration the acquisition function is maximized to decide the next sample to query. Where an higher value of the objective function is predicted by the Gaussian process, that means exploitation toward the optimum, and where the prediction uncertainty is high, that means exploration of the objective domain, the acquisition function is high in value. Computed the maximum of the acquisition function, the objective is sampled. Then the Gaussian process can be updated, and the process repeated.

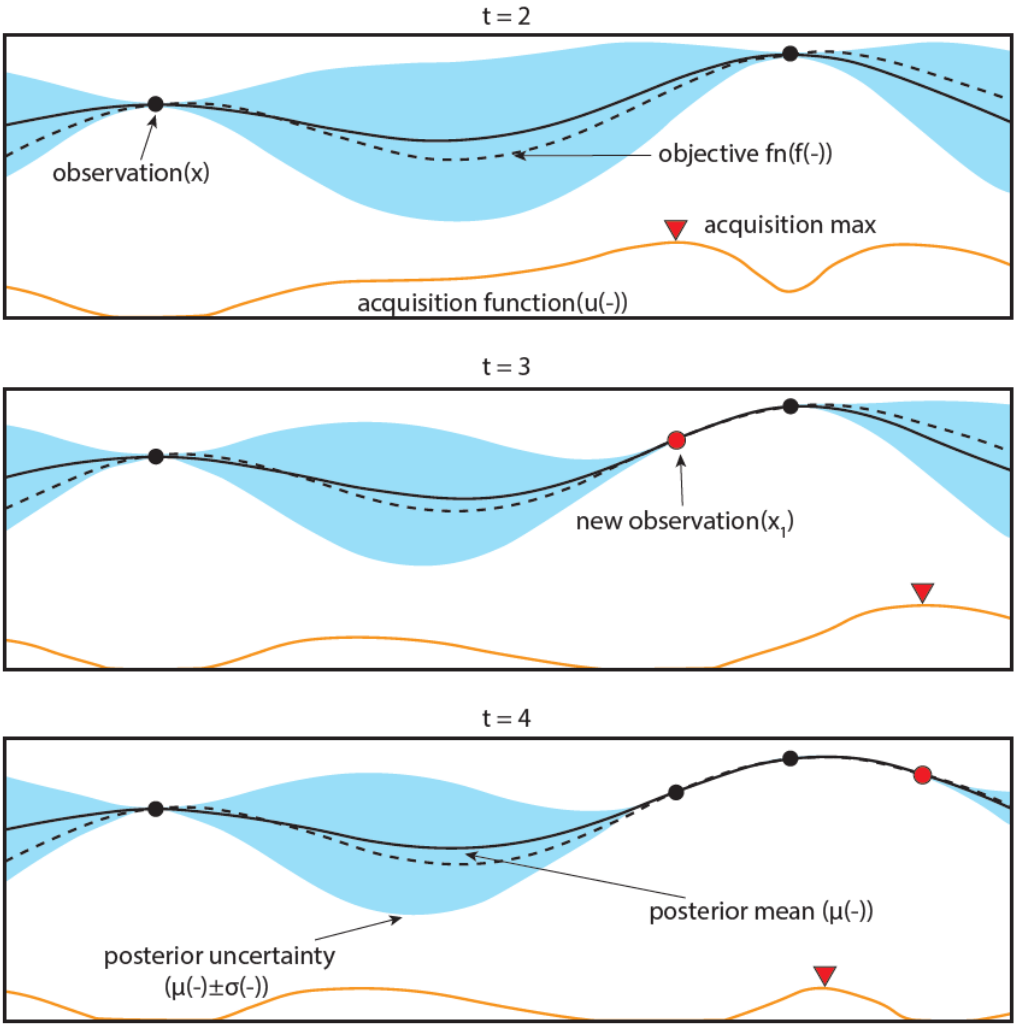

Figure 2: An example of using Bayesian optimization on a 1D design problem.

3.1. Gaussian Processes Priors

A Bayesian method depends on prior distribution by definition. The convergence of the optimization method to the optimum occurs if the acquisition function is continuous and approximately minimizes the risk (defined as the expected deviation from the global minimum at a fixed sample) and conditional variance converges to zero (or to an appropriate positive minimum value in the presence of noise) if and only if the distance to the nearest observation is zero [131, 132].

As defined in [132], the framework for Gaussian process prior must meet the following conditions: (i) the objective function is continuous; (ii) the prior is homogeneous; (iii) the optimization is independent of the m^{th} differences.

A Gaussian process is an extension of the multivariate Gaussian distribution to infinite dimension stochastic process for which any finite combination of dimensions will be a Gaussian distribution [129].

As a Gaussian distribution is a distribution over a random variable, completely specified by its mean and covariance, the GP is a distribution over functions, completely specified by its mean function m and covariance function k:

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')) \tag{11}$$

Typically, is assumed that the prior mean is zero function $m(x) = 0$. Alternatives prior means can be found in [133, 134]. A very popular choice for defining the covariance function k (also referred as *Kernel* function) is the squared exponential function:

$$k(x_i, x_j) = \exp\left(-\frac{1}{2}\left\|x_i - x_j\right\|^2\right) \tag{12}$$

This function approaches to 1 as values get close together and 0 as they get further apart. As a consequence, two points that are close together is expected to have a large influence each other, whereas distant points have almost none [129]. Other expression of Kernel functions ca be found in the next paragraph.

According to what was said, in order to sample from the prior is needed to choose $\{x_{1:t}\}$ and sample the values of the function at these indices to produce the pairs $\{x_{1:t}, f_{1:t}\}$, where $f_{1:t} = f(x_{1:t})$.

The function values are drawn according to a multivariate normal distribution $\mathcal{N}(0, \boldsymbol{K})$, where the kernel matrix is given by:

$$\boldsymbol{K} = \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_t) \\ \vdots & \ddots & \vdots \\ k(x_t, x_1) & \cdots & k(x_t, x_t) \end{bmatrix} \tag{13}$$

The diagonal values are 1 because each point is perfectly correlated with itself (only possible in noise-free environment).

In the classical engineering problems, data from an external model are used to fit the GP and get the posterior. Considering the observations $\{x_{1:t}, f_{1:t}\}$ known, we want to use BO to decide the next candidate x_{t+1}. The value of the objective function at this arbitrary point is $f_{t+1} = f(x_{t+1})$.

By the properties of Gaussian processes, $f_{1:t}$ and f_{t+1} are jointly Gaussian:

$$\begin{bmatrix} f_{1:t} \\ f_{t+1} \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \boldsymbol{K} & \boldsymbol{k} \\ \boldsymbol{k}^T & k(x_{t+1}, x_{t+1}) \end{bmatrix}\right) \tag{14}$$

Where:

$$\boldsymbol{k} = [k(x_{t+1}, x_1) \quad k(x_{t+1}, x_2) \quad \cdots \quad k(x_{t+1}, x_t)] \tag{15}$$

Using the Sherman-Morrison-Woodbury formula [131], can be easily derived an expression for the predictive distribution:

$$P(f_{t+1} | \mathcal{D}_{1:t}, x_{t+1}) = \mathcal{N}\big(\mu_t(x_{t+1}), \sigma_t^2(x_{t+1})\big) \tag{16}$$

Where

$$\mu_t(x_{t+1}) = \boldsymbol{k}^T \boldsymbol{K}^{-1} f_{1:t} \tag{17}$$

$$\sigma_t^2(x_{t+1}) = k(x_{t+1}, x_{t+1}) - \boldsymbol{k^T K^{-1} k} \tag{18}$$

$\mu_t(\cdot)$ and $\sigma_t^2(\cdot)$ are the sufficient statistics of the predictive posterior distribution $P(f_{t+1}|\mathcal{D}_{1:t}, x_{t+1})$.

Having regard for the discussion proposed by Brochu et al. [129], Fig. 3 illustrates a mono-dimensional Gaussian process considering three observations. The black line represents the Gaussian process surrogate mean prediction of the objective function. The area filled with blue colour represent the GP mean plus and minus the GP variance $\mu(x_i) \pm \sigma(x_i), i = 1, 2, 3$. The black dots along the prediction indicate the values of the objective function (calculated on specific samples) known a priori. As expected, in the neighbourhood of this points, the GP variance is sensibly low because the knowledge about the objective function is maximum. The superimposed Gaussians correspond to the GP mean and standard deviation $\mu_t(\mathrm{x_i})$ and $\sigma_t^2(\mathrm{x_i})$ $i = 1, 2, 3$ of the prediction at the points considered.

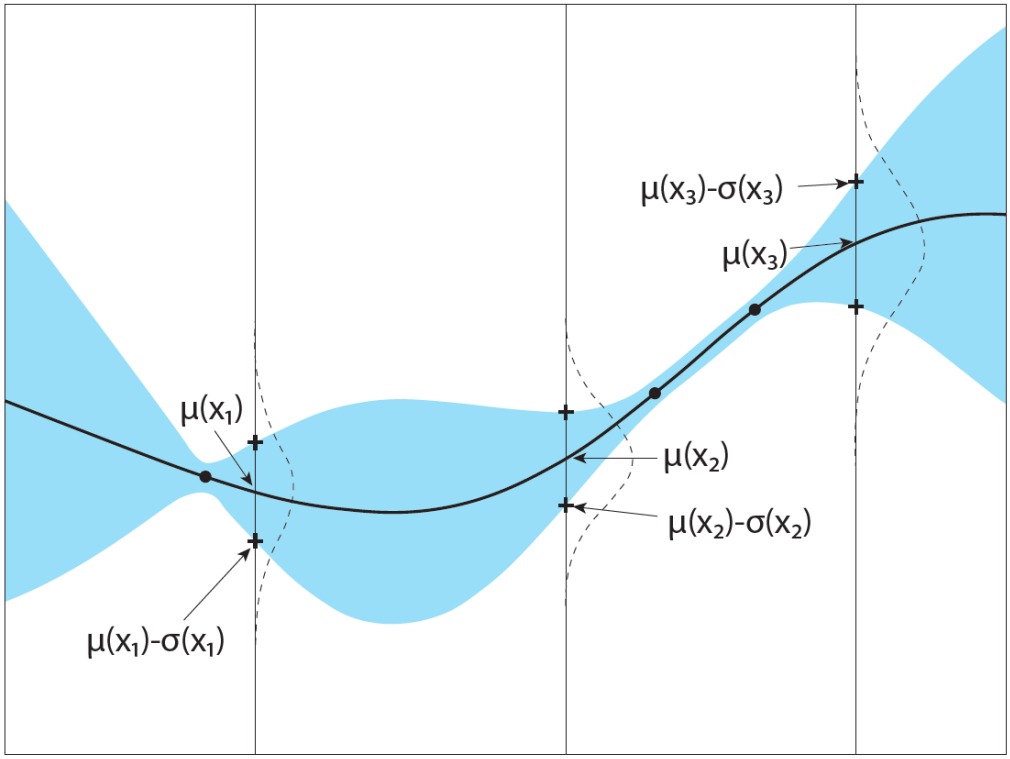

Figure 3: 1D GP with three observations

3.1.1. Kernel Functions

In machine learning, a kernel is usually defined in the contest of the kernel trick. Kernel trick is a method of using a linear classifier to solve a non-linear problem. Basically, this approach transforms linearly inseparable data randomly distributed to linearly separable ones. The kernel function is what is applied on each data instance to map the original non-linear observations into a higher-dimensional space in which they become separable.

The choice of kernel function is important as it determines the smoothness properties of samples drawn from it [129]. In machine learning literature can be found numerous kernel formulations (see, e.g., [135, 138] for an overview). However, the squared exponential and the Matérn kernel are the most common covariance function for Gaussian Processes.

The squared exponential function presented in the Gaussian process dissertation in Eq. (12), must be generalized by adding *hyperparameters.* In an isotropic model, it can be done with a single hyperparameter θ, which controls the width of the kernel:

$$k(x_i, x_j) = \exp\left(-\frac{1}{2\theta^2}\left\|x_i - x_j\right\|^2\right) \tag{19}$$

Regarding the anisotropic models, a very common choice is the squared exponential kernel with a vector of automatic relevance determination (ARD) hyperparameters θ [135]:

$$k(x_i, x_j) = \exp\left(-\frac{1}{2}(x_i - x_j)^T diag(\theta)^{-2}(x - x')\right) \tag{20}$$

Where $diag(\theta)$ is a diagonal matrix with d entries θ along the diagonal. It can be observed that if a particular θ_l has a small value, the kernel becomes indipendent of l-th input, effectively removing it automatically. As a consequence, irrelevant dimensions are discarded.

Considering the analysis conducted by Brochu et al. in [129], Fig.4 shows examples of different hyperparameter values on the squared exponential function and what functions sampled from those values look like. Typically, the hyperparameter values are learned by “seeding" with a few random samples and maximizing the log-likelihood of the evidence given θ [111, 139, 140, 135]. This can often be aided with an informative hyperprior on the hyperparameters, often a log normal prior [141, 142].

Another important kernel for Bayesian optimization is the Matérn kernel [136, 137], where a smoothness parameter ς has been introduced to allow greater flexibility in modelling functions:

$$k(x_i, x_j) = \frac{1}{2^{\varsigma-1}\Gamma(\varsigma)}\left(2\sqrt{\varsigma}\|x_i - x_j\|\right)^{\varsigma} H_{\varsigma}\left(2\sqrt{\varsigma}\|x_i - x_j\|\right) \tag{21}$$

Where $\Gamma(\cdot)$ and $H_{\varsigma}(\cdot)$ are the Gamma function and the Bessel function of order ς. If $\varsigma \to \infty$ the Matérn kernel tends to the squared exponential kernel and if $\varsigma = 0.5$ it reduces to the unsquared exponential kernel.

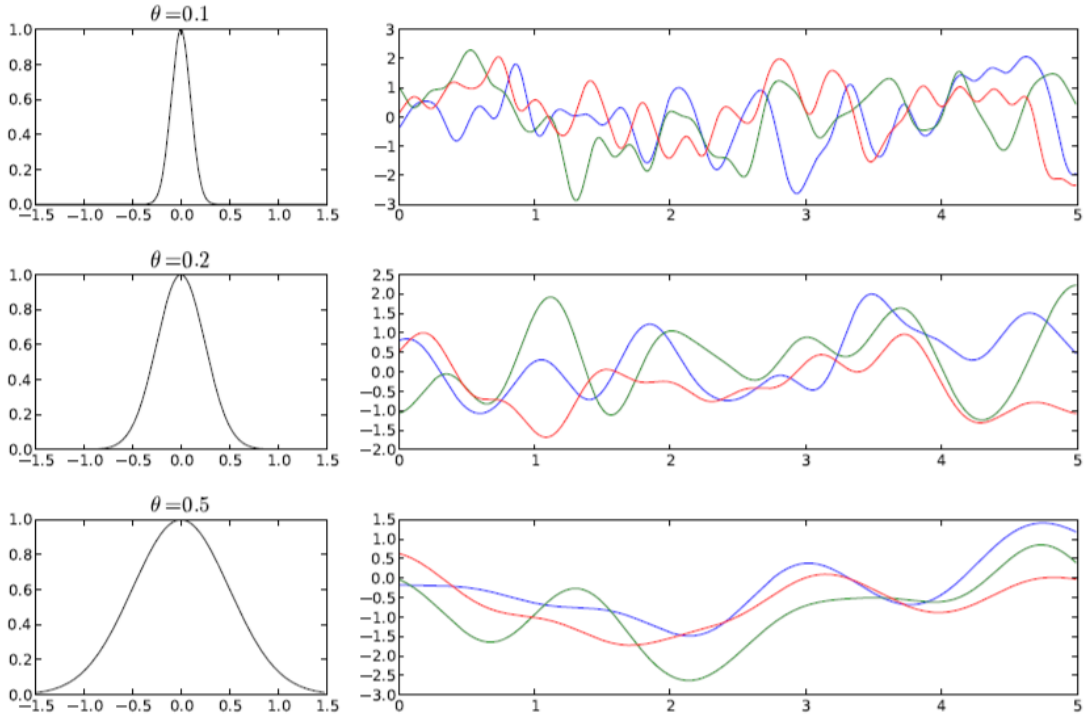

Figure 4: The effect of changing the kernel hyperparameters. Specifically, the exponential kernel is considered. The hyperparameter θ is considered equal to 0.1, 0.2 and 0.5. The left graphs show the function $k(0, x)$. The right graphs show three one-dimensional functions sampled from a Gaussian process with the hyperparameter value.

3.1.2. Noise

The model we have used so far assumes that we have perfectly noise-free observations. In real life, this is rarely possible, and instead of observing $f(x)$, we can often only observe a noisy transformation of $f(x)$ [129]. The simplest transformation can be considered when $f(x)$ is corrupted with Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma_{noise}^2)$ [135]. If is additive, the noise distribution can be easily added to the Gaussian distribution $\mathcal{N}(0, \boldsymbol{K})$ and so can be defined:

$$y_i = f(x_i) + \epsilon_i \tag{22}$$

Since the mean is zero, the kernel can be replaced by the following expression for the noisy observations of $f(\cdot)$:

$$K = \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_t) \\ \vdots & \ddots & \vdots \\ k(x_t, x_1) & \cdots & k(x_t, x_t) \end{bmatrix} + \sigma_{noise}^2 I \tag{23}$$

This yields the predictive distribution:

$$P(y_{t+1} | \mathcal{D}_{1:t}, x_{t+1}) = \mathcal{N}(\mu_t(x_{t+1}, \sigma_t^2(x_{t+1}) + \sigma_{noise}^2) \tag{24}$$

And the sufficient statistics:

$$\mu_t(x_{t+1}) = \boldsymbol{k}^T [\boldsymbol{K} + \sigma_{noise}^2 I]^{-1} y_{1:t}$$

$$\sigma_t^2(x_{t+1}) = k(x_{t+1}, x_{t+1}) - \boldsymbol{k}^T [\boldsymbol{K} + \sigma_{noise}^2 I]^{-1} \boldsymbol{k}$$

In a noisy environment change the definition of EI acquisition functions. Instead of using the best observation, we use the distribution at the sample points, and define as the incumbent, the point with the highest expected value,

$$\mu^{+} = argmax_{x_i \in x_{1:t}} \mu(x_i)$$

This avoids the problem of attempting to maximize expected improvement over an unreliable sample.

3.1.3. Kriging

This paragraph briefly discusses the synergy between Kriging and Bayesian optimization. A more detailed analysis is given in [137, 138]. In many modelling techniques in statistics and machine learning, it is assumed that samples drawn from a process with independent, identically distributed residuals, typically, $\epsilon \sim \mathcal{N}(0, \sigma_{noise}^2)$:

$$y(x) = f(x) + \epsilon$$

However, in kriging the assumption is that the errors are not independent but are spatially correlated: where errors are high is expected that nearby errors will also be high. Kriging is a combination of a linear regression model and a stochastic model fitted to the residual errors of the linear model. The residual is modelled with a zero-mean Gaussian process, so ϵ is actually parameterized by x: $\epsilon(x) \sim \mathcal{N}(0, \sigma^2(x))$.

The regression model depends on the type of kriging. In simple kriging, f is modelled with the zero-function making it a zero-mean GP model. In ordinary kriging, f is modelled with a constant but unknown function. In universal kriging models, f is modelled with a polynomial of degree k with bases m and coefficients β, so that

$$y(x) = \sum_{j=1}^{k} \beta_j m_j(x) + \epsilon(x)$$

Is clear that kriging and Bayesian optimization are related. But some key differences can be highlighted. In Bayesian optimization, models are usually fit through maximum likelihood. In kriging, models are usually fit using a variogram, a measure of the average dissimilarity between samples versus their separation distance.

Fitting is done using least squares or similar numerical methods, or interactively, by an expert visually inspecting the variogram plot with specially designed software. Kriging also often restricts the prediction model to use only a small number of neighbours, making it fit locally while ignoring global information. Bayesian optimization normally uses all the data in order to learn a global model.

3.2. Acquisition Functions for Bayesian Optimization

Defined a prior model that represent our belief about the black-box function f given $\mathcal{D}_{1:t}$. The acquisition function guides the sampling in the search for the optimum.

Typically, where the acquisition function has higher values corresponds to potentially lower values of the objective function, whether because the prediction is high, the uncertainty is great, or both. Maximizing the acquisition function allows to select the next candidate at which to evaluate the objective function.

In the early literature [143], maximizing the *probability of improvement* over the incumbent $f(x^+)$ was the best practice, where $x^+ = argmax_{x_i \in x_{1:t}} f(x_i)$ so that:

$$PI(x) = P\big(f(x) \geq f(x^+)\big)$$

$$= \Phi\left(\frac{\mu(x) - f(x^+)}{\sigma(x)}\right) \tag{25}$$

Where $\Phi(\cdot)$ is the normal cumulative distribution function, called also maximum probability improvement (MPI) or P-algorithm.

A problem with the formulation in Eq. (25) is that is purely exploitation. In fact, points that have a high probability of being infinitesimally greater than $f(x^+)$ will be drawn over points that offer larger gains but less certainty. In order to also consider exploration, a trade-off parameter $\xi \geq 0$ has been introduced in the formulation:

$$PI(x) = P(f(x) \geq f(x^+) + \xi)$$

$$= \Phi\left(\frac{\mu(x) - f(x^+) - \xi}{\sigma(x)}\right) \tag{26}$$

The choice of ξ is left to the user. In [143] is recommended a schedule for the trade-off parameter so that it started fairly high early in the optimization to drive exploration and decrease toward zero as the algorithm continued.

The impact of the trade-off parameter in different domains have been studied in several publications (see, e.g., [144, 145, 141]). A significant property of this formulation for perpetual and preference models is that while maximizing $PI(\cdot)$ is still greedy, it selects the point most likely to offer an improvement of at least ξ [129].

An alternative acquisition function would be one that considers not only the probability of improvement, but the magnitude of the improvement a point can potentially yield. In particular, we want to minimize the expected deviation from the true maximum $f(x^*)$, when choosing a new trial point [129]:

$$x_{t+1} = argmin_x \mathbb{E}(\|f_{t+1}(x) - f(x^*)\| \,|\mathcal{D}_{1:t})$$

$$= argmin_x \int \|f_{t+1}(x) - f(x^*)\| \, P(f_{t+1}|\mathcal{D}_{1:t}) df_{t+1} \tag{27}$$

Considering this formulation, the decision process considers only one-step ahead choices. In order to plan two steps ahead, recursion can be applied:

$$x_{t+1} = argmin_x \mathbb{E}(\min_{x'} \mathbb{E}\|f_{t+2}(x') - f(x^*)\| \,|\mathcal{D}_{t+1}) \,|\mathcal{D}_{1:t}) \tag{28}$$

However, applying this procedure of dynamic programming for many steps ahead can be computationally expensive.

Mockus *et al.* [146] proposed the alternative of maximizing the expected improvement with respect to $f(x^+)$. The definition of the expected improvement function is [146]:

$$I(x) = \max\{0, f_{t+1}(x) - f(x^+)\} \tag{29}$$

Where $I(x)$ is positive when the prediction is higher than the best value known thus far. Otherwise, $I(x)$ is set to zero. By maximizing the expected improvement, the new query point can be found:

$$x = argmax_x \mathbb{E}(\max\{0, f_{t+1}(x) - f(x^+)\} | \mathcal{D}_t) \tag{38}$$

The likelihood of improvement I on a normal posterior distribution characterized by $\mu(x), \sigma^2(x)$ can be computed from the normal density function:

$$\frac{1}{\sqrt{2\pi}\sigma(x)} \exp\left(-\frac{(\mu(x) - f(x^+) - I)^2}{2\sigma^2(x)}\right)$$

The expected improvement is the integral over this function:

$$\mathbb{E}(I) = \int_{I=0}^{I=\infty} I \frac{1}{\sqrt{2\pi}\sigma(x)} \exp\left(-\frac{(\mu(x) - f(x^+) - I)^2}{2\sigma^2(x)}\right) dI$$

$$= \sigma(x)\left[\frac{\mu(x) - f(x^+)}{\sigma(x)} \Phi\left(\frac{\mu(x) - f(x^+)}{\sigma(x)}\right) + \phi\left(\frac{\mu(x) - f(x^+)}{\sigma(x)}\right)\right] \tag{30}$$

The expected improvement can be evaluated analytically [111, 146], yelding:

$$EI(x) = \begin{cases} \big(\mu(x) - f(x^+)\big)\Phi(Z) + \sigma(x)\phi(Z) & if\ \sigma(x) > 0 \\ 0 & if\ \sigma(x) = 0 \end{cases} \tag{31}$$

Defining:

$$Z = \frac{\mu(x) - f(x^+)}{\sigma(x)}$$

Where $\phi(\cdot)$ and $\Phi(\cdot)$ denote the PDF and CDF of the standard normal distribution, respectively.

Following the discussion proposed by Brochu et al. [129], In Fig. 5 a typical expected improvement scenario is illustrated. The figure is the same presented in Fig. 3 previously but additionally showing the region of probable improvement. In this example, the maximum observation is considered at x^+. The blue filled area, representing the GP mean plus and minus the GP variance $\mu(x_i) \pm \sigma(x_i), i = 1, 2, 3$, can be used as a measure of the improvement, I(x). As expected, the model predicts almost no possibility of improvement by observing at $x_1$or x_2, while sampling at x_3 is more likely to improve on $f(x^+)$.

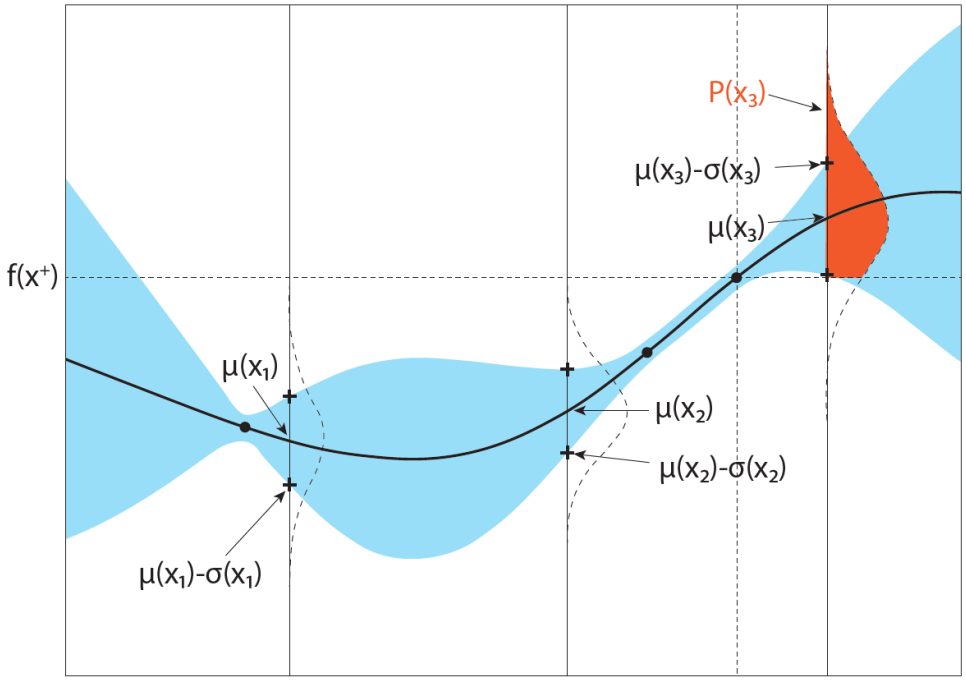

Figure 5: typical expected improvement scenario.

4. Multifidelity Methods Background and Overview

The term fidelity referring to a physical model describes how the model itself is able to estimate reality more or less accurately. Nowadays, in the preliminary design phase of complex systems the use of basic physical models or empirical tools is preferred, but these tools in the face of a fast computation, are not able to predict important physical phenomena involved. The need to describe reality more faithfully leads to the desire to use high fidelity models in the preliminary design phases. In fact, complex phenomena would be better described, redesign costs due to lack of performance satisfaction or because the system behaves in an unexpected way would be reduced if the complex system were modelled more accurately. However, the problem of using high fidelity models is related to the computational cost involved. High-fidelity models can consist of physical experiments or numerical simulations, both of which are expensive and time consuming. In the early stages of design, the evaluation of numerous configurations is important and therefore this is usually only possible with the use of low fidelity models. Multi fidelity frameworks offer the opportunity to overcome the problems exposed above. By effectively combining information from low-fidelity and high-fidelity models, multi-fidelity methods aim to accelerate design exploration providing accurate information with respect to physical reality.

Models are used to support many aspects of computational science and engineering. One primary purpose of a model is to characterize the input-output relationship of the system of interest. The input describes the system properties and environmental conditions and the output describes quantity of interest to the task at hand. Usually in engineering applications evaluating a model means performing numerical simulation that implements the model, computes a solution, and thus maps an input onto an approximation of the output [13].

Mathematically, a model is a function $f: Z \rightarrow Y$ that maps an input $z \in Z$ to an output $y : Y$, where $Z \subseteq \mathbb{R}^d$ is the domain of the inputs of the model, with dimension $d \in \mathbb{N}$ and $Y \subseteq \mathbb{R}^{d'}$ is the codomain of the model, with dimension $d' \in \mathbb{N}$. Model evaluations incur computational costs $c \in \mathbb{R}_+$ that tipically increase with the accuracy of the approximation of the output, where $\mathbb{R}_+ = \{x \in \mathbb{R} : x > 0\}$.

In many applications, different models with different levels of accuracy and computational cost are available. We define low-fidelity model $f_{lo}: Z \rightarrow Y$ as a model that estimates the same output with lower accuracy and a cost $c_{lo} \in \mathbb{R}_+$. The cost $c_{hi} \in \mathbb{R}_+$ of the high-fidelity model is typically higher than the low fidelity one but also the accuracy of information is higher.

An important class of optimization methods are surrogate-based optimization (SBO) in which the search process is simulation-based and at each new sampling point, rather than invoke the high-fidelity model, a less computationally expensive surrogate model is query.

According to Eldred et al. [25], surrogate models leveraged in SBO can be obtained in three ways: data-fit models, hierarchical models, and projection-based models. Surrogate data fit is a non-physics-based approximation of a set of data generated by the original model, using interpolation or regression techniques. The model hierarchy surrogate is physic-based but is derived as a lower fidelity model (e.g. coarser discretization, reduced element order, relaxed solver tolerances, omitted physics etc.) and used in place of the high-fidelity model. The projection-based surrogate is derived directly from the high-fidelity model using a reduced basis and projection of the original high-dimensional system down to a small number of generalized coordinates.

The classification proposed by Eldred et al. is significant in SBO based on multifidelity surrogate. In this case, the surrogate model is built combining information from multiple sources, known a priori a fidelity hierarchy of the models. These multiple information sources can be obtained from the High-fidelity model, that is our maximum accuracy believe, by data-fit models, projection-based models, and hierarchical models.

Peherstorfer et al. [13] reviewed MF approaches within the context of uncertainty quantification and defined three categories of methods approaching the combination between data from multiple sources: “adaptation”, “fusion” and “filtering”.

The adaptation strategy enriches the low fidelity model with information from the high-fidelity model while the computation is done. One example is global optimization with efficient global optimization (EGO) presented in [34, 35, 101, 102, 103], where the adaptation process is applied to a kriging model in each iteration of the optimization. A second example is the correction of low fidelity model outputs with updates coming from the high-fidelity model [36, 37] or with Gaussian process models to predict better the output of the high-fidelity model [38, 104].

The fusion strategy evaluates low and high-fidelity models and then combine information from all outputs. An example is Cokriging multi-fidelity method where a surrogate model is derived from multiple high and low fidelity information sources [39, 40, 41, 105]. Another example is the control variate framework [42, 43, 44, 106, 107], where the variance of Monte Carlo estimators is reduced by the correlation between high and low fidelity models.

The last model management strategy is based on filtering, where the high-fidelity model is invoked following the evaluation of a low-fidelity filter. One example of MF filtering strategy is the two stage MCMC algorithms [45, 46, 108]. Another example is the exploration of the stochastic space with low-fidelity model to derive sampling points at which the high-fidelity model is then evaluated [47]. A third example is the MF importance sampling, where the high-fidelity model sampling is guided by an importance sampling biasing distribution constructed with a low-fidelity model [48].

Beran et al. [24] considered MF analysis or design of a system when different mathematical descriptions (i.e. different physics represented by different governing equations, boundary conditions or parametric attributions) are used in cooperation during design or analysis procedure. Therefore, MF methods are considered a class of approaches that manipulate a set of information sources to accelerate computational task. The information sources quantify system response using both computational and non-computational approaches.

Consequently, the authors noted that multi-information-source (MIS) (i.e. synergistic use of solutions of the same equations with different meshes considered or cooperative use of information source with surrogate extracted numerically from the information source [26]) methods are similar in form to MF when the relevant responses are high-level quantities of interest. Multi-information-source approaches optimize expensive black-box objective functions while optionally accessing cheaper biased noisy approximations referred as information sources [109]. The information sources quantify system response using both computational and non-computational approaches.

The categorization of potential MIS adaptations proposed is: “governing equations”, “coupling”, “geometry”, “discretization”, “convergence” and “data”. The governing equations are derived from modelling assumptions and refinements: higher or lower fidelity models can be considered. More or less coupling between physics disciplines and more or fewer details in geometry description of the problem can be also included. Numerical adaptations are resumed in finer or coarser discretization and tighter or looser tolerances in convergence. The last category is referred to use more or less data from a particular information source or making the decision to use more or fewer information sources.

Beran et al categorize various MF approaches in: “by quantity of interest (BQ) “, “by intermediate variables (BIV)”, “by system element (BSE)”, “by physical zone (BPZ)” and “by machine learned calibration (MLC)”.

In BQ approaches the quantities of interest are sampled in different points of the parameter space and computed using models of different fidelity for one or more system elements. In [27, 28] can be found both design oriented and data centric examples. BIV approaches use a data-driven surrogate-refinement process (similar to BQ) but sample intermediate variables across the information sources. In [29, 30] are shown analysis oriented and data centric examples.

In BSE fidelity across system elements sharing a common discipline can be changed and are computed across the information sources. Analysis oriented and physics-centric example are in [31].

BPZ approaches model part of the physical domain across the information sources, so the highest fidelity model may not be solved over the entire domain relevant to the calculation. An example can be found in [32].

In MLC approaches states across the information sources are correlated. In [33] there are examples design oriented and physics centric.

In this thesis we will refer to multi-fidelity methods in all cases in which two or more sources of information are categorized according to hierarchical levels. We consider two key properties of multi-fidelity methods (in accordance with what is reported by Peherstorfer et al. [13]):

1. Low-fidelity models $f_{lo}^{(1)}, \ldots, f_{lo}^{(k)}$ that provide useful approximations of the high-fidelity model f_{hi}. An important element of this property is the use of explicit low-fidelity models that approximate the same output quantity as the high-fidelity model.
2. A model management strategy that distributes work among the models while providing theoretical guarantees that establish the accuracy and/or convergence of the outer-loop result.

4.1. Multifidelity Optimization

Countless heuristic techniques have been used in literature to optimize a high-fidelity function using information from low-fidelity functions. In this thesis the multi-fidelity optimization approaches (MFO) will be considered.

Multi-fidelity optimization approaches generally aim to identify the global optimum of the high-fidelity function, but formal mathematical proofs that this will happen are not always available. These approaches are referred as heuristic methods, ranging from problem-specific formulations to computational strategies that calculate a probability of finding an improved high-fidelity function value.

In contrast, non-heuristic methods are defined as those in which, given a set of requirements for the initial design and behaviour of the high-fidelity and low-fidelity functions, there is a mathematical guarantee that the multi-fidelity method will achieve the optimum of the high-fidelity function. However, the non-heuristic multi-fidelity methods could take significantly longer than the single-level fidelity function given the acquisition of expensive high-fidelity data.

Two main approaches are considered in the discussion of MFO: global and local. Global optimization methods search the best design across the feasibility domain, while local optimization methods try to find the closest and best performing design compared to all other nearby designs. The benefits of global methods lie in the fact that it is not necessary to estimate the gradient of the high-fidelity function. Frequently, in fact, the gradient of a high-fidelity function cannot be estimated or is determined with low accuracy. Examples of this are when a high-fidelity function is a simulation with a finite convergence tolerance, it is a black-box function, the output contains random noise, it can occasionally fail to provide a solution, or it is a physical experiment. In these cases, the high-fidelity gradient is not available and cannot be accurately estimated. However, using global MFOs typically requires considerably more high-fidelity function evaluations when compared to local methods. In the following paragraphs some of the global and local methods which can be found in literature are presented.

Commonly, the multifidelity optimization process can be referred as offline/online or active learning searching for the optimum. Offline/online techniques built a surrogate model, combining information from models at different fidelities before the iterative optimization process, and leverage the surrogate in the searching for the optimum. Active learning approaches built an initial surrogate in the same way done by offline/online techniques. The difference is that during the iterations, the information obtained during the sampling are leveraged to update the surrogate model, guiding the exploration and exploitation towards the optimum.

4.1.1. Global Multifidelity Optimization

Typically, gradient-free optimization frameworks interpolate high-fidelity objective function with several strategies. Offline-online methods represent a significant category. This technique provides a data set from different information sources offline (e.g. experimental, numerical or empirical models results) and then built the surrogate function online, combining data from high and low fidelity models by means of a Gaussian Process, in order to approximate the objective function. An example of this methods is Cokriging surrogate models. Cokriging method [49, 50] is commonly known as an extension of kriging including multiple levels of fidelities in the surrogate construction. The typical implementation of Cokriging methods in optimization is to create a global surrogate model of the expensive function. As shown in [51], update iteratively the global Cokriging models during optimization often results in finding a high-fidelity optimum faster than with other calibration models, even when the number of calibration points is limited from the computational cost [52]. The main problem related to Cokriging approaches is that the convergence to high-fidelity optimum is not guaranteed and without controlling the location of calibration points and the optimization steps taken, also the local convergence is not ensured. Another class of global methods are Goal-driven methods, where a Bayesian framework is used in order to drive the adaptive sampling. Several Bayesian framework-based approaches can be found in the literature. Many of these turn out to be derivations from the Efficient Global Optimization (EGO) technique. In EGO approaches the mean of the Gaussian process interpolates the value of the high-fidelity function value, while the mean square error of the Gaussian process models the uncertainty in the high-fidelity function value. This error estimate is zero at all locations where the value of the high-fidelity function is known and increase with the distance away from sample points. Optimization is then performed on the Gaussian process model, and the high-fidelity function is sampled at locations likely to reduce to the value of the function over the current observed minimum. This method can be made globally convergent, as illustrated in [53], and the high-fidelity derivative is not required. However, this method may be globally biased and attempt to explore the entire design space. In addition, EGO is sensitive to the initial high-fidelity samples [53] and to the exact metric of selecting points likely to improve the high-fidelity function value [54]. Examples of approaches related to EGO can be found in [55, 56] where in searching performance improving points both improvement and risk are considered objectives in multi-objective methods and Regis et al. work [57], a provably convergent formulation that considers both the value of the Gaussian process and the distance from the previous interpolation point. Kennedy and O'Hagan enabled EGO to be used in a multi-fidelity contest, calibrating lower-fidelity models with Gaussian process models to best predict the output of high-fidelity models [58]. They also performed a complete Bayesian uncertainty analysis of calibrating computer models to true physical processes and determined where the uncertainties arise [59]. The challenge with global optimization is that to

proving that a global optimum within the domain has been reached is necessary demonstrating that the sequence of trial points of the algorithm becomes dense, or that function has been evaluated in all arbitrarily small intervals of the domain. With the limited exception of some cases (where there are known bounds on smoothness) global optimization methods requires an immense number of evaluations in order to find the optimum. However, local optimization techniques can guarantee the best design in the neighbourhood of a starting point often quickly and with few evaluations.

4.1.2. Local Multifidelity Methods

A typical example of local multi-fidelity methods is the use of Trust regions algorithms. Trust regions have been frequently implemented in optimization problems and are probably convergent. Commonly trust region-based algorithm optimize a function using its gradient and Hessian at the centre of a trust region as a quadratic surrogate model of the function. The trust region then determines the allowable step length that may be taken based on how well the function can be represented by a quadratic surrogate. A significant extension of trust region to a general multi-fidelity optimization approach is given by Alexandrov et al. [60, 61, 62]. They demonstrate that general low-fidelity models can be adjusted to be a surrogate model upon which optimization is performed in a manner provably convergent to an optimum of the high-fidelity function. The requirement for the surrogate is that at the centre of each trust region the surrogate model and the high-fidelity function must have equal value and equal gradient, referred in literature as first-order consistency. Alexandrov et al. Proposed a multiplicative and an additive correction to adjust and arbitrary low-fidelity function so that a lower-fidelity surrogate satisfying the first order consistency requirement can be created [38]. When a high-fidelity gradient is not available and cannot be estimated accurately, the first-order consistency criteria cannot be satisfied. In this case, interpolation-based surrogate models are also a common choice. Examples include that by Conn et al. [63, 64] or Marazzi et al. [65] and are based on polynomial interpolants and trust regions. Further work by Conn et al. on interpolation [66] and on derivative-free trust regions methods [67, 68] have been used by Wild et al. to develop a derivative-free optimization algorithm that is provably convergent to a high-fidelity optimum also using a radial-basis function interpolant [69, 70]. Then March et al. [71] combined the probably convergent optimization frameworks of Wild et al. [69, 70] and Conn et al. [66] with Bayesian model calibration ideas to result in a probably convergent multi-fidelity optimization approach that does not require high-fidelity gradient information. These latter techniques reduce the number of high-fidelity calls required to find a local minimum compared to other previous methods. However, the methods are sensitive to the quality of low-fidelity model dramatically with the number of design variables and the number of calibration points used to build the radial basis function model.

4.2. Open Issues in Multifidelity Optimization

In this paragraph are summarized some of the most critical challenges in developing a multi-fidelity algorithm in design optimization contest following the discussion proposed by [24]:

1. *Curse of Dimensionality:* the growth of computational cost with problem dimension, multidisciplinary, physics coupling and in general system complexity.
2. *Noise and Localized Behaviours:* surrogate models constructed to fit sampled data are susceptible to spurious oscillations (errors) in presence of noise. Noise can be generated by experimental data or a consequence of numerical errors in analysis. Also changes in physical behaviour are relevant because models describing physical outcomes may have difficulties in coping with these transitions. The consequence is an increased model stiffness and number of samples required.
3. *Inconsistent Parameterization:* parameter inconsistency between model, for example, when model at lower fidelity levels use coarser geometric descriptions requiring fewer parameters.
4. *Implementation Complexity and Computational Efficiency:* overly complex and stiff procedures requiring expert knowledge of the multi-fidelity method; method; overly intrusive procedures requiring expert knowledge of the application when integrated with the multi-fidelity method; distribution barriers arising from the use of legacy software, proprietary software, or software with elaborate build requirements, and computational inefficiencies in method implementation (and potentially formulation) related to, for example, parallelization and scaling.

5. Multifidelity Bayesian Optimization

Multifidelity optimization frameworks leverage the availability of analytical models characterized by different level of fidelity. In most cases, high-fidelity data includes information produced by stochastic process, physical models or real-life experiments that closely match the context of interest.

However, these data are expensive to obtain in terms of both time and money, which limits the amount of data that can be obtained and so can significantly impair the ability of the model to produce valid estimates. Low-fidelity data are information coming from a stochastic process, physical models or real-life experiments that deviates from the real-world system of interest (e.g. introducing approximations, neglecting physical effects, coarser discretization, and resolutions of numerical models). This information is inexpensive to acquire, so it is possible to elicit larger amount of data.

Multifidelity Bayesian Optimization enrich the Gaussian process combining a large amount of cheap low-fidelity data with a reduced number of expensive and accurate high-fidelity information. This potentially increase the knowledge of the objective function leveraging the effectively enrichment of low-fidelity data with high-fidelity data.

5.1. Multifidelity Gaussian Processes

The Gaussian process regression can be extended to combine different sources of information in a single probabilistic model. Assuming that $y^{(1)}(x), \dots, y^{(M)}(x)$ observation values are available at M different levels of fidelity, where $y^{(1)}$ is the lowest fidelity and $y^{(M)}(x)$ the highest. The training dataset $\mathcal{D}_n = \left\{\left(x_i, y^{(m)}(x_i), m_i\right)\right\}_{i=1}^{n}$ is then composed by the paired input/output observation $(x_i, y^{(m_i)}(x_i))$, generated by the m_i unknown mapping function $y^{(m)}(x) = f^m(x) + \epsilon$, where the measurement noise $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon)$ is assumed to have the same distribution over the fidelities.

One of the possibilities mentioned in the literature is the Gaussian process regression (MF-GP) autoregressive scheme proposed by Kennedy and O'Hagan [8]. The following assumptions are posed by the authors:

1. Different levels of the same code are correlated in some way. Extra complexity is usually achieved by expanding simple models, so that each level of code should share some basic features.
2. The codes have a degree of smoothness, in the sense that the output values for similar inputs are reasonably close. If the codes are extremely rough, then individual runs can only provide information about the output in a small surrounding neighbourhood, and the advantage of the Bayesian model is minimal.
3. Prior beliefs about each level of the code can be modelled using Gaussian process.
4. Each code output is scalar. Computer codes often produce multivariate time series output, so the method must be generalised assuming multivariate normality for the outputs.

The lowest fidelity function prior can be represented as a Gaussian process prior:

$$f^{(1)} = GP(0, k_1(x, x')) \tag{32}$$

Where k_1: $\mathbb{R}^{d\times d} \rightarrow \mathbb{R}$ is the kernel function. The higher fidelities function can be defined recursively as:

$$f^{(m)}(x) = \rho f^{(m-1)}(x) + \delta^{(m)}(x) \quad m = 2, \dots, M \tag{33}$$

where ρ is a constant factor that sales the contribution of the preceding fidelity to the following one, and $\delta^{(m)}(x) \sim GP(0, k_m(x, x'))$ models the bias between fidelities.

According to the autoregressive formulation:

$$cov\left[f^{(m)}(x), f^{(m-1)}(x') \middle| f^{(m-1)}(x)\right] = 0 \quad \forall x \neq x' \tag{34}$$

Which can be interpreted as a Markov property: given the point $f^{(m-1)}(x)$ nothing more cam be learnt about $f^{(m)}(x)$from any other model evaluation $f^{(m-1)}(x')$, for $x \neq x'$ [8, 46]. A covariance function between a pair of samples $\left\{\left(x_i, y^{(m_i)}(x_i), m_i\right), \left(x_j, y^{(m_j)}\left(x_j\right), m_j\right)\right\}$ can be written as:

$$k\left((x_i, m_i), \left(x_j, m_j\right)\right) = cov\left[f^{(m_i)}(x_i), f^{(m_j)}(x_j)\right] \tag{35}$$

As previous described, $\boldsymbol{K} \in \mathbb{R}^{n \times n}$ is the kernel matrix, such that $\boldsymbol{K}(i,j) = k\left((x_i, m_i), \left(x_j, m_j\right)\right)$. The predictive distribution of the MF-GP is defined by the predictive mean and variance:

$$\mu^{(m)}(x) = k_n^{(m)}(x)^T(\boldsymbol{K} + \sigma_\epsilon I)^{-1} y \tag{36}$$

$$\sigma^{2(m)}(x) = k\big((x,m),(x,m)\big) - k_n^{(m)}(x)^T(\boldsymbol{K} + \sigma_\epsilon I)^{-1} k_n^{(m)}(x) \tag{37}$$

Where

$$k_n(x) = \left(k\big((x,m),(x_1,m_1)\big), \dots, k\big((x,m),(x_n,m_n)\big)\right)^T$$
$$y = \left(y^{(m_1)}(x_1), \dots, y^{(m_n)}(x_n)\right)^T.$$

5.2. Multifidelity Acquisition Function

In the multi-fidelity context, given the availability of different models with different levels of fidelity, the Bayesian optimization process will not only have to determine the next candidate for each iteration, but will also have to choose which level of fidelity it is appropriate to query. Several approaches to this challenge can be found in literature. Some of the most significant will be presented in the following chapters.

5.2.1. Multifidelity Predictive Entropy Search

The multi-fidelity acquisition function α should be constructed to enable the multi-fidelity BO algorithm to jointly and naturally optimize the non-trivial trade-off between exploitation vs. exploration over the target and auxiliary functions with varying fidelities for finding or improving the belief of the global target maximize. In order to do this, multi-fidelity predictive entropy search (MFES) approaches follow the idea of information-based acquisition function proposed by Henning et al. and Hernandez et al. in [156, 157]. The selection of the next candidate is based on maximize the expected reduction of the negative differential entropy of the posterior distribution $p(x_*|\mathcal{D}_n)$ of the location x_* of the global maximum (in [156, 157] the optimization problem is posed with the maximization of the objective function). The corresponding acquisition function is:

$$\alpha_n(x) = H[p(x_*|\mathcal{D}_n)] - \mathbb{E}_{p(y|\mathcal{D}_n, x)}[H[p(x_*|\mathcal{D}_n \cup \{(x, y)\})]] \tag{38}$$

Where $H[p(x)] = -\int p(x) log p(x) dx$ represents the differential entropy of its argument and the expectation above is taken with respect to the posterior predictive distribution of y given x. The exact evaluation of Eq. (38) is infeasible in practice. A direct evaluation of Eq. (38) is possible only after performing many approximations [158]. To avoid this, predictive entropy search approach rewrite Eq. (38) as the mutual information between x_* and y given $\mathcal{D}_n$. Since the mutual information is a symmetric function, $\alpha_n(x)$ can be made explicit:

$$\alpha_n(x) = H[p(y|\mathcal{D}_n, x)] - \mathbb{E}_{p(x_*|\mathcal{D}_n)}[H[p(y|\mathcal{D}_n, x, x_*)]] \tag{39}$$

Where $p(y|\mathcal{D}_n, x, x_*)$ is the posterior predictive distribution for y given the observed data $\mathcal{D}_n$ and the location of the global maximiser of f . A multifildelity BO algorithm has to repeatedly selects the next candidate and the level of fidelity to query. The Eq. (39) can be extended to a multifidelity framework [159]:

$$\alpha(y_X, \langle x, i\rangle) = H(y_i(x)|y_X) - \mathbb{E}_{p(x_{*t}|y_X)}[H(y_i(x)|y_X, x_{*t}] \tag{40}$$

Where $\langle x, i\rangle$ is the next input for evaluating the *i*-th function f_i at x that maximizes the Eq. (40) given the past observations (X, y_X). The first Gaussian predictive/posterior entropy term can be computed analytically:

$$H(y_i(x)|y_X) \triangleq 0.5 \log\left(2\pi e\left(\sigma^2_{\langle x,i\rangle|X} + \sigma^2_{n_i}\right)\right) \tag{41}$$

Where

$$\sigma^2_{\langle x,i\rangle|X} \triangleq \Sigma_{\{\langle x,i\rangle\}\{\langle x,i\rangle\}|X}$$

More details can be found in [159].

5.2.2. Multifidelity Max-Value Entropy Search

This approach employs the information-based framework [156, 157]. In the case of MFBO the information gain for identifying the maximum of the highest fidelity function $f_* = \max_{x \in \mathcal{A}} f_x^{(M)}$ is derived by observing an arbitrary fidelity observation. Considering the case that a query is sequentially issued after the previous one is observed, which can be referred to as *sequential querying* (in [125] can be found also a *parallel querying* approach). Defining a training data set $\mathcal{D}_t$, to decide the next sample x_{t+1} and level of fidelity m_{t+1} the acquisition function is defined, recalling the formulation of Takeno et al. in [125], as:

$$\alpha(x, m) = I\left(f_*; f_x^{(m)} \middle| \mathcal{D}_t\right) / \lambda^{(m)} \tag{42}$$

Where $I\left(f_*; f_x^{(m)} \middle| \mathcal{D}_t\right)$ is the mutual information between f_* and $f_x^{(m)}$ conditioned on $\mathcal{D}_t$ and $\lambda^{(m)}$ is the querying cost associated at the m-th fidelity. By maximizing $\alpha(x, m)$ we obtain a pair of the input x and the fidelity m which maximally gains information of the optimal value f_* of the highest fidelity per unit cost.

The mutual information can be written as the difference of the entropy [125]:

$$\begin{aligned} & I\left(f_*; f_x^{(m)} \middle| \mathcal{D}_t\right) \\ & = H\left(f_x^{(m)} \middle| \mathcal{D}_t\right) - \mathbb{E}_{(f_* | \mathcal{D}_t)}[H(f_x^{(m)} | f_*, \mathcal{D}_t] \end{aligned} \tag{43}$$

Where $H(\cdot \mid \cdot)$ is the conditional entropy of $p(\cdot \mid \cdot)$. The first term in the right-hand side of Eq. (43) can be derived analytically for any fidelity m: $H\left(f_x^{(m)} \middle| \mathcal{D}_t\right) = \log\left(\sigma_x^{(m)} \sqrt{2\pi e}\right)$ where $e = exp(1)$. The second term in Eq. (43) takes the expectation over the maximum f_*. Since an analytical formula is not known for this expectation, a possibility is to employ Monte Carlo estimation by sampling f_* from the current GP regression:

$$\mathbb{E}_{(f_*|\mathcal{D}_t)}[H(f_x^{(m)}|f_*,\mathcal{D}_t] \approx \sum_{f_* \in F_*} \frac{H(f_x^{(m)}|f_*,\mathcal{D}_t)}{|F_*|} \tag{44}$$

Where F_* is a set of sampled f_*. For more details please refer to [125].

5.2.3. Multifidelity Expected Improvement

Multi-fidelity expected improvement (MFEI) is based on the expected improvement formulation presented before, corrected as an integrated search criterion that determines both location and fidelity level of the subsequent evaluation. The following MFEI formulation presented by Huang et al. in [160] and implemented with an improved search algorithm in [110], is here reported:

$$MFEI(x,m) = \mathbb{E}[\max\left(f^{(M)}(x) - f^{(M)}(x^+), 0\right] \alpha_1(x,m)\alpha_2(x,m)\alpha_3(x,m) \tag{45}$$

Where the first term is simply the EI evaluated at the highest fidelity. The utility functions $\alpha_1(x,m), \alpha_2(x,m)\ and\ \alpha_3(x,m)$ are defined as:

$$\alpha_1(x,m) = corr\left[f^{(m)}(x), f^{(M)}(x)\right] \tag{46}$$

$$\alpha_2(x,m) = 1 - \frac{\sigma_\epsilon}{\sqrt{\sigma^{2(m)}(x) + \sigma_\epsilon^2}} \tag{47}$$

$$\alpha_3(x,m) = \frac{\lambda_M}{\lambda_m} \tag{48}$$

The term $\alpha_1(x,m)$ is designed to account for the reduction in reward when a lower-fidelity evaluation is used. When $l = m$ the term $\alpha_1(x,m)$ should be equal to one. Also, $\alpha_1(x,m)$ should be zero when sample at (x,m)) exists already and there is not random error, as such a replicate brings no additional benefit. In Eq. (45) $\alpha_1(x,m)$ is the correlation between the posterior estimate of system m and the posterior estimate of system M at location x, because can be interpreted as the fraction of uncertainty on system M that can be eliminate once system m is known.

The purpose of the term $\alpha_2(x,m)$ is to adjust EI when outputs of system m contain random errors. It accounts for the diminishing return of additional replicates as the prediction becomes more accurate. This factor is equal to the relative reduction in the posterior standard deviation after a new replicate is added. This factor equals one when the variance of the random errors is zero.

The term $\alpha_3(x,m)$ is the ratio between the cost-per-evaluation on the higher fidelity system and that on system m. It represents an adjustment to the sampling strategy based on the evaluation costs. With the expected gains equal, a cheaper data point is preferred to a more expensive one

To summarize, each modifier term in Eq. (45) has an important benefit. If $\alpha_1(x,m)$ is excluded, we will always choose on the lowest-fidelity system as it is cheaper. If $\alpha_2(x,m)$ is excluded, we will see many unnecessary replicates when random errors exist. And if $\alpha_3(x,m)$ is excluded, the cost-per-evaluation will be out of the picture, so the highest fidelity will always be chosen.

6. Numerical Implementation of Algorithms

This chapter resents an overview of the numerical implementation of the single fidelity expected improvement (SFEI) algorithm and the multi-fidelity expected improvement (MFEI) algorithm. Both SFEI and MFEI algorithm are implemented in Matlab in the form of function packages. The algorithms will be presented in the form of meta code.

6.1. Single Fidelity Bayesian Optimization

The single fidelity Bayesian optimization algorithm is based on the expected improvement approach in Eq. (31).

In Tab.3 is presented the meta code of the algorithm. The input is the black-box function $f(x)$ to be optimized and a Matlab struct opt that includes the algorithm setup. More details of opt struct is given in Tab.2. The setup elements are presented in the Matlab nomenclature, so an element of the algorithm setup is defined as $opt.setup\_element$. The term $opt.dims$ the dimension of the objective function is specified. The parameters $opt.mins$ and $opt.maxes$ respectively report the minimum and the maximum value assumed by the objective function variables. The maximum number of iterations to be performed are defined in $opt.\max\_iters$. The parameter $opt.grid\_size$ define the number of points of the sampling grid. In order to perform the Gaussian process, the mean function, the kernel function, and the inference method are essential to characterized and are defined respectively in $opt.meanfunc, opt.covfunc$, and $opt.infmethod$.

struct: "***opt***"
$opt.dims$: specifies the number of parameters of the objective function.
$opt.mins$: the minimum value of each parameter.
$opt.maxes$: the maximum value of each parameter.
$opt.\max\_iters$: number of iterations of Bayesian optimization to perform.
$opt.grid\_size$: number of candidate points to sample in.
$opt.meanfunc$: mean function used in Gaussian Process.
$opt.covfunc$: Kernel function used in Gaussian Process.
$opt.infmethod$: Inference method.

Tabel 2: Matlab struct opt *that includes the algorithm setup.*

At the beginning of the optimization process, an initial candidate hyper grid is drawn from a random Latin hypercube.

As presented in chapter 2, Latin hypercube sampling aims to spread the sample points evenly across all possible values. It partitions each input distribution into N intervals of equal probability and select one sample from each interval.

Once the initial samples x_1 are determined, the initial values $f(x_1)$ of the objective function can be computed and the main Bayesian Optimization single fidelity loop can start.

Considering the *i*-th iteration, the mean μ and variance σ^2 of the Gaussian process are determined by enriching the posterior with the information about the objective computed in the previous step. The posterior is obtained given the inference method, the mean function, and the covariance function.

The inference method for Gaussian likelihood reduces to compute mean and covariance of a multivariate Gaussian. The mean function selected is the constant value and the kernel function selected is the linear kernel.

The next computational step is the definition of the expected improvement (see Eq. (31) for more details) and the maximization in order to select the next sample x_{i+1} of the iteration $i + 1$.

Finally, samples and values matrix are uploaded with respectively the new selected sample x_{i+1} and the value of the objective in that sample $f(x_{i+1})$.

After all iterations, the minimum value of the objective $argmin(values)$ and the corresponding minimum sample $argmin(samples)$ are determined.

SFEI Algorithm

$[minsample, minvalue] = SFEI(f(x), opt)$

1: Draw initial candidate grid from a random Latin hypercube

$$hyper\ grid = Latinhypercube(opt.dims, opt.gridsize, opt.mins, opt.maxes)$$
$$\rightarrow hyper\ grid \rightarrow x_1 \rightarrow\ samples = [samples; x_1]$$

2: Get values from the first samples

$$\rightarrow f(x_1) \rightarrow values = [values; f(x_1)]$$

3: Main Bayesian Optimization Single Fidelity Loop

$\boldsymbol{for}\ i = 1{:}\ opt.\max\ \_iters$

$$f(x) \sim GP$$

$[\mu, \sigma^2] = GP(hypergrid, samples, values, opt.infmethod, opt.meanfunc, opt.covfunc)$

$$best = \min\ (values)$$
$$EI = computeEI(\mu, \sigma^2, best)$$
$$\max(EI) \rightarrow x_{i+1}$$

4: Upload Samples and Values

$$samples = [samples; x_{i+1}]$$

$$values = [values; f(x_{i+1})]$$

end

5: Get minvalue and minsample

$$minsample = \min(samples)$$

$$minvalue = \min(values)$$

Table 3: SFEI Algorithm meta code.

6.2. Multifidelity Bayesian Optimization

The Multifidelity Bayesian optimization algorithm is based on the multifidelity expected improvement formulation presented in Eq. (45).

The initial candidate grid is drawn from a random Latin hypercube and the setup parameters are summed up in a Matlab struct as the SFEI algorithm.

Given the different fidelity levels of the objective function, the following matrix are defined in the meta code presented in Tab. 4:

$$F(x) = \left[f^{(1)}(x), \dots, f^{(m)}(x), \dots, f^{(M)}(x)\right]$$

$$samples_1 = \left[samples_1^{(1)}, \dots, samples_1^{(m)}, \dots, samples_1^{(M)}\right]$$

$$values_1 = \left[values_1^{(1)}, \dots, values_1^{(m)}, \dots, values_1^{(M)}\right]$$

$$m = 1, \dots, M \qquad levels\ of\ fidelity$$

Where $samples_1$ is a matrix containing the initial samples related to the m-th level of fidelity, $values_1$ are the initial value of the m-th level of fidelity objective function calculated in the corresponding sample vector.

Computed the initial samples and values, the main Bayesian optimization multifidelity loop starts performing the iterations.

Considering the *i*-th iteration, the Gaussian process posterior is derived from the data computed in the previous iterations from all the m-th fidelity levels of the objective function.

The next step is computing the terms $\alpha_1\left(m, x_i^{(m)}\right), \alpha_2\left(m, x_i^{(m)}\right)$ that appears in the MFEI formulation. The cost vector $\alpha_3(m)$ is derived from knowledge-based considerations, experiments, and computational times analysis. More details are given in chapter 7, where the algorithm is tested on analytical experiments.

Defined these parameters, the multifidelity expected improvement is computed. The next candidate $x_{i+1}^{(m)}$ and the m-th fidelity level of the objective function to query are determined by maximizing the MFEI.

At the end of the iterations, the minimum value of the objective function $argmin(values)$ and the corresponding minimum sample $argmin(samples)$ are determined.

MFEI Algorithm

$[minsample, minvalue] = MFEI(F(x), opt)$

1: Draw initial candidate grid from a random Latin hypercube

$$hyper\ grid = Latinhypercube(opt.dims, opt.gridsize, opt.mins, opt.maxes)$$

$$\rightarrow hyper\ grid \rightarrow x_1^{(m)} \rightarrow\ samples_1^{(m)} = \left[samples^{(m)}; x_1^{(m)}\right] \rightarrow samples_1 \quad m = 1, .., M$$

2: Get values from the first samples

$$\rightarrow f^{(m)}\left(x_1^{(m)}\right) \rightarrow values^{(m)} = \left[values^{(m)}; f^{(m)}\left(x_1^{(m)}\right)\right] \rightarrow values_1 \quad m = 1, .., M$$

3: Main Bayesian Optimization Multifidelity Loop

$\boldsymbol{for}\ i = 1{:}\ opt.\max\_iters$

$$f(x) \sim GP$$

$$\left[\mu^{(m)}, \sigma^{2(m)}\right] = GP(hypergrid, samples_i, values_i, \ldots$$
$$opt.infmethod, opt.meanfunc, opt.covfunc), \quad m = 1, .., M$$
$$\alpha_1(m, x_i) = corr\left[f^{(m)}(x_i), f^{(M)}(x_i)\right] \; m = 1, .., M$$

$$\alpha_2(m, x_i) = 1 - \frac{\sigma_\epsilon}{\sqrt{\sigma^{2(m)}(x) + \sigma_\epsilon^2}} \quad m = 1, .., M$$

$$\alpha_3(m) = \frac{\lambda_M}{\lambda_m} \quad m = 1, .., M$$

$$compute \; \mathbb{E}[\max\left(f^{(M)}(x_i) - f^{(M)}\left(argmin\, F(x_i)\right)\right]$$

$$compute \; MFEI(x, m)$$

$$\max(MFEI(x, m)) \rightarrow x_{i+1} \; \& \; m_{fidelity} level$$

4: Upload Samples and Values

$$samples_{i+1} = [samples_i; x_{i+1}]$$
$$values_{i+1} = [values_i; f^{(m)}(x_{i+1})]$$

end

5: Get minvalue and minsample

$$minsample = \min\,(samples)$$
$$minvalue = \min\,(values)$$

Table 4: MFEI Algorithm meta code.

7. Analytical Experiments

In this chapter are presented numerical experiments demonstrating the performance of the multifidelity expected improvement (MFEI) algorithm compared to a standard single fidelity expected improvement (SFEI) framework.

In the following paragraphs a description of the experiments and of the experiment setup is given, including the benchmarks description and the tests configurations; Then will be discussed the results of both MFEI and SFEI algorithm.

The tests are made on popular benchmark problems to test the efficiency and robustness of the proposed algorithm. The test functions are contrived and with no physical meaning, which is useful in demonstrating optimization methodologies. The benchmark functions were selected to exemplify different types of correlations among the fidelity levels, described in the following.

Consistently with the already used notation, $y^{(m)}(x)$ is the objective function and $m = 1, \dots, M$ indicate the level of fidelity considered, with M the highest-fidelity. MFEI, SFEI algorithms and the numerical experiments are implemented in Matlab.

7.1 Forrester Function

The first benchmark analytical function is the popular Forrester function [169]. This function is used to represent a multimodal objective function landscape, i.e. one where a search routine could become trapped in a local minimum. It is a 1-dimensional nonlinear function defined over the domain $[0, 1]$:

$$y^{(2)}(x) = (6x - 2)^2 \sin(12x - 4) \tag{49}$$

The minimum is at $x^* \backsimeq 0.727549$ and $f(x^*) \backsimeq -6.02074$. The low fidelity level is given by the linear mapping:

$$y^{(1)}(x) = 0.5y^{(2)}(x) + 10(x - 0.5) \tag{50}$$

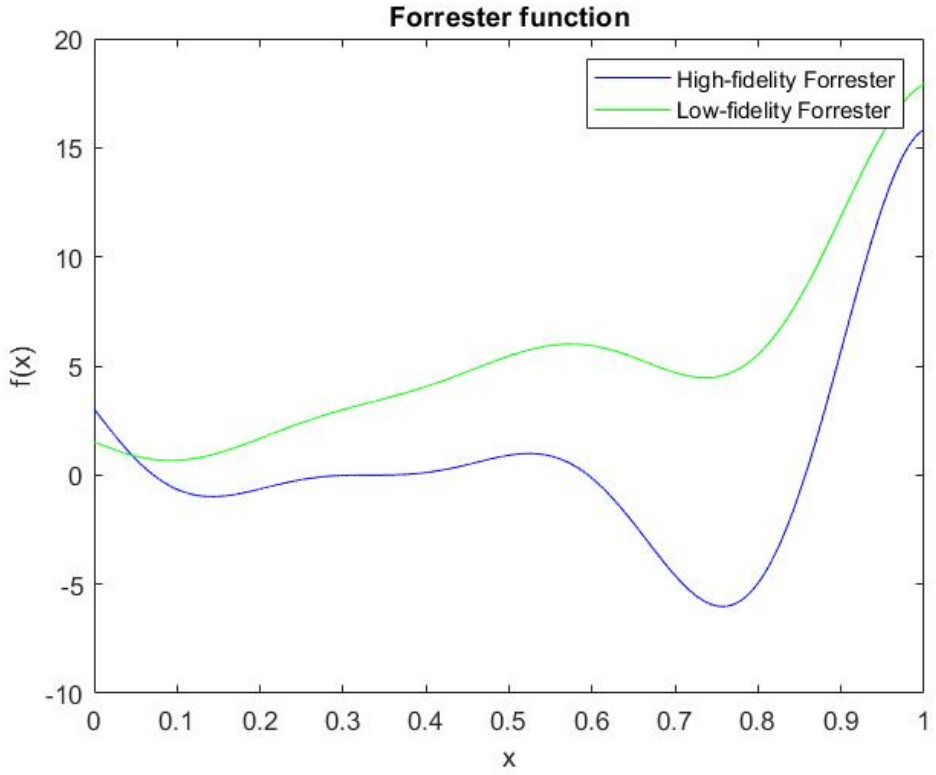

Figure 6: Forrester function. Both high-fidelity and low-fidelity representation are plotted.

7.1.1. Experiments Results

In this paragraph, the results of the SFEI and MFEI algorithm applied to the Forrester function (Eq. (49)) are considered.

In Fig.7 are reported the median and the interval of the error result of 100 single fidelity tests performed with the SFEI algorithm, for which single test an initial DOE of $n_0 = 2$ points is selected. The number of iterations performed for test are 70 with a grid of 10000 points. The error is calculated considering at each iteration the overall minimum found so far.

In the graph presented, the median shows a smooth trend in reaching the zero, highlighting the balance between exploration of the design space and exploitation moving to the global minimum. Most of the tests conducted arrive at an error of 0 in just over 50 iterations, but overall, about 60 iterations are a plausible estimate of the computational performance of the algorithm.

The test results of the MFEI algorithm are obtained under the same conditions reported for the single fidelity tests. The DOE selected is of 5 low fidelity points and 2 high fidelity points. Specifically, two levels of fidelity of the Forrester function reported in Eq. (49) and Eq. (50) are considered.

The term $\alpha_3 = [4, 1]$ was determined from the experience gained with MFEI algorithm tests considering different computational costs associated with the two high and low fidelity functions.

In Fig.8 it can be noted that not in all iterations the error tends to zero. Indeed, the interval remains up to zero with the iterations. The algorithm tends to stay in the local minima of the low fidelity function $x \backsimeq 0.08481\ y \backsimeq 0.6717$ and $x \backsimeq 0.7384\ y \backsimeq 4.463$ and in the local minimum of the high-fidelity function $x \backsimeq 0.1394\ y \backsimeq -0.9848$.

However overall, 93% of tests exhibit zero error after a maximum of 64 iterations. Furthermore, was found on average an invocation of the high-fidelity function of 7.8 times, suggesting a decrease in the total computational cost of tests compared to the SFEI case. Indeed, on average, the duration of a test using the MFEI algorithm was 126 seconds, against 318 seconds of the tests conducted with the SFEI algorithm.

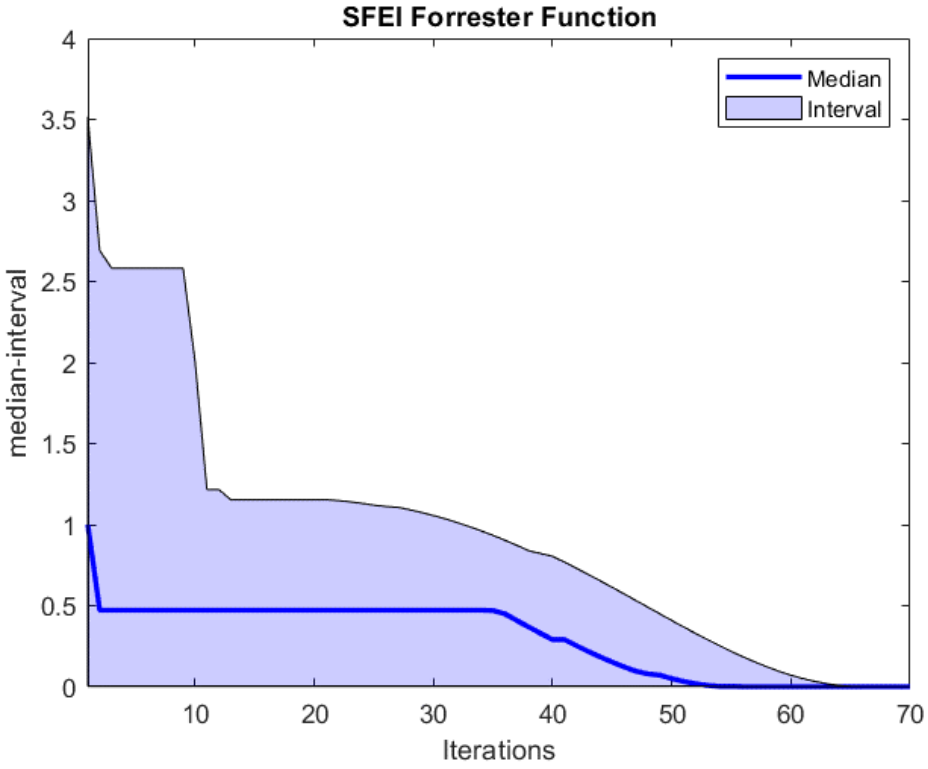

Figure 7: Median/Interval-Iterations results of the SFEI algorithm applied to the Forrester function optimization problem.

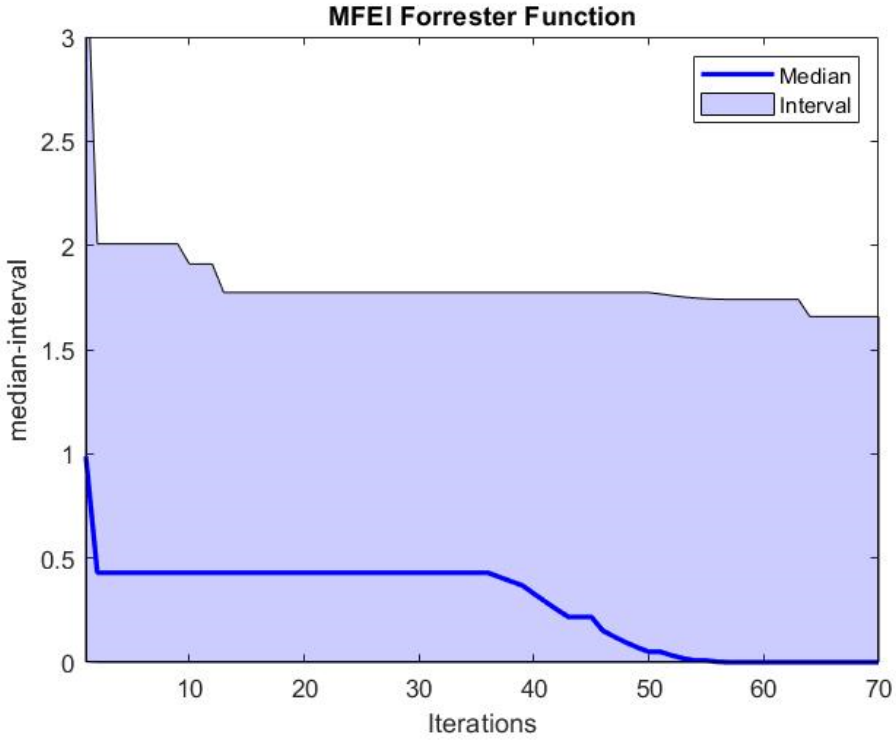

Figure 8: Median/Interval-Iterations results of the MFEI algorithm applied to the 2-fidelity Forrester function optimization problem.

7.2. Sinusoidal Squared 1D Function

The second benchmark is a sinusoidal squared 1-dimensional function [170], with domain in the interval $[0, 1]$. The high-fidelity function is defined as:

$$y^{(2)}(x) = \left(x - \sqrt{2}\right)\left(y^{(1)}(x)\right)^2 \tag{51}$$

which is a non-linear function of the low fidelity variant, given by:

$$y^{(1)}(x) = \sin(8\pi x) \tag{52}$$

The minimum is at $x^* \backsimeq 0.0619147$ and $f(x^*) \backsimeq -1.35201$.

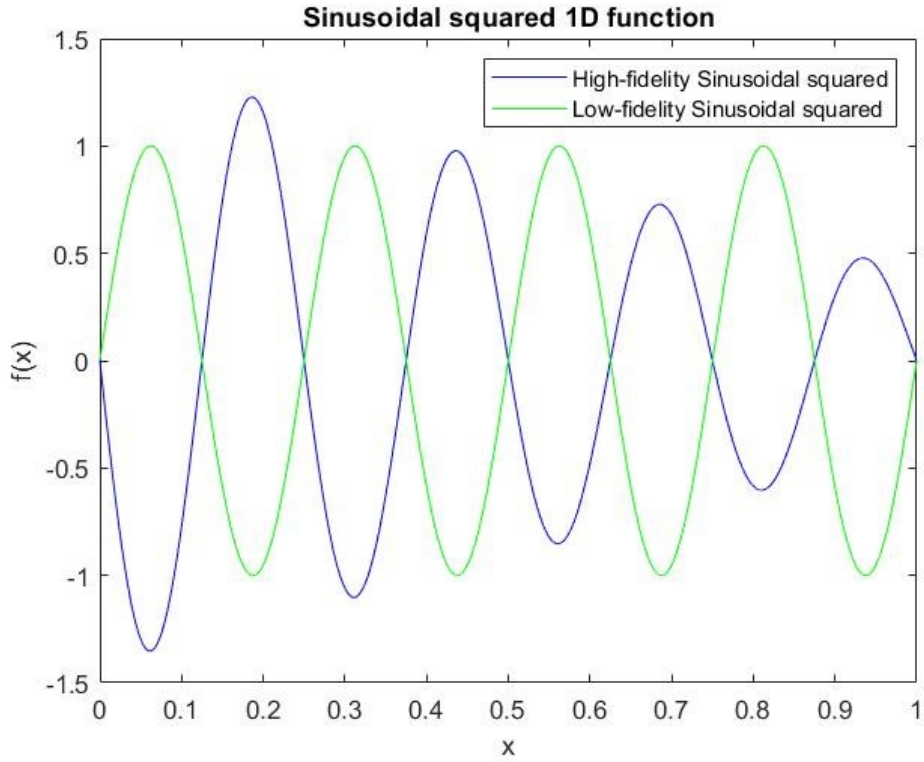

Figure 9: Sinusoidal squared 1D function. Both high-fidelity and low-fidelity representation are plotted.

7.2.1. Experiments Results

In this paragraph, the results of the SFEI and MFEI algorithm applied to the sinusoidal squared 1D function (Eq. (51)) are considered.

In Fig. 10 are reported the median and the interval of the error result of 100 single fidelity tests performed with the SFEI algorithm, for which single test an initial DOE of $n_0 = 2$ points are selected. The number of iterations performed for test are 100 with a grid of 10000 points. The error is calculated considering at each iteration the overall minimum found so far.

In the graph presented it can be seen that overall, the median trend tends towards zero with a considerably reduced number of iterations. In fact, in most cases a number of iterations less than 30 is sufficient to reach the minimum of the function.

In Fig.11 are reported the results of the MFEI algorithm considering the 2 level of fidelity reported in the paragraph above. The tests are carried out under the same conditions reported for the experiments of the SFEI algorithm. The DOE selected is of 5 low fidelity points and 2 high fidelity points.

The term $\alpha_3 = [2, 1]$ was determined from the experience gained with MFEI algorithm tests taking into account different computational costs associated with the two high and low fidelity functions as in the case of the Forrester function.

Overall, the median decreases with the number of iterations, until it reaches about zero. However, it can be noted that unlike what has been seen for the SFEI algorithm, the achievement of the null value requires a considerably higher number of iterations. Furthermore, the median seems to assume constant trends due to the achievement of the local minima of the low-fidelity function and the consequent stationing in those points.

This behavior is predictable since the low-fidelity function presents the local minima in correspondence with the local minima and maxima of the high-fidelity function alternatively. When the stationing in the local minima of the low-fidelity function is presented, it has been noted that the invocation of the maximum fidelity function is necessary in order to allow the algorithm to converge towards the global high-fidelity minimum. Nonetheless, 99% of the test cases considered converge towards the global low of the high-fidelity function

On average, the number of invocations of the high-fidelity function was 3.8 suggesting a decrease in the computational cost of the tests. Despite the significantly higher number of iterations to reach the global minimum, the computation time associated with the MFEI algorithm averages 164 seconds against 187 seconds for the SFEI algorithm.

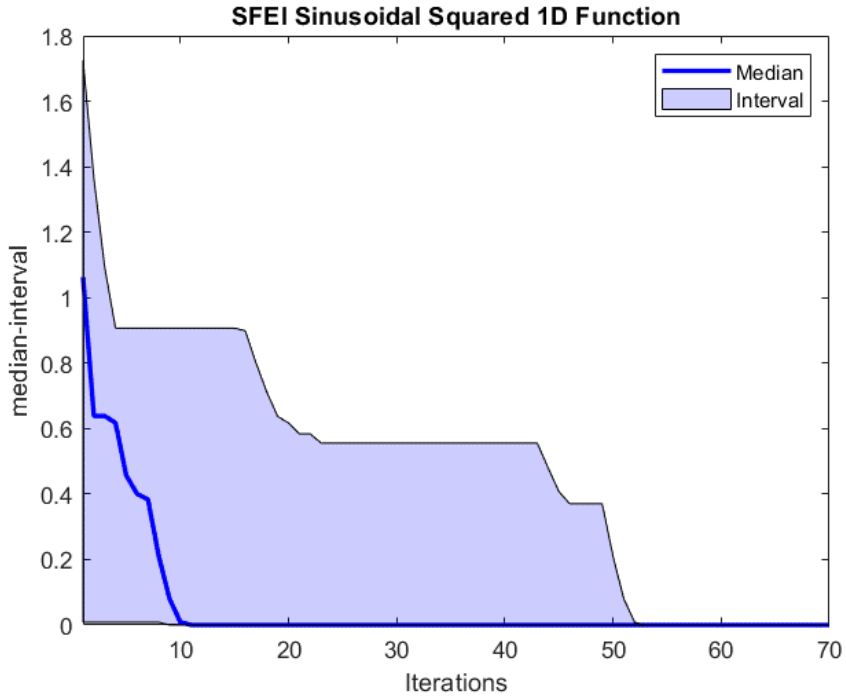

Figure 10: Median/Interval-Iterations results of the SFEI algorithm applied to the sinusoidal squared 1D function optimization problem

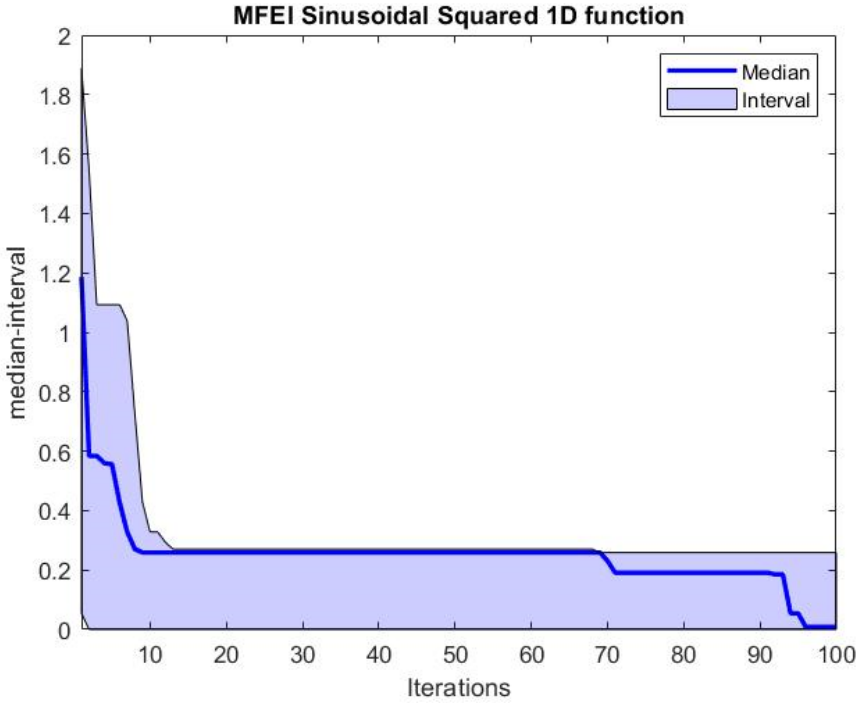

Figure 11: Median/Interval-Iterations results of the MFEI algorithm applied to the 2-fidelity sinusoidal squared 1D function optimization problem.

7.3. Rosenbrock Function

The third benchmark problem is the d-dimensional Rosenbrock function [171], a non-convex function with domain in the interval $[-2, 2]^d$ defined as:

$$y^{(M)}(x) = \sum_{i=1}^{d-1} (1 - x_i)^2 + 100(x_{i+1} - x_i)^2 \tag{53}$$

Where $x = [x_1, \dots, x_d] \in \mathbb{R}^d$. The global minimum $f(x^*)$ lies in a narrow, parabolic valley and is located at $x^* = [1, \dots, 1]^d$.

The MFEI algorithm has been tested considering $d = 2, 4, 8$ and with different experimental setups.

7.4. 2D Rosenbrock Function

Recalling the Eq. (53), the highest fidelity function is defined as:

$$y^{(M)}(x) = (1 - x_1)^2 + 100(x_2 - x_1)^2 \tag{54}$$

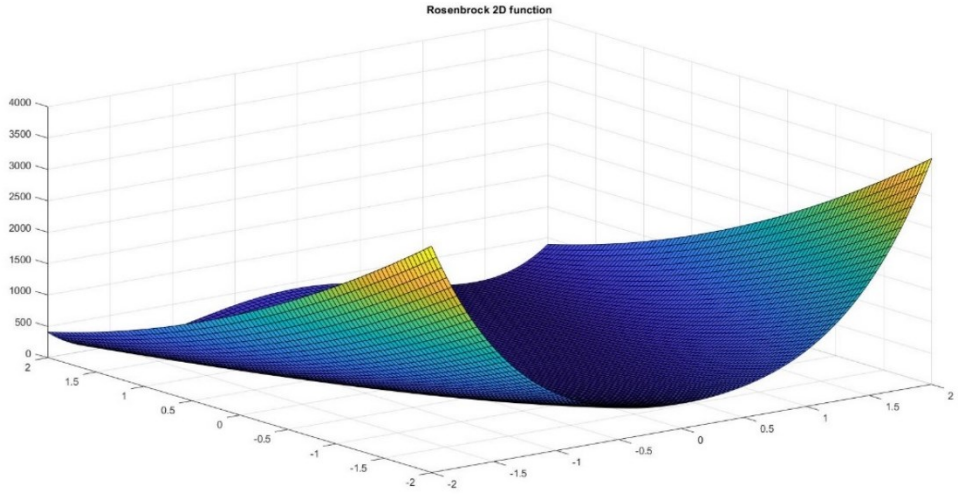

Figure 12: Rosenbrock 2D function

The experimental setups considered are distinct from the low fidelity functions considered:

- *Experiment 1:* Low-fidelity function linear mapping defined as [172]:

$$y^{(1)}(x) = \frac{y^{(2)}(x) - 4.0 - \sum_{i=1}^{2} 0.5x_i}{3.0 + \sum_{i=1}^{2} 0.25x_i} \tag{55}$$

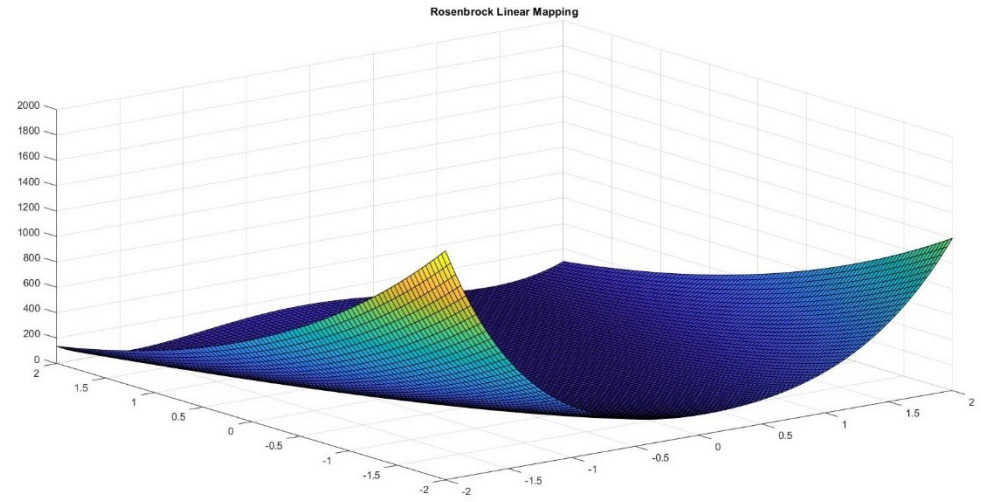

Figure 13: Low-fidelity linear mapping function.

- *Experiment 2:* Low-fidelity functions defined as the linear mapping in Eq. (56) and the quadratic 2-dimensions function defined in Eq. (57):

$$y^{(1)}(x) = \frac{y^{(M)}(x) - 4.0 - \sum_{i=1}^{2} 0.5x_i}{3.0 + \sum_{i=1}^{2} 0.25x_i} \tag{56}$$

$$y^{(2)}(x) = x_2^2 + x_1^2 \tag{57}$$

- *Experiment 3:* Low-fidelity functions defined by March et al. [173]:

$$y^{(1)}(x) = x_1^2 + x_2^2$$

$$y^{(2)}(x) = x_2^4 + x_1^2 \tag{58}$$

$$y^{(3)}(x) = -x_1^2 - x_2^2 \tag{59}$$

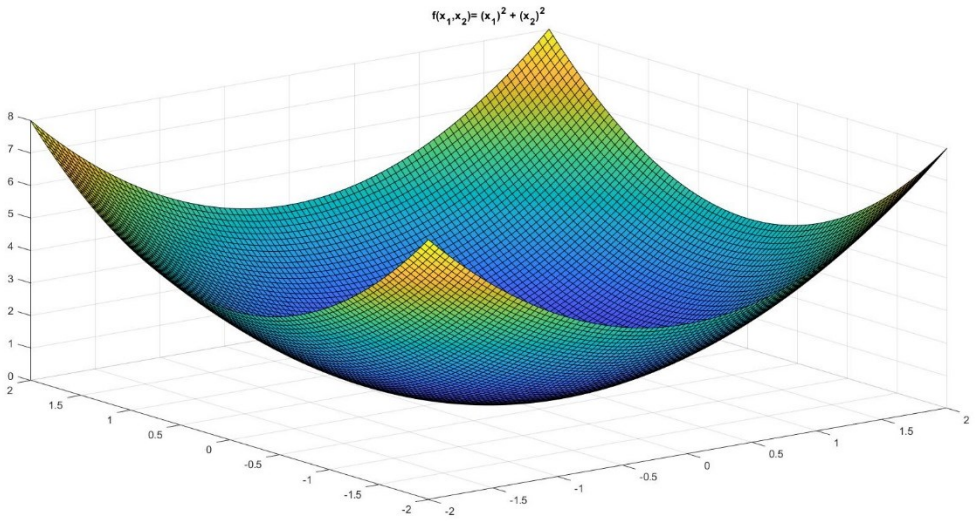

Figure 14: Low-fidelity function defined as $f(x) = x_1^2 + x_2^2$

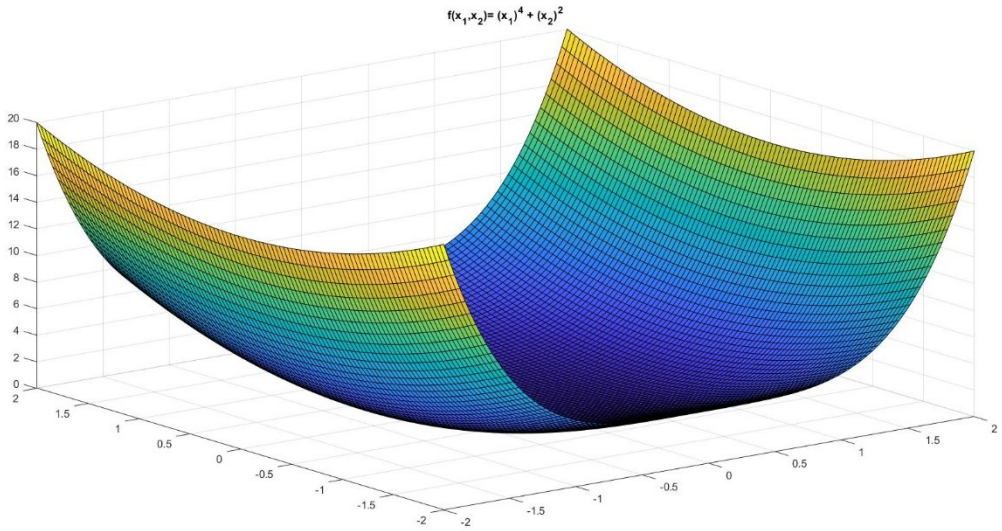

Figure 15: Low-fidelity function defined as $f(x) = x_1^4 + x_2^2$

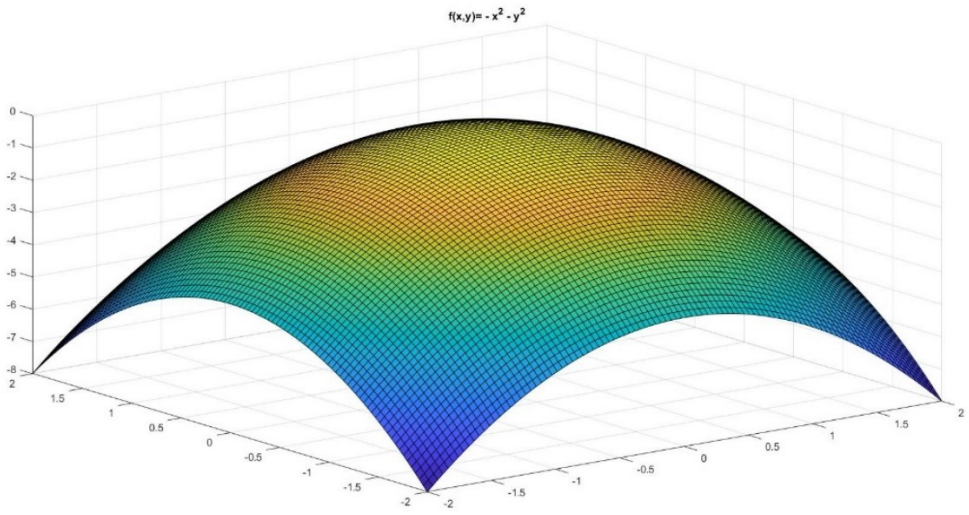

Figure 16: Low-fidelity function defined as $f(x) = -x_1^2 - x_2^2$

7.4.1. Experiments Results

This section reports the results of the SFEI and MFEI algorithm considering the experiments conducted on the 2-dimensional Rosenbrock function and mentioned above.

- *Experiment 1*

  Fig. 17 shows the median and interval of the error achieved by the SFEI algorithm considering the high-fidelity Rosenbrock 2D function. The total number of tests conducted are 100.

  For each test, the number of iterations carried out was 100 with a subdivision of the grid into 100,000 points. As in the previous test functions, the error is calculated taking into account the overall minimum reached during the iterations.

  Observing the median curve, it can be seen that the number of average iterations necessary to achieve the zero value is of 37 runs. However, some tests require significantly more time with up to 86 iterations.

  Considering the results obtained by the MFEI algorithm in experiment 1 with a linear mapping as a low-fidelity function.

  In Fig. 18 are reported the median-interval trend over the 100 tests. The experiment setup is the same of the SFEI algorithm reported above.

  The term $\alpha_3 = [4, 1]$ was selected after some numerical tests considering different elements of the cost vector. On average, the convergence of the tests occurs in 30 iterations but compared to the SFEI case, a greater variability of the interval and therefore of the overall minimum reached up to that computational step can be observed.

  This is explained by considering that the linear mapping of the low fidelity function has a higher computational efficiency than the high-fidelity function. Consequently, the algorithm is facilitated in the exploratory process, allowing to decrease considerable initial errors (higher than in the single fidelity case) in a contained number of iterations. On average, the number of invocations of the high-fidelity function is 11.2 explaining the different computational time jointly with the decrease of the iterations number in achieving convergence. The computation time associated with the MFEI algorithm averages 17 minutes against 31 minutes for the SFEI algorithm.

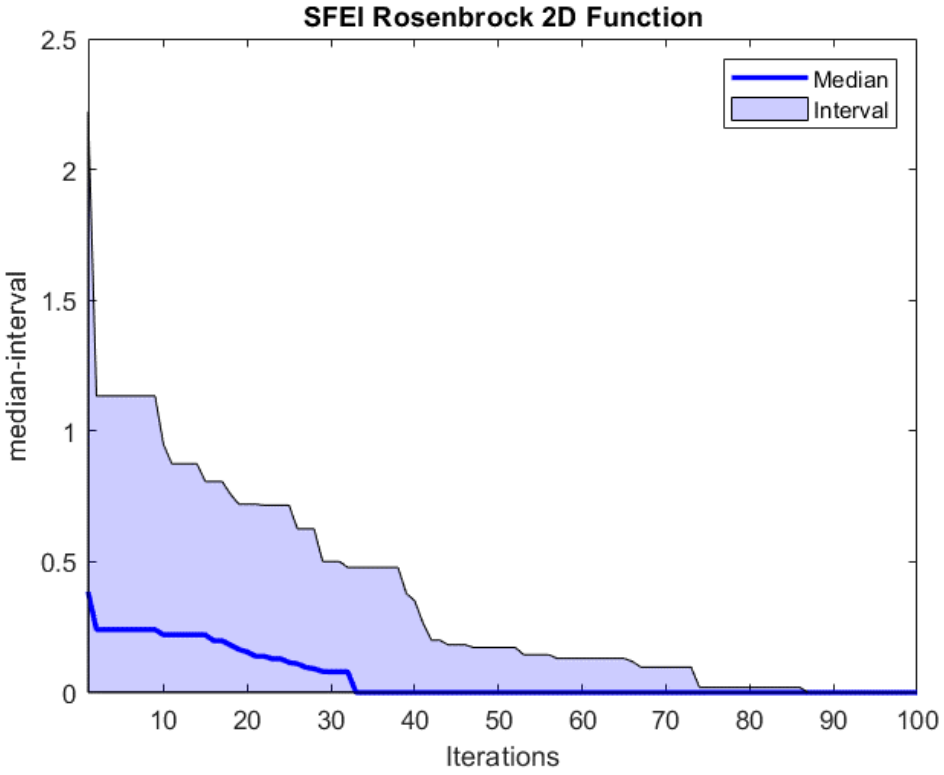

Figure 17: Median/Interval-Iterations results of the SFEI algorithm applied to the Rosenbrock 2D function optimization problem

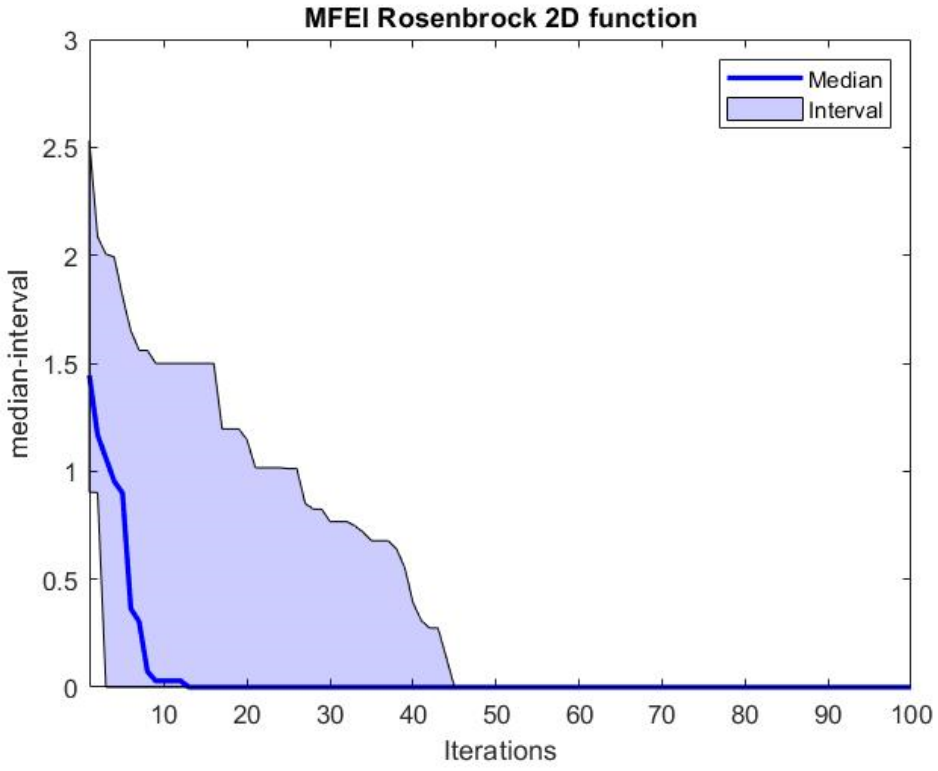

Figure 18: Median/Interval-Iterations results of the MFEI algorithm applied to the experiment 1 of the multifidelity Rosenbrock 2D function optimization problem.

- *Experiment 2*

Experiment 2 considers three levels of fidelity: the 2D linear mapping of the Rosenbrock function, the two-dimensional quadratic function and the Rosenbrock 2D high-fidelity function.

The numerical conditions are the same as in experiment 1. In these tests the cost vector $\alpha_3 = [4, 2, 1]$ is selected in order to emphasize the cost of the quadratic function over the linear mapping function.

In Fig. 19 are presented the median and interval of the error resulting in the experiment 2 optimization problem.

Compared to the previous case, the number of iterations necessary to clear the error is greater. This can be attributed to the invocation of the quadratic function which, being non-linear, tends to reduce the efficiency of the algorithm. On average, function $y^{(1)}$ was called 57.4 times, function $y^{(2)}$ 29.3 times and function $y^{(3)}$ was called 13.3 times. The computation time associated with the MFEI algorithm averages 22 minutes against 31 minutes for the SFEI algorithm.

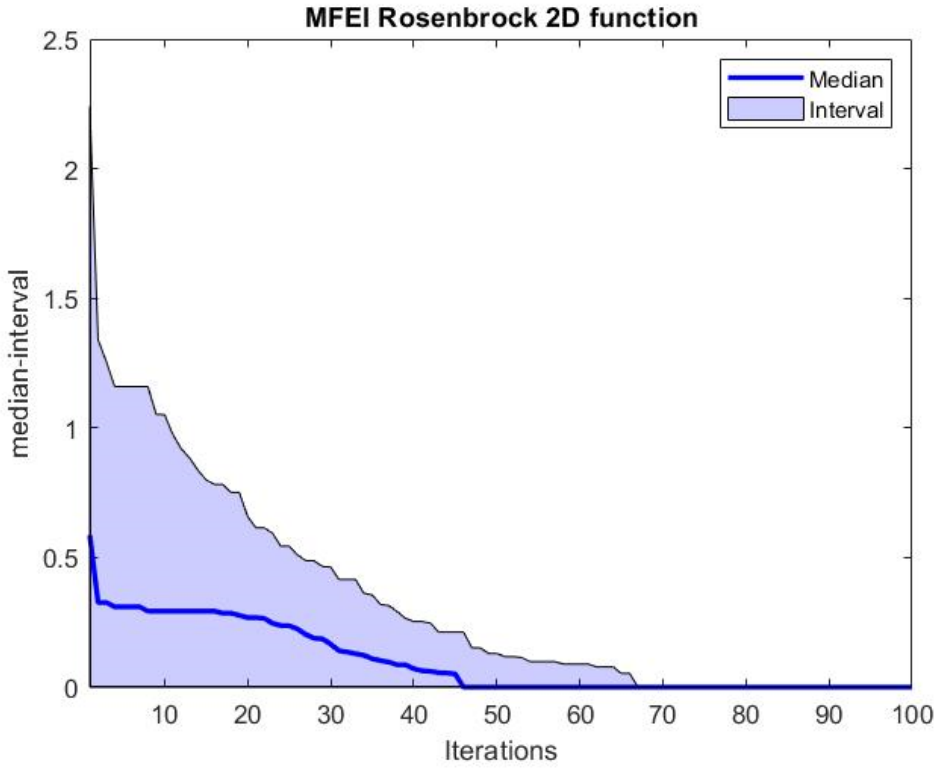

Figure 19: Median/Interval-Iterations results of the MFEI algorithm applied to the experiment 2 of the multifidelity Rosenbrock 2D function optimization problem.

- *Experiment 3*

In experiment 3, four levels of fidelity are considered. The cost vector $\alpha_3 = [2, 4, 2.5, 1]$ was selected after different tries and calibration.

In Fig.20 is shown the median-interval trend of the error resulting in the experiment 3 optimization problem.

A further increase in the iterations necessary to achieve convergence can be observed. This is due to the absence of linear mapping, making the algorithm less effective in the search process. Furthermore, 6% of the tests do not reach the null error condition, as the interval trend demonstrate, but keep staying in the neighborhood of the global minimum of the function.

Probably with an even greater number of iterations, total convergence of the tests would have been achieved.

On average, function $y^{(1)}$ was called 35.6 times, function $y^{(2)}$ 22.8 times, function $y^{(3)}$ 25.9 times, and function $y^{(4)}$ was called 15.7 times. The computation time associated with the MFEI algorithm averages 24 minutes against 31 minutes for the SFEI algorithm.

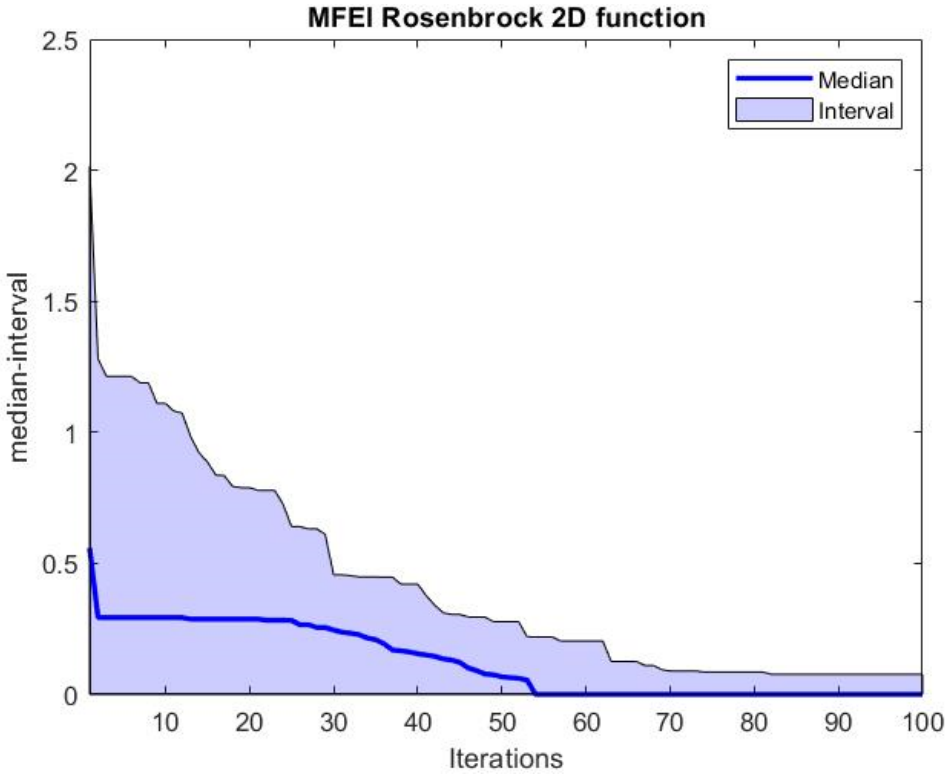

Figure 20: Median/Interval-Iterations results of the MFEI algorithm applied to the experiment 3 of the multifidelity Rosenbrock 2D function optimization problem.

7.5. 4D Rosenbrock Function

The high-fidelity function is defined as:

$$y^{(M)}(x) = \sum_{i=1}^{3}(1 - x_i)^2 + 100(x_{i+1} - x_i)^2 \tag{60}$$

The experimental setups considered are the following:

- *Experiment 1:* Low-fidelity function linear mapping defined as:

$$y^{(1)}(x) = \frac{y^{(M)}(x) - 4.0 - \sum_{i=1}^{4} 0.5x_i}{3.0 + \sum_{i=1}^{4} 0.25x_i} \tag{61}$$

- *Experiment 2:* Low-fidelity functions defined as the linear mapping in Eq. (61) and the quadratic 4-dimensions function:

$$y^{(1)}(x) = \frac{y^{(M)}(x) - 4.0 - \sum_{i=1}^{4} 0.5x_i}{3.0 + \sum_{i=1}^{4} 0.25x_i}$$

$$y^{(2)}(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 \tag{62}$$

- *Experiment 3:* low-fidelity function defined as the quadratic 4-dimensions function:

$$y^{(2)}(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2$$

7.5.1. Experiments Results

This section reports the results of the SFEI and MFEI algorithm considering the experiments conducted on the 4-dimensional Rosenbrock function mentioned above.

- *Experiment 1*

In experiment 1 two levels of fidelity are considered. The lowest fidelity function is a linear 4-dimensional mapping. The initial conditions are the same as those imposed in the tests of the 2D Rosenbrock function.

The cost vector $\alpha_3 = [4, 1]$ was selected to emphasize the lower computational cost of $y^{(1)}$, on the base of several tests of the algorithm.

In Fig.21 is presented the error median and the interval over the test iterations. As already seen for the two-dimensional case, the median trend decreases rapidly with the number of iterations. Furthermore, the null value is reached in all the tests carried out. This suggests, as noted earlier, that the low-fidelity function defined as a linear mapping allows for greater effectiveness of the algorithm.

On average, the low-fidelity function $y^{(1)}$ was called 88.9 times and the high-fidelity function $y^{(2)}$ was called 11.1 times. The computation time associated with the MFEI algorithm averages 43 minutes against 61 minutes for the SFEI algorithm.

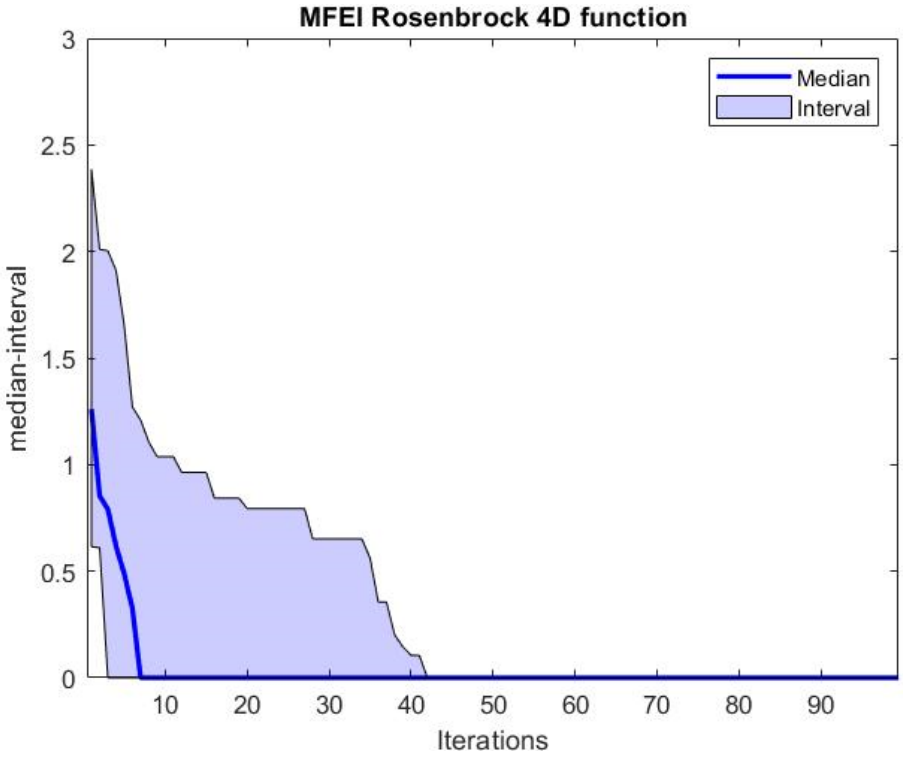

Figure 21: Median/Interval-Iterations results of the MFEI algorithm applied to the experiment 1 of the multifidelity Rosenbrock 4D function optimization problem.

- *Experiment 2*

In experiment 2 three levels of fidelity are considered. Function $y^{(1)}$ is a linear mapping, $y^{(2)}$ is a four dimensions quadratic function and $y^{(3)}$ is the Rosenbrock 4D function.

The initial conditions are the same as those imposed in the tests of the 2D Rosenbrock function. In these tests the cost vector $\alpha_3 = [4, 2, 1]$ is selected for the same considerations in experiment 2 of the 2D Rosenbrock function.

In Fig. 22 it can be seen how the effectiveness of the algorithm has slightly worsened. Although the median reaches values close to zero in a few iterations, the value of the interval is considerable. In fact, convergence is achieved in a greater number of iterations than in experiment 1. Furthermore, 3% of the tests do not reach the null error condition. On average, the function $y^{(1)}$ was called 59.5 times, the function $y^{(2)}$ 28.7 times and the high-fidelity function $y^{(2)}$ was called 11.8 times. The computation time associated with the MFEI algorithm averages 56 minutes against 61 minutes for the SFEI algorithm.

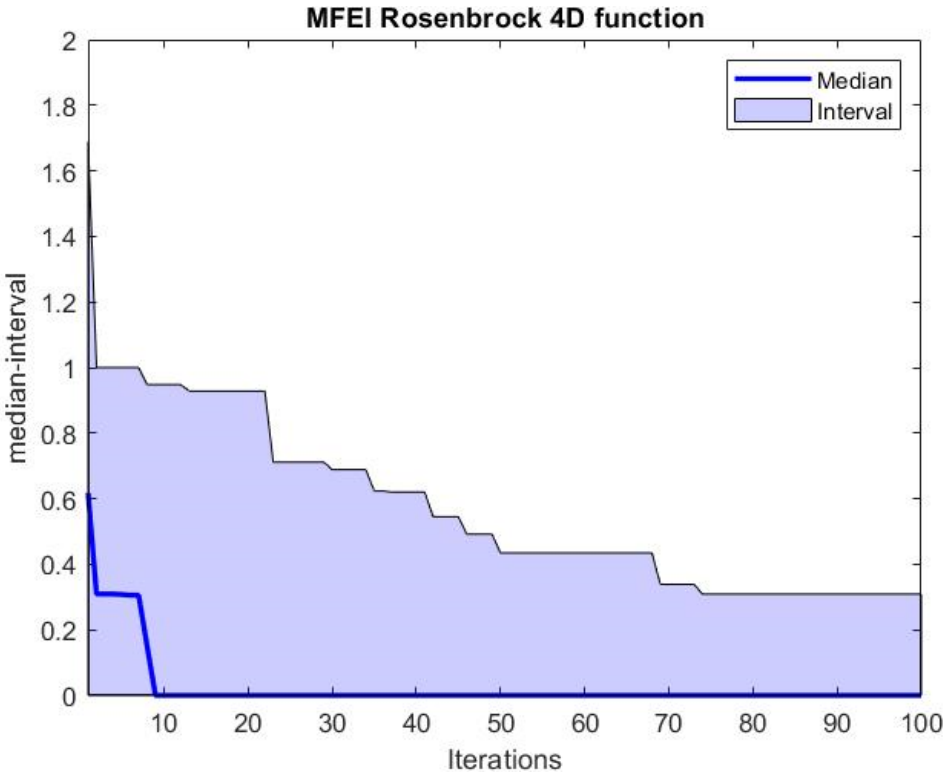

Figure 22: Median/Interval-Iterations results of the MFEI algorithm applied to the experiment 2 of the multifidelity Rosenbrock 4D function optimization problem.

- *Experiment 3*

Considering the experiment 3, two levels of fidelity are taken into account. The function $y^{(1)}$ is the 4D quadratic function and $y^{(2)}$ is the 4D Rosenbrock function. The initial conditions are the same as those imposed in the tests of the 2D Rosenbrock function. The cost vector selected is $\alpha_3 = [4, 1]$.

In Fig.23 it is shown how the performance of the algorithm is even worse than in the previous case. In most of the tests conducted, the median tends to zero in a significantly greater number of iterations. Furthermore 4% of the tests do not reach convergence, as the interval suggest. On average, the low-fidelity function $y^{(1)}$ was called 82.7 times, and the high-fidelity function $y^{(3)}$ was called 17.3 times. The computation time associated with the MFEI algorithm averages 69 minutes against 61 minutes for the SFEI algorithm.

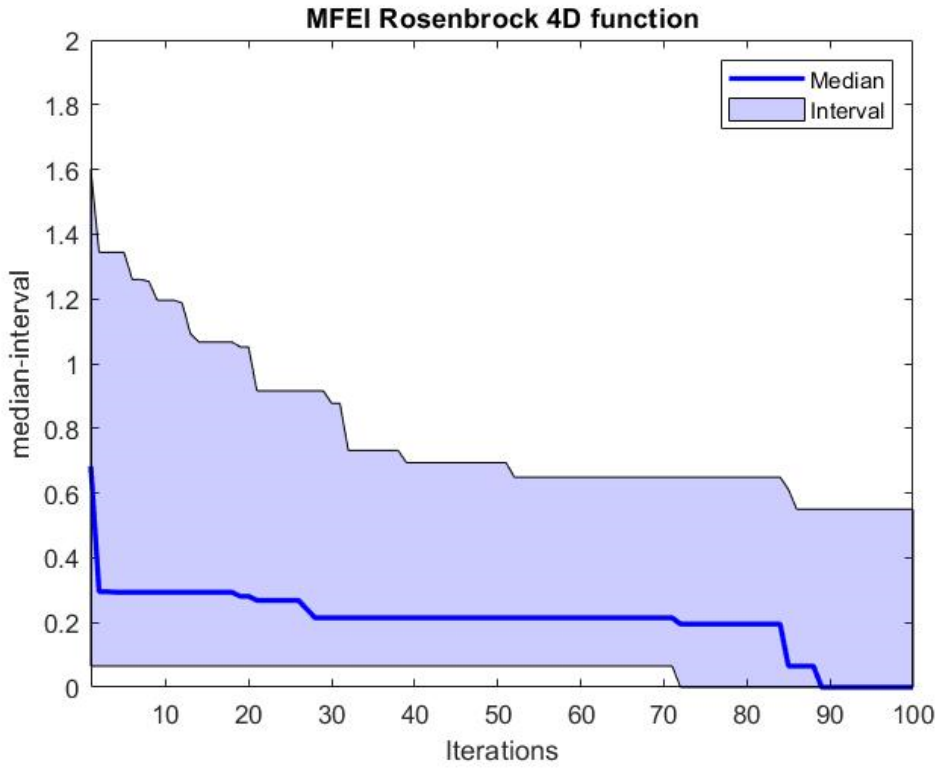

Figure 23: Median/Interval-Iterations results of the MFEI algorithm applied to the experiment 3 of the multifidelity Rosenbrock 4D function optimization problem.

7.6. 8D Rosenbrock Function

The high-fidelity function is defined as:

$$y^{(M)}(x) = \sum_{i=1}^{7} (1 - x_i)^2 + 100(x_{i+1} - x_i)^2 \tag{63}$$

The experimental setups considered are the following:

- *Experiment 1:* Low-fidelity function linear mapping defined as:

$$y^{(1)}(x) = \frac{y^{(M)}(x) - 4.0 - \sum_{i=1}^{8} 0.5x_i}{3.0 + \sum_{i=1}^{8} 0.25x_i} \tag{64}$$

- *Experiment 2:* Low-fidelity functions defined as the linear mapping in Eq. (64) and the quadratic 8-dimensions function:

$$y^{(1)}(x) = \frac{y^{(M)}(x) - 4.0 - \sum_{i=1}^{8} 0.5x_i}{3.0 + \sum_{i=1}^{8} 0.25x_i}$$

$$y^{(2)}(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2 + x_7^2 + x_8^2 \tag{65}$$

- *Experiment 3:* Low-fidelity functions defined as the quadratic 8-dimensions function:

$$y^{(2)}(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2 + x_7^2 + x_8^2$$

7.6.1. Experimental Results

This section reports the results of the SFEI and MFEI algorithm considering the experiments conducted on the 8-dimensional Rosenbrock function mentioned above.

- *Experiment 1*

In experiment 1 two levels of fidelity are considered. The lowest fidelity function is a linear 8-dimensional mapping. The initial conditions are the same as those imposed in the tests of the 2D Rosenbrock function. The cost vector $\alpha_3 = [4, 1]$ is selected based on previous low-dimensional experiments.

In Fig.24 the error median and the interval over the test iterations is presented. As already seen for the lower dimension cases, the median trend decreases rapidly with the number of iterations. Nevertheless, the interval is different from zero at the end of the iterations, since 2% of cases do not reach convergence. This suggests that with the increase of the dimensionality of the problem, the algorithm tends to perform less effectively.

On average, the low-fidelity function $y^{(1)}$ was called 83.4 times and the high-fidelity function $y^{(2)}$ was called 16.6 times. The computation time associated with the MFEI algorithm averages 78 minutes against 108 minutes for the SFEI algorithm.

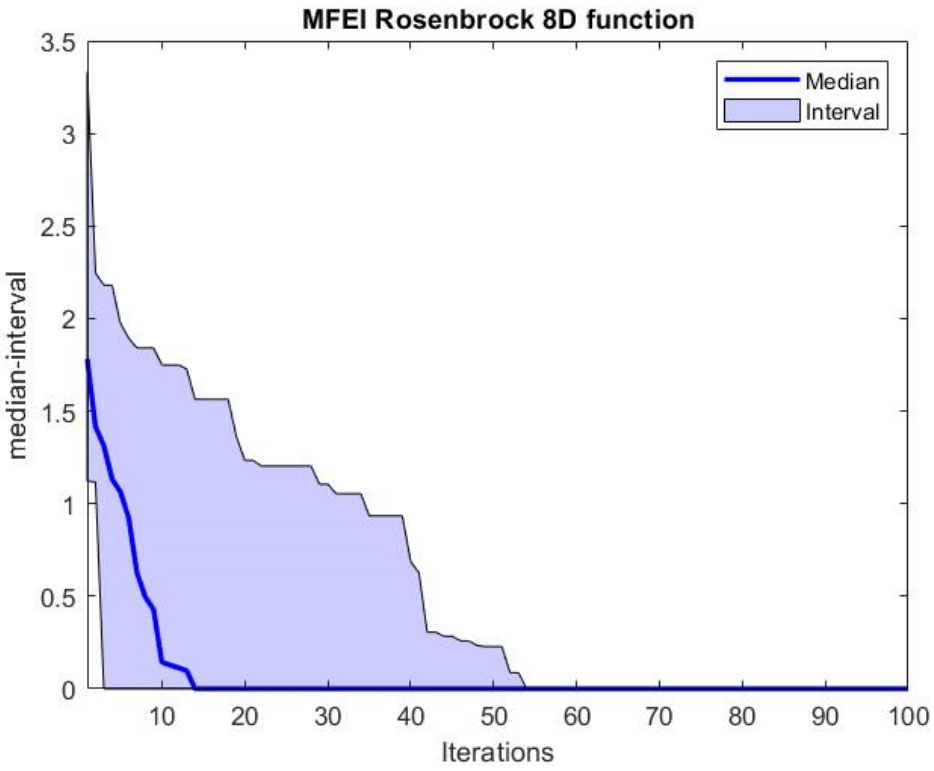

Figure 24: Median/Interval-Iterations results of the MFEI algorithm applied to the experiment 1 of the multifidelity Rosenbrock 8D function optimization problem.

- *Experiment 2*

In experiment 2 three levels of fidelity are considered. Function $y^{(1)}$ is a linear mapping, $y^{(2)}$ is an eight dimensions quadratic function and $y^{(3)}$ is the Rosenbrock 8D function. The initial conditions are the same as those imposed in the tests of the 2D Rosenbrock function. In these tests the cost vector $\alpha_3 = [4, 2, 1]$.

In Fig. 25 it can be seen how the effectiveness of the algorithm has worsened, both for the presence of the quadratic function and for the increase in dimensionality of the problem. In fact, convergence is achieved in a greater number of iterations than in experiment 1. Furthermore, 10% of the tests do not reach the null error condition. On average, the function $y^{(1)}$ was called 68.3 times, the function $y^{(2)}$ 23.6 times and the high-fidelity function $y^{(2)}$ was called 8.1 times. The computation time associated with the MFEI algorithm averages 93 minutes against 108 minutes for the SFEI algorithm.

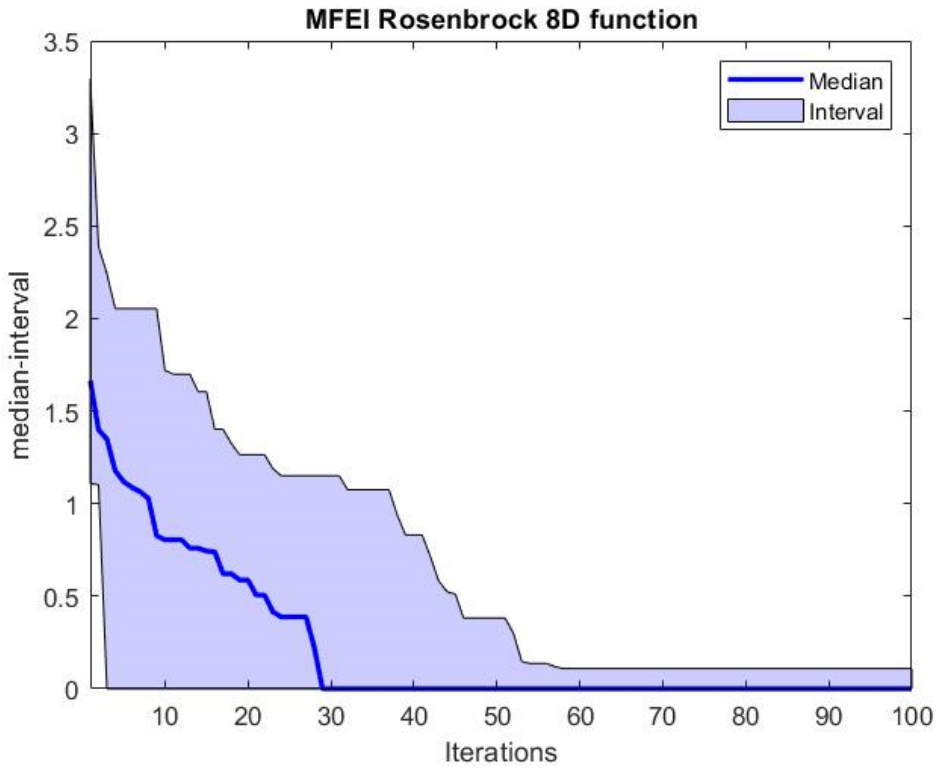

Figure 25: Median/Interval-Iterations results of the MFEI algorithm applied to the experiment 2 of the multifidelity Rosenbrock 8D function optimization problem.

- *Experiment 3*

Considering the experiment 3, two levels of fidelity are taken into account. The function $y^{(1)}$ is the 8D quadratic function and $y^{(2)}$ is the 8D Rosenbrock function. The initial conditions are the same as those imposed in the tests of the 2D Rosenbrock function. The cost vector selected is $\alpha_3 = [4, 1]$.

In Fig.26 it is shown how the performance of the algorithm is even worse than in the previous case. In most of the tests conducted, the mean tends to zero in a significantly greater number of iterations. However, only the 1% of the tests didn't reach the convergence.

On average, the low-fidelity function $y^{(1)}$ was called 84.9 times, and the high-fidelity function $y^{(2)}$ was called 15.1 times. The computation time associated with the MFEI algorithm averages 98 minutes against 108 minutes for the SFEI algorithm.

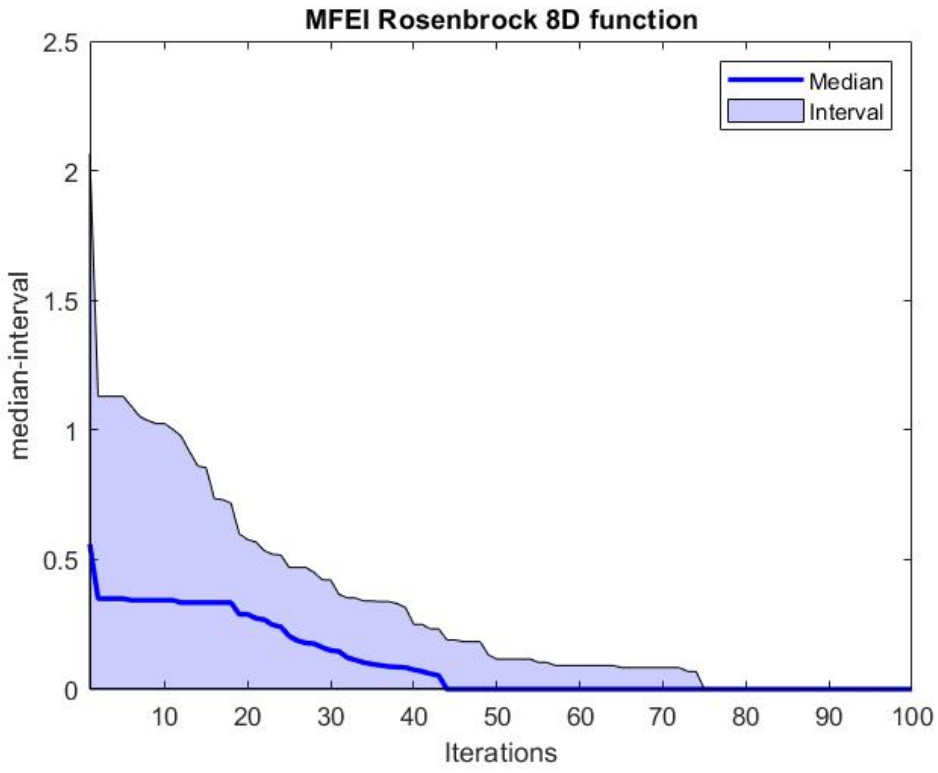

Figure 26: Median/Interval-Iterations results of the MFEI algorithm applied to the experiment 3 of the multifidelity Rosenbrock 8D function optimization problem.

8. Re-Entry Vehicle Optimization Problem

Returning to Earth from other planetary bodies or Low Earth Orbit (LEO) entails an extreme hypersonic environment during atmospheric entry. This involves entering Earth's atmosphere at high velocities ranging from 10 km/s to 15 km/s with corresponding Mach numbers from 30 to 50, while withstanding 3000+ K temperatures at and near the stagnation point. The heat shield, which faces the freestream flow and protects the entry vehicle, is the primary source of the vehicle's hypersonic aerothermodynamic performance (i.e. the aerodynamic forces, moments, and heat transfer). The rest of the vehicle is secondary since it is covered with regions of extremely low pressure, due to flow separation, and since it experiences significantly lower heat fluxes.

For these systems, aerodynamics and thermodynamics phenomena are strongly coupled and relate to structural dynamics and vibrations, chemical non equilibrium phenomena that characterize the atmosphere, specific re-entry trajectory, and geometrical shape of the body.

Blunt bodies are common geometric configurations used in planetary re-entry (e.g. Apollo Command Module, Mars Viking probe, etc.). These geometries permit to obtain high aerodynamic resistance to decelerate the vehicle from orbital speeds along with contained aerodynamic lift for trajectory control. The large radius-of-curvature allows to reduce the heat flux determined by the high temperature effects behind the shock wave.

Given the multidisciplinary nature of the phenomena and the Multiphysics domain, modelling the physical phenomena involved in the return phase represents a demanding challenge. The design and optimization of blunt bodies and the re-entry trajectories would largely benefit from accurate analysis of the re-entry flow-field through high-fidelity representations of the aerodynamic and aerothermodynamic phenomena.

However, those high-fidelity representations are usually in the form of computer models for the numerical solutions of PDEs (e.g. Navier-Stokes equations, heat equations, etc.) which require significant computational effort and are commonly excluded from preliminary multidisciplinary design and trade-off analysis.

To address this challenge, the application of Multi-fidelity methods is explored in the design optimization process. Multi-fidelity methods allow to accelerate the exploration and evaluation of designs in optimization problems through the use of different physical models characterized by different levels of fidelity and associated computational cost. High-fidelity data are produced by a process that can closely match the operational contest of interest. Low-fidelity data are produced by a process that deviates in terms of accuracy from the real-world case of interest.

The benefit of using low-fidelity data is that are cheaper to acquire, in terms of computational cost, than the high-fidelity information. However, despite the possible elicitation of large amount of data, low fidelity models may not be adequate to predict the real contest of operation of the system analysed.

By effectively combining low-cost information from low-fidelity models and a reduced number of invocations of more expensive computationally high-fidelity models, multi-fidelity methods help speed up the process of identifying the optimal design reducing the total simulative computational cost. In particular, the implementation of a multi-fidelity method based on active learning will be investigated. This choice is linked to the possibility of reducing the number of points to be sampled and optimizing the choice of points to be sampled in the optimization process.

In the following paragraphs the problem of optimizing an atmospheric re-entry vehicle will be faced. Specifically, the optimization problem will consist in defining an appropriate set of forces (therefore an appropriate re-entry trajectory) and thickness of the thermal protection system (therefore an appropriate structural design) in order to minimize a black-box objective function dependent on the mass of propellant burned, mass of the thermal protection system (TPS) and the temperature reached by the heat shield. The MFEI algorithm presented in Chapter 6 and extensively tested on analytic test-functions in Chapter 7 has been implemented with an original formulation to address the re-entry optimization problem.

8.1. Re-Entry Vehicle Design Structure Matrix

In Fig. 27 is presented the design structure matrix (DSM) of the re-entry optimization problem. Several workflows are defined supporting the multidisciplinary design problem described above. The various analyses include:

1. *Propulsion system:* modelling the physics of space vehicle thrusters for attitude control and orbital manoeuvres.
2. *Trajectory:* definition and computation of the equations of atmospheric re-entry motion.
3. *Aerodynamic and thermodynamic:* modelling the dynamics of the chemically unstable gas and the interaction with the vehicle. Determine the phenomena and thermal flows on the structure. Specifically, two levels of fidelity are considered for this multidisciplinary block: a low-fidelity model based on simplified formulations and a high-fidelity model based on the numerical resolution of the Navier-Stokes equations.
4. *Thermal protection system structure and temperature:* implementation of a suitable structural model for the TPS of a re-entry vehicle to determine structural sizing and thermal loads.

The input data to the optimization loop are a set of design optimization variables along with data required to configure the various analysis. The design optimization variables are the parameters sampled by that the algorithm in order to reach the minimum of the objective function.

In Fig. 27 the optimization variables are the propellant mass burned during the manoeuvres and the TPS structural mass and temperature, indicated by orange blocks. The data needed for the analyses are shown in the green blocks. This information are not optimization variables and therefore it does not vary during the optimization loop. In the diagonal of the DSM are placed the models that define the multidisciplinary analyses conducted in the optimization loop. Physical model outputs are shown with grey blocks and are related according to the relationships listed in chapter 2. Nested loops are shown in the lower triangle of the DSM. Specifically, one loop is identified by the mass of propellant in output from the propulsion system block, the other is constituted by the TPS structure mass and temperature in output from the TPS structure and temperature block. In the following paragraphs the models adopted in the optimization process are described.

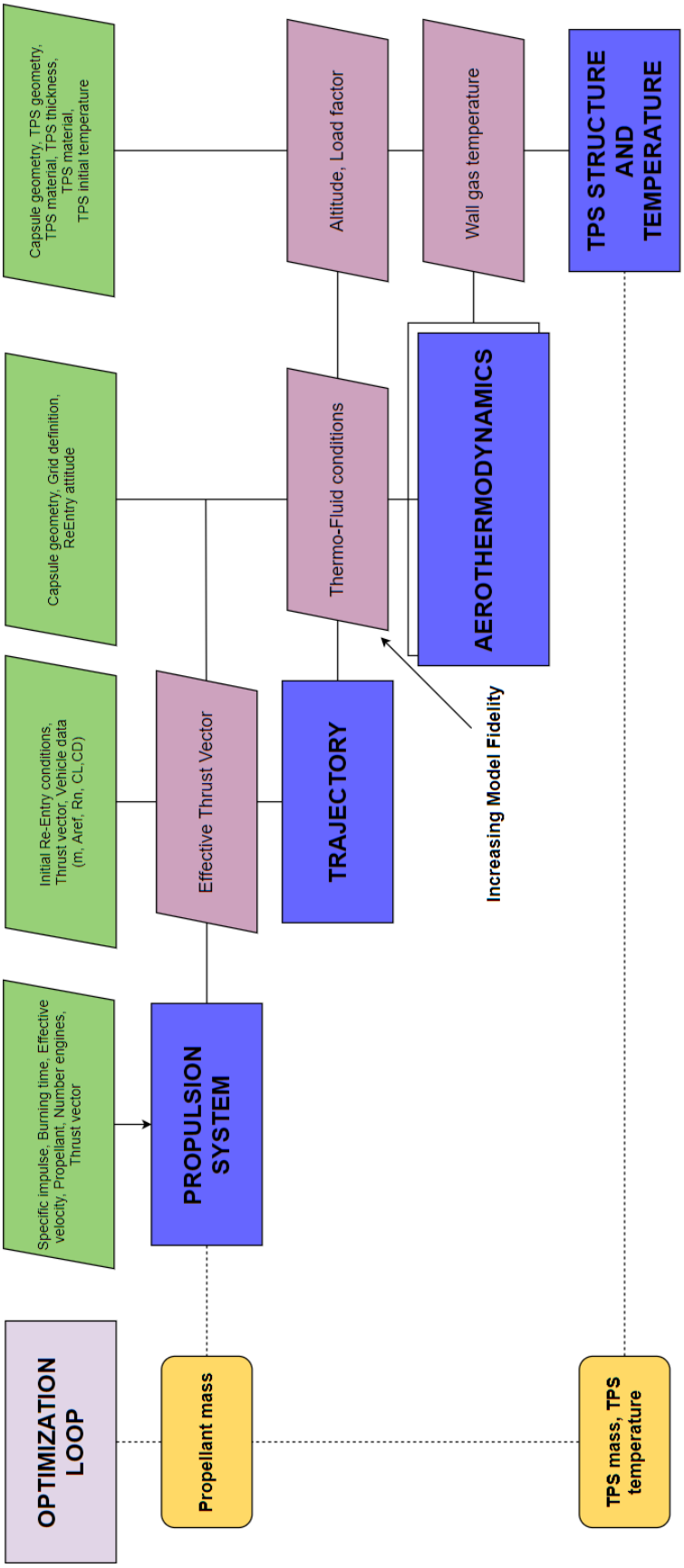

Figure 27: Design Structure Matrix Re-entry problem.

8.2. Atmospheric Re-Entry From Space: The Hypersonic Regime

A Hypersonic flow is a flow moving many times faster than the speed of sound. As the definition of hypersonic flow, there is not a precise Mach number that sets the lower bound of the hypersonic regime. It is commonly accepted that hypersonic flow occurs when the free-stream Mach number is larger than a value ranging from 5 to 7. Hypersonic flight is the flight regime that characterizes the entry of a space vehicle in a planetary atmosphere (terrestrial or extra-terrestrial). It is also the flight regime of propelled hypersonic vehicles that at this time have been or are being developed mainly for technology demonstration and military purposes.

The difference between hypersonic flow and supersonic flow ($1 < M < 5$) has great impact on vehicle design [2]. For example, the shock wave that is generated will be closer to the vehicle surface, and a thick layer in which large frictional forces are present is generated. The flow has a high velocity and the kinetic energy is high. Due to the frictional forces, the air is slowed down, and heat is generated. This can result in high temperatures, in the order of 10,000K behind a normal shock. When air is heated up, the chemistry changes [4].

Regular air is mainly dominated by concentrations of oxygen (O_2) and nitrogen (N_2). Around 800K, air is excited due to vibrational energy, which affects gas properties.

Following the dissertation proposed by Anderson et al. [1787], Fig.28 shows the ionisation and dissociation diagram of oxygen and nitrogen. At 2500K oxygen begins to dissociate, breaking the molecule apart into two atoms ($2O$). At 4000K, nitrogen also begins to dissociate into atomic nitrogen ($2N$). Finally, at 9000K the air is completely dissociated and starts to ionise, a process where an electron ($e-$) moves freely around, resulting in positively charged nitrogen and oxygen ions (N^+ and O^+).

The mixture of species (N_2, N, N^+, O_2, O, O^+) has enough energy to react with either the surface material of the vehicle or with other species itself, creating, for example, NO as well. Especially atomic oxygen is highly corrosive and is damaging to materials. The ionised species create a layer of plasma around the body, which is responsible for the communication blackout in re-entry.

Fig.28 shows the Shuttle velocity-altitude map. Nitrogen and oxygen are already dissociated at the beginning of the trajectory of the Space Shuttle. When the velocity is decreased and the altitude decreases, only oxygen is still dissociated while nitrogen has recombined to nitrogen molecules. After the oxygen dissociation phase, the vibrational excited molecules are still present. In the final leg of the re-entry trajectory, the velocity and altitude are decreased to a point where no chemical effects occur. Chemically reacting flows must be considered when considering hypersonic flow.

The presence of different chemical species in a flow calls for a different gas model than the well-known perfect gas model. The specific heat ratio γ is not constant anymore and the value of cp becomes dependent of temperature. Forward and backward reaction rates need to be considered. These reaction rates are determined experimentally, and the coefficients are difficult to compute. To deal with these chemical mechanisms (non-equilibrium flow), engineering methods have been derived to either treat the chemical composition as fixed in time and space (frozen flow) or only fixed in time (equilibrium flow) [178].

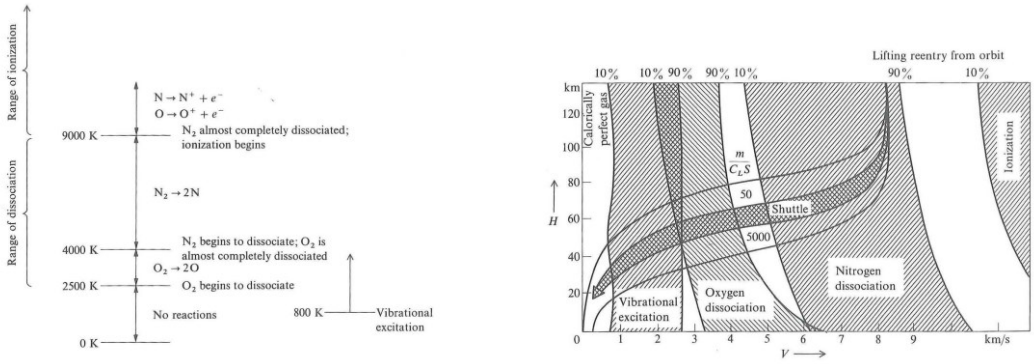

Figure 28: on the right dissociation and ionisation diagram of N_2 and O_2 ; on the left the velocity-altitude map with Shuttle re-entry corridor and areas of vibrational excitation, dissociation and ionisation. The Images are taken from the dissertation of Anderson et al. [178]

8.3. Re-Entry Vehicle Geometry

Blunt body configurations are the most common geometry employed for entry into planetary atmosphere. Examples of manned blunt-body entry vehicles include the Mercury, Gemini and Apollo capsules and SpaceX Crew Dragon capsule. Examples of unmanned flight test or interplanetary probe blunt-body entry vehicles are more numerous and include the Viking Pioneer, FIRE II, ARD, OREX, Stardust, etc. [1].

In broad terms, a blunt-body entry vehicle is comprised of a large heat shield that protects a smaller crew cabin or robotic probe payload. The heat shield is generally axisymmetric with either a large-angle, sphere cone geometry (e.g., the Mars Viking probe) or a large radius-of-curvature spherical cap (e.g., the Apollo command module), although asymmetric shapes have sometimes been considered (e.g., the cancelled Aero assist Flight Experiment).

The geometry of the heat shield produces large amounts of aerodynamic drag that decelerate the vehicle from orbital or interplanetary speeds. A small amount of aerodynamic lift for manoeuvrability and cross range capability may also be provided by offsetting the centre-of-gravity of the vehicle to trim it at a nonzero angle of attack [1].

8.3.1. Ballistic Coefficient

The re-entry vehicle's size and shape help determine the *ballistic coefficient* (BC) and the amount of lift it will generate [3]:

$$BC = \frac{W}{C_d A} \tag{66}$$

Where W is the vehicle weight, A is the cross-sectional area and C_d is the drag coefficient. The hardest component of BC to determine for re-entry vehicles is the drag coefficient, C_D, which depends mainly on the vehicle's shape. At low speeds, analysing a model of the vehicle in a wind tunnel and take specific measurements it's enough to determine C_D. But at re-entry speeds approaching 25 times the speed of sound, wind tunnel testing is not practical because no tunnels work at those speeds. Instead, mathematical models of hypersonic flow must be considered to find C_D. The most accurate of these approaches is *computational fluid dynamics* (CFD).

The Ballistic Coefficient is the single most important parameter in controlling flight trajectory during entry [3]. Heating and deceleration are less intense for a low BC value (low weight and/or high drag and large frontal area) than for a high BC value (high weight and/or low drag and small frontal area) since the entry occurs high in the atmosphere where the air is less dense. Early Inter-Continental Ballistic Missiles (ICBM) with highly blunted sphere-cone-cylinder-flare geometries utilized this re-entry method. Thermal protection for these early warheads was a massive metallic heat shield, which merely provided a "heat sink" for the short heating pulse at high altitudes.

It was soon discovered that delivery accuracy could be improved by increasing the values of BC using slightly blunted slender sphere-cone geometries thus increasing the impact velocity so that the final descent phase was less affected by winds. Thermal protection was provided by allowing the material at the surface of the heat shield to melt or vaporize thus transferring much of the heat back into the atmosphere. This method of thermal protection is referred to as "ablation," and the material that is applied to the vehicle's outer surface is called an "ablator."

Representative ballistic earth entry trajectories are presented in the analysis proposed by Adams et al. [3]. In Fig.29 are presented the results based on the application of a point mass ballistic entry computer program using the 1976 U.S. standard atmosphere model [181]. Initial entry conditions are:

- Altitude = 250,000 ft
- Velocity = 22,500 ft/sec
- Flight Path Angle = 12 deg

with four values of BC, namely 100, 500, 1000, and 5000 lb_f/ft^2. A BC value between 100 and 500 is representative of the early ICBM highly blunted sphere-cone-cylinder-flare geometry, while a BC value of 1000 to 5000 is representative of the slightly blunted slender sphere-cone geometry used in modern re-entry vehicles [3].

The stagnation point heat transfer is for a sphere having a nose radius of 1.0 ft using the Fay-Riddell correlation. Note how peak deceleration, dynamic pressure, dynamic energy, stagnation point pressure, and stagnation point heat transfer are shifted to a lower altitude with increasing BC. Entry times vary from slightly over three minutes for a BC = 100 value to slight less than one minute for a BC = 5000 value. Range distance is about 160 miles for the lowest value of BC (below about 50,000 ft in altitude the vehicle has slowed to subsonic velocities and literally falls out of the sky), and increases to 190, 200, and 210 miles for the other three BC values in increasing order.

The flight path angle at entry controls range distance and entry times, with shallow angles increasing range distance and flight time. Further observe that modern ballistic re-entry vehicles with BC values on the order of 5000 impact the earth's surface at hypersonic conditions [3].

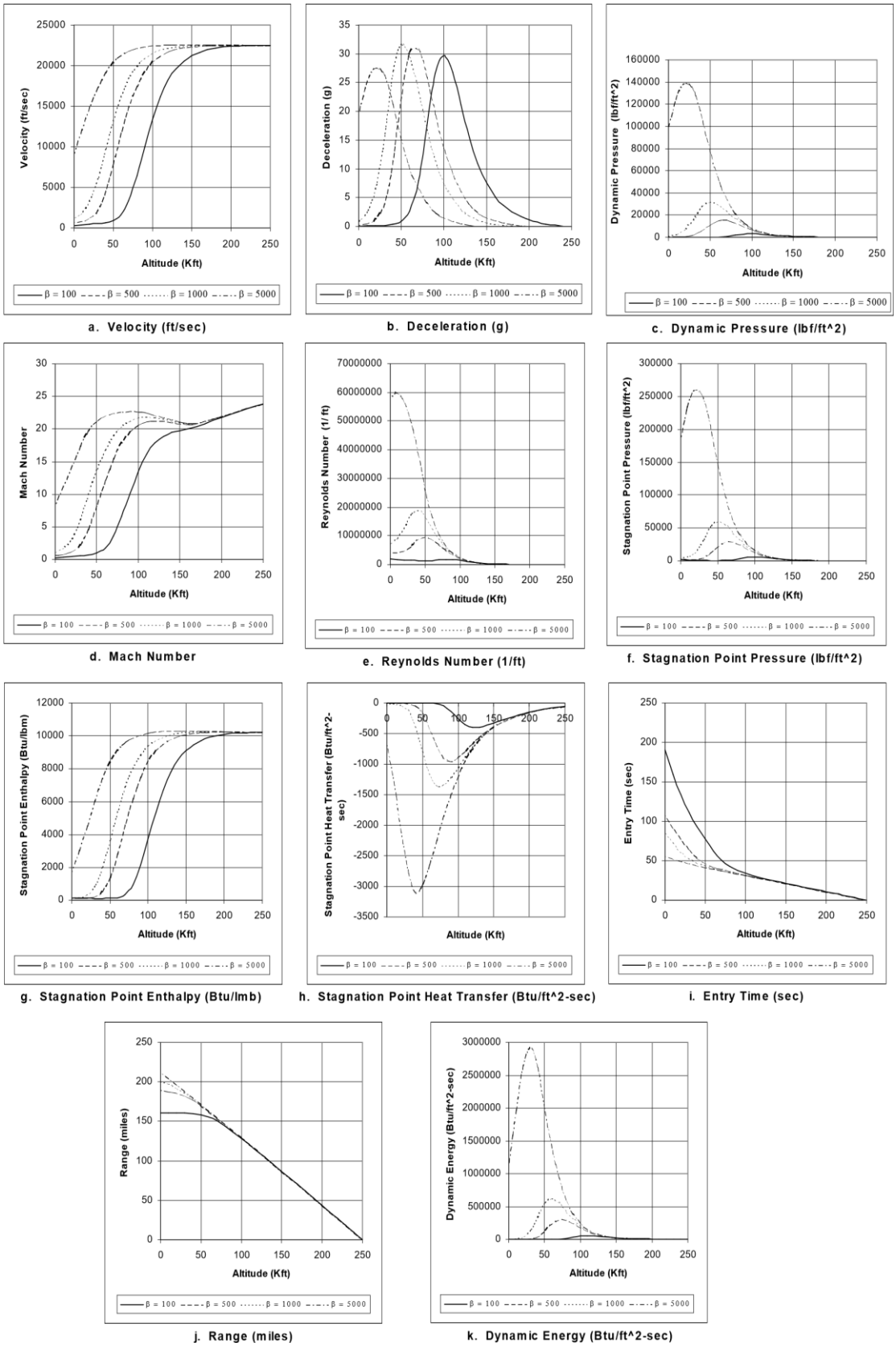

Figure 29: Ballistic Earth Entry Trajectory. The Images are taken from the work of Adams et al. [3].

8.3.2. Orion Crew Module Geometry

In this thesis, Orion Crew Module geometry is considered in the re-entry optimization problem. The Project Orion Crew Exploration Vehicle (CEV) was defined by NASA's Exploration Systems Architecture Study as NASA's next manned space vehicle (Fig.30) [174].

The CEV will support NASA's exploration missions by providing crew access to the International Space Station, the moon, and Mars. The geometry of the CEV (Fig.31) is similar to that of Apollo – a spherical segment heat shield that protects a truncated-cone shaped crew compartment – but is considerably larger. The maximum diameter (current design iteration) of the CEV is 5 m, as compared to that of 3.912 m for Apollo.

The design of the CEV TPS must account for the high heating rates generated at lunar return velocities and the aerothermodynamic challenges of non-equilibrium thermo-chemistry, turbulent flow, and radiation transport [175, 176, 177].

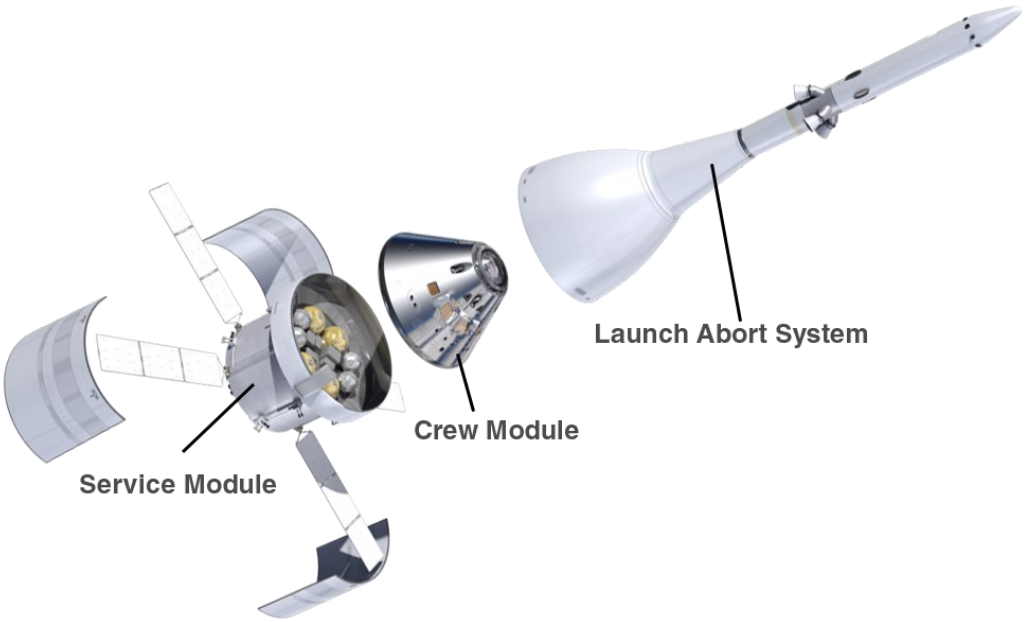

Figure 30: Orion Crew Exploration Vehicle (CEV). Image from NASA website [185].

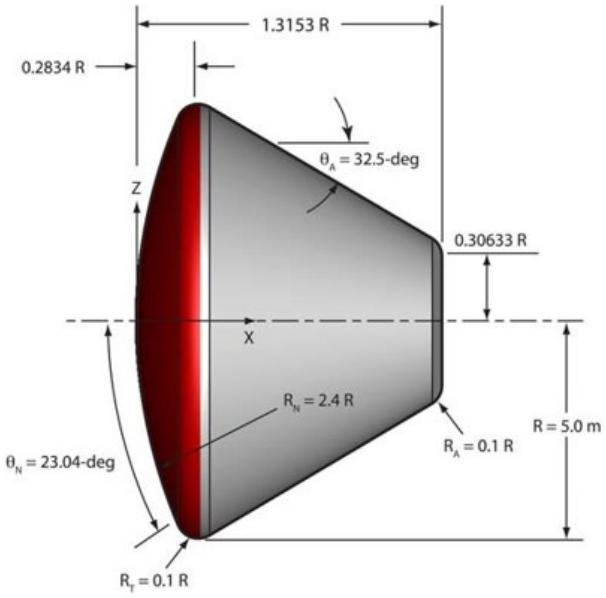

Figure 31: Orion CEV Crew Module Dimensions. Image reproduced from the dissertation of Adhams et al. [174].

8.4. Re-Entry Dynamics

Several fundamental concepts regarding re-entry trajectory formulation have been applied. In the following paragraphs, a brief summary of them is provided including the equations of motion, atmospheric and gravitational model, and the computation of load factor. The following dissertation is taken from Hirshel et al. work [177].

8.4.1. General Equation for Planetary Flight

Considering an unpowered and uncontrolled flight, Newton's second law can be formulated for an inertial system[1], $\boldsymbol{O}, x_0, y_0, z_0,$ illustrated in Fig. 32, as follows:

$$m\frac{d\boldsymbol{V}}{dt} = \boldsymbol{F}_A + m\boldsymbol{G} \tag{66}$$

where $\boldsymbol{F}_A$ contains the aerodynamic forces and $\boldsymbol{G}$ represents the gravity.

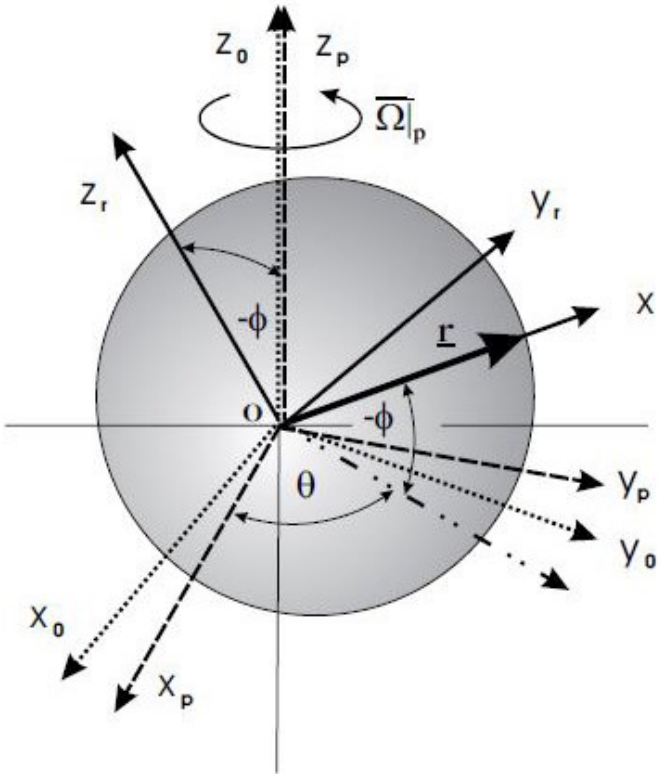

Figure 32: Inertial coordinate system $\boldsymbol{O}, x_0, y_0, z_0$, planet fixed coordinate system $\boldsymbol{O}, x_p, y_p, z_p$, and rotating coordinate system $\boldsymbol{O}, x_r, y_r, z_r$. Image reproduced from Hirshel et al. analysis in [177].

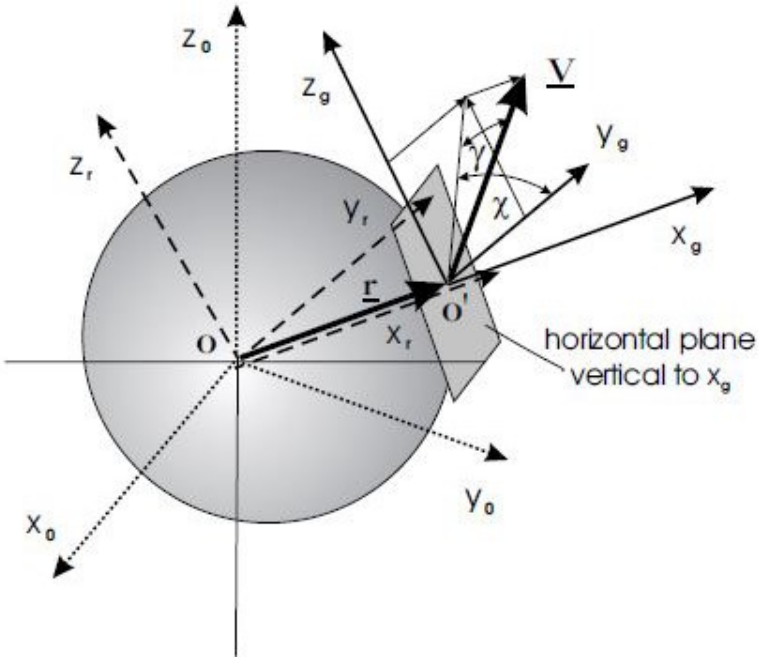

Figure 33: Coordinate system $\boldsymbol{O}', x_g, y_g, z_g$ parallel to coordinate system $\boldsymbol{O}, x_r, y_r, z_r$. Image reproduced from Hirshel et al. analysis in [177].

After various changes of coordinate systems and manipulations [177], as shown in Fig 32, 33 and 34, Eq. (66) can be written in scalar form:

$$\frac{dV}{dt} = -\frac{1}{m}D - g\sin\gamma + \omega^2 r cos^2\phi\,(\sin\gamma - \cos\gamma\, tan\phi\sin\chi)$$

$$V\frac{d\gamma}{dt} = \frac{1}{m}L\,cos\mu_a - g\cos\gamma + \frac{V^2}{r}\cos\gamma + 2\omega V\cos\phi\,cos\chi + \omega^2 r\cos^2\phi\;(\cos\gamma + \sin\gamma\, tan\phi\sin\chi) \tag{67}$$

$$V\frac{d\chi}{dt} = \frac{1}{m}\frac{L\sin\mu_a}{cos\gamma} - \frac{V^2}{r}\cos\gamma\cos\chi\tan\phi + 2\omega V(\tan\gamma\cos\phi\sin\chi - \sin\phi) - \frac{\omega^2 r}{\cos\gamma}\sin\phi\cos\phi\cos\chi$$

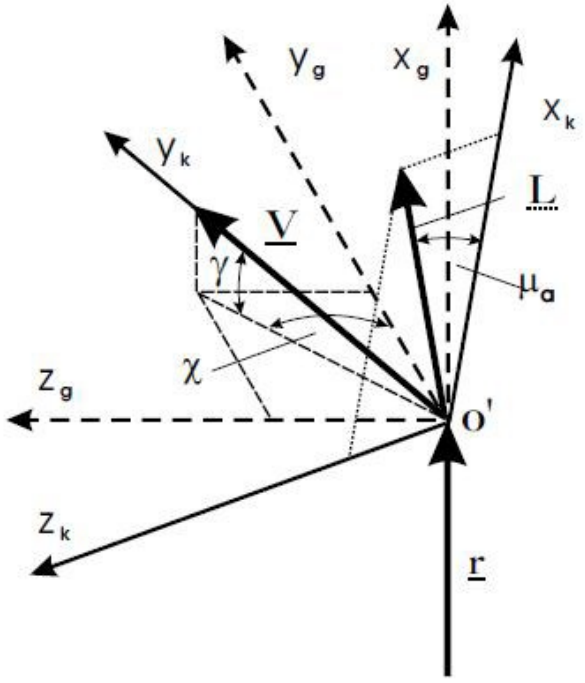

Figure 34: Coordinate system $\boldsymbol{O'}, x_g, y_g, z_g$ and coordinate system $\boldsymbol{O'}, x_k, y_k, z_k$ with the definition of bank angle μ_a and the lift $\boldsymbol{L}$ outside the $\boldsymbol{r}, \boldsymbol{V}$ plane. Image reproduced from Hirshel et al. analysis in [177].

The kinematic equations must be added to Eq. (67):

$$\frac{dr}{dt} = V \sin\gamma$$

$$\frac{d\theta}{dt} = \frac{V \cos\gamma \cos\chi}{r \cos\phi} \tag{68}$$

$$\frac{d\phi}{dt} = \frac{V \cos\gamma \sin\chi}{r}$$

Considering the planet as non-rotating ($\omega = 0$), Eq. (67) can be simplified:

$$\frac{dV}{dt} = -\frac{1}{m}D - g \sin\gamma$$

$$V\frac{d\gamma}{dt} = \frac{1}{m}L\, cos\mu_a - g\cos\gamma + \frac{V^2}{r}\cos\gamma \tag{69}$$

$$V\frac{d\chi}{dt} = \frac{1}{m}\frac{L \sin\mu_a}{cos\gamma} - \frac{V^2}{r}\cos\gamma \cos\chi \tan\phi$$

During the re-entry it is assumed that the flight path azimuth angle χ does not vary. The last equation in Eq. (29) can be neglected:

$$\frac{dV}{dt} = -\frac{1}{m}D - g\sin\gamma$$

$$V\frac{d\gamma}{dt} = \frac{1}{m}L\,cos\mu_a - g\cos\gamma + \frac{V^2}{r}\cos\gamma \tag{70}$$

And Eq. (68) reduces to:

$$\frac{dr}{dt} = V\sin\gamma$$

$$\frac{d\theta}{dt} = \frac{V\cos\gamma}{r} \tag{71}$$

The system of non-linear ordinary differential Eq. (70) and (71) for planar re-entry, in the form of:

$$\frac{dy}{dt} = f(y(t), t) \tag{72}$$

Can be integrated using a *Runge-Kutta method.* The most suitable algorithms from MATLAB ODE solvers is ODE45.

In Fig. 35-36-37 are presented the results of the integration of Eq. (72) performed in Matlab. The outcomes are computed considering an Orion-like vehicle and unpowered re-entry trajectory.

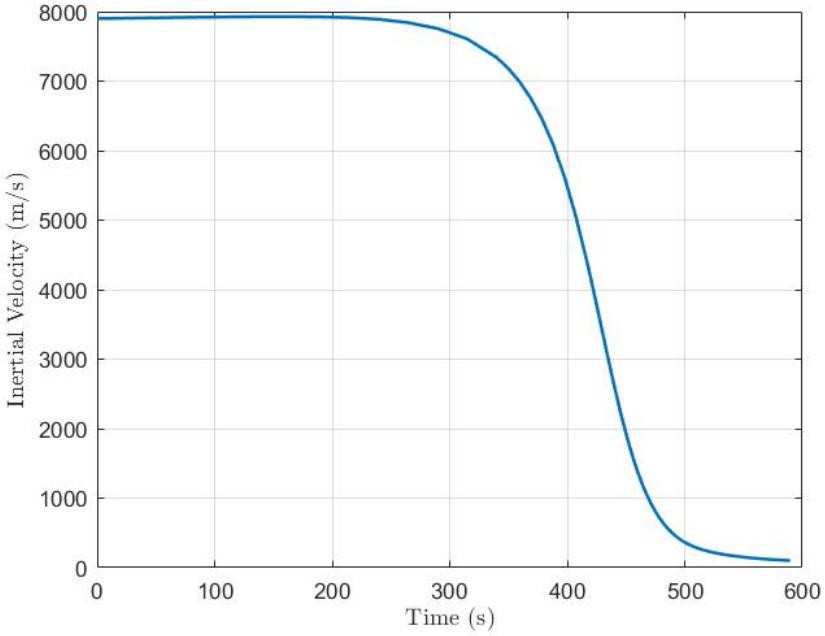

Figure 35: Inertial velocity- time relationship for Orion-like unpowered re-entry vehicle

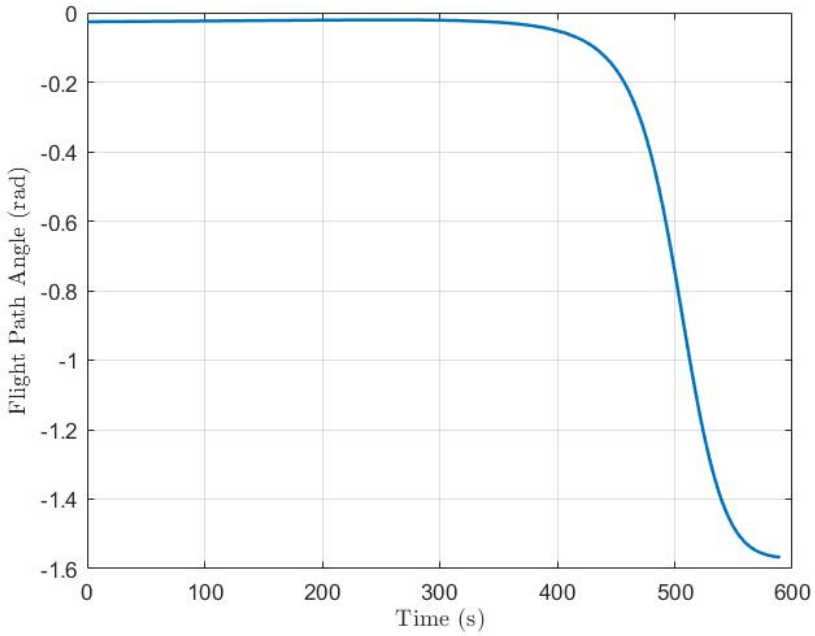

Figure 36: Flight Path Angle-time relationship for Orion-like unpowered re-entry vehicle

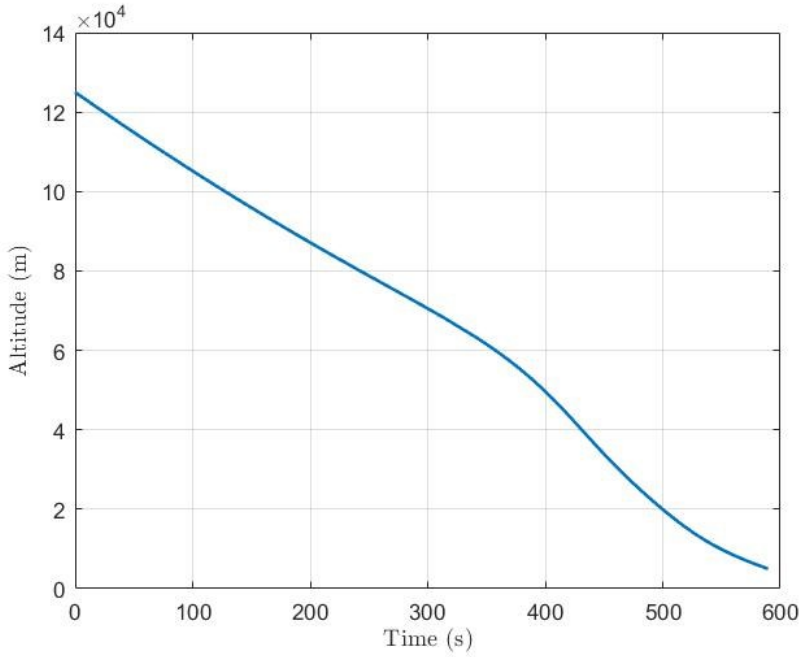

Figure 37: Altitude-time relationship for Orion-like unpowered re-entry vehicle

8.4.2. Atmospheric Model

Different atmospheric models can be considered:

- *1976 COESA model:*

  implemented in the MATLAB *Aerospace Toolbox* as a built-in function *atmoscoesa.* See reference *[181] for more details.*

- *Exponential model:*

$$\rho(H) = \rho_0 \exp(-\beta r) \tag{73}$$

Where $\rho_0 = 1.225\ kg/m^3$ is sea level density and $\beta = 1.40845 \cdot 10^{-4}\ m^{-1}$ is a model parameter.

8.4.3. Gravitational Model

In this work, spherical gravity model is considered. The Newton's law for Universal Gravitation is used to determine gravitational acceleration with altitude:

$$g\,(H) = g_0 \left(\frac{R_E}{R_E + H}\right)^2 \tag{74}$$

Where $g_0 = 9.81\ m/s^2$ is gravitational acceleration at sea level and $R_E = 6.378 \cdot 10^6\ m$ is Earth Radius.

8.4.4. Load Factor

After rearrangement of Eq. (70), the load factor normal n_n and along the flight path n_t can be obtained:

$$n_n = \frac{V}{g}\frac{d\gamma}{dt} \tag{75}$$

$$n_t = \frac{1}{g}\frac{dV}{dt} \tag{76}$$

In Fig. 38-39 are presented the normal and tangential load factor trends with altitude. The vehicle considered is an Orion-like capsule and the re-entry is performed with unpowered trajectory.

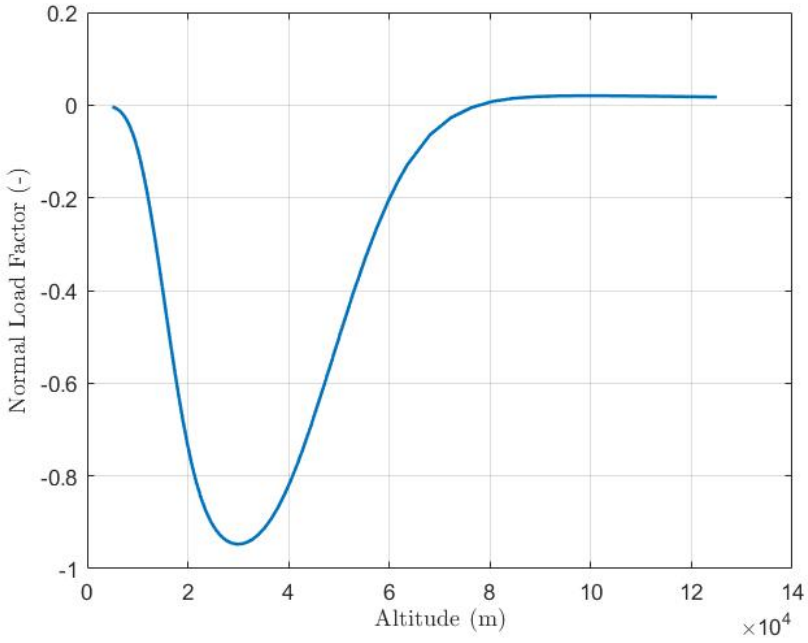

Figure 38: Normal load factor-time relationship for Orion-like unpowered re-entry vehicle.

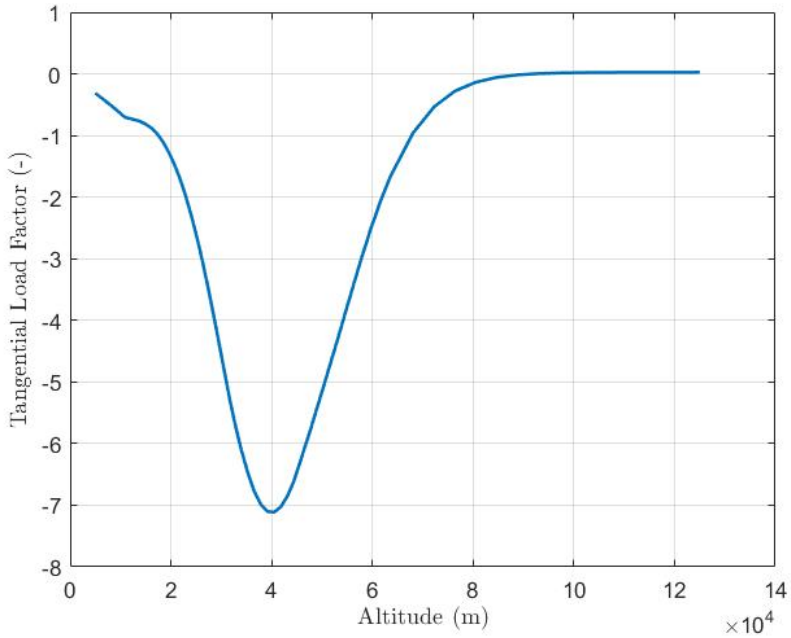

Figure 39: Tangential load factor-time relationship for Orion-like unpowered re-entry vehicle.

8.5. Flow Regimes

Knudsen number is defined as:

$$Kn = \frac{MFP}{L_c} \tag{77}$$

Where MFP is the mean free path between successive collisions of air molecules and L_c is the flowfield characteristic length, approximated with the body characteristic length. The mean free path can be expressed as:

$$MFP = \frac{1}{\sqrt{2}\pi\sigma_d^2 n_d} \tag{78}$$

Where $\sigma_d = 3.7 \cdot 10^{-10} m$ is the effective diameter of the gas particles and n_d is the number density that can be expressed as ratio between the air density and the molecule mass. The *Knudsen number* can be used to distinguish among three flow field regimes [178]:

- *Free Molecular Regime:* $Kn > 10$

  The Free Molecular Regime is characterized by a large mean free path and consequently by a long relaxation time. The continuum hypothesis does not hold, and it is necessary to consider the state of the single particles and their interaction with other particles and with boundaries. The *direct simulation Monte Carlo* (DSMC) method is required to simulate flow in the Free Molecular Regime.

- *Transition Regime:* $0.01 < Kn < 10$

  In the Transition Regime both the molecule-surface collisions and the intermolecular forces are important. The *direct simulation Monte Carlo* (DSMC) method is used in the upper limit of Transition Regime. The *Navier-Stokes* methods can be extended to the lower limit of Transition Regime.

- *Continuum Regime:* $Kn < 0.01$

  In the Continuum Regime the flow has a very short relaxation time and the macroscopic properties can be considered to vary continuously. *Navier-Stokes* methods are appropriate in the Continuum Regime.

8.5.1. Mach Number

The *Mach Number* allows to establish how important the compressibility effects of the fluid under examination are. In physics, the Mach number defines the ratio between a macroscopic speed V and the propagation speed a of longitudinal sound waves in the considered medium:

$$Ma = \frac{V}{a} \tag{79}$$

The motion around bodies can be classified in different conditions which correspond to different fluid dynamic behaviours, depending on the local Mach number:

- *Subsonic Regime:* $Ma < 0.8$.
- *Transonic Regime:* $0.8 < Ma < 1.3$.
- *Supersonic Regime:* $1.3 < Ma < 5.0$.
- *Hypersonic Regime:* $5.0 < Ma < 10.0$.
- *High-hypersonic Regime:* $10.0 < Ma < 25.0$.
- *Re-entry Speeds Regime:* $Ma > 25.0$.

8.5.2. Reynolds Number

The Reynolds number physically represents the ratio between the inertia and viscous forces acting on a fluid particle moving with speed V inside the same fluid:

$$Re = \frac{\rho V l_c}{\mu} \tag{80}$$

Where ρ is the air density, l_c is the body characteristic length and μ is the fluid dynamic viscosity. The dynamic viscosity can be estimated using *Sutherland's Law* [9]:

$$\mu = \mu_{ref} \left(\frac{T}{T_{ref}} \right)^{\frac{3}{2}} \frac{T_{ref} + S}{T + S} \tag{81}$$

Where T_{ref} is the reference temperature, μ_{ref} is the fluid dynamic reference viscosity and $S = 110\ K$ is a constant. The Reynolds number allows to evaluate whether the flow of a fluid is in a laminar regime (in correspondence with which there are lower values of the Reynolds number) or in a turbulent regime (in correspondence with higher values of the parameter). This transition between laminar and turbulent regime can be predicted by exploiting the Moody diagram, with which the viscous friction coefficient can be calculated starting from the values of the Reynolds number and the relative roughness. The Reynolds number values are to be considered "low" or "high" in relation to a specific system, in which they are fixed:

- The geometry of the body hit by the flow.
- The nature of the fluid.
- The operating conditions (temperature and pressure) under which the experience takes place.

8.6. Low Fidelity Aerothermodynamics Model

In order to speed up the computation some important assumptions have been made in modelling aerodynamic and thermodynamic physics. Specifically, the aerodynamic model introduces the *Oswatitsch Mach number indipendence principle* while the thermodynamic model exploits the semi-empirical correlations of Tauber-Sutton and Sutton-Grave to solve heat equation.

8.6.1. Aerodynamic Forces

A body of any shape immersed in a fluid current is affected by a fluid-dynamic action whose resultant in general will be directed obliquely with respect to the flight speed V. The force F is the resultant of the local actions that, due to the relative motion, are exerted on each external surface element of the body. For ease of study, the F is decomposed according to the directions of the reference wind axes in two components:

- *Lift*: is defined as the aerodynamic force acting perpendicular to the direction of the wind.

$$L = \frac{1}{2}\rho V^2 A_{ref} C_L \tag{82}$$

- *Drag:* is defined as the aerodynamic force acting in the direction of the wind.

$$D = \frac{1}{2}\rho V^2 A_{ref} C_D \tag{83}$$

Where A_{ref} is the area of the midship section of the vehicle, C_D is the drag coefficient and C_L is the lift coefficient. These coefficients are obtained by integrating the pressure distribution over the body surface.

The main approximation of the trajectory model is that C_L and C_D remain constant with altitude. At large Mach numbers, given a defined body geometry, force coefficients tend to become independent of the Mach number, as illustrated in Fig. 40. This is also true for the bow shock shape, the streamlines patterns, the sonic lines, the Mach lines in the supersonic region of the flow, the pressure coefficient and, in general, all force and moment coefficients. Physically, this may be interpreted saying that, as sound speed tends to infinity, the flow field “freezes” into

a limiting flow field. This principle can be found in literature as the *Oswatitsch Mach number indipendence principle* [10].

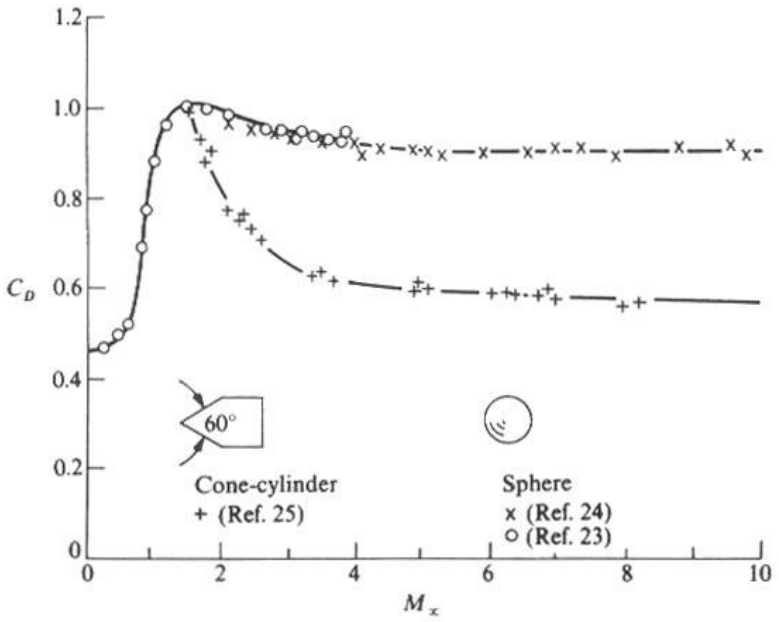

Figure 40: C_D tends to an asymptote as the Mach number increases.

8.6.2. Stagnation Point Pressure

Stagnation point pressure is the static pressure at a stagnation point in a fluid flow. In fluid dynamics, the stagnation point is a point in the flow field where the local instantaneous velocity is zero and all kinetic energy is converted into pressure energy isentropically. The stagnation point is on the surface of the object, in the flow field, which causes the flow to stop. So, stagnation point pressure is equal to the sum of the free-stream dynamic pressure $\frac{1}{2}\rho V^2$ and free-stream static pressure p_∞ [2]:

$$p = p_\infty + \frac{1}{2}\rho V^2 \tag{84}$$

8.6.3. Adiabatic Wall Temperature

Adiabatic wall temperature is the temperature acquired by a wall in liquid or gas flow if the condition of thermal insulation is observed on it. The thermal insulation condition can be defined as the zero-wall normal temperature gradient condition: $\left(\frac{\partial T}{\partial n}\right)_w = 0$. It is sometimes called equilibrium temperature or recovery temperature. The concept of adiabatic wall temperature is used in the field of high velocity aerodynamics. The temperature profile in the boundary layer of a high-velocity gas flow over an adiabatic surface is displayed in Fig.41, Curve 1. Curve 2 shows a typical distribution of temperature for the case where heat is being added through the surface and Curves 3 and 4 are typical of cases where heat is being removed from the fluid via the surface

For a compressible gas, adiabatic wall temperature can be calculated as [182]:

$$T = T_\infty(1 + r_{air}\frac{\gamma_{air} - 1}{2}M_\infty^2) \tag{85}$$

Where $\gamma_{air} = 1.4$ is the isoentropic coefficient and r_{air} is the recovery coefficient. r_{air} is not a constant but depends on the character of the flow on the surface, the flow regime, and the thermal properties of the medium. For simple geometries, its value can be estimated [182]:

- $r_{air} = 1$ at the front stagnation point of the body.
- $r_{air} = \sqrt{Pr}$ in a laminar boundary layer with $0.5 < Pr < 10$.
- $r_{air} = \sqrt[3]{Pr}$ in a turbulent boundary layer with Prandtl number close to 1

For supersonic flows when variations of the thermal properties of gas become significant, the above relations hold for the enthalpy field [182]:

$$h_r = h_\infty + \frac{r_{air}V_\infty^2}{2} \tag{86}$$

In estimating the recovery coefficient, the Prandtl number is chosen at the reference *Eckert enthalpy* [11]*:*

$$h^* = h_\infty 0.5(h_w - h_\infty) + 0.22(h_r - h_\infty) \tag{87}$$

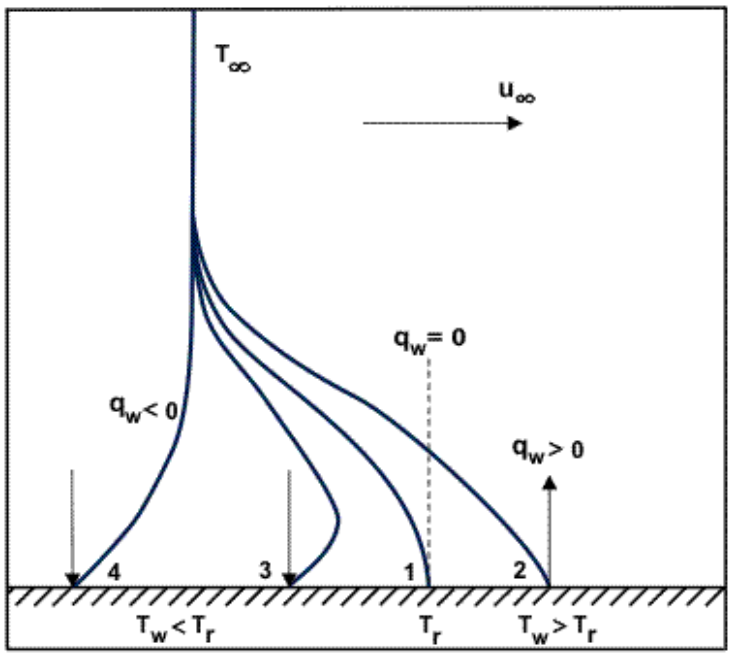

Figure 41: Temperature profiles in the boundary layer of a high-velocity gas flow. The Image is taken from Dorrance et al. work in [182]).

8.6.4. General Heat Equation

The general heat equation is a parabolic partial differential equation that describes the distribution of heat (or temperature change) in each region over time:

$$\rho C_P t_z \frac{\partial T}{\partial t} = \nabla(k t_z \nabla T) - q_r - q_c + q_s \tag{88}$$

Where the left term is the non-stationary one, while the right term is the sum of the conductive term, the radiative term $q_r = \sigma\varepsilon(T^4 - T_a^4)$, the convective term $q_c = h(T - T_a)$ and the source term q_s.

In the following paragraphs are presented experimental correlations to approximate the terms in Eq. (88).

In Fig. 42 is reported the outcome of Eq. (88) considering empirical formulations for the heat terms. The vehicle considered is an Orion-like capsule and the re-entry is performed with uncontrolled trajectory.

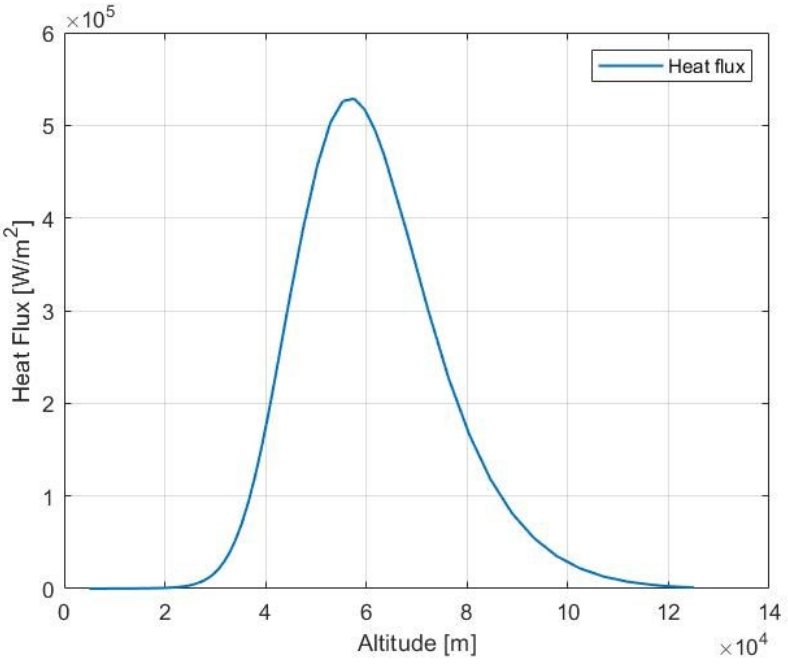

Figure 42: Heat Flux-Altitude relationship for Orion-like unpowered re-entry vehicle. Experimental formulations are considered in computing the heat equation.

8.6.5. Tauber-Sutton Correlation

The Radiative heat transfer is computed using the *Tauber-Sutton* radiative heating correlation for Earth re-entry [8]:

$$q_{rad} = C\ R_n^a\ \rho^b f(V) \tag{89}$$

Where q_{rad} is the radiative heat-transfer rate into flight body, C, a, b are constants based on the planetary atmosphere, R_n is the hemispherical nose radius, ρ is the freestream density and $f(V)$ is a tabulated function of velocity [8].

V (m/s)	f (V)
9000	1.5
9250	4.3
9500	9.7
9750	19.5
10000	35
10250	55
10500	81
10750	115
11000	151
11500	238
12000	359
12500	495
13000	660
13500	850
14000	1065
14500	1313
15000	1550
15500	1780
16000	2040

Table 5: Radiative heating velocity function for earth.

8.6.6. Sutton-Grave Correlation

The Convective heat transfer is computed using the *Sutton-Grave* convective heating correlation for Earth re-entry [12]:

$$q_{conv} = k \sqrt{\frac{\rho}{R_n}} \left(\frac{V_\infty}{1000}\right)^3 \tag{90}$$

Where q_{conv} is the convective heat-transfer rate into flight body, k is a constant based on planetary atmosphere, ρ is the free stream density and V_∞ is the flight velocity.

8.7. High Fidelity Aerothermodynamics Model

SU2 Computational Fluid Dynamic (CFD) is the solver used in this work to closely predict the Aerothermodynamics variables of interest. SU2 is a suite of open-source software tools written in C++ for the numerical solution of partial differential equations (PDE) and performing PDE-constrained optimization. In the following paragraphs the physical models present in SU2 are explained in detail.

8.7.1. Navier-Stokes Equations

The Navier-Stokes equations are considered to be the complete governing flow equations. The equations are derived by applying the conservation laws (conservation of mass, momentum, and energy) on a given control volume. The resulting equations[2], including what contribution each term provides, yield [178]:

$$\frac{\partial \rho_i}{\partial t} + \nabla \cdot (\rho \boldsymbol{V}) = 0 \tag{91}$$

$$\frac{\partial \rho_i}{\partial t} + \nabla \cdot (\rho_i \boldsymbol{V_i}) = \frac{d\omega_i}{dt} \tag{92}$$

$$\frac{\partial (\rho \boldsymbol{V})}{\partial t} + \rho(\boldsymbol{V} \cdot \nabla \boldsymbol{V}) + \nabla p + \rho \boldsymbol{f} + \nabla \cdot \left[\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \lambda \nabla \cdot \boldsymbol{V} \right] = 0 \tag{93}$$

$$\frac{\partial (\rho E)}{\partial t} + \nabla \cdot (\rho \boldsymbol{V}) E = \nabla \cdot (k \nabla T) - \nabla \cdot \left(\sum_i \rho_i \boldsymbol{V}_i h_i \right) - \nabla \cdot \boldsymbol{q_R} - \nabla \cdot (p \boldsymbol{V}) + \nabla \cdot \left(\boldsymbol{V} \cdot \tau_{ij} \right) \tag{94}$$

The Navier-Stokes equations include viscous effects, radiative effects, chemical reactions, species continuity in Eq. (92), diffusion and conduction. To fully simulate a flow, all these terms are necessary, especially for the hypersonic high temperature

flow considered in this thesis. However, creating a CFD software package to take all terms into account that can be validated and verified is difficult. In addition, the computational time would increase drastically. SU2 has capabilities of computing chemically reacting flow, however, this feature is not fully developed yet. Since the software is not specialised towards hypersonic flows, the following quantities from Eq. (92) and (94) are not included:

- Species continuity in Eq. (30),
- Body force,
- Transport of energy due to chemical diffusion,
- Transport of energy due to radiation.

The resulting governing equations for the analysis of this thesis are [178]:

$$\frac{\partial \rho_i}{\partial t} + \nabla \cdot (\rho \boldsymbol{V}) = 0 \tag{95}$$

$$\frac{\partial (\rho \boldsymbol{V})}{\partial t} + \rho(\boldsymbol{V} \cdot \nabla \boldsymbol{V}) + \nabla p + \nabla \cdot \left[\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \lambda \nabla \cdot \boldsymbol{V} \right] = 0 \tag{96}$$

$$\frac{\partial (\rho E)}{\partial t} + \nabla \cdot (\rho \boldsymbol{V}) E = \nabla \cdot (k \nabla T) - \nabla \cdot (p \boldsymbol{V}) + \nabla \cdot \left(\boldsymbol{V} \cdot \tau_{ij} \right) \tag{97}$$

8.7.2. Stagnation Point Heat Flux: Fay-Riddle Correlations

A well-accepted engineering method to determine the stagnation point heat flux on a vehicle in a hypersonic flow is the method derived by Fay and Riddell. The equations are based on the transformed boundary layer energy equation. The assumptions associated with Fay and Riddell are [6] listed below and visualised in Fig. 43.

- The shock-layer is partially dissociated and the flow conditions at the outer edge of the boundary layer are those for local thermodynamic and chemical equilibrium. The chemical reactions only take place in the boundary layer and the thermodynamic state of the flow outside the boundary layer does not change over time.
- Regions of nonequilibrium, equilibrium or frozen flow may be present in the boundary layer, depending on the characteristic time for a fluid element to move through the flow field and the characteristic time for a fluid element to reach chemical equilibrium.
- The inviscid velocity distribution at the outer edge of the boundary layer in the stagnation region is given by the incompressible relation:

$$u_e = ax, \qquad a = \left(\frac{du_e}{dx}\right)_s \tag{98}$$

- The wall can be equilibrium catalytic or non-catalytic. That means that the surface material influences the chemical composition, such that the chemical composition at the wall are the local equilibrium values at the temperature and pressure at the wall (equilibrium catalytic) or the surface material has no influence on the chemical composition at all (non-catalytic).
- The gas is assumed to consist of heavy particles and light particles (binary mixture, molecules, or atoms). This is a simplification to diffusion, such that an average for the transport coefficients can be assumed for the molecules and atoms separately.

The equations derived by Fay and Riddell can be divided into three specific types of flow: an equilibrium boundary layer, a frozen boundary layer with equilibrium catalytic wall and a frozen boundary layer with a non-catalytic wall. The equations yield [6]:

- Equilibrium boundary layer (spherical nose):

$$q_w = 0.76Pr^{-0.6}\,(\rho_e\mu_e)^{0.4}(\rho_w\mu_w)^{0.1}\sqrt{\left(\frac{du_e}{dx}\right)_s}\,(h_{0_e} - h_w) \times \left[1 + (Le^{0.52} - 1)\left(\frac{h_D}{h_{0_e}}\right)\right] \tag{99}$$

- Frozen boundary layer with an equilibrium catalytic wall (spherical nose):

$$q_w = 0.76Pr^{-0.6}\,(\rho_e\mu_e)^{0.4}(\rho_w\mu_w)^{0.1}\sqrt{\left(\frac{du_e}{dx}\right)_s}\,(h_{0_e} - h_w) \times \left[1 + (Le^{0.52} - 1)\left(\frac{h_D}{h_{0_e}}\right)\right] \tag{100}$$

- Frozen boundary layer with a non-catalytic wall (spherical nose):

$$q_w = 0.76Pr^{-0.6}\,(\rho_e\mu_e)^{0.4}(\rho_w\mu_w)^{0.1}\sqrt{\left(\frac{du_e}{dx}\right)_s}\,(h_{0_e} - h_w) \times \left[1 + (Le^{0.52} - 1)\left(\frac{h_D}{h_{0_e}}\right)\right] \tag{101}$$

The stagnation-point velocity gradient is given using a pressure distribution based on the Newtonian theory:

$$\left(\frac{du_e}{dx}\right)_s = \frac{1}{R}\sqrt{\frac{2(p_e - p_\infty)}{\rho_e}} \tag{102}$$

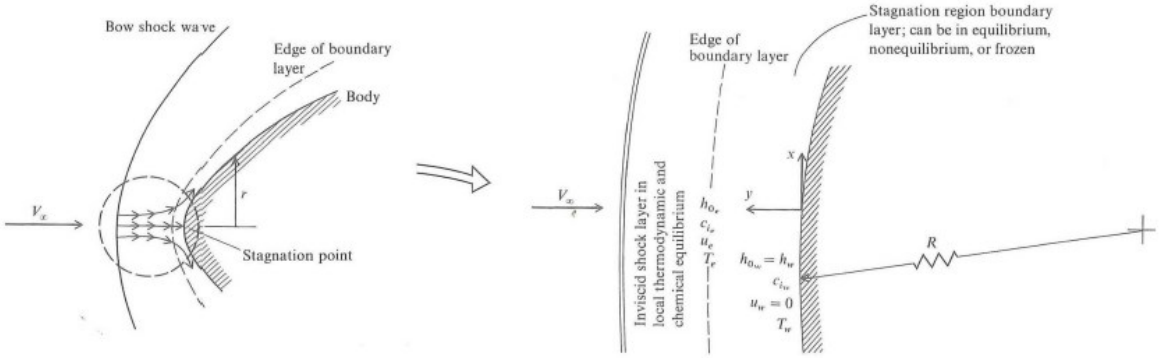

Figure 43: Overview of the stagnation region flow model. The figure on the left shows the bow shock wave and the stagnation region where the shock wave can be assumed a normal shock. The figure on the right shows a close-up of the stagnation region, where outside the boundary layer an inviscid shock layer is present. In the boundary layer, equilibrium, non-equilibrium, or frozen flow can be present. Images taken from Anderson et al. [178].

8.7.3. Spatial Discretization

Spatial discretisation methods are required to cast the governing flow equations (for example Navier-Stokes) into a discretised form for the use of CFD. For hypersonic flow, there are two main spatial discretisation techniques for the convective fluxes that are useful, the central scheme and the upwind scheme. Both methods discretise the governing flow equations to solve the state at a certain cell vertex.

A central scheme is based on the Taylor expansion. Performing a Taylor expansion around a first order derivative yields [12]:

$$\left(\frac{\partial u}{\partial x}\right)_i = \frac{u_{i+1} - u_i}{\Delta x} - \frac{\Delta x}{2}(u_{xx})_i - \frac{\Delta x^2}{6}(u_{xxx})_i + \cdots = \frac{u_{i+1} - u_i}{\Delta x} + \mathcal{O}(\Delta x) \qquad (103)$$

Using the same Taylor expansion, a relation for u_{i-1} can be derived:

$$\left(\frac{\partial u}{\partial x}\right)_i = \frac{u_i - u_{i-1}}{\Delta x} - \frac{\Delta x}{2}(u_{xx})_i - \frac{\Delta x^2}{6}(u_{xxx})_i + \cdots = \frac{u_i - u_{i-1}}{\Delta x} + \mathcal{O}(\Delta x) \quad (104)$$

When these forward and backward difference equations are added up, the central difference equation is obtained:

$$\left(\frac{\partial u}{\partial x}\right)_i = \frac{u_{i+1} - u_{i-1}}{2\Delta x} + \mathcal{O}(\Delta x^2) \quad (105)$$

In the central scheme, the evaluated grid point is computed using grid points in front and behind the central point. The finite difference representation of the first derivate is of order two. The central scheme does not generate two independent solutions of the equations, which physically means that upstream variables can be changed by this scheme. This causes instability in the solution; thus, artificial dissipation must be added to increase the stability. An example of a central scheme with artificial dissipation is the Jameson-Schmidt-Turkel scheme, which is also present in SU2.

Upwind schemes are biased towards computing the current grid point by using the grid points upstream of the current grid point [12]:

$$u_i = f(u_{i-1}, u_{i-2}, \dots, u_{i-n}) \quad (106)$$

The upwind scheme considers the wave propagation direction, which the central schemes do not. This makes the upwind scheme suitable for an accurate computation of the boundary layer. For capturing discontinuities, upwind schemes are particularly interesting. When considering supersonic flow, the information can only propagate in downstream direction, thus using only upstream values to compute a certain grid point state is valid. The downside of upwind schemes is notable in the higher-order accuracy schemes where limiters are needed to prevent oscillations near discontinuities.

8.7.4. Time Integration

There are various numerical time integration methods available. In SU2, the Euler method and Runge-Kutta method are considered. Numerical integration methods are used to solve ordinary or partial differential equations. An integration method may have either an implicit scheme, an explicit scheme or both. Explicit schemes are generally easier to implement and to solve. For example, let x_{n+1} denote the state of a differential equation at $t = t_{n+1}$ (the new state) and x_n at $t = t_n$ (the old state). Then, for an explicit scheme, the new state would be calculated using:

$$x_{n+1} = f(x_n, x_{n-1}, x_{n-2}, \dots) \tag{107}$$

Implicit schemes are more difficult to implement, as the new state is also included in the function to compute the new state. Implicit schemes take more time to compute, as the new state has to be computed iteratively, thus more function evaluations are necessary to compute the new state. Consider again the old and new states as mentioned above, then an implicit scheme would be:

$$x_{n+1} = g(x_{n+1}, x_n, x_{n-1}, \dots) \tag{108}$$

Although explicit schemes seem more interesting since they are easier and require less computation time, implicit schemes are more stable than explicit schemes. Moreover, implicit schemes may attain a higher order of accuracy, while using a larger step size than explicit schemes. Decreasing the step size in explicit schemes to attain a higher order has a negative effect: as the error in computation decreases (truncation error), the rounding error due to the small step size increases. In this section, the explicit schemes of the integration methods are discussed.

The Euler method is the simplest integration method available. The Euler integrator is evaluated at the beginning of the step size interval. The method requires sufficient initial values to the differential equation. For example, consider an ordinary differential equation of the form [179]:

$$\frac{dx}{dt} = x'(t) = \lambda(t)x(t), \quad t > 0, \qquad x(0) = c \tag{109}$$

For small step sizes, this method proves to be rather accurate. However, as the step size is increased, the error is also increased. An advantage of the Euler method is that it is fast.

The Runge-Kutta method is a higher order integration method. To compute a next time step, it uses information about the previous time step, similar to the Euler method. The Runge-Kutta integrator is different from the Euler method. Whereas Euler only evaluates the new time step based on information at the beginning of the step interval, the Runge-Kutta method uses information about different locations from the step interval. The general Runge-Kutta scheme is given by the following equations [180]:

$$x_{n+1} = x_n + h\sum_{i=1}^{s} b_i k_i \tag{110}$$

Where

$$k_i = f\left(t_n + c_i h_i x_n + h\sum_{j=1}^{s} a_{ij} k_j\right) \tag{111}$$

And

$$c_i = \sum_{j=1}^{s} a_{ij}\ , i = 1{:}\,s \tag{112}$$

In these equations s denotes the number of stages the Runge-Kutta integrator uses. The coefficients $c_i, b_i\ and\ a_i$ are given in Butcher arrays [179] per Runge-Kutta stage.

As mentioned before, implicit methods are more difficult to implement, require more computational time and are more stable than explicit methods. Larger time steps can be taken whilst maintaining higher order accuracy. In SU2, the explicit Euler and Runge-Kutta schemes and implicit Euler scheme is implemented. To achieve convergence, a small-time step (CFL number) is required with a fourth order Runge-Kutta explicit scheme. Due to the small-time step, the order of accuracy is reduced for this method (due to rounding errors), and stability is not guaranteed. Hence, the implicit Euler method is used as a larger time stepping value can be used, the solution tends to converge, and the order of accuracy remains unchanged.

8.8. Thermal Protection System Temperature Evaluation

Thermal protection systems are the features incorporated into a spacecraft's design to protect it from the severe aerodynamic heating during high-speed travel through planetary atmospheres. The determination of the temperature reached by the system is an important design evaluation to size it correctly. In the following paragraphs are presented the geometry modelling and the resolution of the thermal protection system heat equation. The dissertation is taken from the work of Scaramuzzino et al. in [183].

8.8.1. Thermal Protection System Geometry Modelling

In first approximation for preliminary sizing the Thermal protection system (TPS) can be modelled as an arc of circumference, assuming that the vehicle cane be assimilated to a sphere with a radius equal to the nose radius R_N. The model is show in Fig. 44.

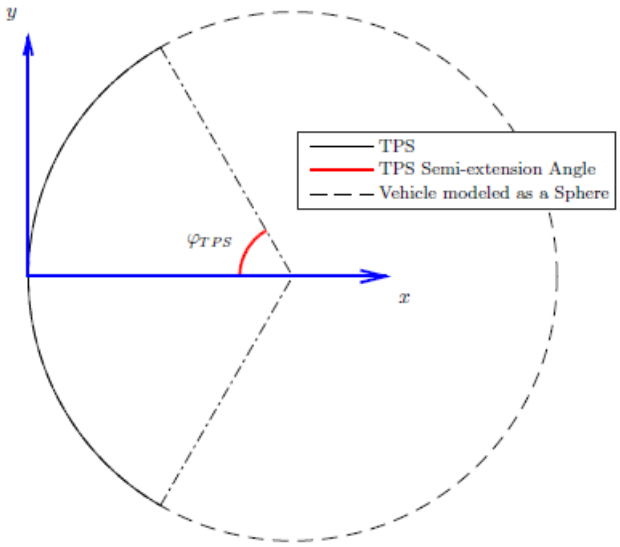

Figure 44: TPS modelling. Image from the work of Scaramuzzino et al. in [183].

Given this TPS model, the Heat equation in Eq. (89) can be specialized for the two-dimensional case by considering uniform thermal conductivity k and thickness t_z and no convective term [183]:

$$\rho\, C_P\, t_z \frac{\partial T}{\partial t} = k\, t_z \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) - \sigma\varepsilon\, (T^4 - T_a^4) + q_s \tag{113}$$

Eq. (29) is non-linear due to the presence of the radiative term. In order to use a standard finite element approximation, a linearization must be performed [183]:

$$\rho\, C_P\, t_z \frac{\partial T}{\partial t} = k\, t_z \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) - 4\sigma\varepsilon T_a^3 T + 4\sigma\varepsilon T_a^4 + q_s \tag{114}$$

Further step is to discretize the TPS and specialize Eq. (114) for each element. In Fig. 45 is illustrated the discretization of the TPS using four elements and the global numeration adopted for the node.

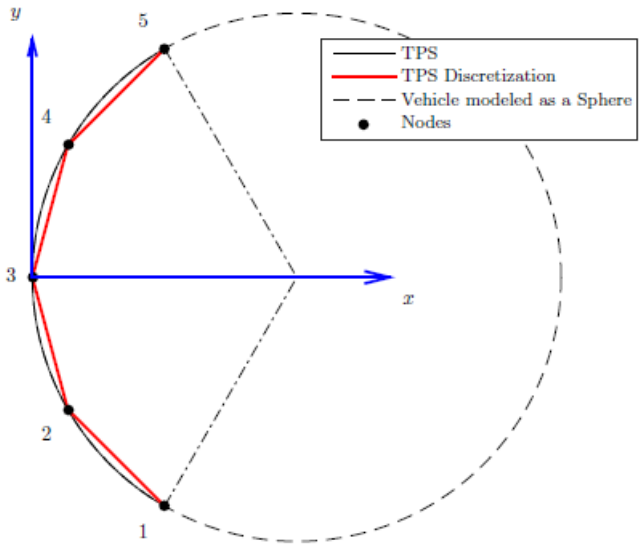

Figure 45: TPS discretization, example with four elements. Image from the work of Scaramuzzino et al. in [183].

Considering a single element, as shown in Fig.46, if equidistant nodes are used [183]:

$$n_n = n_e + 1 \tag{115}$$

$$\varphi = \varphi_e = \frac{2\varphi_{TPS}}{n_e} \tag{116}$$

$$l = l'_e R_n \phi_e \tag{117}$$

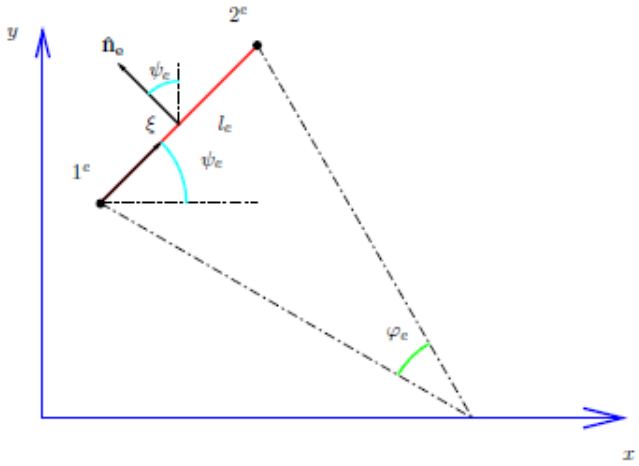

Figure 46: TPS element. Image from the work of Scaramuzzino et al. in [183].

$$\varphi_e = \arctan\left(\frac{y_2^e - y_1^e}{x_2^e - x_1^e}\right), \qquad e = 1{:}\,n_e \tag{118}$$

$$\widehat{\boldsymbol{n_e}} = \{-sin\psi_e \; cos\psi_e\} \tag{119}$$

$$x = x_1^e + \xi cos\psi_e \tag{120}$$

$$y = y_1^e + \xi sin\psi_e \tag{121}$$

Using the *chain rule* is possible to change coordinates:

$$\frac{\partial}{\partial x} = \frac{\partial \xi}{\partial x}\frac{\partial}{\partial \xi} = \frac{1}{cos\psi_e}\frac{\partial}{\partial \xi} \tag{122}$$

$$\frac{\partial}{\partial y} = \frac{\partial \xi}{\partial x}\frac{\partial}{\partial \xi} = \frac{1}{sin\psi_e}\frac{\partial}{\partial \xi} \tag{123}$$

$$\frac{\partial^2}{\partial x^2} = \frac{\partial}{\partial x}\left(\frac{1}{cos\psi_e}\frac{\partial}{\partial \xi}\right) = \frac{1}{\cos^2 \psi_e}\frac{\partial^2}{\partial \xi^2} \tag{124}$$

$$\frac{\partial^2}{\partial y^2} = \frac{\partial}{\partial y}\left(\frac{1}{sin\psi_e}\frac{\partial}{\partial \xi}\right) = \frac{1}{\sin^2 \psi_e}\frac{\partial^2}{\partial \xi^2} \tag{125}$$

Using Eq. (124) and (125), Eq. (114) becomes [183]:

$$\rho\, C_P\, t_z \frac{\partial T}{\partial t} = k\, t_z \left(\frac{1}{\cos^2 \psi_e} + \frac{1}{\sin^2 \psi_e}\right)\frac{\partial^2 T}{\partial \xi^2} - 4\sigma\varepsilon T_a^3 T + 4\sigma\varepsilon T_a^4 + q_{s_e} \tag{126}$$

Recalling that:

$$\frac{1}{\cos^2\psi_e}+\frac{1}{\sin^2\psi_e}=\frac{1}{\sin^2\psi_e\cos^2\psi_e}=\frac{4}{\sin(2\psi_e)} \tag{127}$$

Eq. (126) becomes:

$$\rho\, C_P\, t_z\frac{\partial T}{\partial t}=\frac{4kt_z}{\sin(2\psi_e)}\frac{\partial^2 T}{\partial \xi^2}-\,4\sigma\varepsilon T_a^3 T+4\sigma\varepsilon T_a^4+q_{s_e} \tag{128}$$

Introducing:

$$k_e=\frac{4k}{\sin(2\psi_e)} \tag{129}$$

Eq. (128) becomes:

$$\rho\, C_P\, t_z\frac{\partial T}{\partial t}=k_e t_z\frac{\partial^2 T}{\partial \xi^2}-\,4\sigma\varepsilon T_a^3 T+4\sigma\varepsilon T_a^4+q_{s_e} \tag{130}$$

8.8.2. Discretization

Using the *method of mean weighted residuals* is possible to write the weak formulation of the problem [183]. Eq. (130) is likely to be written as:

$$f\big(T(\xi,t)\big) = 0 \tag{131}$$

Supposing that Eq. (45) is satisfied only in a finite number of points, while everywhere else only in average. This can be done multiplying Eq. (45) by a *weight function* $W(\xi)$, integrating over the domain and forcing it to be equal to zero:

$$\int_0^{l_e} W(\xi) f\big(T(\xi,t)\big) d\xi = 0 \tag{132}$$

The temperature along each element is approximated using the *separation of variables technique,* with the *shape function* N:

$$T(\xi,t) = N(\xi)\theta_e(t) \tag{133}$$

Applying the *Galerkin method,* using the shape function as test function:

$$\int_0^{l_e} N^T(\xi)\, f\big(T(\xi,t)\big) d\xi = 0 \tag{134}$$

$$\int_0^{l_e} N^T(\xi) \left[\rho\, C_P\, t_z \frac{\partial T}{\partial t} - k_e t_z \frac{\partial^2 T}{\partial \xi^2} + 4\sigma\varepsilon T_a^3 T - (4\sigma\varepsilon T_a^4 + q_{s_e})\right] d\xi = 0 \tag{135}$$

Let us analyse the terms of Eq. (135):

- Non-stationary term:

$$\int_0^{l_e} N^T(\xi)\,\rho\,C_P\,t_z\frac{\partial T}{\partial t}d\xi = \rho\,C_P\,t_z\int_0^{l_e} N^T(\xi)\,N(\xi)d\xi\frac{\partial \vartheta_e}{\partial t} = \boldsymbol{A_e\dot{\theta}_e} \tag{136}$$

- Conductive term:

$$\int_0^{l_e} N^T(\xi)\,k_e t_z\,\frac{\partial^2 T}{\partial \xi^2}d\xi$$

$$= -k_e t_z\left[N^T(\xi)\frac{\partial T}{\partial \xi}\right]_0^{l_e} + k_e t_z\int_0^{l_e}\left[\frac{\partial N}{\partial \xi}\right]^T\frac{\partial T}{\partial \xi}d\xi =$$

$$-k_e t_z\left[N^T(\xi)\frac{\partial T}{\partial \xi}\right] + k_e t_z\int_0^{l_e} N_{/\xi}^T N_{/\xi}d\xi\vartheta_e =$$

$$\boldsymbol{b_e + K_e^{cond}\vartheta_e} \tag{137}$$

- Radiative term:

$$\int_0^{l_e} N^T(\xi)\,4\sigma\varepsilon T_a^3 T d\xi = 4\sigma\varepsilon T_a^3\int_0^{l_e} N^T(\xi)N(\xi)d\xi\vartheta_e = \boldsymbol{K_e^{rad}\vartheta_e} \tag{138}$$

- Constant term:

$$-\int_0^{l_e} N^T(\xi)\left(4\sigma\varepsilon T_a^4 + q_{s_e}\right)d\xi = -(4\sigma\varepsilon T_a^4 + q_{s_e})\int_0^{l_e} N^T d\xi = -f_e(t) \tag{139}$$

Since T_a and q_{s_e} change with velocity and altitude and consequently with time, $f_e(t)$ is time dependent.

In conclusion, for each element the following equation can be written:

$$\boldsymbol{\dot{A}_e}\dot{\vartheta}_e + \boldsymbol{K_e}\vartheta_e = f_e(t) \tag{140}$$

Where:

$$\boldsymbol{K_e} = \left(\boldsymbol{K_e^{cond}} + \boldsymbol{K_e^{rad}}\right) \tag{141}$$

Considering linear shape function in Fig.47, Eq. (133) becomes:

$$T(\xi,t) = [\,N_1(\xi) \quad N_2(\xi)]\{\vartheta_{1e}(t) \quad \vartheta_{2e}(t)\}' = \left[1 - \frac{\xi}{l_e} \quad \frac{\xi}{l_e}\right]\{\vartheta_{1e}(t) \quad \vartheta_{2e}(t)\}' \tag{142}$$

Where $\vartheta_{1e}(t)$ and $\vartheta_{2e}(t)$ are the nodal temperatures.

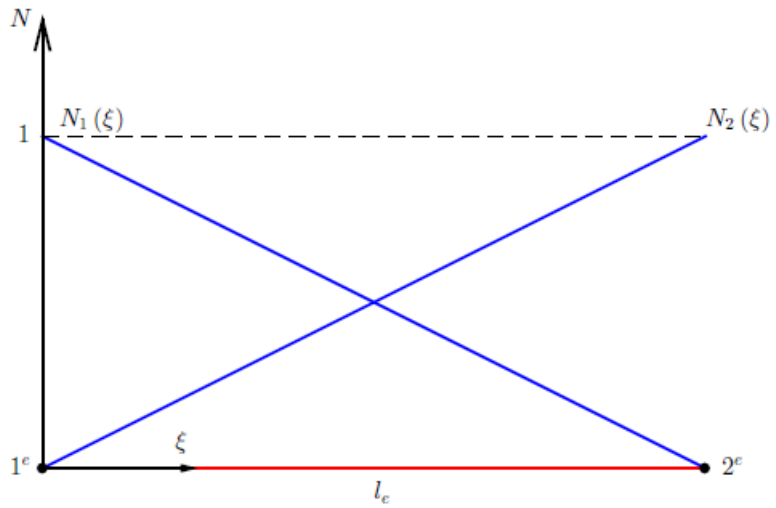

Figure 47: Linear shape functions. Image from the work of Scaramuzzino et al. in [183].

In this way the matrices can be evaluated for each element:

$$N_{/\xi} = [\,-\frac{1}{l_e} \quad \frac{1}{l_e}] \tag{143}$$

$$\begin{aligned}\int_0^{l_e} N^T(\xi)N(\xi)d\xi \\ &= \int_0^{l_e} \begin{bmatrix} 1-\frac{\xi}{l_e} \\ \frac{\xi}{l_e} \end{bmatrix} \begin{bmatrix} 1-\frac{\xi}{l_e} & \frac{\xi}{l_e} \end{bmatrix} d\xi \\ &= \int_0^{l_e} \begin{bmatrix} \left(1-\frac{\xi}{l_e}\right)^2 & \frac{\xi}{l_e}-\frac{\xi^2}{l_e^2} \\ \frac{\xi}{l_e}-\frac{\xi^2}{l_e^2} & \frac{\xi^2}{l_e^2} \end{bmatrix} d\xi = \begin{bmatrix} \frac{l_e}{3} & \frac{l_e}{6} \\ \frac{l_e}{6} & \frac{l_e}{3} \end{bmatrix} = \frac{l_e}{6}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}\end{aligned} \tag{144}$$

$$\begin{aligned}\int_0^{l_e} N_{/\xi}^T\ N_{/\xi} d\xi = \int_0^{l_e} \begin{bmatrix} -\frac{1}{l_e} \\ \frac{1}{l_e} \end{bmatrix} \begin{bmatrix} -\frac{1}{l_e} & \frac{1}{l_e} \end{bmatrix} d\xi \\ &= \int_0^{l_e} \begin{bmatrix} \frac{1}{l_e^2} & -\frac{1}{l_e^2} \\ -\frac{1}{l_e^2} & \frac{1}{l_e^2} \end{bmatrix} d\xi = \begin{bmatrix} \frac{1}{l_e} & -\frac{1}{l_e} \\ -\frac{1}{l_e} & \frac{1}{l_e} \end{bmatrix} = \frac{1}{l_e}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\end{aligned} \tag{145}$$

$$\int_0^{l_e} N^T\, d\xi = \int_0^{l_e} \begin{bmatrix} 1-\frac{\xi}{l_e} \\ \frac{\xi}{l_e} \end{bmatrix} d\xi = \begin{Bmatrix} \frac{l_e}{2} \\ \frac{l_e}{2} \end{Bmatrix} = \frac{l_e}{2}\begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \tag{146}$$

$$\boldsymbol{A_e} = \rho C_P t_z \int_0^{l_e} N^T(\xi) N(\xi) d\xi = \frac{\rho C_P t_z l_e}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \tag{147}$$

$$\boldsymbol{K_e^{cond}} = k_e t_Z \int_0^{l_e} N_{/\xi}^T N_{/\xi} d\xi = \frac{k_e t_z}{l_e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{148}$$

$$\boldsymbol{K_e^{rad}} = 4\sigma\varepsilon T_a^3 \int_0^{l_e} N^T(\xi) N(\xi) d\xi = \frac{4\sigma\varepsilon T_a^3 l_e}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \tag{149}$$

$$f_e(t) = (4\sigma\varepsilon T_a^4 + q_{s_e}) \int_0^{l_e} N^T d\xi = \frac{\left(4\sigma\varepsilon T_a^4 + q_{s_e}\right) l_e}{2} \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \tag{150}$$

A further approximation is introduced to make $\boldsymbol{K_e^{rad}}$ non time dependent:

$$K_e^{rad} = \frac{4\sigma\varepsilon T_a^3|_{h=0} l_e}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \tag{151}$$

8.8.3. Boundary Conditions

Two different boundary conditions can be considered [183]:

- *Dirichlet boundary conditions:* the temperature at the boundary is imposed:

$$T|_1(t) = \theta_1(t) = \overline{T_1}(t) \tag{152}$$

$$T|_{n_n}(t) = \theta_{n_n}(t) = \overline{T_{n_n}}(t) \tag{153}$$

- *Neumann boundary conditions:* the heat flux at the boundary is imposed:

$$-k_1 t_z \frac{\partial T}{\partial \xi}|_1 (t) = \overline{q_1}(t) \tag{154}$$

$$-k_{n_n} t_z \frac{\partial T}{\partial \xi}|_1 (t) = \overline{q_{n_n}}(t) \tag{155}$$

Neumann boundary conditions are satisfied with the weak formulation of the problem.

For the TPS sizing only Dirichlet boundary conditions are considered, formulated as follows [183]:

$$T|_1 (t) = \theta_1(t) = T_a (t) \tag{156}$$

$$T|_{n_n} (t) = \theta_{n_n}(t) = T_a (t) \tag{157}$$

The temperature at the boundary is imposed to be equal to the atmospheric one.

8.8.4. Initial Condition

Considering the stationary problem, the initial condition can be formulated as follows [183]:

$$\nabla(kt_z \nabla T) - q_r - q_c + q_s = 0 \tag{158}$$

If conductive and convective terms are neglected, Eq. (158) becomes:

$$q_r = q_s \tag{159}$$

The source term q_s is the sum of convective heating q_{gw} and the shock-layer gas radiation heat flux q_{rg} [178].

$$q_s = q_{in} = q_{gw} + q_{rg} \tag{160}$$

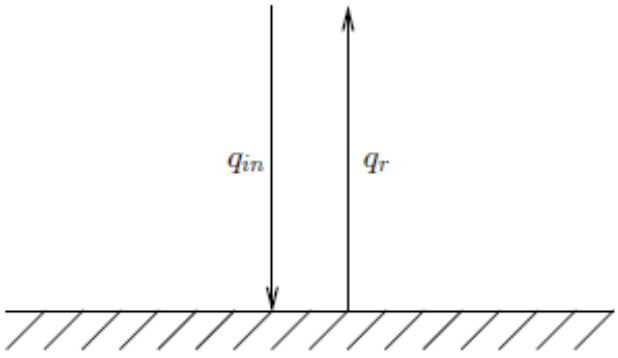

Figure 48: Schematic description of the thermal state of the surface in stationary condition, without convective and conductive heat fluxes. Image from the analysis of Anderson et al. in [178].

The source term of each element corresponds to the projection of q_{in} along the normal versor of the element itself:

$$q_{se} = q_{in}\,\hat{\boldsymbol{i}} \cdot \boldsymbol{n_e} \tag{161}$$

Considering the low fidelity aerothermodynamic model, the term q_{in} can be easly evaluated for the stagnation point through the correlation Eq. (112) and (113). Regarding the high fidelity aerothermodynamic model, the term q_{in} is computed by a CFD simulation.

8.8.5. Assembly

Considering Eq. (54), premultiply by the transpose of the virtual temperature $\delta\theta_e \neq 0$ and sum over all the elements [183]:

$$\sum_{e=1}^{n_e} \delta\theta_e^T \left(\boldsymbol{A_e}\dot{\theta}_e + \boldsymbol{K_e}\theta_e - f_e(t) \right) = 0 \tag{162}$$

Using the *incidence matrices* Ω_e is possible to shift from the local nodes numeration to the global one [183]:

$$\theta_e = \Omega_e \theta \qquad e = 1{:}\,n_e \tag{163}$$

$$\dot{\theta}_e = \Omega_e \dot{\theta} \qquad e = 1{:}\,n_e \tag{164}$$

$$\delta\theta_e = \Omega_e \delta\theta \qquad e = 1{:}\,n_e \tag{165}$$

Using Eq. (163), (164) and (165), Eq. (162) becomes:

$$\sum_{e=1}^{n_e} \delta\theta^T \Omega_e^T \left(\boldsymbol{A_e} \Omega_e \dot{\theta} + \boldsymbol{K_e} \Omega_e \theta - f_e(t) \right) = 0 \tag{166}$$

$$\delta\theta^T \left(\underbrace{\sum_{e=1}^{n_e} \Omega_e^T \boldsymbol{A_e} \Omega_e \dot{\theta}}_{A} + \underbrace{\sum_{e=1}^{n_e} \Omega_e^T \boldsymbol{K_e} \Omega_e \theta}_{K} + \underbrace{\sum_{e=1}^{n_e} \Omega_e^T f_e(t)}_{f(t)} \right) = 0 \tag{167}$$

Since $\delta\theta$ is arbitrary, Eq. (167) becomes [183]:

$$\boldsymbol{A}\dot{\theta} + \boldsymbol{K}\theta = f(t) \tag{168}$$

The discretization in space allows to transform a system of partial *differential equations* (PDE) into a system of *ordinary differential equations* (ODE).

8.8.6. Time Integration

The time derivative is approximated using finite differences [183]. Assuming a constant time step Δt:

$$t^{n+1} = t^n + \Delta t = t^1 + n\Delta t \tag{169}$$

Using forward differences, the time derivative can be approximate as follows:

$$\dot{\theta} = \frac{\theta^{n+1} - \theta^n}{\Delta t} \tag{170}$$

Using ϑ-method for the integration, Eq. (168) becomes:

$$[\boldsymbol{A}]\left\{\frac{\theta^{n+1} - \theta^n}{\Delta t}\right\} + [\boldsymbol{K}]\{\vartheta\theta^{n+1} + (1-\vartheta)\theta^n\} = \{\vartheta f^{n+1} + (1-\vartheta)f^n\} \tag{171}$$

Solving the Eq. (171) for θ^{n+1}:

$$\begin{aligned}\underbrace{([A] + \Delta t\, \vartheta[K])}_{[B]}\{\theta^{n+1}\} \\ = \underbrace{([A] - \Delta t(1-\vartheta)[K])}_{[C]}\{\theta^n\} + \underbrace{\{\Delta t\vartheta f^{n+1} + \Delta t(1-\vartheta)f^n\}}_{\{d\}}\end{aligned} \tag{172}$$

$$[\boldsymbol{B}]\{\theta^{n+1}\} = [\boldsymbol{C}]\{\theta^n\} + \{d\} \tag{173}$$

The value given to the parameter ϑ define a different integration method. The most common are summarized in Tab.6.

ϑ	Method type	Method name
0	Explicit	Foward Euler
1	Implicit	Backward Euler
1/2	Semi-implicit	Crank-Nicolson

Table 6: Integration methods for different values of the ϑ parameter.

8.9. Propulsion System

Depending on their missions, space vehicles present rocket propulsion systems. Rocket propulsion is used for primary propulsion, for example along the flight path for orbit insertion or orbit change manoeuvres, and secondary propulsion, for example attitude control, spin control, momentum wheel and gyro unloading, stage separation and the settling of liquid in tanks. A space vehicle usually has a series of different rocket propulsion systems, some often very small. For spacecraft attitude control about three perpendicular axes, each in two rotational directions, the system must allow the application of pure torque for six modes of angular freedom, thus requiring a minimum of 12 thrust chambers. More complex manned spacecrafts have 40 to 80 rocket units in all its stages [184].

8.9.1. Definitions

Here are presented some common definitions in rocket science:

- *Total Impulse I_T:* defined as the thrust force F (which can be a function of time) integrated over the burning time t [184]:

$$I_T = \int_0^t F\, dt \tag{174}$$

Total impulse is proportional to the total energy released by all the propellant in a propulsion system.

- *Specific impulse:* is the total impulse per unit of weight of propellant [184]:

$$I_s = \frac{I_T}{g_o \int \dot{m}\, dt} \tag{175}$$

Where $g_0 = 9.8066\ m/s^2$ is the standard acceleration of gravity at sea level and $\dot{m}$ is the total mass flow rate of propellant. Specific impulse is an important figure

of merit of the performance of a rocket propulsion system. A higher number means better performance.

- *Thrust:* is the force produced by a rocket propulsion system acting upon a vehicle. Thrust is given as:

$$F = \dot{m}v_2 + (p_2 - p_3)A_2 \tag{176}$$

The first term is the momentum thrust represented by the product of the propellant mass flow rate and its exhaust velocity relative to the vehicle v_2. The second term represent the pressure thrust consisting of the product of the cross-sectional area at the nozzle exit A_2 and the difference between the exhaust gas pressure at the exit and the ambient fluid pressure.

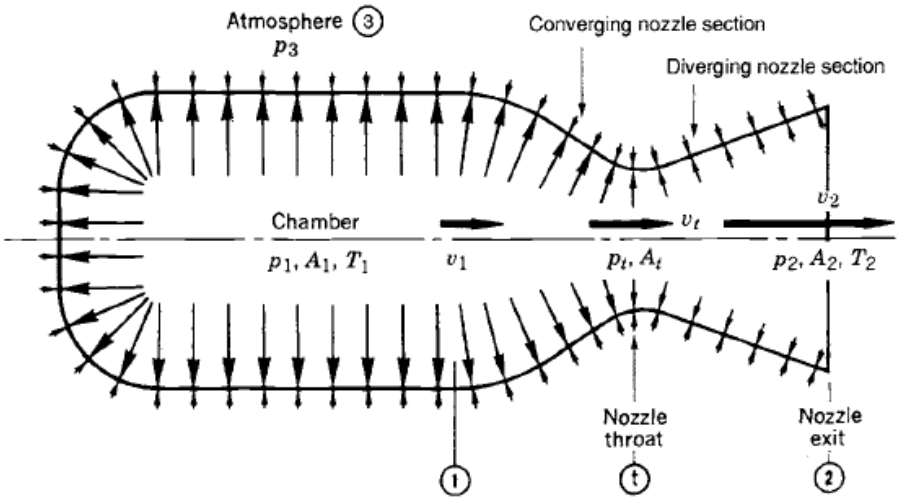

Figure 49: pressure, velocity, and cross-sectional area at different stations over the chamber axis. Image taken from Sutton et al. in [184].

- *Effective exhaust velocity* is the average equivalent velocity at which propellant is ejected from the vehicle. It is defined as [184]:

$$c = I_s g_0 = F/\dot{m} \tag{177}$$

The effective exhaust velocity as defined in Eq. (177) applies to all rockets that thermodynamically expand hot gases in a nozzle and, indeed, to all mass expulsion systems. Considering a constant propellant mass flow, Eq. (177) becomes:

$$c = v_2 + \frac{(p_2 - p_3)A_2}{\dot{m}} \tag{178}$$

The second term of Eq. (177) on the right-hand side is usually small in relation to v_2. So, the effective exhaust velocity is usually close in value to the actual exhaust velocity. When $c = v_2$ the Eq. (176) can be rewritten as [184]:

$$F = \dot{m}c \tag{179}$$

- *Burned propellant mass:* defined as the propellant mass flow integrated over burning time:

$$m_P = \int_0^t \dot{m}\, dt \tag{180}$$

The definitions presented in this paragraph will be used to estimate the mass of propellant burned during the re-entry manoeuvre of the considered space vehicle.

8.9.2. Impulsive Orbital Manoeuvres

A simplified orbital manoeuvre model is assumed which is applied with good approximation to chemical propulsion (considered in the return vehicle object of the thesis). The model assumes the application of an infinite thrust in infinitesimal times. As a result, only a variation of the spacecraft velocity V is produced without a perturbation of the orbital radius during the burn. Considering the re-entry phase, aerodynamic and gravitational forces must be considered. The Newton's second law, reported in Eq. (86), can be written as follows:

$$m\frac{d\boldsymbol{V}}{dt} = \boldsymbol{T} + m\boldsymbol{g} + \boldsymbol{D} + \boldsymbol{L} \tag{181}$$

Considering the forces as shown in Fig. 50, Eq. (100) becomes:

$$\frac{dV}{dt} = \frac{T}{m}cos\alpha - \frac{D}{m} - gsin\varphi = \frac{T}{m} - \frac{T}{m}(1 - cos\alpha) - \frac{D}{m} - gsin\varphi \tag{182}$$

Where φ is the flight path angle and α is the thrust angle. Integrating Eq. (182) over the burning time:

$$\Delta V = \underbrace{\int_0^t \frac{T}{m}dt}_{\Delta V\ ideal} - \underbrace{\int_0^t \frac{T}{m}(1 - cos\alpha)dt}_{misalignment\ losses} - \underbrace{\int_0^t \frac{D}{m}dt}_{drag\ losses} - \underbrace{\int_0^t gsin\varphi dt}_{gravitational\ losses} \tag{183}$$

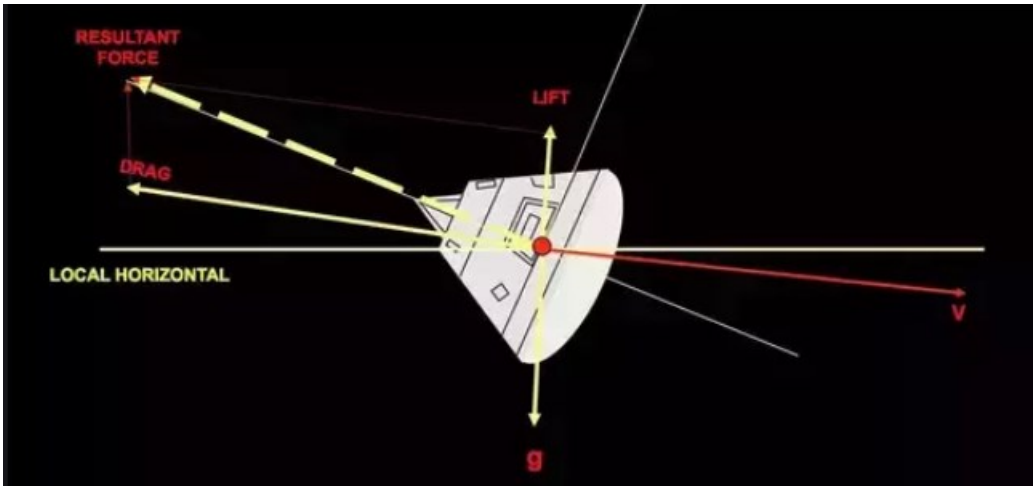

Figure 50: Re-entry vehicle aerodynamic forces.

8.9.3. Rocket Equation

The rocket equation, also commonly referred as the *Tsiolkovsky equation,* describes the motion of spacecrafts with variable mass:

$$\Delta V = c \ \ln \frac{m_i}{m_f} \tag{184}$$

Where m_i is the initial rocket mass and m_f is the final rocket mass (after the burning time). The rocket equation is derived considering only the motion subject to the thrust, without aerodinamic or gravitational force.

However, Eq. (184) can be effectively applied to the analysis of orbital manoeuvres, if performed with chemical thrusters. In fact, it allows both to determine which orbit can be reached with a given quantity of propellant, and to determine, in its inverse form (shown below), how much propellant m_p is needed to reach a given orbit (that is, to acquire a given variation in the velocity value ΔV):

$$m_p = m_i \left[1 - \exp\left(-\frac{\Delta V}{g_0 I_s}\right)\right] \tag{185}$$

In the application to orbital manoeuvres, it is assumed in particular that the manoeuvre takes place in an impulsive manner: both the variation in the speed value and the ignition phase of the engine are treated as if they were instantaneous. This hypothesis is quite accurate for short-term ignitions, such as those used in course correction or orbital insertion manoeuvres. As the duration of the rocket ignition increases, however, the result loses accuracy due to the effects of gravity on the vehicle over the duration of the manoeuvre.

8.9.4. Thrusters

The space vehicle, considered for the test problem, has both primary and secondary propulsion systems. Chemical rocket engines for re-entry and attitude control are considered. The engine uses hypergolic propellant consisting of monomethyl hydrazine (MMH) as a propellant and dinitrogen tetra oxide (NTO) as an oxidant. They have the ability to be turned on multiple times and can precisely control thrust. For primary propulsion, the engine system consists of eight primary thrusters. For secondary propulsion, a cluster of eighteen secondary thruster are considered. In both cases propulsion system can vary thrust level from 20% to 100%.

In Tab. 7 are summarized the technical specifications of the thrusters considered:

Specifications	Primary thruster	Secondary thruster
Propellant	NTO/MMH	NTO/MMH
Max Thrust (Vacuum)	73 kN	400 N
Specific Impulse	253 s	300 s
Burning time	25 s	25 s
Number	8	18

Tabel 7: Thruster technical specifications.

8.9.5. Equation for Planetary Controlled Flight

The general equations for planetary flight must be modified to consider the thrust vector. Thrust vector T can be decoposed in a component in the direction of the velocity T_V and a component normal to the velocity T_N:

$$T = \{ T_V \quad T_N \} \tag{186}$$

The Eq. (90) becomes:

$$\frac{dV}{dt} = -\frac{1}{m} D - g \sin\gamma + T_V$$

$$V \frac{d\gamma}{dt} = \frac{1}{m} L \cos\mu_a - g \cos\gamma + \frac{V^2}{r} \cos\gamma + T_N \tag{187}$$

The thrust component T_V is given by the engines used in primary propulsion, while the thrust component T_N is given by the engines used in secondary propulsion. Since the propulsion system is powered by chemical propellant, impulsive manoeuvres are considered. The equations are integrated with MATLAB solver ODE45 from initial time t_0 to final time t_f with a temporal step Δt. Since the chosen time step is greater than the typical burning time of the thrusters, the components of the thrust vector will be considered different from zero only in a time step, simulating an impulsive manoeuvre.

8.10. Multifidelity Re-entry Optimization Algorithm

The optimization problem is considered by means of the algorithm presented in chapter 6. However, some important changes have been made to adapt the multi-fidelity expected improvement algorithm to solve the atmospheric re-entry problem. The following paragraphs present the objective black box function the alternative formulation introduced for the multi-fidelity expected improvement, the algorithm implemented including meta code and flow charts describing how the algorithm operates.

The optimization problem is defined in the following way:

$$argmin_x f(x)$$

Given the following constraints:

$$Fixed\ vehicle\ geometry$$

$$Fixed\ TPS\ material$$

$$Fixed\ propellants$$

$$Fixed\ initial\ reentry\ conditions$$

$$Thrust\ \neq 0 \qquad if\ \ 90\ km < altitude < 120\ km$$

$$TPS\ temperature < material\ melting\ temperature$$

$$propellant\ mass\ burned < propellant\ mass\ stocked$$

$$Residuals\ of\ thermofluid\ CFD\ anaysis < 10^{-6}$$

Where $f(x)$ is the objective black box function. The capsule geometry, TPS material, propellants and initial re-entry conditions are fixed. The thruster can fire only in a specific range of altitude in order to avoid failure due to thermofluidic continuous regime, stressful for the propulsive system. The TPS temperature must be under the material melting temperature to guarantee the survival of the payload. The thrust needed to optimize the trajectory must be allowable by the stock propellant mass so the propellant mass burned must be less than tank mass. Finally, an important constraint is the residual of CFD analysis in order to guarantee the convergence of computation.

8.10.1. Objective Function Definition

In the setup of the atmospheric re-entry problem, the objective function (the black-box function) to be minimized is defined as follows:

$$f(x) = v1\frac{m_{TPS}}{m_{TPS0}} + v2\frac{T_{TPS}}{T_{TPS0}} + v3\frac{m_p}{m_{p0}} \tag{188}$$

$$v1 = 0.4 \quad v2 = 0.4 \quad v3 = 0.2$$

$$m_{TPS0} = 2000\ kg \quad T_{TPS0} = 1500K \quad m_{p0} = 150\ kg$$

$$x = \{F_n, F_t, s_{TPS}\}$$

Where m_{TPS} is the TPS mass computed, m_{TPS0} is the TPS mass of reference, T_{TPS} is the TPS temperature computed, T_{TPS0} is the TPS temperature of reference, m_p is the propellant mass burned, m_{p0} is the propellant mass of reference, F_n is the thrust normal to velocity vector, F_t is the tangential thrust to velocity vector and s_{TPS} is the TPS thickness.

The reference variables are chosen on the basis of experimental data from atmospheric re-entry vehicles. The multiplicative coefficients $v1, v2$ and $v3$ selection is given by knowledge-based considerations. In fact, it is believed that the temperature and mass of the TPS are more important when compared to the mass of propellant consumed. This derives from considerations related to the survival of the vehicle in the re-entry phase and the reduction of costs in the launch phase.

8.10.2. Re-Entry Multi-Fidelity Expected Improvement

Having regard to the nomenclature presented in Chapter 5, the Re-Entry Multi-Fidelity Expected Improvement formulation is the following:

$$ReEntryMFEI(x,m) = MFEI(x,m)\alpha_4(x,m) \tag{189}$$

$$m = 1, \dots, M \qquad levels\ of\ fidelity$$

Where $MFEI(x,m)$ is the multi-fidelity expected improvement presented in Eq. (45). The $\alpha_4(x,m)$ is a term defined as follows:

$$\alpha_4(x,m) = \begin{cases} 1 \quad if\ m = 1, \dots, M-1 \\ \beta\dfrac{h}{h_0} \quad if\ m = M\ and\ 35\ km \leq h \leq 65km\ \ h_0 = 50km \\ 1 \quad if\ m = M\ and\ h < 35km\ or\ h > 65km \end{cases}$$

Where h is the altitude of the vehicle, h_0 is the reference altitude and $\beta = 200$ is the multiplicative parameter. The reference height h0 is chosen on the basis of empirical considerations. In fact, h0 represents the typical altitude at which thermal phenomena are most important.

Consequently, the algorithm is incentivized to define the thermal loads by means of the higher-fidelity model when the altitude is considered critical for the vehicle's survival.

The value assigned to β is defined on the basis of tests carried out on the algorithm. The outcomes suggest that $\beta = 200$ is a satisfactory value.

8.10.3. Multifidelity Re-entry Algorithm

In this paragraph is described the algorithm implemented in Matlab. Specifically, the meta code and the flow chart of the algorithm is presented. The MFEI Re-entry code is implemented in Matlab in the form of functions packages. The algorithm setup is presented in a Matlab struct opt and listed in Tab. 2 (For more details please refer to Chapter 6.).

In the specific optimization problem considered, the aerothermodynamic models are hierarchically divided into two levels of fidelity, as described in the previous paragraphs. In the following is illustrated the meta-code presented in Tab. 8.

The initial candidates are sampled by a Latin hypercube sampling method and stored in the $samples$ variable. The values of the objective function are computed from the initial candidates and stored in the $values$ variable.

After the initialization process, the multifidelity Bayesian optimization loop starts.

Considering a generic iteration, defined the thrust vector and the thickness of TPS, the trajectory and the propellant mass burned are computed. The re-entry trajectory is computed in Matlab with ODE45. Consequently, the outcome is defined as a matrix whose dimension depends on the integration time step used.

The next computational step is evaluating the aerothermodynamic conditions in all the trajectory points and select which aerothermodynamic fidelity model to query. First the Gaussian process in enriched by the information about the objective function obtained from the previous computational step. Then the GP mean and variance is computed.

The next step is evaluating the MFEI acquisition function. In the specific problem considered, the MFEI is evaluated in each point of the trajectory (in the generic Bayesian optimization loop iteration) in order to determine the next aerothermodynamic fidelity model to query at the following point of the trajectory.

To account this, in all the points is evaluated and maximized the MFEI considering the formulation in Eq. (189).

Once the aerothermodynamic variables are computed for all the trajectory points, the structural and thermal analysis of TPS is performed and TPS temperature and mass are calculated.

Consequently, the value of the objective function for the iteration considered is evaluated and stored in $values$ matrix. Considering the values of MFEI calculated in each trajectory point, the absolute maximum identifies the next sample to query in the following iteration. The new sample is then stored in $samples$ matrix.

MFEI Re-Entry Algorithm

1: Draw initial candidate grid from a random Latin hypercube

$$hyper\ grid = Latinhypercube(opt.dims, opt.gridsize, opt.mins, opt.maxes)$$

$$\rightarrow hyper\ grid \rightarrow x_1^{(m)} \rightarrow\ samples_1^{(m)} = \left[samples^{(m)}; x_1^{(m)}\right] \rightarrow samples_1 \quad m = 1, 2$$

2: Get values from the first samples

$$\rightarrow f^{(m)}\left(x_1^{(m)}\right) \rightarrow values^{(m)} = \left[values^{(m)}; f^{(m)}\left(x_1^{(m)}\right)\right] \rightarrow values_1 \qquad m = 1, 2$$

3: Main Bayesian Optimization Multifidelity Loop

$\boldsymbol{for}\ i = 1{:}\ opt.\max\ \_iters$

4: Compute re-entry trajectory

$$trajectory(i) = \{V(i), h(i), r(i), \theta(i), \gamma(i)\}$$

5: Compute Propellant mass burned mp

6: Select the Aerothermodynamic fidelity model to query:

$\boldsymbol{for}\ j = 1{:}\ length\left(trajectory(i)\right)$

$$\left[\mu^{(m)}, \sigma^{2(m)}\right] = GP(hypergrid, samples_i, values_i, \ldots \\ opt.infmethod, opt.meanfunc, opt.covfunc), \quad m = 1, 2$$

$$\alpha_1(m, j) = corr\left[f_{aeroth}^{(m)}(trajectory(j)), f_{aeroth}^{(M)}(trajectory(j))\right] \quad m = 1, 2$$

$$\alpha_2(m, j) = 1 - \frac{\sigma_\epsilon}{\sqrt{\sigma^{2(m)}(trajectory(j)) + \sigma_\epsilon^2}} \qquad m = 1, 2$$

$$\alpha_3(m) = \frac{\lambda_M}{\lambda_m} \qquad m = 1, 2$$

$$\alpha_4(trajectory(j), m) = \begin{cases} 1 \quad if\ m = 1 \\ \beta \dfrac{h}{h_0} \quad if\ m = 2\ and\ 35\ km \leq h \leq 65km\ \ h_0 = 50km \\ 1 \quad if\ m = 2\ and\ h < 35km\ or\ h > 65km \end{cases}$$

$$compute\ \mathbb{E}[\max\left(f_{aeroth}^{(2)}(trajectory(j)) - f_{aeroth}^{(2)}\left(argmin\ f_{aeroth}^{m}(trajectory(j))\right)\right]$$

$$compute\ MFEI(trajectory(j), m)$$

$$\max(MFEI(trajectory(j), m)) \rightarrow\ \ m_{fidelity} level\ at\ j + 1\ step$$

end

7: Compute TPS temperature T_{TPS} and TPS mass m_{TPS}

8: Choose next candidate

$$\max(MFEI(trajectory(i), m)) \rightarrow\ \ x_{i+1} \qquad for\ m = 1{,}2$$

9: Upload Samples and Values

$$samples_{i+1} = [samples_i; x_{i+1}]$$
$$values_{i+1} = [values_i; f^{(m)}(x_{i+1})]$$

end

10: Get minvalue and minsample

$$minsample = \min(samples)$$
$$minvalue = \min(values)$$

Table 8: MFEI algorithm specialized for re-entry vehicle optimization problem.

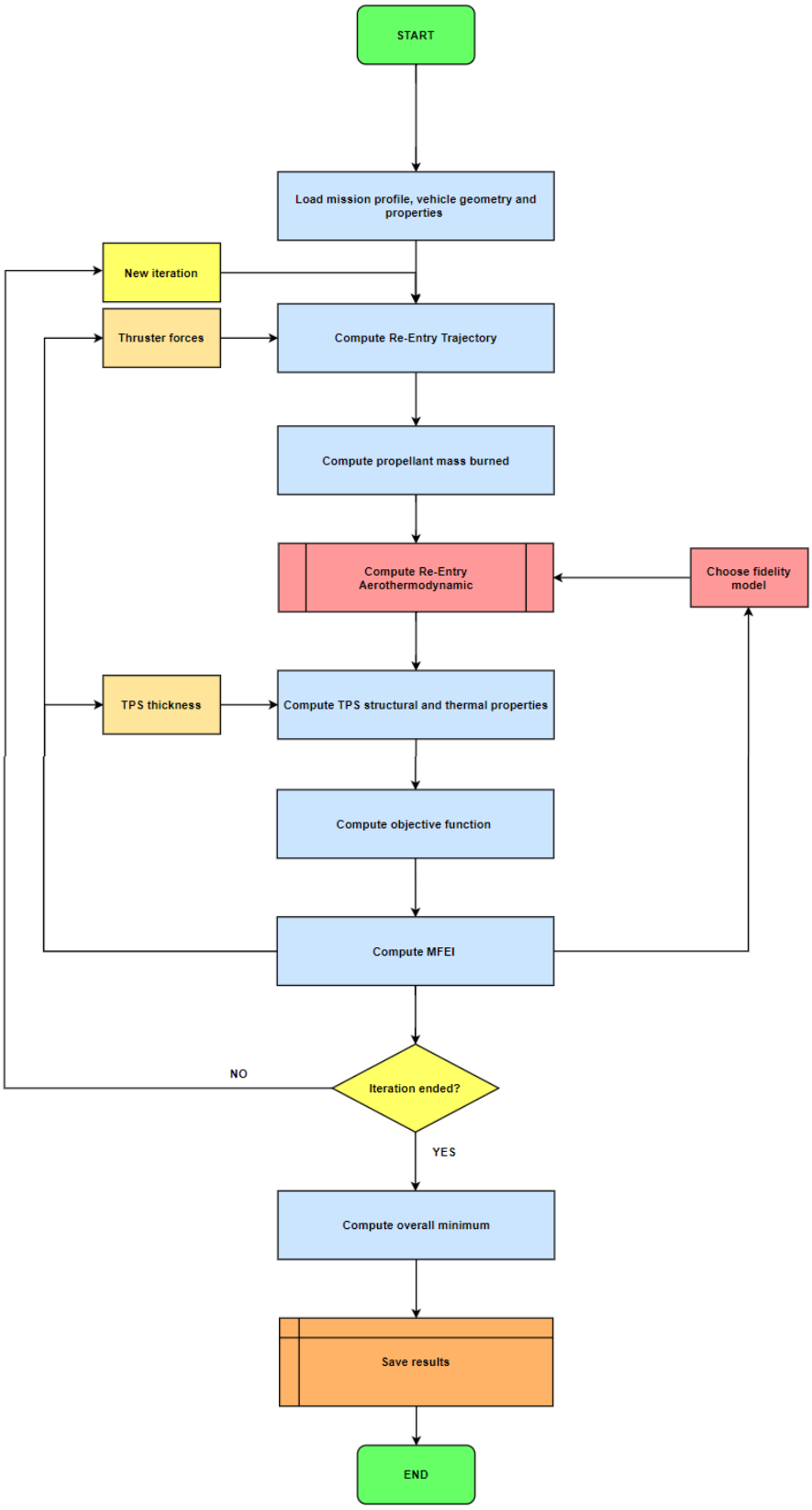

Figure 51: Re-entry optimization problem flow-chart

8.11. Multifidelity Optimization Results

In this paragraph are presented the results of the MFEI algorithm formulated for the re-entry optimization problem. The results of the optimization problem are summed up in the following:

$$argmin_x f(x) = 0.8332$$

$$at \quad x = \{3.4100 \cdot 10^4, 1.8041 \cdot 10^3, 0.0105\} \quad iteration = 97$$

$$Highfidelity\ calls = 153$$

$$lowfidelity\ calls = 1324$$

$$computation\ time = 561\ hours$$

Specifically, the physical outcomes are the following:

$$F_T = 3.4100 \cdot 10^4 N$$

$$F_N = 1.8041 \cdot 10^3 N$$

$$s_{TPS} = 0.0105\ m$$

$$argmax(T_{TPS}) = 1230.8\ K$$

$$m_{TPS} = 1465.8\ kg$$

$$m_p = 86.8429\ kg$$

In Fig. 52 is reported the value of the objective function at each iteration. It can be noted that at first the trend of the objective function turns out to be variable. This is because the algorithm tends to favour the exploration phase. However, as the iterations increase, the variability of the objective function is significantly reduced. This can be interpreted as a prevailing of the exploitation phase towards the minimum.

In Fig. 53 are presented the global minimum at each iteration. Although there are some sections where the global minimum does not decrease for some iterations, overall, it decrements with a significant rate. Moreover, in the last iterations the decrease of the global minimum seems to be faster, synonymous with the fact that

with a higher number of iterations the algorithm could have further decreased the value found.

In Fig. 54 are reported the samples over the iterations. With the red point is indicated the sample corresponding to the global minimum. There is a great variability of the forces applied by the thrusters and of the thickness of the TPS. This is because the application of suitable forces allows to reduce the temperature reached by the materials. However, there are few samples with very high forces. This is because as the tangential force increases above the speed, the tangential load factor tends to increase considerably, forcing an increase in the structural mass. The thickness of the TPS, on the other hand, tends to vary within a limited and globally small range of values. This is understandable from the fact that the temperatures reached by the TPS are contained due to the applied forces, thus leading to better tolerance to thermal loads and reducing the structural mass of the TPS itself.

In Fig. 55 the trend of the mass of the TPS is presented with the number of iterations. As mentioned before, the variability of the contained structural thickness and the reduced tangential load factor led to a globally decreasing trend in the number of iterations.

In Fig. 56 the variation of the maximum structural temperature of the TPS is presented with the performed iterations. The trend is substantially increasing. This is because in the first iterations the level of the applied forces is excessive. On one hand, high forces have made it possible to drastically reduce the structural temperature, on the other hand, however, the structural mass increases considerably to be able to withstand the high tangential load factor. Consequently, going towards the optimum the temperature of the TPS increases in the face of lower forces and lower structural masses accordingly.

In Fig. 57 is shown the trend of the mass of propellant burned as the iterations proceed. Due to the discussion made above on the applied forces, the mass of propellant consumed tends to decrease.

The results computed by the trajectory model are shown in Fig. 58. Specifically, the time of the beginning of the graphs refers to the altitude of 85 km, where the continuum hypothesis is valid and therefore the models implemented.

The pressure, the thermal flow and the wall temperature on the gas side are presented respectively in Fig. 59-60-61. Specifically, the blue curve is calculated considering the low-fidelity aerothermodynamic model. The values indicated by the green points, on the other hand, are obtained by solving a CFD simulation. In the iteration corresponding to the optimum, the high-fidelity aerothermodynamic model was invoked three times. Following the formulation presented for the α_4 parameter in Eq. (XX), the high-fidelity model is required in the altitude interval where the thermal flows are more stressful for the structure. In fact, the maximum heat flux is computed by means of a CFD simulation. This choice is due to the design importance given by the heat flux for the structural dimensioning of the TPS.

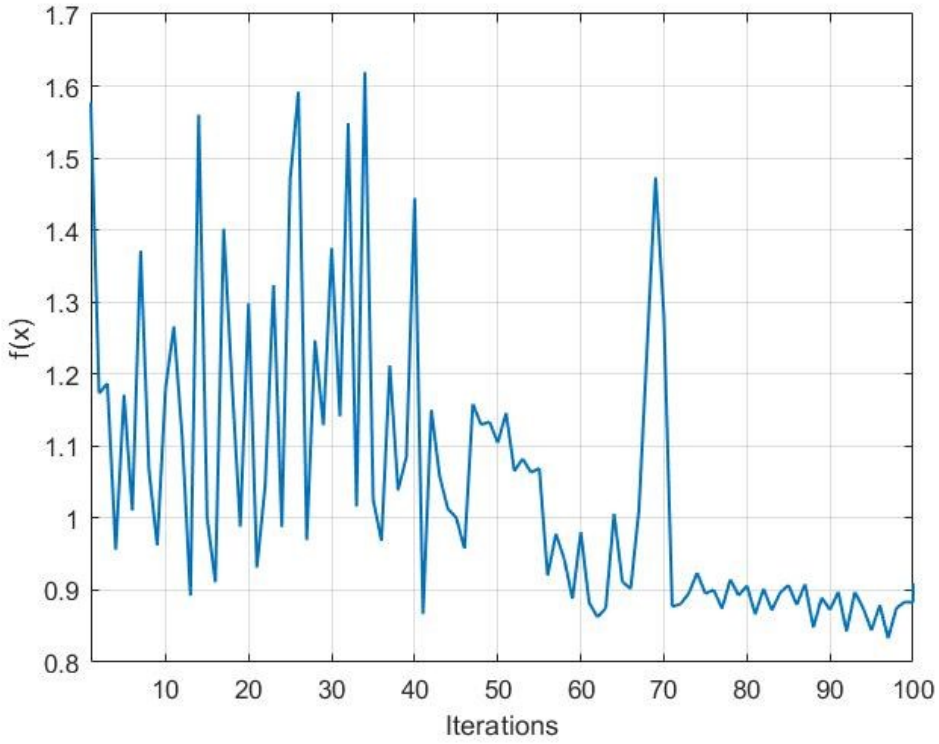

Figure 52: $f(x)$-iterations.

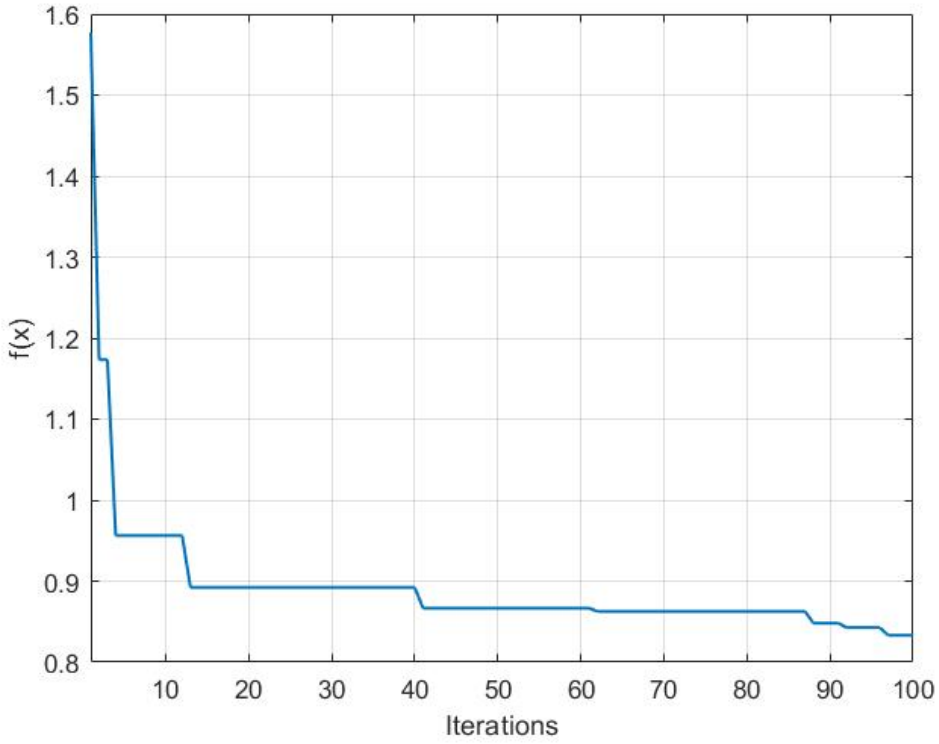

Figure 53: $f(x)$ overall minimum-iterations

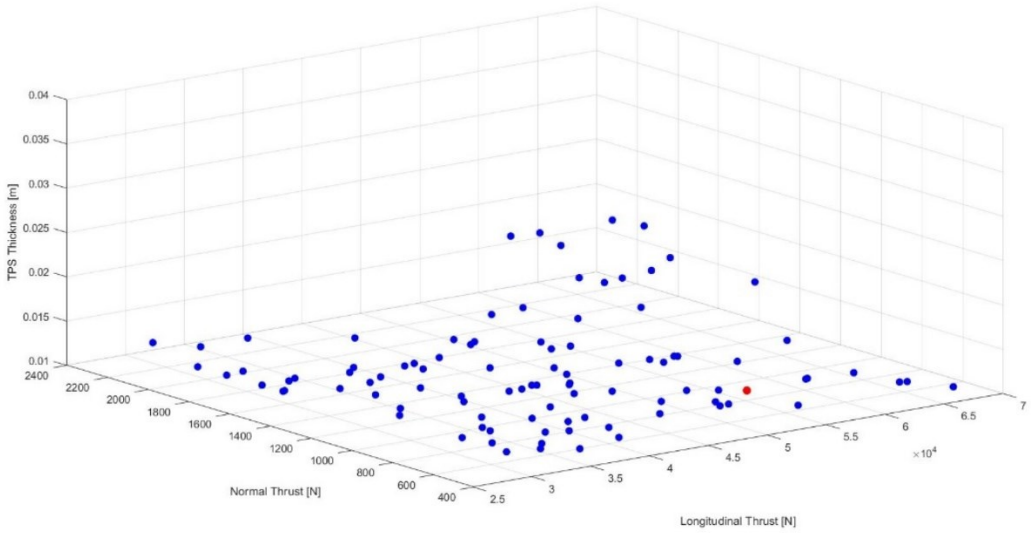

Figure 54: sample points selected during iterations. In red is reported the sample corresponding to the overall minimum of the objective function.

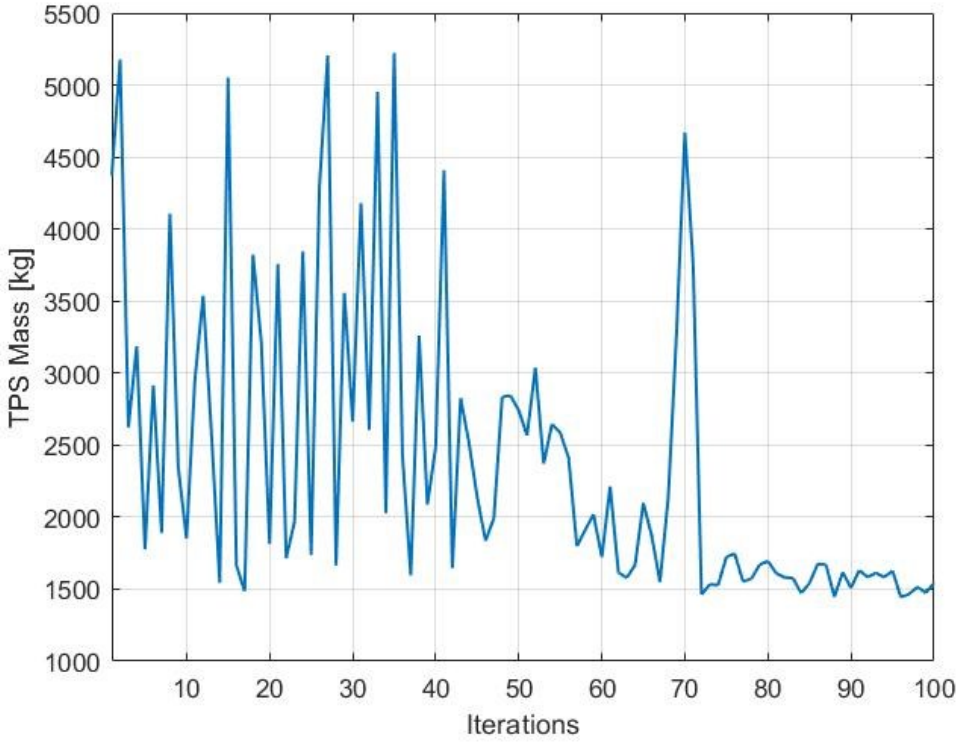

Figure 55: TPS mass-iterations

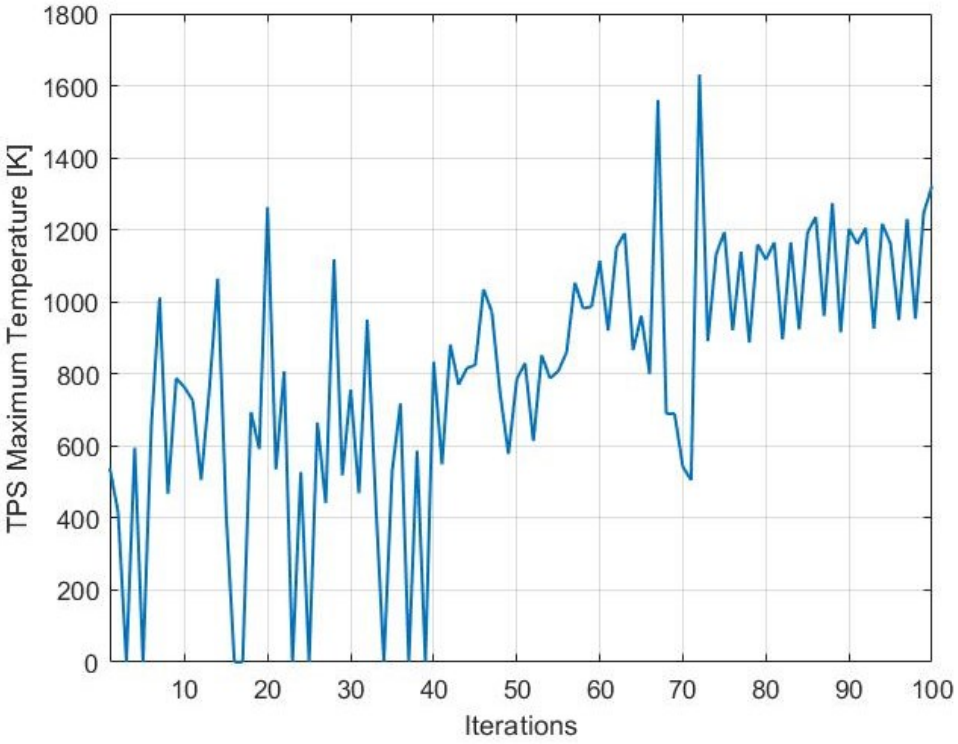

Figure 56: TPS maximum temperature-iterations

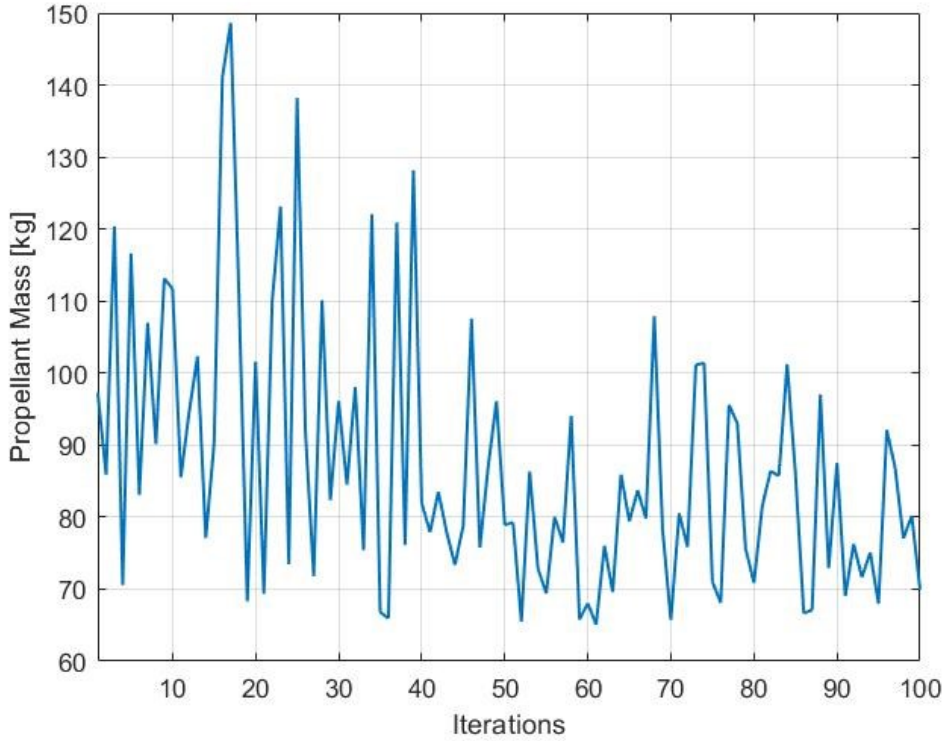

Figure 57: Propellant mass-iterations

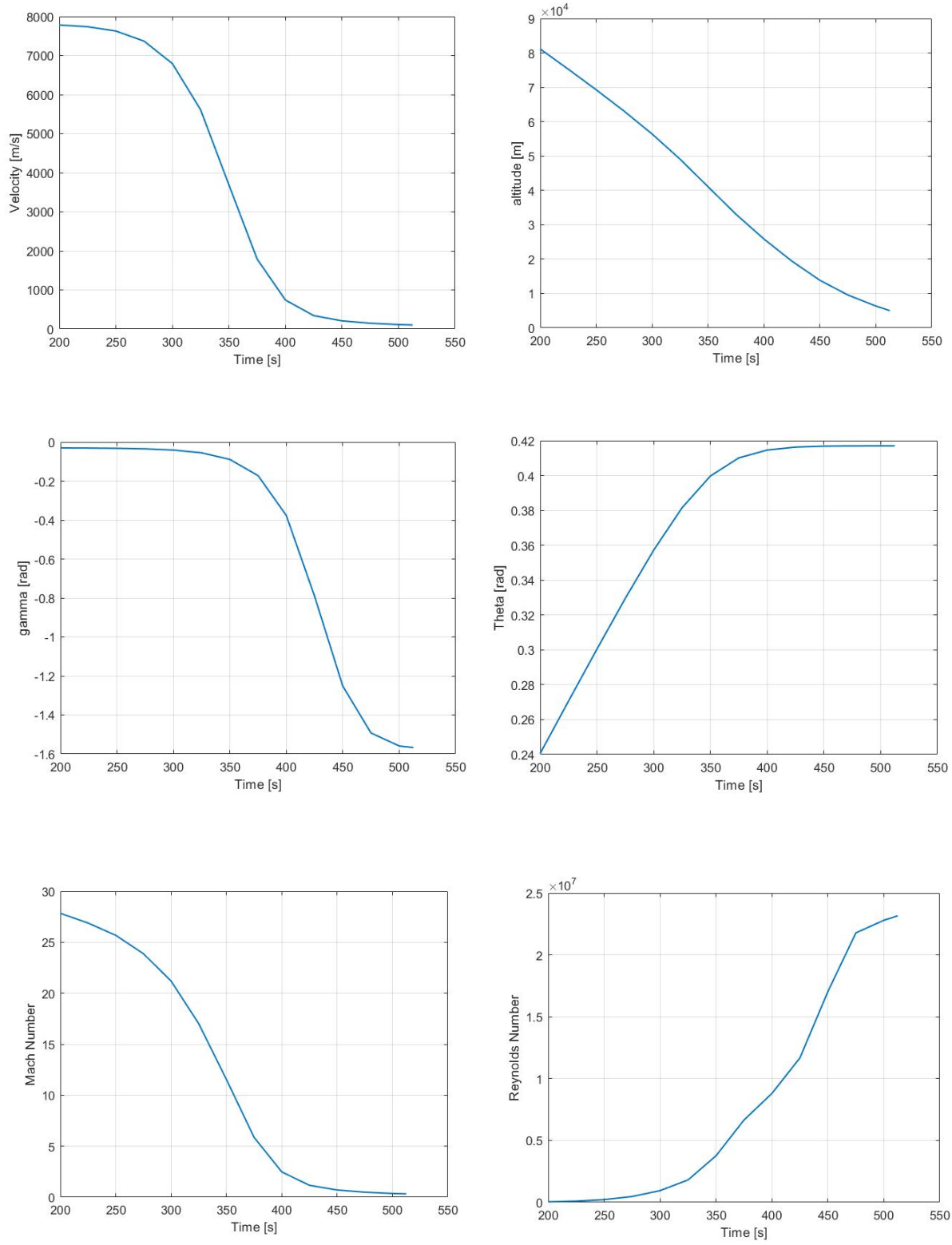

Figure 58: velocity, altitude, gamma, theta, Reynolds, and Mach number over re-entry time.

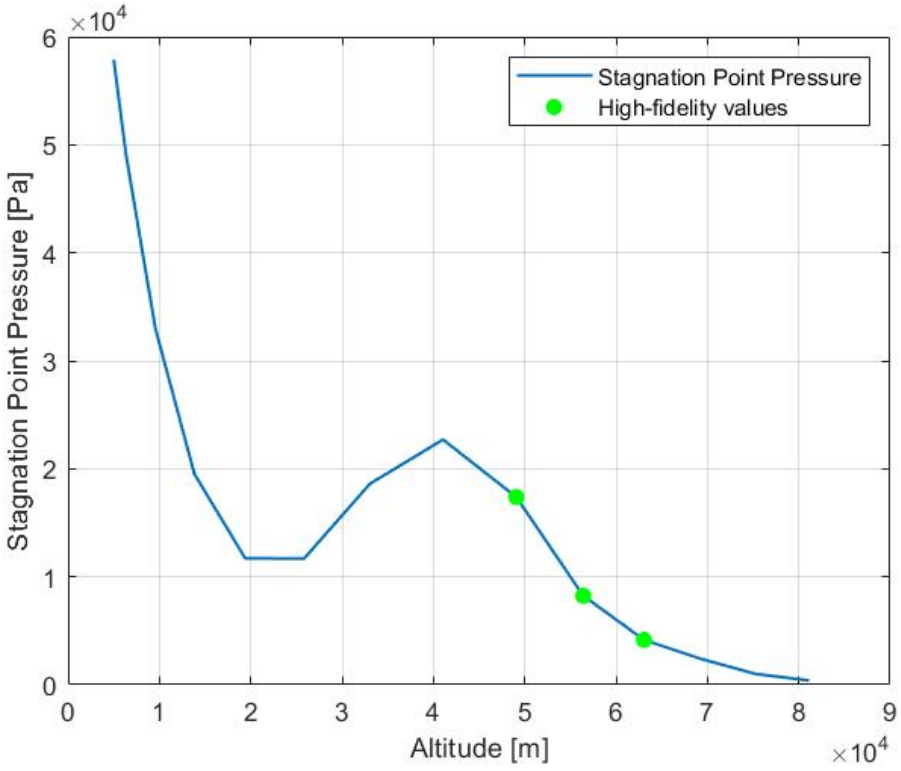

Figure 59: Stagnation point pressure-altitude. The green points are the high-fidelity values computed during the optimum iteration.

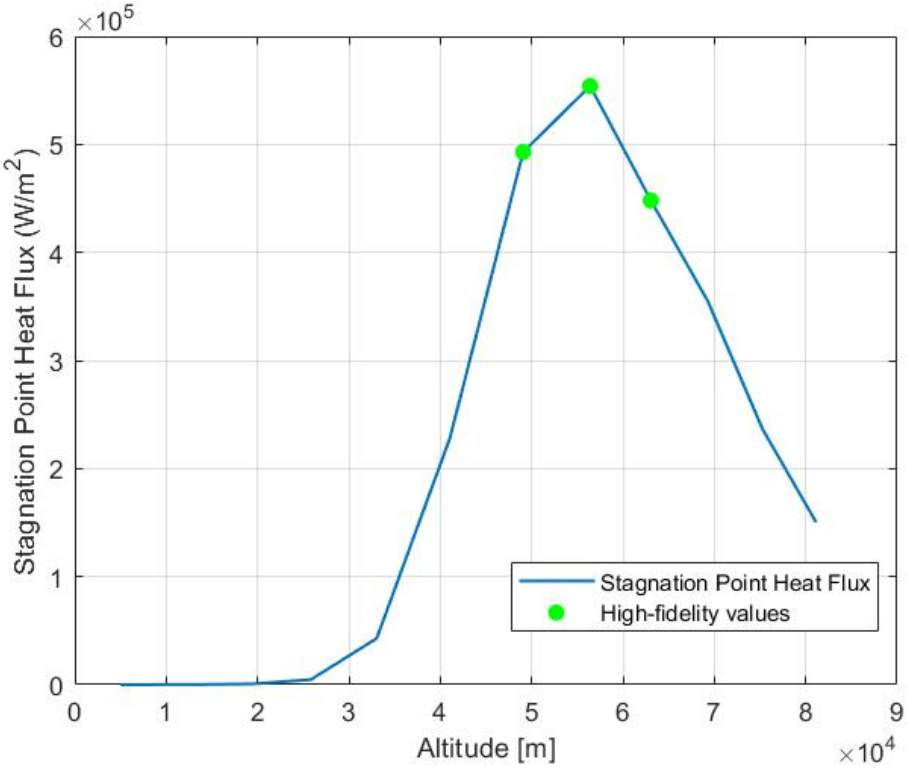

Figure 60: Stagnation point heat flux-altitude. The green points are the high-fidelity values computed during the optimum iteration.

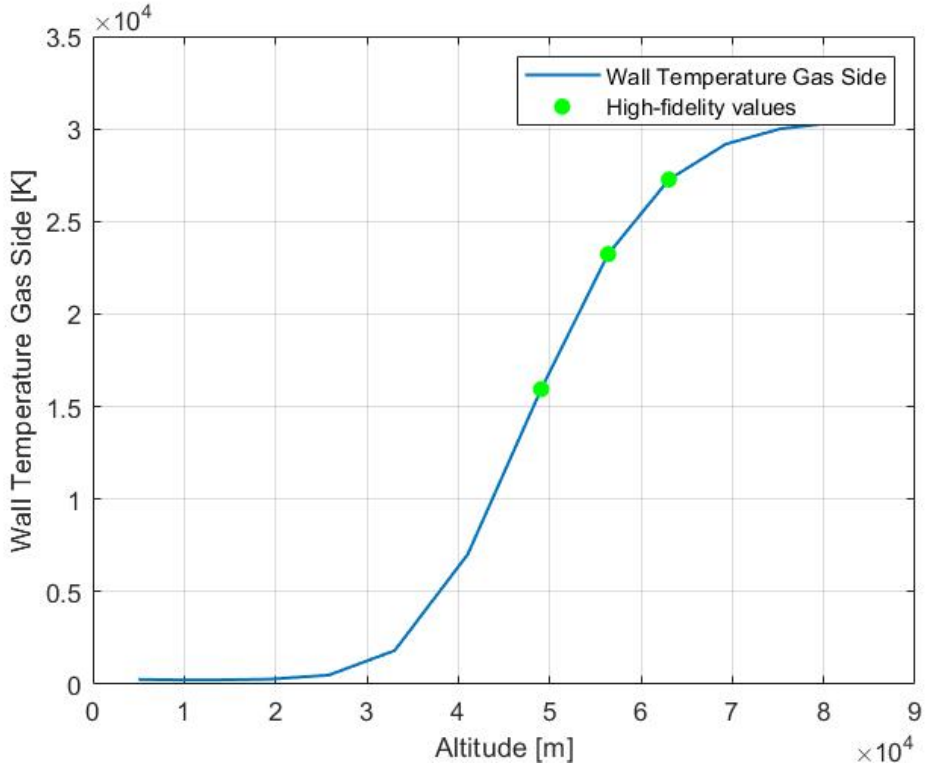

Figure 61: Stagnation point heat flux-altitude. The green points are the high-fidelity values computed during the optimum iteration.

8.12. Comparison Between Single and Multifidelity Algorithm

In this paragraph is presented a comparison between the single fidelity expected improvement algorithm, presented in chapter 6, and the original multifidelity algorithm described in this chapter. The aim is to determine if, from the only tests performed, the multifidelity algorithm is more performing than the single fidelity algorithm considering the re-entry optimization problem. The single fidelity expected improvement algorithm is implemented considering the low-fidelity aerothermodynamic model.

The results are presented in Tab.9:

	SFEI algorithm	**MFEI algorithm**
Iterations	100	100
Overall minimum	1.0816	0.8332
Tangential thrust	40,902 kN	34.1 kN
Normal thrust	2.0917 kN	1.804 kN
TPS thickness	0.0250 m	0.0105 m
TPS mass	2265.9 kg	1465.8 kg
TPS max temperature	2018.3 K	1230.8 K
Propellant mass	114.114 kg	86.8429 kg
Computation time	18 minutes	561 hours

Table 9: SFEI and MFEI algorithm results comparison.

Overall, the MFEI algorithm appears to perform better than the SFEI. This can be derived from the fact that the global minimum of the objective function, and therefore the consequent design variables, are smaller in the case of the multifidelity algorithm. Since the objective function has been defined so that the value 1 corresponds to average data of atmospheric re-entry capsules, two considerations can be made: the low-fidelity model would seem to be the most conservative and that the information coming from the CFD simulations allow an enrichment of information and therefore a better re-entry mission.

In Fig. 62 a comparison is presented between the value of the objective function assumed for each sample in the case of MFEI and SFEI algorithms. In particular, the performance of the SFEI algorithm shows, in the first calculations performed, an exploration phase characterized by a strong variability of the objective function. However, with increasing iterations the exploitation towards the minimum shows a reduced fluctuation of the objective.

This is proven by the trends of the global minima with the number of iterations. The SFEI algorithm shows in Fig. 63 a decreasing trend that is not excessively accentuated, if compared to the trend of the MFEI algorithm. However, as in the multifidelity case, it can be assumed that by increasing the number of iterations, the minimum can further decrease.

The sampling carried out in the iterations are shown in Fig. 64. The SFEI algorithm shows a sampling characterized by a greater variability of the thickness of the TPS. Furthermore, the values themselves are also significantly higher. This can be interpreted by the fact that the low fidelity aerothermodynamic model has in output higher thermal loads than the high-fidelity model. To cope with the higher thermal loads, the thickness of the TPS must necessarily increase. In Fig. 65-66 what has just been said can be noted. Although both the mass of the TPS and the structural temperature have an average decreasing trend, the values reached are significantly higher than the results obtained with the multifidelity algorithm. In Fig. 67 the propellant mass computed performing the SFEI algorithm shows the same trend.

In Fig.68 are reported the outcomes of the trajectory model to characterize the optimum re-entry profile.

The comparisons between the single fidelity and multifidelity trends of pressure, heat flux and gas side wall temperature at the point of stagnation are shown in Fig. 69-70-71 respectively. Focusing on the heat flux, the single fidelity result shows a peak of $6.333 \cdot 10^5\ W/m^2$ at the altitude of $5.771 \cdot 10^4\ m$ while the multifidelity output shows a maximum heat flux of $5.541 \cdot 10^5\ W/m^2$ at the altitude of $5.901 \cdot 10^4$ m. Considering the wall temperature, on average the values obtained with the SFEI algorithm are greater than the one obtained with MFEI strategy, confirming what has been said earlier about the single fidelity optimization result. The same result can be observed in the stagnation point pressure trend.

In conclusion, the single fidelity algorithm leverages effectively the low-fidelity aerothermodynamic models, allowing design exploration to be carried out in a short time given the limited calculation expense. However, the global minimum value of the objective function is above 1 (mean reference value for atmospheric re-entry vehicles). This suggests that aerothermodynamic physical models are conservative, because tend to overestimate thermodynamic variables such as heat flux or wall temperature. Consequently, the optimum design obtained with the SFEI algorithm overestimates the structural mass, the temperature of the TPS and the mass of

propellant burned. By contrast, the multifidelity algorithm is computationally less efficient given the much longer computation times. However, the information coming from high fidelity numerical simulations allows to enrich the knowledge of the objective function, leading to a better optimal design.

Some considerations can be made on the implementation of a SFEI algorithm with high fidelity model. In this work, suitable tests were not done due to a high time required and computational cost. However, some considerations can be made. The model that allows the computation of the trajectory has been set in order to have 23 points in the range of heights of interest for the problem. Consequently, an iteration of the SFEI or MFEI algorithm involves evaluating the aerothermodynamic model 23 times. Based on this consideration and the computational performance of the MFEI algorithm known, the time required to perform 100 iterations by implementing the multifidelity algorithm corresponds to approximately the time required to perform 6 iterations by implementing the single high-fidelity model.

Therefore, two important implications can be outlined: the multifidelity algorithm is computationally more efficient than the single high-fidelity algorithm and allows to improve the solution obtained with the single low-fidelity algorithm; While it can be said with some confidence that 6 single high-fidelity iterations are not enough to achieve convergence, nothing can be said about the level of accuracy achievable by performing 100 iterations with the high-fidelity model.

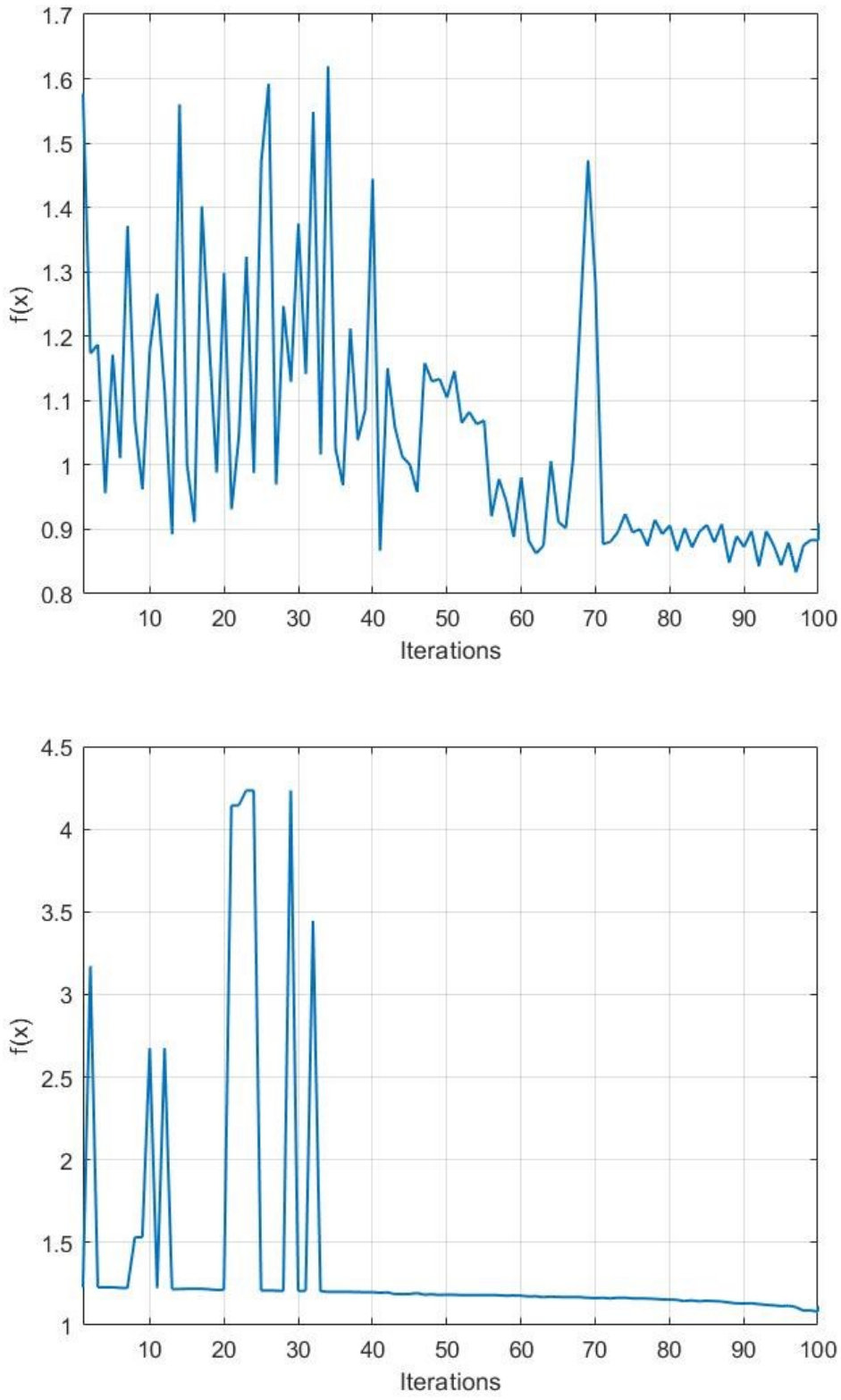

Figure 62: $f(x)$-iterations. The graph above is the result of the MFEI algorithm. The lower one of the SFEI algorithm.

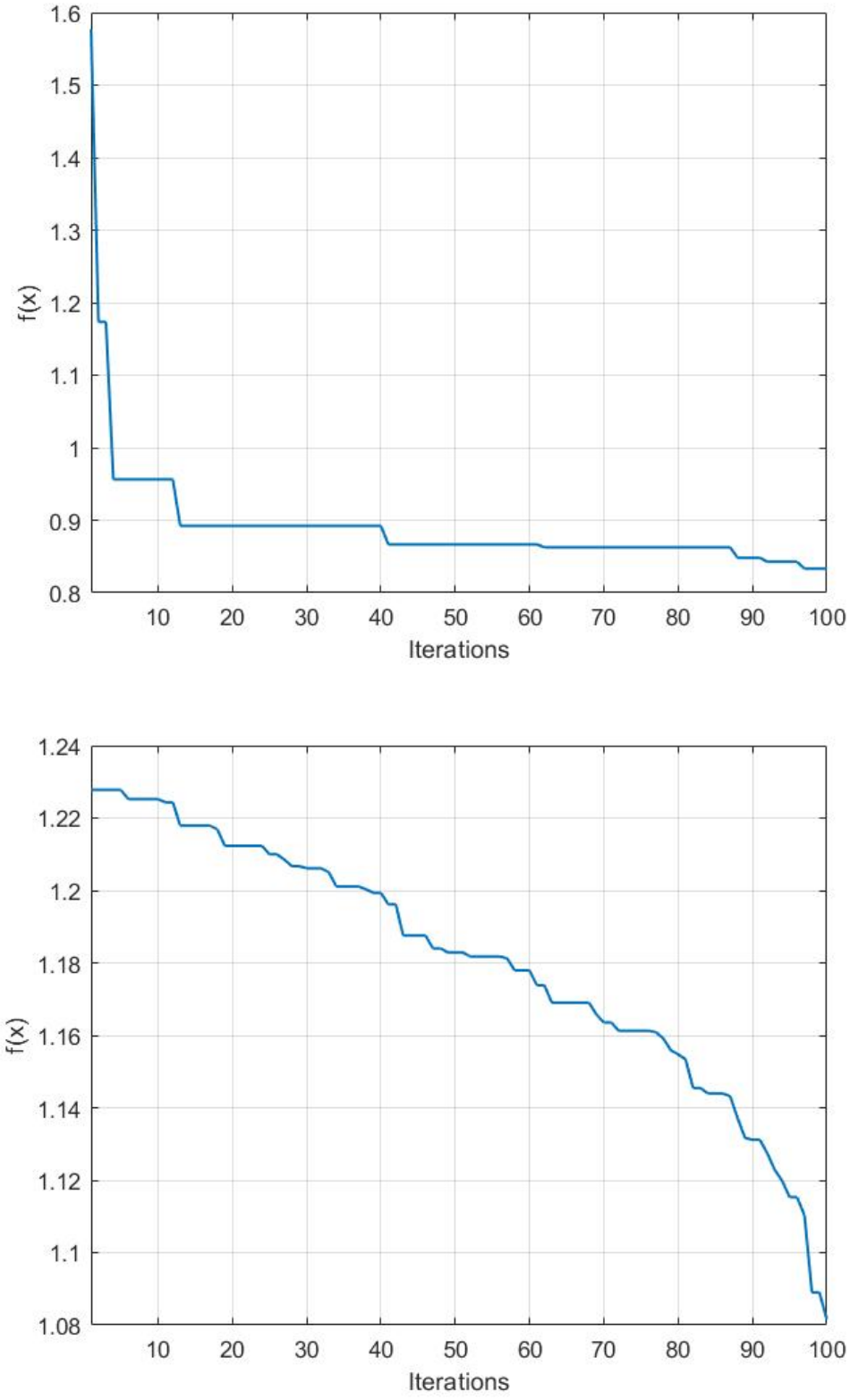

Figure 63: $f(x)$ overall minimum-iterations. The graph above is the result of the MFEI algorithm. The lower one of the SFEI algorithm.

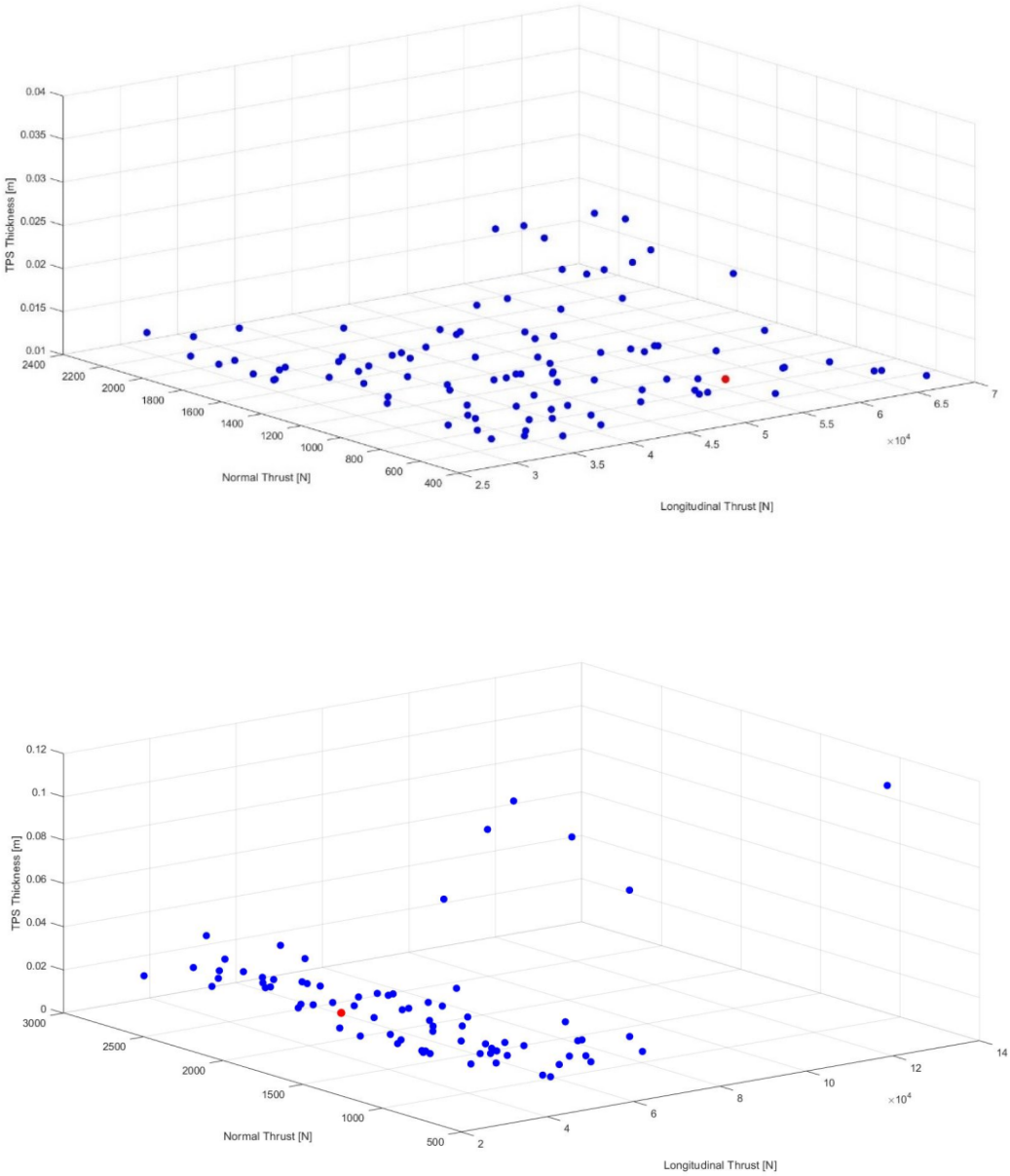

Figure 64: sample points selected during iterations. In red is reported the sample corresponding to the overall minimum of the objective function. The graph above is the result of the MFEI algorithm. The lower one of the SFEI algorithm.

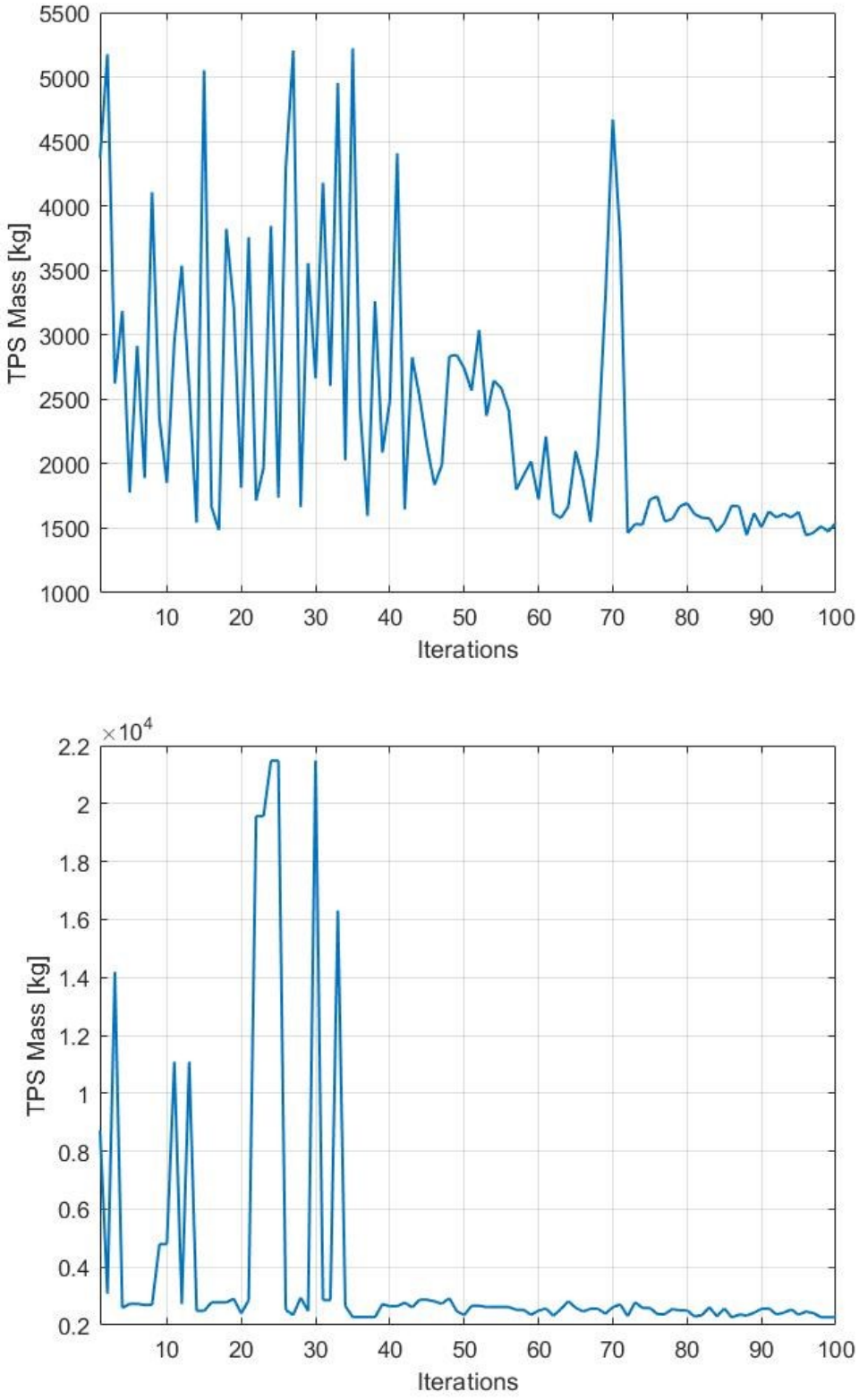

Figure 65: TPS mass-iterations. The graph above is the result of the MFEI algorithm. The lower one of the SFEI algorithm.

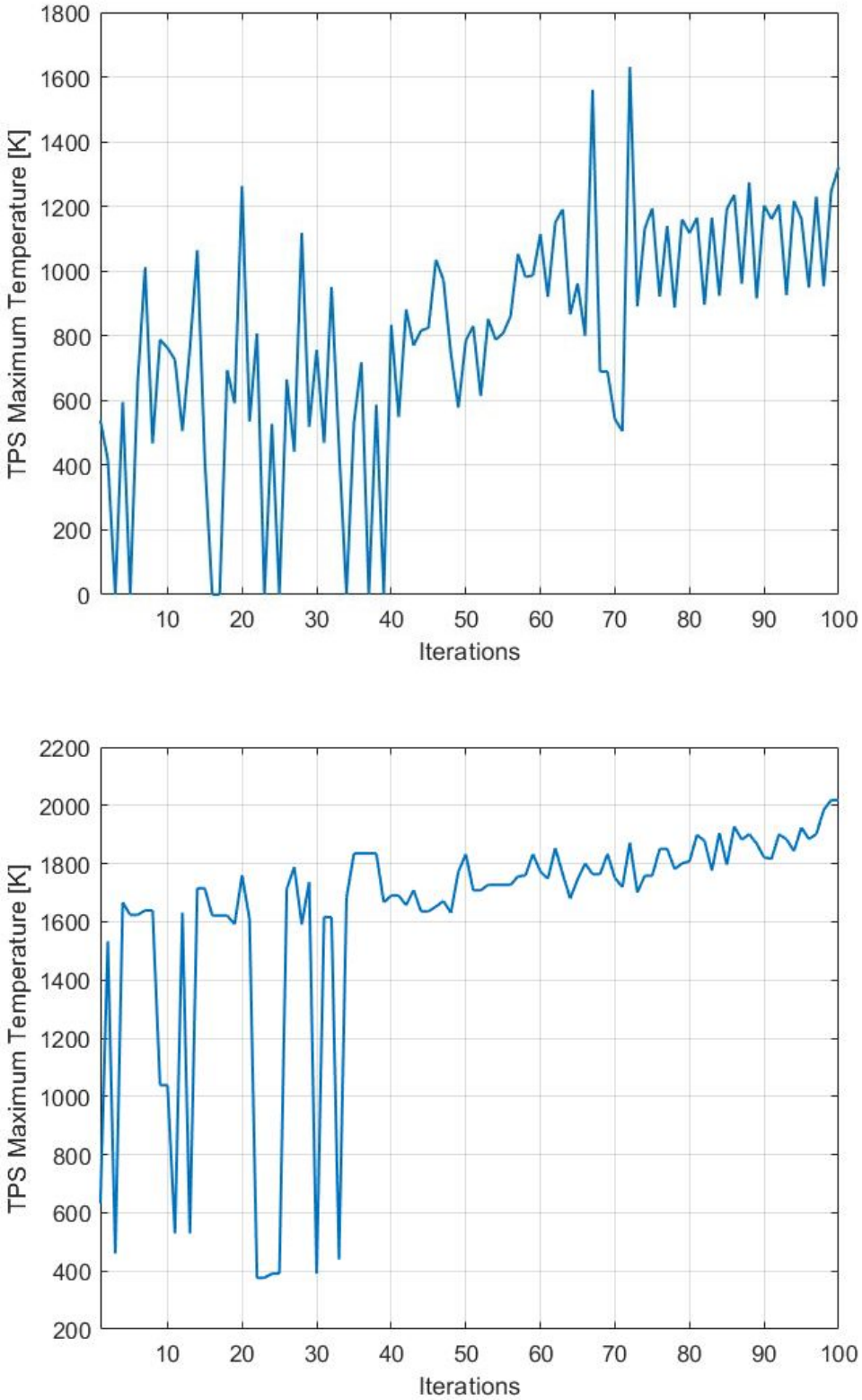

Figure 66: TPS maximum temperature-iterations. The graph above is the result of the MFEI algorithm. The lower one of the SFEI algorithm.

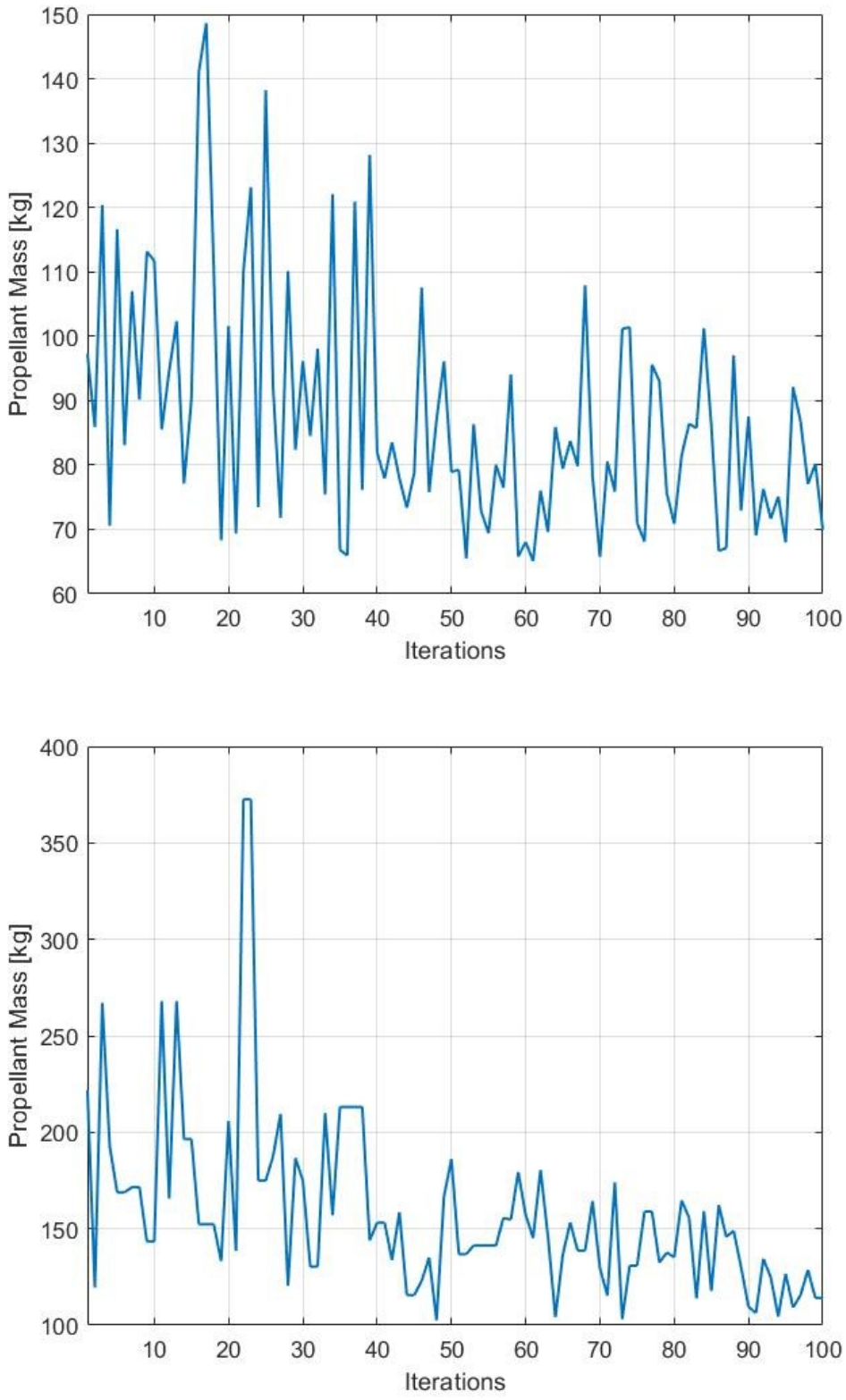

Figure 67: Propellant mass-iterations. The graph above is the result of the MFEI algorithm. The lower one of the SFEI algorithm.

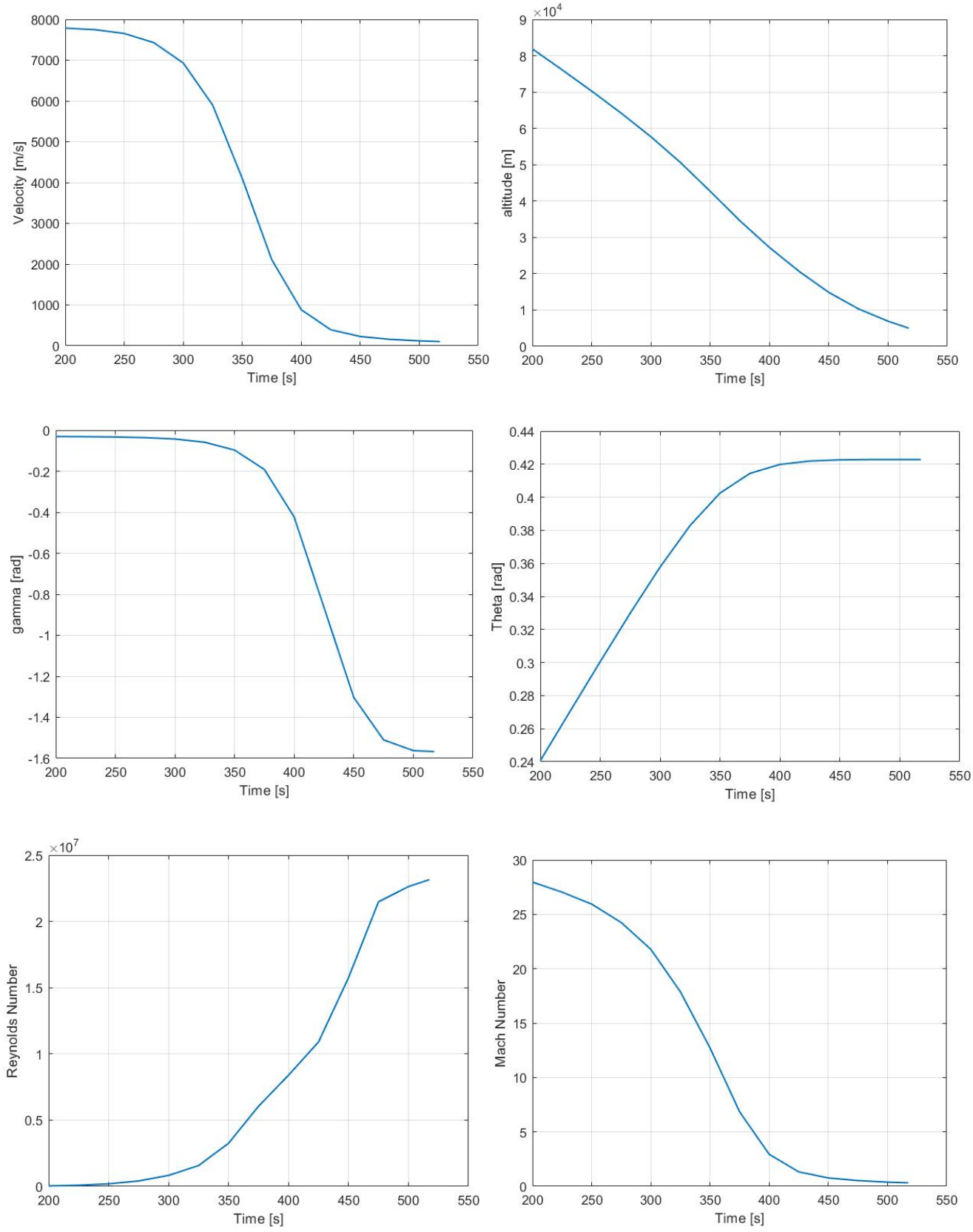

Figure 68: trajectory model outcomes.

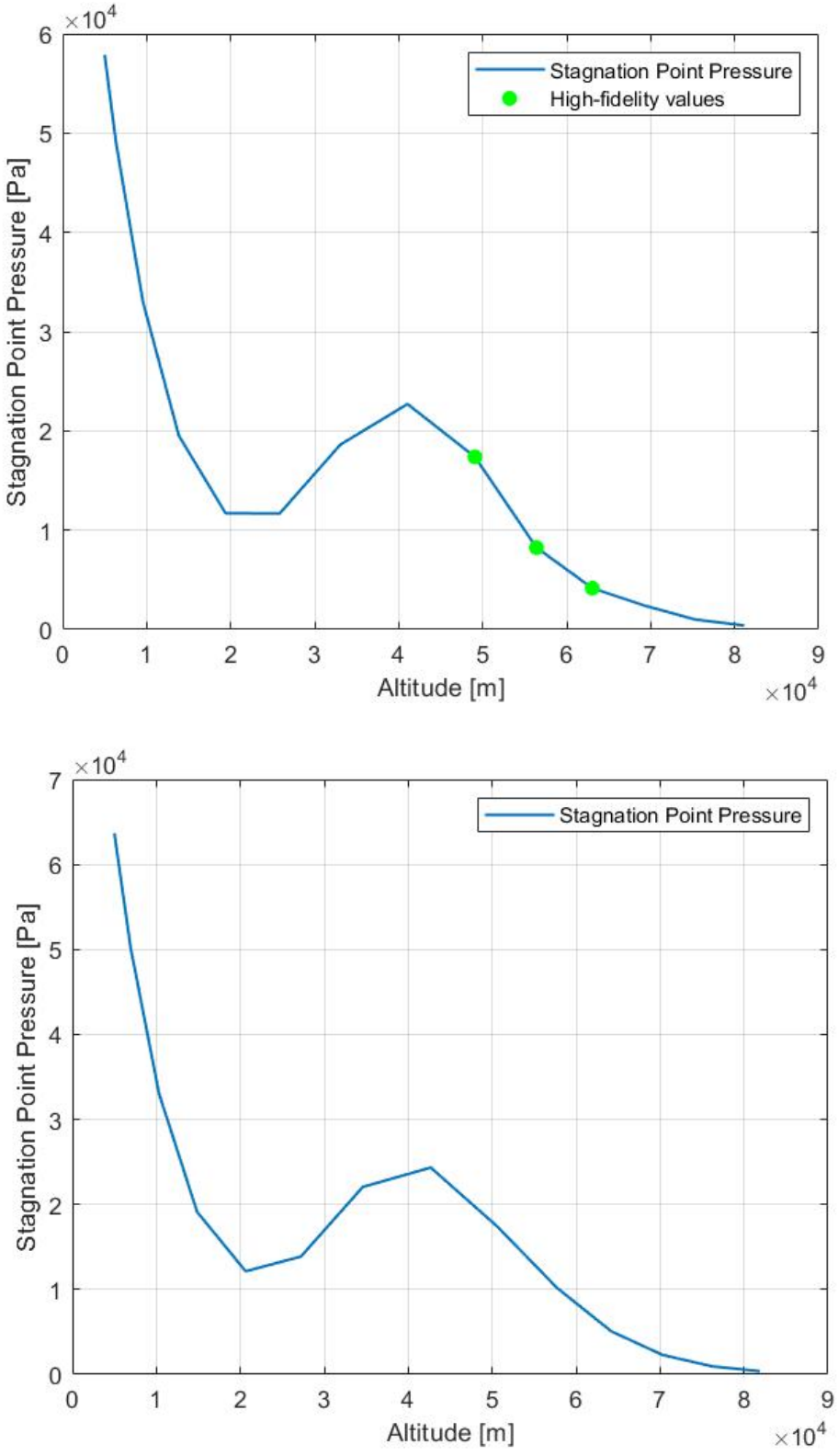

Figure 69: Stagnation point pressure-altitude. The graph above is the result of the MFEI algorithm. The lower one of the SFEI algorithm.

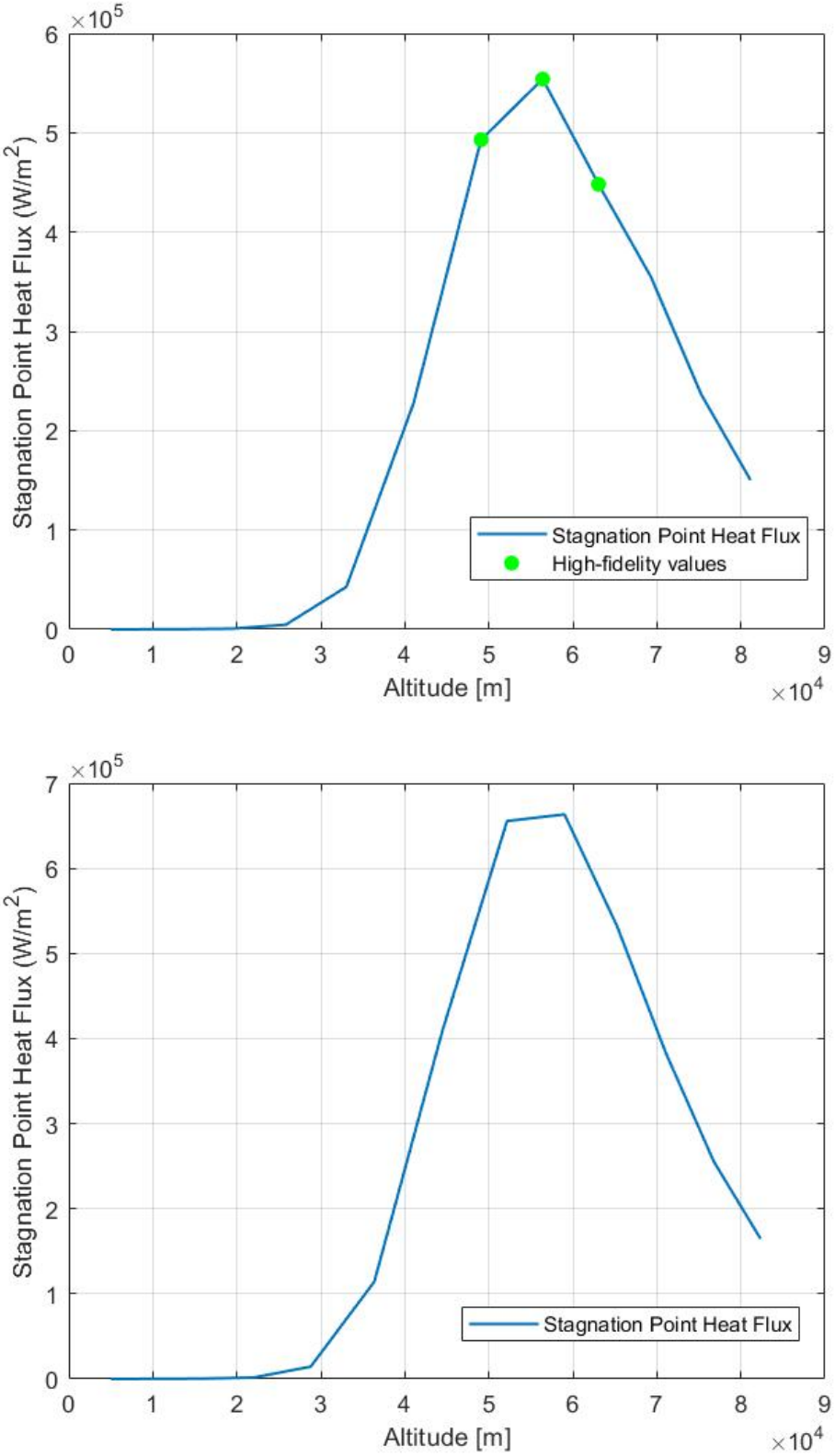

Figure 70: Stagnation point heat flux-altitude. The graph above is the result of the MFEI algorithm. The lower one of the SFEI algorithm.

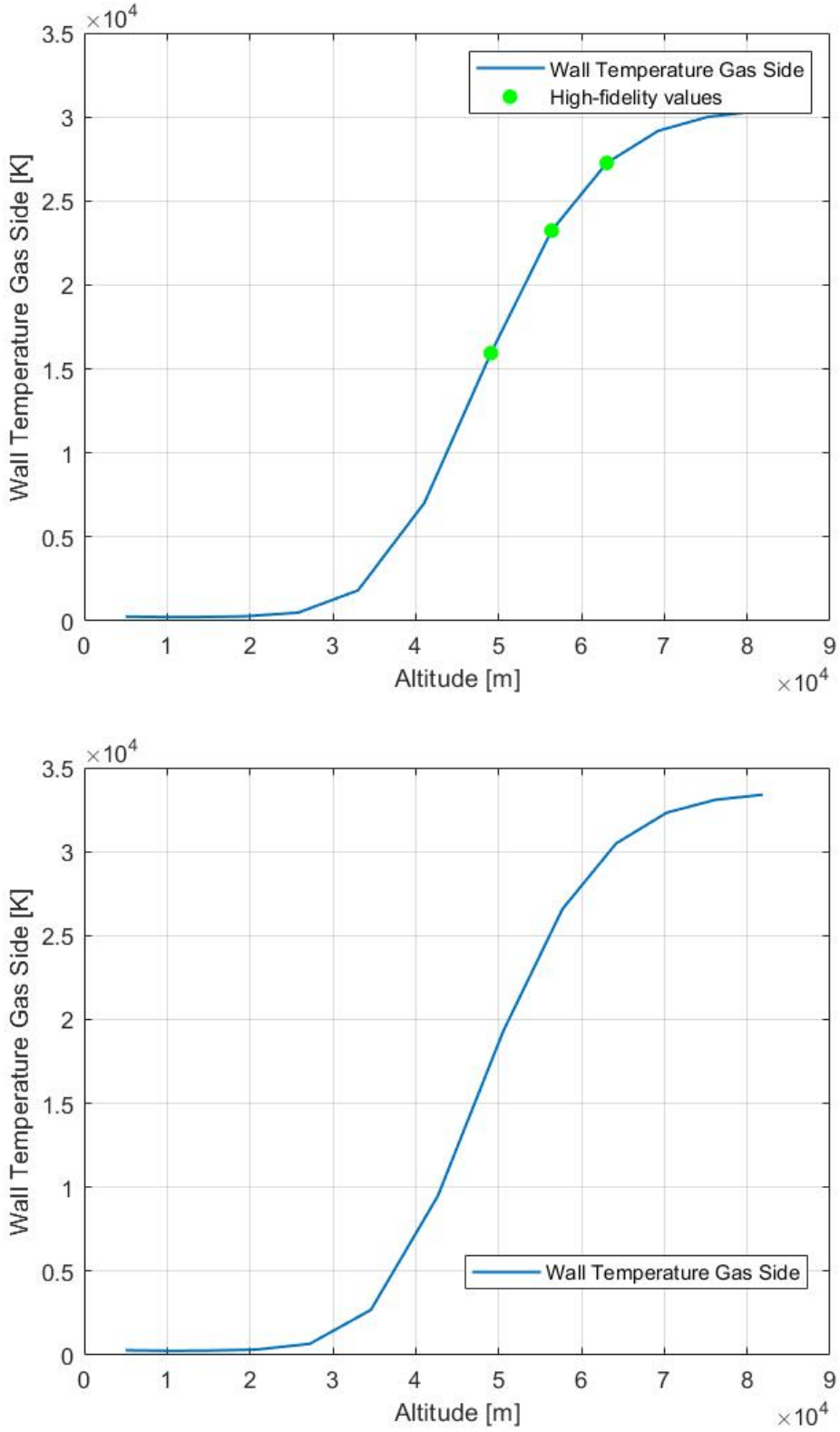

Figure 71: Stagnation point wall temperature gas side-altitude. The graph above is the result of the MFEI algorithm. The lower one of the SFEI algorithm

9. Conclusions

The objectives of this thesis are to investigate the leverage of multifidelity methods to speed-up the design optimization process and to develop an innovative multifidelity algorithm for the multidisciplinary design optimization (MDO) of a vehicle performing an atmospheric controlled re-entry.

This thesis has explored multifidelity methods in order to reduce the time expense in the design process and acquire significant information about the system in the early stages of the process. Specifically, a multifidelity expected improvement algorithm is implemented and compared to the performance of a single fidelity expected improvement algorithm in order to address an optimization problem. The tests are conducted on analytical benchmark objective functions and related low-fidelity formulations, given the low computational expense of simulations and knowledge about the analytical optimum of the test functions. The single fidelity algorithm invokes the higher fidelity formulation of the analytical functions. The multifidelity algorithm leverages the hierarchical set of different fidelity level functions.

The results show that the multifidelity framework enriches the information elicited from low-fidelity models with high-fidelity data, guiding effectively towards the optimum. Moreover, the computational expense is inferior compared to the single fidelity algorithm. In fact, we can observe that the high-fidelity model is queried a lower number of times than low-fidelity models during the optimization process. The tests show that the algorithm tends with a probability of less than 7% to detect the minimum of the low-fidelity functions as global minimum of the optimization problem, instead of the benchmark functions minimum. However, this behaviour was expected given the use of significantly less data from the benchmark functions.

In addition, this thesis proposed an innovative multifidelity algorithm to address the multidisciplinary design optimization problem of controlled re-entry vehicle.

The objective function is dependent on propellant mass burned, thermal protection system (TPS) mass and structural temperature. The design variables are the thrust tangential and normal to the trajectory and the TPS thickness. The optimization constraints are the capsule geometry, TPS material, propellants and initial re-entry conditions. Moreover, the thruster can fire only in a specific range of altitude in order to avoid failure due to the hypersonic thermofluidic regime, stressful for the propulsive system; The TPS temperature must be under the material melting temperature to guarantee the survival of the payload; The thrust needed to optimize the trajectory must be allowable by the stock propellant mass; The residual of CFD analysis must be less than 10^{-6} in order to guarantee the convergence of computation.

Specifically, two levels of fidelity are considered for the aerodynamic and aerothermodynamic model, given the importance of this physical phenomena for

the problem. In fact, this physical discipline has a dominant impact on design of TPS structure, TPS material selection and trajectory definition of a re-entry vehicle. The low-fidelity aerodynamic and aerothermodynamic model is defined by experimental and empirical correlations. The high-fidelity model is given by computational fluid dynamic (CFD) simulations for the aerodynamic physics and Fay-Riddle correlations for thermal loads estimate.

The original multifidelity algorithm is based on the MFEI approach. The formulation of the multifidelity expected improvement was modified to consider the aerodynamic and aerothermodynamic physical phenomena involved in atmospheric re-entry. The optimization process is conducted considering 100 iterations for both the original MFEI and SFEI algorithm.

The results show that the proposed MFEI algorithm is capable to achieve a better optimization results than the SFEI algorithm based on low-fidelity aerothermodynamic model. The formulation of the objective function it is dimensionless and was designed to return 1 when the propellant mass, temperature and mass of the TPS correspond to average values for re-entry spacecrafts. Specifically, the minimum of the objective function is 0.8332 performing the multifidelity approach and 1.0816 executing the single fidelity approach. As a consequence, the optimal design achieved by leveraging the multifidelity method has a TPS structural mass, TPS structural temperature and propellant mass burned significantly lower than the single low-fidelity outcome. This suggest that the innovative algorithm is capable to enrich low-fidelity data with high-fidelity information to improve the knowledge about the objective function.

Analysing the computational effort outcoming from the MDO problem, a comparison between the original MFEI algorithm and SFEI based on high-fidelity aerothermodynamic model can be conducted. It is possible to note that the computational expense of the multifidelity algorithm in performing 100 iterations corresponds to the calculation time necessary to perform 6 iterations considering the single high-fidelity algorithm.

Many future developments for further research relating to the multifidelity methods applied to re-entry optimization problems can be considered. First the high-fidelity representation of the aerothermodynamic physics would largely benefit from the use of CFD solvers capable to implement vibrational and chemical non-equilibrium phenomena to further refine the solution. Moreover, the experiments conducted on the MDO problem were carried out with a single run of the algorithm due to the excessive computational cost. It would be significant to extend the number of runs in order to have a statistic on the performance of the innovative MFEI and SFEI algorithm considering both the low and high-fidelity cases. The time discretization implemented in the numerical integration of the equations of planetary re-entry flight can be refined in order to obtain more detailed results. The selected time

discretization is the result of a trade-off between the accuracy of results and the computational resources available.

In conclusion, this thesis demonstrated the ability of multifidelity methods to speed up the search for the optimum in an optimization problem. Furthermore, it has been shown that the innovative multi-fidelity formulation proposed for the MDO problem of atmospheric re-entry, allows to effectively and efficiently combine low fidelity information with accurate and computationally expensive data, in order to obtain a sensitive improvement to the design of a re-entry vehicle.

Bibliography

[1] Davies, C., Planetary Mission Entry Vehicles Quick Reference Guide, Version 3.0, NASA SP-2006-3401,2006.

[2] J. J. Bertin, Hypersonic Aerothermodynamics, AIAA Education Series, 1994.

[3] Adams, J.C. Atmospheric Re-Entry, 2003.

[4] J. J. Martin, Atmospheric reentry: an introduction to its science and engineering, AIAA Education Series, 1966.

[5] Hollis, Brian & Borrelli, Salvatore, Aerothermodynamics of blunt body entry vehicles. Progress in Aerospace Sciences. Pages 48-49. 2012.

[6] J. A. Fay, F. R. Riddel, Theory of Stagnation Point Heat Transfer in Dissociated Air, Journal of the Aerospace Sciences, Volume 25, Issue 2, February 1958.

[7] K. Sutton, R. A. Graves, A General Stagnation Point Convective Heating Equation for Arbitrary Gas Mixture, NASA TR-376, 1971.

[8] M. E- Tauber, K. Sutton, Stagnation Point Radiative Heating Relations for Hearth and Mars Entries, Journal of Spacecraft and Rockets, Volume 28, Issue 12, December 1957.

[9] Sutherland, W. The viscosity of gases and molecular forces, Philosophical Magazine, 1893

[10] Oswatitsch, K. Ähnlichkeitsgesetz für Hyperschallströmung. ZAMP, vol. II, pp. 249–264, 1951. (Similarity Laws for Hypersonic Flow. Royal Institute of Technology. Stockholm. Sweden. KTH-AERO TN 16, 1950.

[11] Echert, E. R. G. and Drake, R. M. Analysis of Heat and Mass Transfer, McGraw-Hill, New York, 1972.

[12] Anderson, J.D. Computational Fluid Dynamics: The Basics with Applications. McGraw-Hill Education, 1995.

[13] B, Willcox K, Gunzburger M, Survey of Multifidelity Methods in Uncertainty Propagation, Inference, and Optimization. Siam Rev 60(3):550–591, 2018.

[14] Meliani, M., Bartoli, N., Lefebvre, T., Bouhlel, M.A., J., Martins, and Morlier, J., "Multi-fidelity efficient global optimization: Methodology and application to airfoil shape design," AIAA Aviation 2019 Forum, AIAA, 2019.

[15] D. Huang, T. T. Allen, W. I. Notz, and R. A. Miller, Sequential kriging optimization using multiple-fidelity evaluations, Structural and Multidisciplinary Optimization, vol. 32, no. 5, pages 369-382, 2006

[16] D. R. Jones, M. Schonlau, and W. J. Welch, Efficient global optimization of expensive black-box functions, Journal of Global optimization, vol. 13, no. 4, pp. 455-492, 1998.

[17] F. Grassi, G. Manganini, M. Garraffa, L. Mainini, Resource Aware Multifidelity Active Learning for Efficient Optimization, 2020

[18] E. Brochu, V. M. Cora, and N. de Freitas, A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning, 2010.

[19] P. I. Frazier, A tutorial on bayesian optimization, arXiv preprint arXiv:1807.02811, 2018.

[20] Beran P, Bryson D, Thelen A, Diez M, Serani A, Comparison of Multi-Fidelity Approaches for Military Vehicle Design. AIAA 2020-3158, 2020

[21] Huang, D., Allen, T.T., Notz, W.I. et al. Global Optimization of Stochastic Black-Box Systems via Sequential Kriging Meta-Models. J Glob Optim 34, pages 441–466 (2006).

[22] Zan Yang, Haobo Qiu, Liang Gao, Xiwen Cai, Chen Jiang & Liming Chen, A surrogate-assisted particle swarm optimization algorithm based on efficient global optimization for expensive black-box problems, Engineering Optimization, pages 549-566, 2019.

[23] Peherstorfer B, Willcox K, Gunzburger M, Survey of Multifidelity Methods in Uncertainty Propagation, Inference, and Optimization. Siam Rev 60(3): pages 550–591, 2018.

[24] Beran P, Bryson D, Thelen A, Diez M, Serani A, Comparison of Multi-Fidelity Approaches for Military Vehicle Design. AIAA 2020-3158, 2020.

[25] Eldred, M.S., Dunlavy, D.M., Formulations for Surrogate-Based Optimization with Data Fit, Multifidelity, and Reduced-Order Models, 2006

[26] Giles, M. B., "Multilevel Monte Carlo methods," Vol. 24, 2015, pp. 259–328.

[27] Pellegrini, R., Serani, A., Broglia, R., Diez, M., and Harries, S., "Resistance and Payload Optimization of a Sea Vehicle by Adaptive Multi-Fidelity Metamodeling," 2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, American Institute of Aeronautics and Astronautics, 2018.

[28] Chaudhuri, A., Jasa, J., Martins, J., and Willcox, K. E., "Multifidelity Optimization Under Uncertainty for a Tailless Aircraft," 2018 AIAA Non-Deterministic Approaches Conference, American Institute of Aeronautics and Astronautics, 2018.

[29] Thelen, A. S., Leifsson, L. T., and Beran, P. S., “Aeroelastic Flutter Prediction using Multi-fidelity Modeling of the Aerodynamic Influence Coefficients,” AIAA SciTech Forum, American Institute of Aeronautics and Astronautics, 2019.

[30] Thelen, A., Leifsson, L., and Beran, P., “Multifidelity Flutter Prediction Using Local Corrections to the Generalized AIC,” International Forum on Aeroelasticity and Structural Dynamics (IFASD 2019), 2019.

[31] Zhang, W.-W., Ye, Z.-Y., Zhang, C.-A., and Liu, F., “Supersonic Flutter Analysis Based on a Local Piston Theory,” AIAA Journal, Vol. 47, No. 10, 2009, pp. 2321–2328.

[32] Wukie, N. A., Orkwis, P. D., and Schrock, C. R., “A Chimera-based, zonal discontinuous Galerkin method,” 23rd AIAA Computational Fluid Dynamics Conference, American Institute of Aeronautics and Astronautics, 2017.

[33] Fuchi, K., Wolf, E. M., Makhija, D., Wukie, N. A., Schrock, C. R., and Beran, P. S., “Enhancement of Low Fidelity Fluid Simulations using Machine Learning,” AIAA SciTech Forum, American Institute of Aeronautics and Astronautics, 2020.

[34] D. R. Jones, M. Schonlau, and W. J. Welch, Efficient global optimization of expensive black-box functions, J. Global Optim., 13 (1998), pp. 455--492. (Cited on pp. 559, 577).

[35] G. Venter, R. T. Haftka, and J. H. Starnes, Construction of response surface approximations for design optimization, AIAA J., pages 2242-2249, 1998.

[36] N. M. Alexandrov, J. E. Dennis, Jr., R. M. Lewis, and V. Torczon, A trust-region framework for managing the use of approximation models in optimization, Struct. Optim.,15, 1998.

[37] N. M. Alexandrov, R. M. Lewis, C. R. Gumbert, L. L. Green, and P. A. Newman, Approximation and model management in aerodynamic optimization with variable-fidelity models, J. Aircraft, 38, 2001.

[38] M. C. Kennedy and A. O'Hagan, Predicting the output from a complex computer code when fast approximations are available, Biometrika, 87, pages 1-13, 2000.

[39] E. Annels, Geostatistical Ore-Reserve Estimation, in Mineral Deposit Evaluation, Springer, Dordrecht, The Netherlands, pages 175-245, 1991.

[40] D. E. Myers, Matrix formulation of co-kriging, J. Internat. Assoc. Math. Geology, 14 pp. 249-257, 1982.

[41] P. Perdikaris, D. Venturi, J. O. Royset, and G. E. Karniadakis, Multi-fidelity modelling via recursive co-kriging and Gaussian--Markov random fields, Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci., 2015.

[42] P. Bratley, B. L. Fox, and L. E. Schrage, A Guide to Simulation, Springer, 1987.

[43] J. M. Hammersley and D. C. Handscomb, Monte Carlo Methods, Methuen, London, 1964.

[44] L. Nelson, On control variate estimators, Comput. Oper. Res., 14, pp. 219-225, 1987.

[45] J. A. Christen and C. Fox, Markov chain Monte Carlo using an approximation, J. Comput. Graph. Statist., 14 pp. 795-810, 2005.

[46] Fox and G. Nicholls, Sampling conductivity images via MCMC, in The Art and Science of Bayesian Image Analysis, University of Leeds, pp. 91-100, 1997.

[47] Narayan, C. Gittelson, and D. Xiu, A stochastic collocation algorithm with multifidelity models, SIAM J. Sci. Comput., 36 pp. A495-A521, 2014.

[48] Peherstorfer, T. Cui, Y. Marzouk, and K. Willcox, Multifidelity importance sampling, Comput. Methods Appl. Mech. Engrg., 300 pp. 490-509, 2016.

[49] Myers, D. E., "Matrix Formulation of Co-Kriging," Mathematical Geology, Vol. 14, No. 3, 1982, pp. 2495-257

[50] Cressie, N., Statistics for Spatial Data: Wiley Series in Probability and Statistics, Wiley, New York, 1993, pp. 138–142.

[51] S. Leary, A. Bhaskar, and A. Keane. Global approximation and optimization using adjoint computational uid dynamics codes. AIAA Journal, 42(3):631-641, March 2004.

[52] A.I.J. Forrester, A. Sobester, and A.J. Keane. Multifidelity optimization via surrogate modelling. Journal of the Royal Statistical Society, 463(2):3251-3269, September 2007.

[53] R. Jones. A taxonomy of global optimization methods based on response surfaces. Journal of Global Optimization, 21:345-383, 2001.

[54] M. J. Sasena, P. Papalambros, and P. Goovaerts. Exploration of metamodeling sampling criteria for constrained global optimization. Engineering Optimization, pages 263-278, 2002.

[55] Rajnarayan, A. Haas, and I. Kroo. A multifidelity gradient-free optimization method and application to aerodynamic design. AIAA 2008 Multidisciplinary Analysis and Optimization Conference, Victoria, British Columbia, September 10-12 2008.

[56] G. Rajnarayan. Trading Risk and Performance for Engineering Design Optimization Using Multidelity Analyses. PhD thesis, Stanford University, June 2009.

[57] R.G. Regis and C.A. Shoemaker. Constrained global optimization of expensive black-box functions using radial basis functions. Journal of Global Optimization, 31: pages 153-171,2005.

[58] M. Kennedy and A. O'Hagan. Predicting the output from a complex computer code when fast approximations are available. Biometrika, 87(1): pages 1-13, 2000.

[59] M. Kennedy and A. O'Hagan. Bayesian calibration of computer models. Journal of the Royal Statistical Society, 63(3): pages 425-464, 2001.

[60] N. M. Alexandrov, J. Dennis, Jr., R. M. Lewis, and V. Torczon. A trust region framework for managing the use of approximation models in optimization. Technical Report CR-201745, NASA, October 1997.

[61] N. M. Alexandrov, R.M. Lewis, C. Gumbert, L. Green, and P. Newman. Optimization with variable-fidelity models applied to wing design. Technical Report CR-209826, NASA, 1999.

[62] N.M. Alexandrov, R.M. Lewis, C. Gumbert, L. Green, and P. Newman. Approximation and model management in aerodynamic optimization with variable-fidelity models. Journal of Aircraft, pages 1093-1101, 2001.

[63] A.R. Conn, N.I. Gould, and Ph.L. Toint. Trust-Region Methods. MPS/SIAM Series on Optimization. Society for Industrial and Applied Mathematics, Philadelphia, PA,2000.

[64] A.R. Conn, K. Scheinberg, and Ph.L. Toint. Recent progress in unconstrained nonlinear optimization without derivatives. Mathematical Programming, 79:397-414, 1997.

[65] M. Marazzi and J. Nocedal. Wedge trust region methods for derivative free optimization. Mathematical Programming, 91(2): pages 289-305, 2002.

[66] A.R. Conn, K. Scheinberg, and L. Vicente. Geometry of interpolation sets in derivative free optimization. Mathematical Programming, 111(1-2):pages 141-172, 2008.

[67] R. Conn, K. Scheinberg, and L. Vicente. Global convergence of general derivative free trust-region algorithms to first- and second-order critical points. SIAM Journal of Optimization, 20(1): pages 387-415, 2009

[68] R. Conn, K. Scheinberg, and L. N. Vicente. Introduction to Derivative-Free Optimization. MPS/SIAM Series on Optimization. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2009.

[69] S.M. Wild. Derivative-Free Optimization Algorithms for Computationally Expensive Functions. PhD thesis, Cornell University, January 2009.

[70] S.M. Wild and C. A. Shoemaker. Global convergence of radial basis function trust region derivative-free algorithms. SIAM Journal of Optimization,21(3):761-781, 2011.

[71] March and K. Willcox, Provably Convergent Multifidelity Optimization Algorithm Not Requiring High-Fidelity Derivatives. Massachusetts Institute of Technology, Cambridge, Massachusetts.

[72] Georgiades, Alex, Set-Based Design and Optimisation of Aircraft Systems, 2018.

[73] Evin J. Cramer, J. E. Dennis, Jr., Paul D. Frank, Robert Michael Lewis, and Gregory R. Shubin, Problem Formulation for Multidisciplinary Optimization, SIAM Journal on Optimization 4:4, pages 754-776, 1994.

[74] Sobieszczanski-Sobieski, Jaroslaw & Haftka, Raphael, Multidisciplinary aerospace design optimization: Survey of recent developments. Structural Optimization. 14. Pages 1-23. 1997.

[75] S.D. Eppinger and T.R. Browning, Design Structure Matrix Methods and Applications, MIT Press, Cambridge, 2012.

[76] Martins, Joaquim & Lambe, Andrew. Multidisciplinary Design Optimization: A Survey of Architectures. AIAA Journal. 51. Pages 2049-2075. 2013.

[77] Kroo, I. M., MDO for Large-Scale Design, Multidisciplinary Design Optimization: State-of-the-Art, edited by N. Alexandrov and M. Y. Hussaini, SIAM, pp. 22–44, 1997.

[78] Alexandrov, N. M. and Lewis, R. M., Analytical and Computational Aspects of Collaborative Optimization for Multidisciplinary Design, AIAA Journal, Vol. 40, No. 2, pp. 301–309, 2002.

[79] Balling, R. J. and Sobieszczanski-Sobieski, J., Optimization of Coupled Systems: A Critical Overview of Approaches, AIAA Journal, Vol. 34, No. 1, pp. 6–17, 1996.

[80] Lasdon, L. S., Optimization Theory for Large Systems, The Macmillan Company, 1970.

[81] Conejo, A. J., Nogales, F. J., and Prieto, F. J., A Decomposition Procedure Based on Approximate Newton Directions, Mathematical Programming, Vol. 93, pp. 495–515, 2002.

[82] Dantzig, G. B. and Wolfe, P., Decomposition Principle for Linear Programs, Operations Research, Vol. 8, 1960, pp. 101-111.

[83] Benders, J. F., Partitioning Procedures for Solving Mixed Variables Programming Problems, Numerische Mathematik, Vol. 4, pp. 238–252, 1962.

[84] Michelena, N. F. and Papalambros, P. Y., A Network Reliability Approach to Optimal Decomposition of Design Problems, Journal of Mechanical Design, Vol. 2, 1995, pp. 195–204.

[85] Michelena, N. F. and Papalambros, P. Y., A Hypergraph Framework for Optimal Model-Based Decomposition of Design Problems, Computational Optimization and Applications, Vol. 8, pp. 173–196, 1997.

[86] Tosserams, S., Hofkamp, A. T., Etman, L. F. P., and Rooda, J. E., A Specification Language for Problem Partitioning in Decomposition-Based Design Optimization, Structural and Multidisciplinary Optimization, Vol. 42, pp. 707–723, 2010.

[87] Jameson, Antony, Aerodynamic Design via Control Theory. Journal of Scientific Computing. 1988.

[88] Raphson, Joseph, Analysis Æequationum Universalis (in Latin) (2nd ed.). London: Thomas Bradyll. 1697.

[89] Straeter, T. A. On the Extension of the Davidon-Broyden Class of Rank One, Quasi-Newton Minimization Methods to an Infinite Dimensional Hilbert Space with Applications to Optimal Control Problems. NASA Technical Reports Server. NASA.

[90] Nelder, John A.; R. Mead, A simplex method for function minimization. Computer Journal. 7 (4): pages 308–313. 1965.

[91] Spendley, W.; Hext, G. R.; Himsworth, F. R., Sequential Application of Simplex Designs in Optimisation and Evolutionary Operation". Technometrics. 4 (4): pages 441–461. 1962.

[92] Hooke, R.; Jeeves, T.A. “Direct search" solution of numerical and statistical problems". Journal of the ACM. 8 (2): 212–229, 1961.

[93] Haftka, R. T., "Automated Procedure for Design of Wing Structures to Satisfy Strength and Flutter Requirements," Tech. Rep. TN D-7264, NASA Langley Research Center, Hampton, VA, 1973.

[94] Haftka, R. T., Starnes Jr., J. H., Barton, F. W., and Dixon, S. C., "Comparison of Two Types of Optimization Procedures for Flutter Requirements," AIAA Journal, Vol. 13, No. 10, pp. 1333–1339, 1975.

[95] Haftka, R. T., "Optimization of Flexible Wing Structures Subject to Strength and Induced Drag Constraints," AIAA Journal, Vol. 14, No. 8, pp. 1106–1977, 1977.

[96] Haftka, R. T. and Shore, C. P., "Approximate Methods for Combined Thermal/Structural Design," Tech. Rep. TP-1428, NASA, June 1979.

[97] Schmit, L. A., "Structural Design by Systematic Synthesis," 2nd Conference on Electronic Computation, ASCE, New York, NY, pp. 105–132, 1960.

[98] Schmit, L. A. and Thornton, W. A., "Synthesis of an Airfoil at Supersonic Mach Number," Tech. Rep. CR 144, NASA, Jan.1965.

[99] Schmit, L. A., "Structural Synthesis—Its Genesis and Development," AIAA Journal, Vol. 19, No. 10, pp. 1249–1263, 1981.

[100] Schmit Jr., L. A., "Structural Synthesis — Precursor and Catalyst. Recent Experiences in Multidisciplinary Analysis and Optimization," Tech. Rep. CP-2337, NASA, 1984.

[101] Pelamatti, Julien & Brevault, Loïc & Balesdent, Mathieu & Talbi, El-Ghazali & Guerin, Yannick. Efficient global optimization of constrained mixed variable problems. Journal of Global Optimization. 2018.

[102] Zaefferer, Martin & Bartz-Beielstein, Thomas. Efficient Global Optimization with Indefinite Kernels. 2016.

[103] Horowitz, Bernardo & Guimarães, Leonardo & Afonso, Silvana. Efficient Global Optimization of a Structural Frame. 2020.

[104] Song, Jialin & Chen, Yuxin & Yue, Yisong. A General Framework for Multi-fidelity Bayesian Optimization with Gaussian Processes. 2018.

[105] Bertram, Anna & Zimmermann, Ralf. Theoretical investigations of the new Cokriging method for variable-fidelity surrogate modeling: Well-posedness and maximum likelihood training. Advances in Computational Mathematics. 2018.

[106] Gorodetsky, Alex & Geraci, Gianluca & Eldred, Michael & Jakeman, John. A generalized approximate control variate framework for multifidelity uncertainty quantification. Journal of Computational Physics. 2020.

[107] Popov, Andrey & Mou, Changhong & Iliescu, Traian & Sandu, Adrian. A Multifidelity Ensemble Kalman Filter with Reduced Order Control Variates. 2020.

[108] Holden, Lars. Mixing of MCMC algorithms. Journal of Statistical Computation and Simulation. 89. Pages 1-19. 2019.

[109] Poloczek, Matthias & Wang, Jialei & Frazier, Peter. Multi-Information Source Optimization with General Model Discrepancies. 2016.

[110] F. Grassi, G. Manganini, M. Garraffa, L. Mainini, Resource Aware Multifidelity Active Learning for Efficient Optimization, 2020

[111] D. R. Jones, M. Schonlau, and W. J. Welch, Efficient global optimization of expensive black-box functions, Journal of Global optimization, vol. 13, no. 4, pp. 455-492, 1998.

[112] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. K. Tucker, “Surrogate-based analysis and optimization," Progress in aerospace sciences, vol. 41, no. 1, pp. 1-28, 2005.

[113] M. Eldred and D. Dunlavy, Formulations for surrogate-based optimization with data fit, multifidelity, and reduced-order models, in 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, p. 7117, 2006.

[114] T. Robinson, M. Eldred, K. Willcox, and R. Haimes, Surrogate-based optimization using multifidelity models with variable parameterization and corrected space mapping, AIAA Journal, vol. 46, no. 11, pp. 2814-2822, 2008.

[115] I. Forrester and A. J. Keane, “Recent advances in surrogate-based optimization, Progress in aerospace sciences, vol. 45, no. 1-3, pp. 50-79, 2009.

[116] Bhosekar and M. Ierapetritou, Advances in surrogate based modeling, feasibility analysis, and optimization: A review, Computers & Chemical Engineering, vol. 108, pp. 250-267, 2018.

[117] M. C. Kennedy and A. O'Hagan, Predicting the output from a complex computer code when fast approximations are available, Biometrika, vol. 87, no. 1, pp. 1-13, 2000.

[118] I. Forrester, A. Sobester, and A. J. Keane, Multifidelity optimization via surrogate modelling, Proceedings of the royal society a: mathematical, physical and engineering sciences, vol. 463, no. 2088, pp. 3251-3269, 2007.

[119] M. G. Fernandez-Godino, C. Park, N.-H. Kim, and R. T. Haftka, Review of multifidelity models, arXiv preprint arXiv:1609.07196, 2016.

[120] Park, R. T. Haftka, and N. H. Kim, Remarks on multifidelity surrogates, Structural and Multidisciplinary Optimization, vol. 55, no. 3, pp. 1029-1050, 2017.

[121] Peherstorfer, K. Willcox, and M. Gunzburger, Survey of multifidelity methods in uncertainty propagation, inference, and optimization, Siam Review, vol. 60, no. 3, pp. 550-591,2018.

[122] P. S. Beran, D. Bryson, A. S. Thelen, M. Diez, and A. Serani, Comparison of multi-fidelity approaches for military vehicle design, in AIAA AVIATION 2020 FORUM, 2020, p. 3158.

[123] F. A. Viana, R. T. Haftka, and L. T. Watson, Efficient global optimization algorithm assisted by multiple surrogate techniques, Journal of Global Optimization, vol. 56, no. 2, pp. 669-689, 2013.

[124] R. Lam, D. L. Allaire, and K. E. Willcox, Multifidelity optimization using statistical surrogate modeling for non-hierarchical information sources, in 56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, p. 0143, 2015.

[125] S. Takeno, H. Fukuoka, Y. Tsukada, T. Koyama, M. Shiga, I. Takeuchi, and M. Karasuyama, Multi-fidelity bayesian optimization with max-value entropy search, arXiv preprint arXiv:1901.08275, 2019.

[126] K. Williams and C. E. Rasmussen, Gaussian processes for machine learning. MIT press Cambridge, MA, vol. 2, no. 3, 2006.

[127] L. Le Gratiet and J. Garnier, Recursive co-kriging model for design of computer experiments with multiple levels of fidelity, International Journal for Uncertainty Quantification, vol. 4, no. 5, 2014.

[128] K. Cutajar, M. Pullin, A. Damianou, N. Lawrence, and J. Gonzalez, Deep gaussian processes for multifidelity modeling, arXiv preprint arXiv:1903.07320, 2019.

[129] Brochu, V. M. Cora, and N. de Freitas, A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning, 2010.

[130] P. I. Frazier, A tutorial on bayesian optimization, arXiv preprint arXiv:1807.02811, 2018.

[131] J. Mockus. The Bayesian approach to global optimization. In R. Drenick and F. Kozin, editors, System Modeling and Optimization, volume 38, pages 473-481. Springer Berlin Heidelberg, 1982.

[132] J. Mockus. Application of Bayesian approach to numerical methods of global and stochastic optimization. J. Global Optimization, 4(4): pages 347-365, 1994.

[133] Brochu, T. Brochu, and N. de Freitas. A Bayesian interactive optimization approach to procedural animation design. In Eurographics ACMSIGGRAPH Symposium on Computer Animation, 2010

[134] R. Martinez-Cantin, N. de Freitas, E. Brochu, J. Castellanos, and A. Doucet. A Bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot. Autonomous Robots, 27(2):93-103, 2009.

[135] E. Rasmussen and C. K. I. Williams. Gaussian Processes for Machine Learning. MIT Press, Cambridge, Massachusetts, 2006.

[136] Matérn. Spatial Variation. Springer-Verlag, 2nd edition, 1960.

[137] M. L. Stein. Interpolation of Spatial Data: Some Theory for Kriging. Springer Series in Statistics. Springer, 1999.

[138] M. G. Genton. Classes of kernels for machine learning: A statistics perspective. Journal of Machine Learning Research, 2:299-312, 2001.

[139] M. J. Sasena. Flexibility and Efficiency Enhancement for Constrained Global Design Optimization with Kriging Approximations. PhD thesis, University of Michigan, 2002.

[140] T. J. Santner, B. Williams, and W. Notz. The Design and Analysis of Computer Experiments. Springer, 2003.

[141] Lizotte. Practical Bayesian Optimization. PhD thesis, University of Alberta, Edmonton, Alberta, Canada, 2008.

[142] M. Frean and P. Boyle. Using Gaussian processes to optimize expensive functions. In W. Wobcke and M. Zhang, editors, AI 2008: Advances in Artificial Intelligence, volume 5360 of Lecture Notes in Computer Science, pages 258-267. Springer Berlin Heidelberg, 2008.

[143] J. Kushner. A new method of locating the maximum of an arbitrary multipeak curve in the presence of noise. J. Basic Engineering, 86:97 - 106,1964.

[144] Torn and A. Zilinskas. Global Optimization. Springer-Verlag, 1989.

[145] R. Jones. A taxonomy of global optimization methods based on response surfaces. J. Global Optimization, pages 345-383, 2001.

[146] J. Mockus, V. Tiesis, and A. Zilinskas. Toward Global Optimization, volume 2, chapter The Application of Bayesian Methods for Seeking the Extremum, pages 117-128. Elsevier, 1978.

[147] J. Sacks, W. J. Welch, T. J. Welch, and H. P. Wynn. Design and analysis of computer experiments. Statistical Science, 4(4): pages 409-423, 1989.

[148] C. Audet, J. Jr, Dennis, D. W. Moore, A. Booker, and P. D. Frank. Surrogate-model-based method for constrained optimization. In AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 2000.

[149] P. Boyle. Gaussian Processes for Regression and Optimisation. PhD thesis, Victoria University of Wellington, Wellington, New Zealand, 2007.

[150] T. Bartz-Beielstein, C. Lasarczyk, and M. Preuss. Sequential parameter optimization. In Proc. CEC-05, 2005

[151] D. Huang, T. T. Allen, W. I. Notz, and N. Zheng. Global optimization of stochastic black-box systems via sequential Kriging meta-models. J. Global Optimization, 34(3): pages 441-466, March 2006.

[152] Hutter, H. H. Hoos, K. Leyton-Brown, and K. P. Murphy. An experimental investigation of model-based parameter optimisation: SPO and beyond. In Proc. GECCO'09, 2009.

[153] Hutter. Automating the Conguration of Algorithms for Solving Hard Computational Problems. PhD thesis, University of British Columbia, Vancouver, Canada, August 2009.

[154] B. Settles. Active learning literature survey. Computer Science Technical Report 1648, University of Wisconsin-Madison, January 2010.

[155] O'Hagan, A markov property for covariance structures, Statistics Research Report, vol. 98, p. 13, 1998.

[156] P. Hennig and C. J. Schuler. Entropy search for information-efficient global optimization. JMLR, 13:1809–1837, 2012.

[157] J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In Proc. NIPS, pages 918–926, 2014.

[158] P. Hennig and C. J. Schuler. Entropy search for information-efficient global optimization. Journal of Machine Learning Research, 13, 2012.

[159] Y. Zhang, T. N. Hoang, B. K. H. Low, and M. Kankanhalli, Information-based multifidelity bayesian optimization, in NIPS Workshop on Bayesian Optimization, 2017.

[160] D. Huang, T. T. Allen, W. I. Notz, and R. A. Miller, Sequential kriging optimization using multiple-fidelity evaluations, Structural and Multidisciplinary Optimization, vol. 32, no. 5, pp. 369-382, 2006

[161] Little, Max. Random sampling. 2019.

[162] Aronow, Peter & Miller, Benjamin. Learning from Random Samples. 2019.

[163] Gooch, Jan. Random Sampling. 2011.

[164] Ning, Jianhui. Randomized quasi-random sampling/importance resampling. 2020.

[165] Gan, Guojun & Valdez, Emiliano. Latin Hypercube Sampling. 2019.

[166] Iman, Ronald. Latin Hypercube Sampling. 1999.

[167] Hankin, David & Mohr, Michael & Newman, Ken. Adaptive sampling. 2019.

[168] Ngwako, Abby, Adaptive Learning Survey. 2019.

[169] Forrester, A. Sobester, and A. Keane, Engineering Design via Surrogate Modelling: A Practical Guide. Wiley, 2008.

[170] K. Cutajar, M. Pullin, A. Damianou, N. Lawrence, and J. Gonzalez, Deep gaussian processes for multi-fidelity modeling, arXivpreprint arXiv:1903.07320, 2019.

[171] Rosenbrock, An automatic method for finding the greatest or least value of a function, The Computer Journal, vol. 3, no. 3, pp. 175-184, 1960.

[172] D. E. Bryson, A unified, multifidelity quasi-newton optimization method with application to aero-structural design, Ph.D. dissertation, University of Dayton, 2017.

[173] March, Andrew & Willcox, Karen. Provably Convergent Multifidelity Optimization Algorithm Not Requiring High-Fidelity Derivatives. AIAA Journal. 50. Pages 1079-1089, 2012.

[174] Adhams, T.P., Cassady, A.M., MacLean, M. and Holden, M.S., Experimental Studies of the Aerothermal Characteristics of the Project Orion CEV Heat Shield in High Speed Transitional and Turbulent Flows, AIAA Paper 2009-0677, 47th AIAA Aerospace Sciences Meeting, Orlando, FL, USA, January 5-8 2009.

[175] Berger, K.T., Aerothermodynamic Testing of the Crew Exploration Vehicle at Mach 6 and Mach 10, Journal of Spacecraft and Rockets, Vol. 46, No. 4, pp.758-765, July – August 2009.

[176] Berry, S.A., Horvath, T.J., Lillard, R.P., Kirk, B.S. and Cassady, A.M., Aerothermal Testing for Project Orion Crew Exploration Vehicle, AIAA Paper 2009-3842, 41st AIAA Thermophysics Conference, San Antonio, TX, USA, June 22-25 2009.

[177] E.H. Hirschel, C. Weiland, Selected Aerothermodynamics Design Problems of Hypersonic Flight Vehicles, Springer, 2009.

[178] Anderson, J.D. Hypersonic and High Temperature Gas Dynamics. American Institute of Aeronautics and Astronautics, 2006.

[179] Butcher, J. Numerical Methods for Ordinary Differential Equations. John Wiley & Sons, Ltd, 2008.

[180] Griffiths, D. and Higham, D. Numerical Methods for Ordinary Differential Equations: Initial Value Problems. Springer, 2010.

[181] U.S. Standard Atmosphere, NASA TM X-74335, 1976.

[182] Dorrance, W. H. Viscous Hypersonic Flow. Theory of Reacting and Hypersonic Boundary Layers, McGraw-Hill, New York, 1962

[183] Paolo Scaramuzzino, Hypersms Aerothermodynamics tool, MATLAB Central File Exchange. Retrieved August 16, 2020.

[184] Sutton, George P. Rocket Propulsion Elements: An introduction to the Engineering of Rockets. New York, Wiley, 1992.

[185] https://www.nasa.gov/exploration/systems/orion/index.html. Accessed in September 2020.