Politecnico di Torino

Master's Degree in Aerospace Engineering

Master's Degree Thesis

Optimal Control via Direct Methods for Low-Thrust Interplanetary Trajectories

Candidate:
Tommaso Gaspardino

Supervisor:
Prof. Manuela Battipede

TAS-I Advisors:
Dr. Giorgio Fasano
Dr. Giorgio Pesenti

December 2024

Acknowledgments

Giunto alla conclusione di questo lavoro di tesi e di questo percorso universitario, mi preme ringraziare le persone che mi hanno affiancato e sostenuto in questo lungo viaggio.

Innanzitutto, voglio ringraziare la prof. Manuela Battipede per la disponibilità ed il sostegno dimostratimi. Voglio inoltre dedicare uno speciale ringraziamento al Dr. Giorgio Fasano per avermi seguito e supportato continuamente durante lo sviluppo e la stesura di questa tesi, instaurando un dialogo ed un confronto costante, indispensabili per la riuscita del lavoro e fonti di consigli ed insegnamenti che porterò con me nel mio percorso futuro.

Ringrazio il Dr. Giorgio Pesenti per il prezioso aiuto fornito nello sviluppo operativo di questo lavoro. Dal nostro continuo confronto ho avuto modo di imparare moltissimo. Un ringraziamento speciale anche a tutti ragazzi dell'ufficio di Mission Analysis e Operations di TAS-I per avermi fatto sentire parte integrante del gruppo sin dal primo giorno in azienda.

Questa tesi rappresenta però soltanto il taglio del traguardo di una lunga ed intensa corsa durata cinque anni, resa più bella e meno faticosa dalle persone che mi hanno accompagnato lungo tutto il percorso.

Un grande ringraziamento va innanzitutto alla mia famiglia. Grazie mamma Cristina, grazie per la tua immensa pazienza e la tua smisurata fiducia nelle mie potenzialità, ma soprattutto grazie per avermi insegnato la fisica prima ancora che nascessi. Il tuo grande sacrificio nel portarmi in grembo a lezioni ed esami di fisica teorica alla fine ha dato i suoi frutti.

Grazie papà Fiorenzo, il vero ingegnere della mia famiglia. Grazie per avermi insegnato il valore dell'impegnarsi e del lavorare sodo, ma soprattutto grazie per avermi trasmesso quel pizzico di follia indispensabile a vivere a pieno la vita.

Grazie Matteo, negli ultimi anni mi hai dimostrato come ci sia sempre modo di rimettersi in gioco e ribaltare ogni previsione, la tua grande determinazione e la tua costanza sono e saranno per sempre uno stimolo a dare il meglio di me.

Grazie Andrea, sei sempre stato per me un esempio per la tua grande semplicità e profonda sensibilità. Ormai anche tu stai diventando grande, ma che bello divertirci ancora come fossimo quei bambini scalmanati a tirare pallonate al portone del garage di Rocco!

Grazie Chiara, sei e rimarrai per sempre la mia piccola sorellina. Ho una grande responsabilità nei tuoi confronti, che mi motiverà per sempre a dimostrare di essere il fratellone più forte del mondo.

Grazie a tutti gli amici che mi hanno accompagnato e sostenuto in questi anni, da quelli storici di una vita a quelli più recenti, partendo dalla cintura di Torino ed andando fino a Parigi passando per Lecce e la Valle d'Itria. Siete tantissimi e mi sento molto fortunato. Avrò modo di ringraziarvi personalmente uno ad uno in questi giorni di festa.

Una menzione speciale è però obbligatoria alla persona che ha condiviso con me questo percorso come nessun altro. Grazie David, abbiamo affrontato questi anni di università insieme in tutto e per tutto, dalle aule universitarie alle discoteche, dal Fante alle Panche. Sono stati gli anni più belli della mia vita, soprattutto grazie a te.

Ed infine, grazie Ire. Sei l'evento più bello ed inaspettato che mi sia capitato in questi anni universitari. Grazie per avermi sempre offerto una prospettiva diversa dalla mia, spesso diametralmente opposta, e per avermi insegnato l'importanza di saper osservare ed ascoltare gli altri. Effettivamente è già il secondo traguardo che tagliamo insieme, ma non credo sia solamente fortuna, credo proprio che siamo una squadra vincente!

A tutti voi, nuovamente, un ringraziamento sincero.

Settimo Torinese, Dicembre 2024.
Tommaso.

Abstract

The increasing interest in large interplanetary missions, coupled with the need for efficient propellant consumption to achieve substantial DeltaVs, has driven the increasing adoption of electric propulsion systems. While these high specific impulse and low-thrust technologies significantly enhance mission efficiency, they also introduce substantial design challenges. These challenges necessitate the use of more complex and costly trajectory optimization models and tools compared to the traditional methods employed for impulsive maneuver mission analysis.

Within this thesis a flexible low-thrust interplanetary trajectory optimization tool is proposed and developed, designed for seamless integration and implementation in industrial contexts, particularly for preliminary mission analysis studies. To achieve this objective, an optimization approach using direct methods and collocation techniques is adopted.

The mathematical model describing the optimal control problem for interplanetary low-thrust trajectories is presented, with a focus on the reduction of the continuous problem to a finite-dimension nonlinear programming (NLP) problem formulation through direct collocation transcription.

Experimental analyses of the problem are conducted. The model is implemented employing Sequential Quadratic Programming (SQP) algorithms that leverage the sparse structure of the large-scale problem at hand. An environmental analysis is proposed to compare the performance of the *WORHP* solver with the *Matlab Optimization Toolbox*, with a particular focus on the effectiveness of exploiting sparsity in problems of this scale. This analysis aims to demonstrate the importance and necessity of such specific tools in an industrial context to effectively address the problem of low-thrust trajectories optimization.

The optimization tool developed for the low-thrust interplanetary rendezvous maneuver problem is presented. The architecture and operation of the tool are described, with a focus on the global optimum search through a multi-start approach and a refinement of the optimal solution to enhance its accuracy.

Several case studies of interest are described and analyzed to test the developed tool, with a particular focus on missions to Mars and Apophis, either accounting for constant allowable thrust models (e.g., nuclear electric propulsion) or sun-distance-dependent thrust levels (e.g., solar electric propulsion).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Interplanetary missions have always been of scientific interest, since they provide invaluable insights, significantly enhancing our understanding of the solar system and beyond. The key areas of interest related to these missions are summarized below.

- **The origin and evolution of the solar system**: by examining various celestial bodies, scientists can reconstruct the sequence of events that led to the formation of the Sun and its planets. Missions that land on or orbit these bodies allow for detailed analysis of their surface geology and composition, providing insights into their geological history and potential resources.
- **Atmospheric composition and dynamics**: investigating planetary atmospheres provides crucial information on their composition, temperature, and dynamics, which is essential for scientific research and planning future landings.
- **Search for extraterrestrial life**: many interplanetary missions aim to detect signs of life or conditions that could support life, focusing on the presence of water, organic molecules, and other potential indicators of past or present life.
- **Technological advancements**: the advancement of technologies for interplanetary missions frequently results in substantial progress in space exploration and other disciplines. These innovations can have practical applications on Earth, such as advancements in solar power generation and energy storage technologies.

The exploration of our solar system has been marked by significant milestones (see Britannica, 2024[6]), including the first lunar landing by Apollo 11 in 1969, the detailed exploration of Mars by Viking 1 in 1976, and the comprehensive surveys of the outer planets by the Voyager missions. In more recent times, missions such as the ESA's Rosetta and NASA's Dawn have yielded invaluable

Propulsion System	**Specific Impulse [s]**	**Thrust Level [N]**
Liquid Monopropellant (CP)	200-250	0.01-100
Liquid Bipropellant (CP)	300-450	0.01-10^7
Solid Propellant (CP)	200-300	1-10^6
Hybrid Propellant (CP)	250-350	1-10^6
Resistojets (EP)	200-350	0.2-0.3
Arcject (EP)	400-1000	0.2-1
Ion Thrusters (EP)	2000-5000	<0.2
Hall Thrusters (EP)	1500-2000	<2
Pulsed Plasma Thrusters (EP)	600-2000	<0.01
MPD Thrusters (EP)	2000-5000	<2

Table 1.1: Main characteristics of propulsion systems

insights into comets and asteroids, thereby enhancing our understanding of the early solar system.

A significant advancement in space exploration has been the development and utilisation of electric propulsion systems, as reported in Jahn, 1968[21] and ESA Science & Technology, 2019[14]. In contrast to conventional chemical propulsion (CP), electric propulsion (EP) provides a higher specific impulse, thereby facilitating more efficient fuel utilization, accepting the trade-off of significantly lower thrust levels. A comparative analysis of the principal characteristics of chemical and electric propulsion systems is presented in Table 1.1. Several types of electric thrusters have already been employed in the context of solar system space missions. This overview examines the primary types of electric thrusters, highlighting their strengths and weaknesses, and evaluates their suitability for interplanetary missions.

- **Electrothermal Thruster**: Electrothermal thrusters employ electric energy to heat a propellant, which is subsequently expelled to generate thrust. The aforementioned thrusters are further classified as resistojets and arcjets. Resistojets employ electric resistance heaters to heat the propellant, whereas arcjets utilize an electric arc to achieve higher temperatures. The principal advantages of electrothermal thrusters are their straightforward design, their greater thrust output in comparison to other electric thrusters, and their relatively low power requirements. However, they exhibit lower efficiency and specific impulse values than ion and Hall effect thrusters. Consequently, their suitability for interplanetary missions is limited, rendering them more appropriate for short-duration missions or as auxiliary propulsion.

- **Gridded Ion Thruster (GIT)**: Gridded ion thrusters employ electric fields to accelerate positive ions through a grid structure. These thrusters

are distinguished by their high efficiency and specific impulse (Isp), rendering them particularly well-suited for long-duration missions. The principal advantage of GITs is their capacity to attain high efficiency with minimal propellant consumption, which is a vital consideration for interplanetary travel. However, their low thrust output and the complexity of the grid system represent significant drawbacks. Despite these challenges, GITs have been successfully employed in missions such as NASA's Dawn, thereby demonstrating their viability for interplanetary exploration.

- **Hall Effect Thruster (HET)**: Hall effect thrusters employ magnetic fields to accelerate ions within an annular channel. They offer an optimal balance between thrust and efficiency, and their robust design contributes to their reliability. Hall effect thrusters (HETs) are distinguished by their capacity to generate a higher thrust than ion thrusters while maintaining a reasonable level of efficiency. The primary limitation of HETs is the necessity for effective thermal management to prevent overheating. Their balanced performance and reliability render them suitable for interplanetary missions, as evidenced by their deployment in the SMART-1 mission by ESA.

- **Pulsed Plasma Thruster (PPT)**: Pulsed plasma thrusters operate by employing electric discharges to vaporize and accelerate propellant. These thrusters are characterized by a simple and compact design, coupled with low power consumption, rendering them well-suited for deployment on small satellites and for facilitating minor orbital adjustments. However, PPTs have relatively low efficiency and are susceptible to electrode wear, which restricts their long-term viability. In light of these constraints, PPTs are not typically deemed an optimal choice for primary propulsion in interplanetary missions. However, they do offer a viable solution for precise maneuvers.

- **Magnetoplasmadynamic Thruster (MPD)**: Magnetoplasmadynamic thrusters employ magnetic fields to accelerate plasma, resulting in high thrust and the potential for high-power applications. MPDs are capable of producing significant thrust, which is advantageous for missions requiring rapid acceleration. However, they require substantial power and present technical complexities that currently limit their practical use. While MPDs hold promise for future interplanetary missions, further technological advancements are necessary to overcome their current limitations.

Among the various types of electric thrusters, Gridded Ion Thrusters and Hall Effect Thrusters are the most suitable for interplanetary missions. Their high efficiency and low propellant consumption make them ideal for long-duration operations and significant velocity changes (delta-v) required for interplanetary travel. Below are several examples of electric thruster models used in various interplanetary missions.

The ion thrusters utilized in the NASA missions, namely the Deep Space 1 mission to Comet Borely and the Dawn mission to Ceres and Vesta, were of the electrostatic gridded ion engine NSTAR type, exhibiting a thrust of 90 mN. A Hall Plasma Thruster SPT-100 from SNECMA (100 mN thrust) was employed in the Smart 1 mission to the Moon. An advanced gridded electrostatic ion thruster (Figure 1.1) was employed on the Hayabusa1 and 2 missions to near-Earth asteroids. Finally, The BepiColombo spacecraft currently employs QinetiQ T6 ion thrusters (Figure 1.2), representing the most powerful ion engine array ever operated in space, with a combined thrust of up to 290 mN.

Figure 1.1: Artist's impression of Hayabusa2 firing its ion thrusters. Credits: NASA Science, 2024[32]

Figure 1.2: Artist's impression of BepiColombo in Cruise Mode, firing the T6 Thrusters. Credits: ESA Science & Technology, 2024[13]

Nevertheless, the implementation of electric propulsion is not without its inherent difficulties. The low thrust levels inherent to these systems require longer mission durations to achieve the desired velocity changes. Furthermore, the considerable power demands frequently necessitate the deployment of sizable solar arrays or alternative energy sources. Additionally, the continuous thrust provided by electric propulsion introduces additional complexity in trajectory planning and optimization, necessitating the use of sophisticated algorithms and extensive computational resources. The design process is based on modeling and optimization methods related to space engineering applications (see Fasano and Pintér, 2014[16]), by commencing with the formulation of an optimal control problem, the objective of which is to minimize fuel consumption (or minimize the time of flight) while meeting the mission constraints. This necessitates the resolution of challenging problems, which account for the nonlinear dynamics of

the spacecraft and thrust profile. For these reasons, the optimization of low-thrust trajectories poses considerable computational challenges. The nonlinear nature of the problem, when considered alongside the necessity for high precision over extended mission durations, necessitates the utilisation of substantial computational resources. It is imperative that algorithms be capable of efficiently handling large-scale problems, which often involve thousands of variables and constraints. To address these challenges, a variety of techniques, including direct and indirect methods, as well as hybrid approaches, are employed. Despite recent advances, the high sensitivity to initial conditions and the necessity for iterative refinement can result in lengthy computation times and demand the use of robust, high-performance algorithms.

In light of the aforementioned context, the objective of this thesis is to conduct an analysis of direct methods (see Chapter 2) for optimizing low-thrust trajectories, with a particular focus on aligning these methods with the requirements of industry. The thesis is structured as follows.
Chapter 2 provides an overview of optimization theory and its applications in space technology, with a particular emphasis on the application of these principles through mission analysis. The following sections of this chapter present an overview of the general formulations of optimal control problems and the various numerical techniques for their solution. Subsequently, the optimization approach utilized in this thesis is delineated in detail.

Chapter 3 presents a detailed application of optimal control to the low-thrust interplanetary trajectory problem. Initially, the chapter focuses on the dynamic models that are specific to this problem. It then describes the problem statement for trajectory optimization.

Chapter 4 presents the findings of a series of experimental analyses conducted in an industrial environment to assess the performances of commercially available solvers for addressing such complex large-scale optimization problems. Subsequently, the structure of the developed optimization tool is presented, followed by a validation on an Earth-Mars trajectory problem that has already been analyzed in literature (see Casanova-Álvarez, Navarro-Medina, and D.Tommasini, 2024[10]). The chapter concludes presenting the solution refinement process performed on the optimal solutions found, by increasing the model fidelity in order to decrease the approximation of the generated trajectories.

The tool is applied to new case studies of interest in Chapter 5, specifically to generate and study high-precision low-thrust trajectories for Earth-Mars and Earth-Apophis transfers. Indeed, the latter is of particular interest at the industrial level in recent times due to the ongoing design process of missions to Apophis (e.g., Ramses mission, see ESA Space & Safety, 2024[15]). This is because the approach of this asteroid to Earth in the coming years will result in its flyby around Earth in 2029. Consequently, space agencies and industries are engaged in feasibility studies of electrically propelled trajectories for missions to approach and observe the asteroid. The chapter then concludes outlining of the potential future directions opened by the thesis work, focusing on the enhancement of the

developed tool and its extension to more complex scenarios.

In conclusion, the Concluding Remarks section offers comments and conclusions regarding the work completed, emphasizing the achievements and potential opportunities associated with the techniques employed within the developed optimization tool.

Figure 1.3: Depiction of the ESA Ramses mission to Apophis. Credits: ESA Space & Safety, 2024[15]

Chapter 2

Optimization Theory and Space Applications

This thesis addresses complex optimization problems that are non-convex and necessitate global optimization strategies. Based on Becerra, 2013[2], Betts, 2010[4], Bryson and Ho, 1969[7] and Morante, Rivo, and Soler, 2021[29] , this chapter provides a brief overview of optimization, beginning with a brief introduction to optimization problems and the various strategies currently employed. It finally concludes with a detailed presentation of the strategies and methods adopted in this research.

2.1 Introduction to the Optimization Theory

Optimization is a systematic process that seeks to identify the most optimal solution, typically the highest or lowest value, for a given mathematical model or function while adhering to specific constraints. The objective is to find out the most advantageous values for variables that result in optimal outcomes, such as maximizing profit, minimizing costs, improving efficiency, or achieving any other desired objective.

Optimization problems are typically categorized based on several key characteristics, including:

- **Dimensionality**: Problems may be classified as either finite-dimensional, which involves a real or integer decision variables, or infinite-dimensional, where decision variables are functions.

- **Convexity**: In convex problems, if a minimum (or maximum) exists it is a global one, making them generally easier to solve. Non-convex optimization problems are characterized by the presence of local minima, which significantly complicate the search for the optimum (Figure 2.1). It is worthy of note that the majority of space applications deal with non-convexity issues, particularly in the context of trajectory optimization problems.

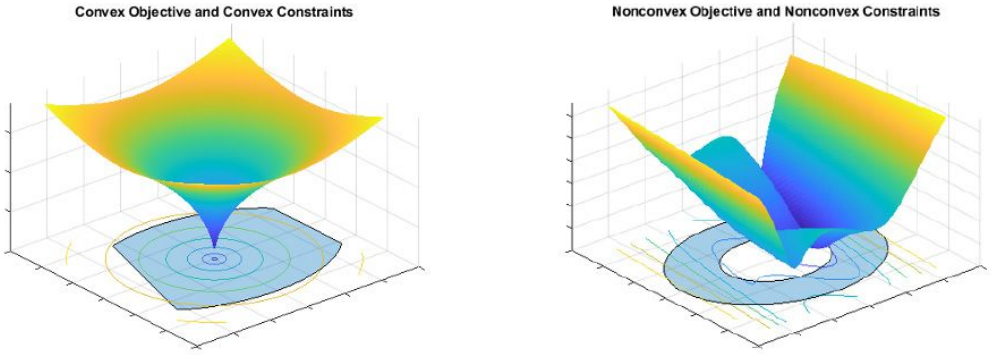

Figure 2.1: Visualization of Convex and Non-Convex problems. Credits: MathWorks.

- **Single/Multiple objectives**: Single-objective optimization is the process of minimizing or maximizing a single objective function. In multi-objective optimization, the objective is to balance conflicting objectives and identify a set of solutions, known as the Pareto front, that represent the optimal trade-off between the conflicting objectives. A solution is defined as Pareto optimal if no other solution can be identified that improves one objective without simultaneously worsening another.

- **Type of cost function and constraints**: The form of the cost function and constraints must be defined. In linear programming (LP), the objective functions and constraints are linear. Nonlinear programming (NLP) encompasses the inclusion of non-linear relationships within either the objective function or the constraints. Quadratic Programming (QP) addresses quadratic objective functions and linear or quadratic constraints.

- **Type of variables** Variables may be real numbers, integers (as in Integer Programming, IP), or a combination of both (as in Mixed Integer Programming, MIP).

2.2 The Optimal Control Problem

An optimal control problem is a mathematical approach focused on the best possible management of dynamic systems. These systems consist of interconnected variables or states that change over time, influenced by control actions. The goal of optimal control is to identify a control strategy (i.e. a set of guidelines or methods for influencing the system) so that a specific performance measure is optimized.

This performance measure is represented by an objective function, which evaluates the efficiency of the system's behavior. This could involve maximizing or minimizing a particular outcome, considering the impact of control actions,

system dynamics, and external factors. The optimal control problem usually includes constraints on the system states, control actions, and other relevant variables.

A widely used formalism for the problem statement is the *Bolza formulation*. The function that needs to be minimized (or maximized) is known as the objective function J_B.

$$J_B = \varphi\left(\boldsymbol{x}_{(0)_+}, \boldsymbol{x}_{1_-}, \ldots, \boldsymbol{x}_{(p-1)_+}, \boldsymbol{x}_{p_-}, t_{(0)_+}, t_{1_-}, \ldots, t_{(p-1)_+}, t_{p_-}\right) + \sum_j \int_{t_{(j-1)_+}}^{t_{j_-}} \phi[\boldsymbol{x}(t), \boldsymbol{u}(t), t]\, dt, \quad j = 1, \ldots, p$$

Where:

- $\boldsymbol{x}(t)$ is the state variable vector;
- $\boldsymbol{u}(t)$ is the control variable vector;
- t is the independent variable;
- $(x_{0_+}, x_{1_-}), \ldots, (x_{(p-1)_+}, x_{p_-})$ are the trajectory arcs;

The state variables adhere to the state equations, which model the system dynamics:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}[\boldsymbol{x}(t), \boldsymbol{u}(t), t]$$

The system is also subject to boundary conditions:

$$\chi\left(\boldsymbol{x}_{(j-1)_+}, \boldsymbol{x}_{j_-}, t_{(j-1)_+}, t_{j_-}\right), \quad j = 1, \ldots, p$$

And possible additional constraints of the following type:

$$\boldsymbol{g}[\boldsymbol{x}(t), \boldsymbol{u}(t), t] \leq 0$$

2.3 Main Numerical Optimization Methods

There are various ways to approach optimization problems and to solve them, this section aims to introduce the main methods most commonly used, focusing particularly on the optimal control problem for low-thrust trajectories.

The importance of an initial analysis of the various techniques available is crucial, as the choice of a certain approach to the problem determines its flexibility, robustness, optimality and automation.

Historically, low-thrust trajectory optimization problems have been framed as continuous optimal control problems (COCP). Generally, there are two main types of approaches: analytical and numerical. Analytical methods yield closed-form solutions for the optimal trajectory, but these solutions can be find only in special cases, making them impractical for most spacecraft trajectory optimization problems. Consequently, most researchers have focused on numerical methods to tackle more complex and realistic problems. For this reason, this section will focus on giving an overview of the state of the art of numerical methods alone to solve the problem at hand.

2.3.1 Approaches

Numerical approaches can be divided into three well-established methods: indirect methods and direct methods. These approaches can be described as follows:

- **Indirect Methods**: The indirect approach (see e.g., Casalino and Colasurdo, 2013[9]) is employed in order to solve the *multipoint boundary value problem (MPBVP)* that arises from the application of Pontryagin's Maximum Principle (PMP).The PMP outlines the first-order necessary conditions that an optimal solution must meet. Its derivation necessitates the determination of the states and costates (see Bryson and Ho, 1969[7]), which must comply with the Euler-Lagrange equation. Significantly, the minimum principle permits the continuous control to be expressed as a function of the state and costate at each instant, either explicitly or numerically. Moreover, a set of supplementary constraints, specifically those of transversality and complementary conditions, must also be fulfilled. Furthermore, Bellman's dynamic programming method (see Bellman, 2013[3]) is classified among the indirect methods. Originally developed for finite problems, it was later extended to continuous problems under very stringent assumptions. However, due to these stringent assumptions, it has seen limited application in the field of trajectory optimization.

- **Direct Methods**: The core concept of direct methods is to convert the COCP into a nonlinear programming problem (NLP), where the objective function is optimized directly (see Becerra, 2013[2]). This conversion process involves discretizing the control variables over a time grid. The objective of an NLP problem is to identify a vector of unknown decision variables

that satisfy a set of nonlinear constraints,which include both equality and inequality conditions. An optimal solution to the NLP must meet the first-order necessary optimality conditions, known as the Karush-Kuhn-Tucker (KKT) conditions (see M.Minoux, 1986[26]).

2.3.2 Techniques to Involve System Dynamics

As space trajectory optimization problems are typically addressed through indirect or direct methods, the primary techniques through which the equations of dynamics are applied within such methods are outlined below.

- **Single Shooting**: The trajectory is computed using numerical propagation methods (Ordinary differential equations (ODEs) solution) starting from t_0 until the final time t_f. In this scenario, the initial state are variables that need to be determined and so optimized, with boundary constraints applied at the conclusion of the integration.
- **Multiple Shooting**: The time interval $[t_0, t_f]$ is divided into $N+1$ subintervals. The trajectory is propagated over each subinterval $[t_i, t_{i+1}]$, with the initial values of the state for each subinterval being unknowns that must be determined. Additionally, continuity conditions must be enforced at the boundaries between each subinterval, as well as boundary constraints applied at the end of the entire trajectory.
- **Collocation**: The states are discretized along a predefined time grid, meaning they are only known at specific discrete points. The system's governing equations are converted into discrete defect constraints (see Section 2.4), which link the values at the start of each subinterval to those at the end. In this case the ODEs are not propagated but approximated by various types of quadrature rules.

2.3.3 Solving Methods

It is now necessary to define the manner in which these NLPs and MPBVPs can be resolved in order to identify optimal solutions. The principal methods of resolution are outlined below.

- **Deterministic Methods (Gradient-Based)**: In a gradient-based method, an initial estimate is made for the unknown decision vector $\boldsymbol{y}$. At the i-th iteration, a search direction $\boldsymbol{p}_i$ and a step length α_i are determined. The search direction indicates the direction along which to adjust the current value $\boldsymbol{y}_i$, while the step length specifies the magnitude of this adjustment.

The update from $\boldsymbol{y}_i$ to $\boldsymbol{y}_{i+1}$ is given by: $\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + \alpha_i \boldsymbol{p}_i$. Iterations continue until the Karush-Kuhn-Tucker (KKT) conditions are satisfied. To compute the search direction, these methods require the user to provide gradient information for the constraints (*Jacobian Matrix*) and, if necessary, the objective function. The most commonly used solving algorithms are cassified as Sequential Quadratic Problems (SQP) or Interior Point Methods (IP) and are implemented in widely known solvers such as MatLab Optimization Toolbox (see MathWorks, 2024[27]), SNOPT (see Gill, Murray, and Saunders, 2005[18]) and WORHP (see Büskens and Wassel, 2013[8]).

- **Heuristic Methods**: The search is conducted using stochastic, metaheuristic or specific heuristic methods that do not rely on gradient information. One of the most well-known types of heuristics is evolutionary algorithms. These algorithms begin by creating a group of potential solutions, known as a population. This population is then repeatedly altered using a set of random rules until a stopping criterion is reached. The candidate with the lowest cost is considered the solution to the problem. Popular stochastic rules include genetic algorithms (GA), which mimic genetic evolutionary processes, and particle swarm optimization (PSO), which is inspired by the behavior of animal swarms.

- **Hybrid Methods**: Hybrid methods combine rules that leverage gradient information with those based on heuristic searches to iteratively improve a solution or a set of potential solutions. Gradient information is used to meet the constraints, while heuristic rules are applied to efficiently explore huge design spaces or manage integer variables. These methods are typically combined in a two-loop system. In the outer loop, the heuristic solver adjusts a subset of decision variables, while in the inner loop, the remaining design parameters are optimized using the gradient-based method.

2.4 Focus on Adopted Optimization Methods

The indirect approach ensures the satisfaction of first-order optimality conditions and provides theoretical insights into the problem's physical and mathematical characteristics. However, it requires explicit derivations of costate and control equations. Additionally, the trajectory can be very sensitive to initial costate guesses, which are not intuitive variables, and the need to reformulate the problem for different state variables, constraints, and dynamics can complicate its application, especially in highly constrained or automated spacecraft trajectory optimization problems.

On the other hand, direct methods do not require deriving first-order necessary conditions and are easier to initialize due to a larger convergence domain

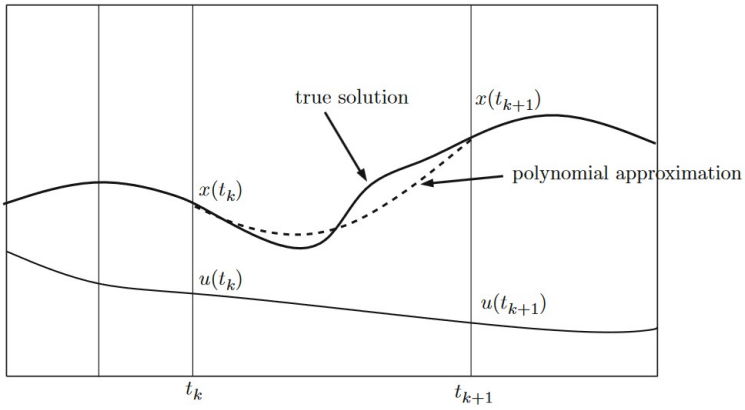

Figure 2.2: Concept visualization of the collocation method

and the intuitive meaning of optimization variables. These methods can handle complex control or state constraints and do not require prior knowledge of the sequence of free and constrained arcs. Nonlinear optimization problems resulting from direct collocation methods can be very large, potentially involving hundreds to tens of thousands of variables and constraints. However, this issue can be handled by exploiting the sparsity structures of the Jacobian matrix of constraints and the objective function gradient of the associated NLP problem, and the availability of efficient methods and software for their solution, as discussed in Chapter 3. Although they may not consistently yield the optimal solution, direct methods at least provide a suboptimal solution that can be extremely useful in an industrial setting, particularly during the initial design phases of new missions.

In view of the previous considerations, this thesis work was undertaken with the objective of developing an industrially functional tool for an initial generation of optimal high-fidelity trajectories for new interplanetary missions in a relatively simple, robust, and effective manner. To this end, direct methods, in particular through collocation techniques, were employed (see Becerra, 2013[2]).
These techniques are based on the discretization of the continuous time domain in a grid of finite points, whereby polynomial functions are employed to approximate the differential equations of dynamics, as shown in Figure 2.2.

In general, the dynamical equations can be written as a set of ordinary differential equations (ODEs), expressed as:

$$\begin{aligned} \boldsymbol{x}(t) &= \boldsymbol{F}[\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{p}, t], \quad t \in [t_0, t_f] \\ \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}[\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{p}, t] \end{aligned} \tag{2.1}$$

where $\boldsymbol{x} : [t_0, t_f] \to \mathbb{R}^{n_x}$ is the state vector function, $\boldsymbol{u} : [t_0, t_f] \to \mathbb{R}^{n_u}$ is the control vector function, $\boldsymbol{p} \in \mathbb{R}^{n_p}$ is a vector of static parameters which are independent of t, and $t \in [t_0, t_f] \subset \mathbb{R}$ is an independent variable, which is usually time.

In particular, we firstly focus on the utilization of the *trapezoidal method* for the initial search for optimal solutions, since it is the easiest and computationally fastest collocation method. It is based on a quadratic interpolation polynomial $\tilde{\boldsymbol{x}}$, of degree $M = 2$ over each discretized interval $[t_k, t_{k+1}], j = 0, ..., N-1$ such that:

$$\tilde{\boldsymbol{x}}(t) = \boldsymbol{a}_0^{(k)} + \boldsymbol{a}_1^{(k)}(t - t_k) + \boldsymbol{a}_2^{(k)}(t - t_k)^2 \tag{2.2}$$

With the coefficients $\boldsymbol{a}_0^{(k)}, ..., \boldsymbol{a}_2^{(k)}$ chosen to match the function at the beginning and the end of the k-th interval:

$$\begin{aligned} \tilde{\boldsymbol{x}}(t_k) &= \boldsymbol{x}(t_k) \\ \tilde{\boldsymbol{x}}(t_{k+1}) &= \boldsymbol{x}(t_{k+1}) \end{aligned} \tag{2.3}$$

Also the time derivatives must match at t_k and t_{k+1}:

$$\begin{aligned} \frac{\mathrm{d}\tilde{\boldsymbol{x}}(t_k)}{\mathrm{d}t} &= \boldsymbol{f}[\boldsymbol{x}(t_k), \boldsymbol{u}(t_k), \boldsymbol{p}, t_k] \\ \frac{\mathrm{d}\tilde{\boldsymbol{x}}(t_{k+1})}{\mathrm{d}t} &= \boldsymbol{f}[\boldsymbol{x}(t_{k+1}), \boldsymbol{u}(t_{k+1}), \boldsymbol{p}, t_{k+1}] \end{aligned} \tag{2.4}$$

By putting together Eq.2.2, Eq.2.3 and Eq.2.4, and evaluating them at t_k and t_{k+1}, it yields:

$$\begin{aligned} \tilde{\boldsymbol{x}}(t_{k+1}) &= \boldsymbol{x}(t_k) + \boldsymbol{f}_k(t_{k+1} - t_k) + \frac{1}{2}(\boldsymbol{f}_{k+1} - \boldsymbol{f}_k)(t_{k+1} - t_k) \\ &= \boldsymbol{x}(t_k) + \frac{1}{2}(\boldsymbol{f}_k + \boldsymbol{f}_{k+1})(t_{k+1} - t_k) \end{aligned} \tag{2.5}$$

Since $\tilde{\boldsymbol{x}}(t_{k+1}) = \boldsymbol{x}(t_{k+1})$, then Eq.2.5 can be expressed as follows:

$$\boldsymbol{z}(t_k) = \boldsymbol{x}(t_{k+1}) - \boldsymbol{x}(t_k) - h_k \frac{1}{2}(\boldsymbol{f}_k + \boldsymbol{f}_{k+1}) = 0, \tag{2.6}$$

where $h_k = t_{k+1} - t_k$ and $\boldsymbol{z}(t_k) = 0$ is the differential defect constraint at node t_k associated with the trapezoidal method. Allowing $j = 0, \ldots, N-1$, Eq.2.6 generates Nn_x differential defect constraints. It is important to note that the trapezoidal method is 2-stage Runge-Kutta method.

The described approximation method prioritizes computational efficiency; however, it suffers from reduced accuracy when the number of discretization intervals is insufficient. Consequently, following an initial coarse search for the optimal trajectory utilizing a limited number of mesh nodes and the trapezoidal method, subsequent solution refinement is necessary to achieve a high-fidelity trajectory. This refinement process, as analyzed and tested in the case studies presented in Chapter 5 and Chapter 5.3, involves progressively increasing

the number of nodes while concurrently increasing the polynomial degree, thus leveraging the *Hermite-Simpson* collocation technique.

The Hermite–Simpson method is based on a cubic interpolating polynomial, such that Eq. (2.23) becomes:

$$\tilde{\boldsymbol{x}}(t) = \boldsymbol{a}_0^{(k)} + \boldsymbol{a}_1^{(k)}(t - t_k) + \boldsymbol{a}_2^{(k)}(t - t_k)^2 + \boldsymbol{a}_3^{(k)}(t - t_k)^3. \tag{2.7}$$

Evaluating the time derivative of the interpolant at the beginning t_k, end t_{k+1}, and midpoint $\bar{t}_k = \frac{t_k + t_{k+1}}{2}$ of the interval, and matching them with the corresponding in Eq.2.1, results in the following expressions:

$$\frac{d\tilde{\boldsymbol{x}}(t_k)}{dt} = \boldsymbol{a}_1^{(k)} = \boldsymbol{f}_k, \tag{2.8}$$

$$\frac{d\tilde{\boldsymbol{x}}(t_{k+1})}{dt} = \boldsymbol{a}_1^{(k)} + 2\boldsymbol{a}_2^{(k)} h_k + 3\boldsymbol{a}_3^{(k)} h_k^2 = \boldsymbol{f}_{k+1}, \tag{2.9}$$

$$\frac{d\tilde{\boldsymbol{x}}(\bar{t}_k)}{dt} = \boldsymbol{a}_1^{(k)} + 2\boldsymbol{a}_2^{(k)} \bar{h}_k + 3\boldsymbol{a}_3^{(k)} \bar{h}_k^2 = \bar{\boldsymbol{f}}_k \tag{2.10}$$

Note that $\bar{\boldsymbol{f}}_k$ depends on the interpolated state at the midpoint of the interval $\tilde{\boldsymbol{x}}(\bar{t}_k)$, and on the control also at the midpoint $\boldsymbol{u}(\bar{t}_k)$. Equations 2.8 and 2.9 result in the following interpolating polynomial:

$$\begin{aligned} \tilde{\boldsymbol{x}}(t) =& \boldsymbol{x}(t_k) + \boldsymbol{f}_k(t - t_k) + \frac{(4\bar{\boldsymbol{f}}_k - \boldsymbol{f}_{k+1} - 3\boldsymbol{f}_k)}{2h_k}(t - t_k)^2 \\ &+ \frac{(\boldsymbol{f}_k + \boldsymbol{f}_{k+1} - 2\bar{\boldsymbol{f}}_k)}{3h_k^2}(t - t_k)^3. \end{aligned} \tag{2.11}$$

Evaluating Eq. 2.11 at the end of the interval gives:

$$\tilde{\boldsymbol{x}}(t_{k+1}) = \boldsymbol{x}(t_k) + \frac{h_k}{6}(\boldsymbol{f}_k + 4\bar{\boldsymbol{f}}_k + \boldsymbol{f}_{k+1}). \tag{2.12}$$

But from the collocation condition $\tilde{\boldsymbol{x}}(t_{k+1}) = \boldsymbol{x}(t_{k+1})$, then Eq.2.12 can be expressed as follows:

$$\boldsymbol{z}(t_k) = \boldsymbol{x}(t_{k+1}) - \boldsymbol{x}(t_k) - \frac{h_k}{6}(\boldsymbol{f}_k + 4\bar{\boldsymbol{f}}_k + \boldsymbol{f}_{k+1}) = 0, \tag{2.13}$$

where $\boldsymbol{z}(t_k) = 0$ is the differential defect constraint at node t_k associated with the Hermite–Simpson method. Evaluating Eq. 2.11 at the midpoint of the interval:

$$\tilde{\boldsymbol{x}}(\bar{t}_k) = \boldsymbol{x}(t_k) + \frac{h_k}{24}(5\boldsymbol{f}_k + 8\bar{\boldsymbol{f}}_k - \boldsymbol{f}_{k+1}). \tag{2.14}$$

Solving Eq. 2.12 for $\bar{\boldsymbol{f}}_k$ and replacing in Eq. 2.14 results in

$$\tilde{\boldsymbol{x}}(\bar{t}_k) = \frac{1}{2}[\boldsymbol{x}(t_k) + \boldsymbol{x}(t_{k+1})] + \frac{h_k}{8}(\boldsymbol{f}_k - \boldsymbol{f}_{k+1}). \tag{2.15}$$

Eq. 2.15 gives the interpolated value of the state at the midpoint of the interval, which is needed to evaluate $\bar{\boldsymbol{f}}_k$. Note that the evaluation of Eq. 2.15 and the substitution of $\tilde{\boldsymbol{x}}(\bar{t}_k)$ into Eq. 2.13 results in the so-called *compressed Hermite–Simpson method*. It generates Nn_x differential defect constraints.

2.4.1 Global Search Method: Multi-Start combined with Gradient-Based SQP

The optimization tool developed in this thesis, as detailed in Chapter 4, employs both deterministic and metaheuristic approaches. Specifically, it utilizes gradient-based SQP algorithms, starting from an initial guess of the decision vector $\boldsymbol{y}$. This approach, beginning from a single arbitrary initial guess, may converge to a local optimal solution without guaranteeing global optimality. Therefore, it necessitates the integration of a local optimization strategy with an external mechanism in order to achieve a global optimization. To this end, a Multi-Start approach was adopted. As illustrated in Figure 2.3, this approach involves the simultaneous generation of multiple initial guesses within the search space, facilitating convergence to various local solutions. By thoroughly exploring the solution space and comparing the solutions systematically, this method may identify the global optimal solution.

This approach is relatively straightforward to implement algorithmically; however, it is computationally more demanding, requiring a higher number of within-domain solution evaluations compared to other global methods. Furthermore, in the context of feasibility studies for interplanetary missions, it offers the added benefit of evaluating and storing sub-optimal solutions. These sub-optimal solutions can serve as alternatives to the optimal solution, particularly when it is necessary considering variations in launch dates or flight durations.

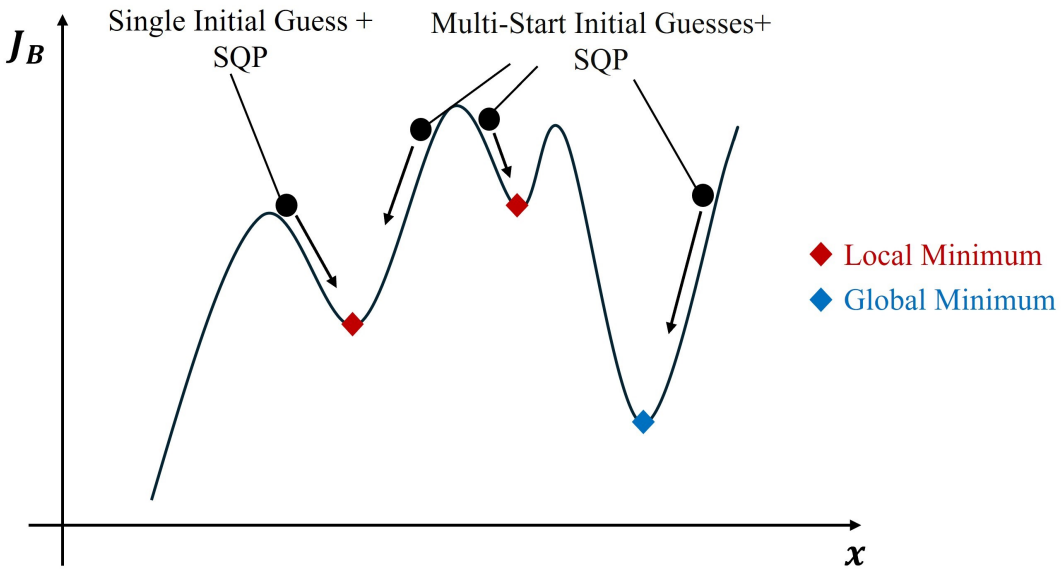

Figure 2.3: Graphical visualization of the multi-start + SQP method

Chapter 3

The Low-thrust Interplanetary Transfer Problem

This chapter aims to delineate and formulate the optimal control model for interplanetary low-thrust trajectories. This mathematical model underpins the implementation of the optimization tool developed within this study, which will be operationally detailed in Chapter 4. The following sections present the dynamic models of interplanetary transfers, consistent with the existing literature (see Vallado, 2001[44], Curtis, 2020[11], Bate, Mueller, and White, 1971[1]), followed by the formulation of the optimization problem, according with Topputo and Aguiar, 2018[42], Betts and Huffman, 1992[5], Pesenti, 2024[36] and Topputo and Zhang, 2014[43].

3.1 Dynamics

In this section, we will discuss the modeling of the complete spaceflight dynamics, also incorporating the mass differential equation. When considering all potential sources of perturbations, the equations are expressed as follows:

$$\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u},t)=\begin{cases}\dot{\boldsymbol{r}}=\boldsymbol{v}\\ \dot{\boldsymbol{v}}=\boldsymbol{a}\\ \dot{m}=-\|\boldsymbol{u}\|\frac{T_{max}}{g_0 I_{sp}}\end{cases}\qquad \boldsymbol{a}=\boldsymbol{a}_{CB}+\sum_{j=1}^{n_{3B}}\boldsymbol{a}_{3Bj}+\boldsymbol{a}_{\text{drag}}+\boldsymbol{a}_{SRP}+\boldsymbol{a}_{Thrust} \tag{3.1}$$

Where m is the mass of the spacecraft, $\boldsymbol{r}$, $\boldsymbol{v}$, $\boldsymbol{a}$ are the position, velocity and acceleration vectors related to the S/C motion in a central body intertial reference frame, thus $\boldsymbol{s}=[\boldsymbol{r},\boldsymbol{v},m]$ is the state vector along the trajectory. Concerning the mass differential equation, $\boldsymbol{u}$ is the control vector (that accounts

for the direction and throttle of the thrust along the path), T_{max} is the maximum allowable thrust level along the trajectory, I_{sp} is the specific impulse of the thruster and $g_0 = 9.81m/s$. On the right side of the acceleration formula in Eq.3.1 we can find the contribution of the central body gravity field ($\boldsymbol{a}_{CB}$), the third-bodies perturbation ($\boldsymbol{a}_{3Bj}$), the atmospheric drag perturbation ($\boldsymbol{a}_{drag}$), the solar radiation pressure ($\boldsymbol{a}_{SRP}$) and finally the continuous thrust ($\boldsymbol{a}_{Thrust}$).

In alignment with the specific objectives of this thesis, this section provides a detailed description of the formulation of accelerations typically considered during the preliminary design of heliocentric segments of low-thrust interplanetary trajectories. Specifically, it addresses the acceleration due to the central body's gravity field, the continuous thrust effect and the perturbations from third bodies. For further details on other perturbations, the reader is referred to the aforementioned literature.

3.1.1 Central Body Gravity Field

To determine the acceleration resulting from the gravity field of a central body, the concept of gravitational potential must be introduced. Since the gravitational field is conservative, it can be associated with a potential. For a spherical body, this potential is expressed as:

$$U(r) = -\frac{\mu}{r} \tag{3.2}$$

The acceleration induced by the potential is evaluated as:

$$\boldsymbol{a}_{CB} = -\nabla U(\boldsymbol{r}) = -\frac{\mu}{r^3}\boldsymbol{r} \tag{3.3}$$

In practice, celestial bodies are not perfectly spherical but exhibit irregular shapes; for instance, it is well known that the Earth approximates a geoid. The impact of these irregularities is considered negligible for the purposes of this thesis and, therefore, is not discussed in detail.

3.1.2 Continuous Thrust Modeling

A key distinguishing and complicating aspect of low-thrust trajectory design is the presence of continuous thrust throughout most of the flight duration. This continuous thrust results in an acceleration on the spacecraft, which can be expressed as:

$$\boldsymbol{a}_{Thrust} = \frac{\boldsymbol{T}}{m} = \boldsymbol{u}\frac{T_{max}}{m} \tag{3.4}$$

Where $\boldsymbol{T}$ represents the thrust vector. The maximum available thrust depends on the type of electric propulsion system being used. For instance, electric

propulsion systems with a constant maximum thrust (T_{max}) are characteristic of Nuclear Electric propulsion (NE) systems or applications where the power generation system is designed to ensure the required power for nominal thruster operation at the farthest point of the trajectory from the Sun. Conversely, for Solar Electric propulsion (SE) systems, the available T_{max} is dependent on the spacecraft distance from the sun and must be modeled accordingly.

The model for determining the thrust level of solar-electric propulsion systems is inherently complex. It is approximately understood that the flux of solar radiation decreases as $1/r^2$, where r represents the distance of the spacecraft from the Sun. Thrust magnitude T_{max} depends on input power $P(r)$, thruster efficiency η, and specific impulse I_{sp}
as follows:

$$T_{max} = 2\eta P(r)/c \ , \tag{3.5}$$

where $c = g_0 I_{sp}$ is the engine exhaust velocity.

Additionally, a cutoff distance r_{max} could be defined, beyond which all the generated electrical power is required to sustain the spacecraft's systems, leaving no power available for propulsion, thus preventing thrust generation. Conversely, there exists a certain value r_{min} at which the available electrical power reaches saturation, resulting in the maximum possible thrust.

Numerous specific mathematical models have been proposed in the literature (e.g., Williams and Coverstone-Carroll, 1997[45], Genta and Maffione, 2015[17], Kluever, 1997[23]); however, the modeling of the thrust profile is generally contingent upon the specific case study and the associated propulsion system specifications.

This study presents a modeling approach that aligns with industry requirements for preliminary mission studies. Notably, many mission analysis guidelines documents provided by clients specify thrust characteristics as constant profiles over various distance ranges, derived from tests, simulations, and measurements at discrete operating points. In mission analysis, to approximate these data more accurately, they are numerically fitted using second-degree polynomials, ensuring that the average value over each sub-interval matches the provided value.

To better understand this process, a representative example of mission analysis fit is shown in Figure 3.1. It is a BepiColombo-based electric propulsion system (see Sutherland, Stramaccioni, and Benkhoff, 2019[40]) using 2xT6 gridded ion thrusters. It will be implemented in the case study detailed in Section 5.1.

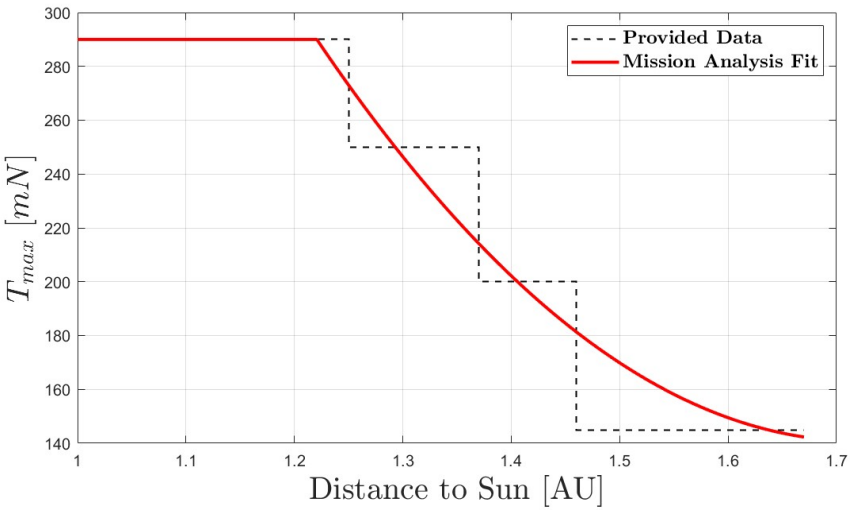

Figure 3.1: BepiColombo-based EP system (2xt6 ion thrusters) Mission Analysis modeling

3.1.3 Third-Body Perturbation

The number of third bodies considered depends on the desired accuracy and the type of orbit a satellite is in. For instance, in very high-altitude earth orbits, accounting for disturbances due to the Sun and the Moon is usually necessary. For interplanetary missions, the initial calculation of the heliocentric trajectory can be performed by neglecting perturbations and considering a restricted two-body problem. However, once this preliminary solution is obtained, it is essential to refine it by incorporating the perturbations caused by other relevant planets. These perturbations may not be negligible, particularly during the initial and final phases of the heliocentric journey.

The contribution of the i-th body is modeled as follows:

$$\boldsymbol{a}_{3Bj} = \mu_j \left(\frac{\boldsymbol{r}_{\mathrm{sat}j}}{r_{\mathrm{sat}j}^3} - \frac{\boldsymbol{r}_j}{r_j^3} \right) \tag{3.6}$$

The gravitational parameter of the j-th body is denoted by μ_j. The position vector of the j-th body relative to the satellite is represented by $\mathbf{r}_{\mathrm{sat}j}$, while $\mathbf{r}_j$ denotes the position vector of the j-th body relative to the central body on which the inertial reference system is centered, as shown in Figure 3.2.

Two terms are identified: the first, known as the direct effect, represents the acceleration induced by the third body on the satellite. The second term, referred to as the indirect effect, identify the interaction between central body and the third body.

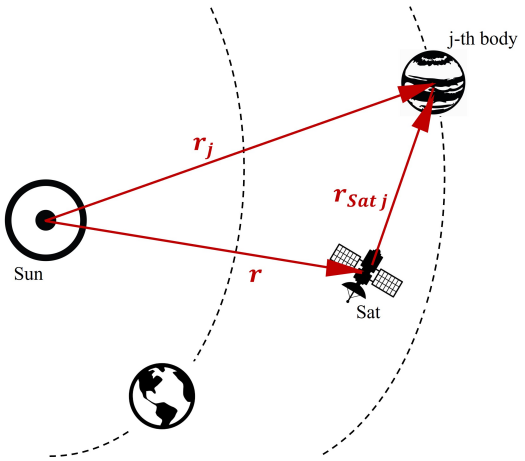

Figure 3.2: Third-Bodies Perturbation geometrical scheme

3.1.4 Patched Conics Approach

In order to efficiently compute complex interplanetary trajectories, typically the concept of *Patched Conics* is used (see Vallado, 2001[44]).
This approach divides the transfer into 3 phases, as shown in Figure 3.3:

- **The escape trajectory**, in which the attractive force of the starting planet on the S/C is predominant over that of the Sun and other planets;
- **The heliocentric trajectory**, in which the Sun's attraction clearly prevails over that of all other celestial bodies;
- **The capture trajectory**, in which the attraction of the Sun and other planets is negligible compared to that of the target planet.

It is possible to study these three phases separately by considering the *Spheres of Influence* (SOI) of each planet as *patch points*. At these points, the analysis transitions from planet-centric to heliocentric at departure and vice versa at arrival.

The sphere of influence of a central body defines a region in space where the motion of a spacecraft is predominantly influenced by the gravity of that body, rather than by any other perturbations. For instance, the Earth's sphere of influence delineates the region of space where Earth's gravitational attraction is more significant than that of the Sun and other planets. A commonly used formula to estimate the radius of this sphere is given by:

$$r_{\mathrm{SOI}} = a_{\oplus}\left(\frac{m_{\oplus}}{m_{\odot}}\right)^{\frac{2}{5}} \tag{3.7}$$

where $m_{\oplus}$ and $m_{\odot}$ are the masses of the planet and the Sun, respectively, and $a_{\oplus}$ is the semi-major axis of planet's orbit around the Sun.

By employing the patched conics approach, it is possible to decouple the analysis of the interplanetary (heliocentric) segment from the planetocentric phases at departure and arrival. This methodology allows for the initial focus on the study and optimization of the heliocentric transfer alone, treating it as a restricted two-body problem and considering the departure and arrival conditions as boundary conditions and/or constraints. Subsequently, it is possible to design the planetocentric trajectories based on the boundary conditions derived from the interplanetary phase analysis. Finally, the three trajectories can be matched a posteriori, enhancing the model fidelity by accounting for perturbations.

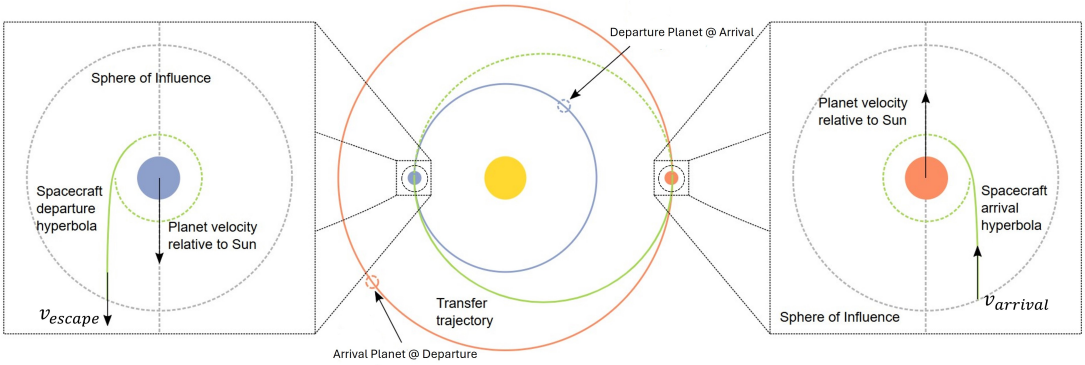

Figure 3.3: Concept of Patched Conics Approach

3.1.5 Adopted Reference Frames

The reference systems utilized in this study are analyzed in this subsection. As the objective of this thesis is to examine only the interplanetary phase of the trajectory and not the ballistic capture phase on the target planet, the primary focus is on the reference system with respect to the starting planet, useful for defining the initial characteristics of the trajectory, and on the reference system employed to describe the entire interplanetary phase.

The terrestrial J2000 reference frame is used for departure. This coordinates system is centered on Earth's center of mass, with the X-axis aligned with the vernal equinox. The Z-axis is Earth's rotation axis at J2000. The Y-axis is perpendicular to the X and Z axes. This reference system is good for modelling the departure phase because it is based on the equatorial plane. It is often used to describe the orbit of Earth, which makes it useful for things like the change in velocity.

For the interplanetary phases, the ECLIPJ2000 reference system is used. This system is centered on the Sun and is also an inertial reference system. It shares the X-axis with the J2000 system but uses the ecliptic plane as its fundamental plane, resulting in a rotation relative to the J2000 system by a specific angle. This reference system is good for the interplanetary phases because most planets

and other small solar system bodies orbit at a low angle to the ecliptic. This allows us to represent their paths in two dimensions by projecting them onto the fundamental plane.

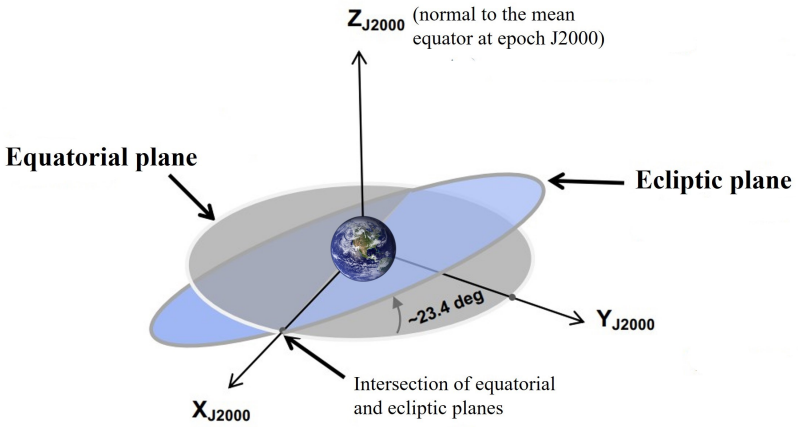

Figure 3.4: Visualizzation of Earth's J2000 Equatorial and Ecliptic Planes. Credits: Navigation and (NAIF), 2023[34]

The SPICE system (see NASA's Navigation and Ancillary Information Facility (NAIF), 2024[33]) is used to determine the orientation of the reference systems and rotation matrices. It was developed by NASA's Navigation and Ancillary Information Facility (NAIF) at the Jet Propulsion Laboratory (JPL). This NASA system is the standard for managing data on planetary missions. It is also used for data analysis on planetary missions by other space agencies.

3.1.6 Ephemerides

In order to study and solve an interplanetary rendezvous problem, it is necessary to be able to define the positions of the celestial bodies concerned as a function of the date of interest. In order to achieve this within the context of this study, ephemeris models are employed in order to determine the positions and velocities of these bodies at specific points in time.

In order to ensure an high-accuracy optimization tool for industrial study purposes, the ephemerides, developed by the Jet Propulsion Laboratory (JPL) and accessible via the SPICE system, are employed. The current lunar orbit is known with submeter accuracy, achieved by fitting lunar laser ranging data with an updated lunar gravity field from the Gravity Recovery and Interior Laboratory (GRAIL) mission. The orbits of the inner planets are determined with sub-kilometer accuracy through radio tracking measurements of orbiting spacecraft. Very long baseline interferometry measurements of spacecraft at Mars

enable the ephemeris orientation to be aligned with the International Celestial Reference Frame with an accuracy of 0.0002 arcseconds. This orientation accuracy limits the orbit uncertainties of the terrestrial planets to a few hundred meters. The orbits of Jupiter and Saturn are determined with accuracies of tens of kilometers by fitting spacecraft tracking data. The orbits of Uranus, Neptune, and Pluto are primarily determined from astrometric observations, where measurement uncertainties due to Earth's atmosphere and star catalog inaccuracies limit position accuracies to several thousand kilometers.

3.2 Optimal Control Problem Formulation

In this section, the development of the mathematical model for the optimal control problem statement is presented. This model forms the foundation of the numerical tool presented in this thesis. Specifically, the general optimal control problem is tailored to address the problem of interplanetary rendezvous maneuvers utilizing continuous thrust and direct transcription methods. Initially, the continuous problem is described in Subsection 3.2.1, followed by the discretization of the problem through collocation, as detailed in Subsection 3.2.2.

3.2.1 Continuous Optimal Control Problem Statement

To effectively model the optimization problem, it is essential to identify the objective function to be minimized, the optimization variables, and the constraints. Initially, the problem must be clearly defined. The aim of this work is to determine the optimal thrust profile (in terms of direction and intensity) to minimize propellant consumption. This trajectory must adhere to interplanetary rendezvous conditions, satisfying both the departure condition from the initial planet (e.g., Earth) and the arrival condition on the target (e.g., Mars).

Additional constraints include the maximum escape velocity from the starting planet, which is linked to the launcher's performances, and the maximum arrival velocity on the target planet, typically set to zero for electric propulsion to facilitate ballistic capture. The optimization must also consider the maximum available thrust along the path, which , depending on the assumptions, may be constant or vary depending on the spacecraft's position relative to the sun. Furthermore, constraints on the departure date, influenced by potential launch window limitations, and the flight duration must be accounted for. The aforementioned considerations are translated into the following mathematical model.

Variables

The following variables are identified:

- t: time;
- t_0: departure date;
- Δt: Time of Flight (TOF);
- $\boldsymbol{r}(t) = [x(t), y(t), z(t)]$: spacecraft position vector along the path;
- $\dot{\boldsymbol{r}}(t) = [\dot{x}(t), \dot{y}(t), \dot{z}(t)]$: spacecraft velocity vector along the path;
- $m(t)$: spacecraft mass along the trajectory;
- $\boldsymbol{u}(t) = [u_x(t), u_y(t), u_z(t)]$: control vector along the trajectory;

The aforementioned vectors $\boldsymbol{r}(t)$,$\dot{\boldsymbol{r}}(t)$, and $\boldsymbol{u}(t)$ are expressed with respect to the ECLIPJ2000 reference frame (see Section 3.1.5). It is important to notice that in this work the thrust control vector $\boldsymbol{u}(t)$ is formulated such that the thrust vector is $\boldsymbol{T}(t) = \boldsymbol{u}(t)T_{max}(t)$. The state vector $\boldsymbol{s}(t) = [\boldsymbol{r}(t), \dot{\boldsymbol{r}}(t), m(t)]$ and the control variable vector $\boldsymbol{u}(t)$ are thus identifiable.

Constraints

The aforementioned variables are subject to box-type constraints (upper and lower bounds) and additional constraints.

The following box-constraints can be identified:

$$t_{0,min} \leq t_0 \leq t_{0,max} \tag{3.8}$$

$$\Delta t_{min} \leq \Delta t \leq \Delta t_{max} \tag{3.9}$$

$$m(t_0) = m_{wet} \tag{3.10}$$

Here, $t_{0,min}$ and $t_{0,max}$ identify the considered launch window limits , m_{wet} represents the wet mass of the S/C at the beginning of the trajectory, m_{min} is the minimum acceptable S/C mass at the end of the trajectory (i.e. when the maximum allowable propellant mass has been consumed), and $t_f = t_0 + \Delta t$.

Additionally, the following constraints can be defined, related to the departure and arrival conditions, where $\boldsymbol{r}_{DP}(t_0)$ and $\boldsymbol{v}_{DP}(t_0)$ are respectively the position vector and the velocity vector of the departure planet at the departure date and $\boldsymbol{r}_{AP}(t_f)$ and $\boldsymbol{v}_{AP}(t_f)$ are the position vector and velocity vector of the

arrival planet at the arrival time:

$$\|\boldsymbol{u}(t)\| \leq 1 \quad \text{for} \quad t \in [t_0, t_f] \tag{3.11}$$

$$\|\boldsymbol{r}(t_0) - \boldsymbol{r}_{DP}(t_0)\| = r_{SOIDP} \tag{3.12}$$

$$\|\dot{\boldsymbol{r}}(t_0) - \boldsymbol{v}_{DP}(t_0)\| \leq v_{escape,max} \tag{3.13}$$

$$\|\boldsymbol{r}(t_f) - \boldsymbol{r}_{AP}(t_f)\| = r_{SOIAP} \tag{3.14}$$

$$\|\dot{\boldsymbol{r}}(t_f) - \boldsymbol{v}_{AP}(t_f)\| \leq v_{arrival,max} \tag{3.15}$$

Where r_{SOIDP} and r_{SOIAP} are the radii of the spheres of influence of respectively departure and arrival planets (computed as shown in Eq.3.7), $v_{escape,max}$ is the maximum allowable escape velocity from the departure planet, and $v_{capture,max}$ is the maximum allowable arrival velocity at the arrival planet's SOI (usually equal to zero).

Finally, the system dynamics are incorporated by imposing the differential dynamical equations (state equations formulated as shown in Section 3.1).

Objective Function

The objective of the optimization is to find the optimal control law to minimize the fuel consumption of the transfer, i.e. to maximize the final mass of the spacecraft $m(t_f)$. It is possible to formulate the objective function as follows

$$min: \quad -m(t_f) \tag{3.16}$$

By accounting for the dynamics of the system (Eq. 3.1) we can rewrite Eq. 3.16 as

$$min: \quad \int_{t_0}^{t_f} \|\boldsymbol{u}(t)\| \frac{T_{max}(t)}{g_0 I_{sp}} \, dt \tag{3.17}$$

3.2.2 Discretized Problem Statement

To numerically solve the continuous problem via direct collocation, it is essential to discretize the problem over a finite set of points, thereby transforming it into a NLP problem. The core of the reduction of the continuous problem to the nonlinear programming problem is the parametrization of all continuous variables and the transcription of the differential equations of dynamics into a finite set of equality constraints. The detailed process of this discretization is elaborated upon as follows.

The reduction of the continuous problem to a finite one is based on the discretization of the continuous variables on a mesh settled up on the time domain. Thus, it is necessary to discretize the (t_0, t_f) interval into an arbitrary number of N sub-intervals, thereby identifying $N+1$ mesh points. The type of discretization employed, along with the chosen value of N, significantly affects the accuracy of the discretized solution in comparison to the continuous one. Consequently, various methods are available in specific literature. The simplest method involves creating an equispaced grid of points; however, this approach may lack precision in regions where the actual solution exhibits strong nonlinearity. In such regions, if the grid is not sufficiently dense, it may fail to adequately capture the solution's behavior. To address this issue, adaptive discretization systems have been developed, which can internally adjust the mesh density based on the nonlinearity trend of the problem in specific areas (e.g., *Legendre-Gauss-Lobatto* techniques, see Hofmann, Morelli, and Topputo, 2022[19]). In this study, an equispatially discretized technique was employed, with a sufficient number of points to enable an adequate description of the phenomenon in a preliminary approximation.

Thus, the time domain is reduced to a finite vector

$$\boldsymbol{t} = [t_0\,, ..., t_i\,, ..., t_N\,] \;\in \mathbb{R}^{N+1} \tag{3.18}$$

Where the time step is $h = \Delta t/N$. Consequently, the continuous variables of the problem will become vectors containing the discrete states of the variables at the grid points, as described below.

Variables

The following discretized variables are identified, by adopting a notation whereby, having identified a specific quantity q, $\underline{q}_k = q(t_k)$ represents the quantity discretized from the continuous problem:

- t_0: departure date;
- Δt: Time of Flight (TOF);
- $\underline{\boldsymbol{r}} = [\underline{x}_0, \ldots, \underline{x}_N, \underline{y}_0, \ldots, \underline{y}_N, \underline{z}_0, \ldots, \underline{z}_N] \;\in \mathbb{R}^{3(N+1)}$;
- $\underline{\dot{\boldsymbol{r}}} = [\underline{\dot{x}}_0, \ldots, \underline{\dot{x}}_N, \underline{\dot{y}}_0, \ldots, \underline{\dot{y}}_N, \underline{\dot{z}}_0, \ldots, \underline{\dot{z}}_N] \;\in \mathbb{R}^{3(N+1)}$;

- $\underline{\boldsymbol{m}} = [\underline{m_0}, ..., \underline{m_i}, ..., \underline{m_N}] \in \mathbb{R}^{N+1}$;
- $\underline{\boldsymbol{u}} = [\underline{u}_{x0}, \ldots, \underline{u}_{xN}, \underline{u}_{y0}, \ldots, \underline{u}_{yN}, \underline{u}_{z0}, \ldots, \underline{u}_{zN}] \in \mathbb{R}^{3(N+1)}$;

It is possible to identify the S/C state vector $\underline{\boldsymbol{s}}_k$ and the control vector $\underline{\boldsymbol{u}}_k$ at each k-th mesh point as follows

$$\underline{\boldsymbol{s}}_k = [\underline{x}_k, \underline{y}_k, \underline{z}_k, \underline{\dot{x}}_k, \underline{\dot{y}}_k, \underline{\dot{z}}_k, \underline{m}_k] \tag{3.19}$$

$$\underline{\boldsymbol{u}}_k = [\underline{u}_{xk}, \underline{u}_{yk}, \underline{u}_{zk}] \tag{3.20}$$

The discretized decision vector containing all the variables is defined as follows

$$\underline{\boldsymbol{d}} = [t_0, \Delta t, \underline{\boldsymbol{r}}, \underline{\dot{\boldsymbol{r}}}, \underline{\boldsymbol{m}}, \underline{\boldsymbol{u}}] \in \mathbb{R}^{(10N+12)} \tag{3.21}$$

These variables are subject to the following discretized constraints.

Constraints

The continuous constraints in the previous section can be reduced to the following discretized ones. The box constraints (3.8) and (3.9) remain unchanged, whereas (3.10) is reformulated as follows:

$$\underline{m}_0 = m_{wet} \tag{3.22}$$

The other constraints are reformulated as follows

$$\|\underline{\boldsymbol{u}}_i\| \leq 1 \quad \text{for} \quad i = 0, ..., N \tag{3.23}$$

$$\|\underline{\boldsymbol{r}}_0 - \boldsymbol{r}_{DP}(t_0)\| = r_{SOIDP} \tag{3.24}$$

$$\|\underline{\dot{\boldsymbol{r}}}_0 - \boldsymbol{v}_{DP}(t_0)\| \leq v_{escape,max} \tag{3.25}$$

$$\|\underline{\boldsymbol{r}}_N - \boldsymbol{r}_{AP}(t_N)\| = r_{SOIAP} \tag{3.26}$$

$$\|\underline{\dot{\boldsymbol{r}}}_N - \boldsymbol{v}_{AP}(t_N)\| \leq v_{arrival,max} \tag{3.27}$$

Finally, the collocation method yields to the transcription of the differential equations (Eq. 3.1) into a finite set of defects constraints $\underline{\boldsymbol{\tau}} = \mathbf{0}$ *with* $\underline{\boldsymbol{\tau}} \in \mathbb{R}^7 \times \mathbb{R}^N$ through techniques shown in Section 2.4.

For instance, in the simpler case of trapezoidal collocation, defects constraints are formulated as follows, according to Eq. 2.6

$$\underline{\boldsymbol{\tau}}_i = \underline{\boldsymbol{s}}_{i+1} - \underline{\boldsymbol{s}}_i - h\frac{1}{2}(\boldsymbol{f}_i + \boldsymbol{f}_{i+1}) = 0 \quad \text{for } i = 0, \ldots, N-1 \tag{3.28}$$

Where $\boldsymbol{f}_k = \boldsymbol{f}(\underline{\boldsymbol{s}}_k, \underline{\boldsymbol{u}}_k, t_k)$ for $k = 0, ..., N$ that represent the discretized values of the dynamics ODEs (Eq. 3.1) evaluated in the mesh points. It is possible to transform the defects matrix into a row vector $\underline{\boldsymbol{\tau}} \in \mathbb{R}^{7N}$. Subsequently, a vector encompassing all non-linear constraints (3.23, 3.24, 3.25, 3.26, 3.27, 3.28) can be defined as follows:

$$\boldsymbol{\zeta} \in \mathbb{R}^{8N+5} \tag{3.29}$$

Within this model, the desired level of accuracy can be adjusted by either increasing the number of nodes in the mesh or by selecting more sophisticated placement techniques (e.g., Hermite-Simpson, see Eq. 2.13) to generate the defect constraints. Additionally, the ODE model used within the defects can be modified to either account for or disregard potential dynamical perturbations (e.g., third-body perturbations).

Objective Function

The objective function can be simply rewritten as

$$minimize: \quad -\underline{m}_N \tag{3.30}$$

3.2.3 Sparsity Implementation

Examining the discretized model reveals that solving an optimization problem using direct transcription and collocation necessitates managing a substantial number of variables and non-linear constraints. The quantity of these elements is contingent upon the discretization refinement and will significantly increase with higher mesh density, as illustrated in Figure 3.5. Specifically, for this problem, the number of variables is given $n_{var} = 10N + 12$, and the number of non-linear constraints is $n_{constr} = 8N + 5$. For instance, with $N = 100$, a plausible value for the initial representative solution in the optimization of interplanetary trajectories, there are 1012 variables and 805 non-linear constraints that the solver must handle. The scale of these values necessitates substantial memory allocation for each evaluation during each iteration of the solution process and considerable computational time to manage and perturb the variables to achieve convergence to the optimal solution.

Specifically, in the context of gradient-based solving techniques, the solver is required to numerically compute finite differences to evaluate the gradient vector

of the objective function. This evaluation is crucial for searching for the optimal solution by attempting to satisfy the KKT conditions (see M.Minoux, 1986[26]). Additionally, the solver must compute the Jacobian matrix of constraints, which is essential for managing the variables to ensure compliance with the imposed non-linear constraints.

The gradient vector of the objective function is identified as $\mathcal{G} \in \mathbb{R}^{10N+12}$, with components defined as follows:

$$\mathcal{G}_k = \frac{\partial J}{\partial \underline{d}_k} \quad \text{for } k = 0, \ldots, 10N + 11 \tag{3.31}$$

where J represents the objective function.

The Jacobian matrix of non-linear constraints is $\mathcal{J} \in \mathbb{R}^{8N+5} \times \mathbb{R}^{10N+12}$, and it represents the matrix containing the partial derivatives of the non-linear constraints with respect to the variables. Thus, its components are defined as follows:

$$\mathcal{J}_{i,k} = \frac{\partial \zeta_i}{\partial \underline{d}_k} \quad \text{for } i = 0, \ldots, 8N + 4 \; ; \; k = 0, \ldots, 10N + 11 \tag{3.32}$$

Where $\boldsymbol{\zeta}$ and $\underline{\boldsymbol{d}}$ are defined respectively in Eq.3.29 and Eq.3.21.

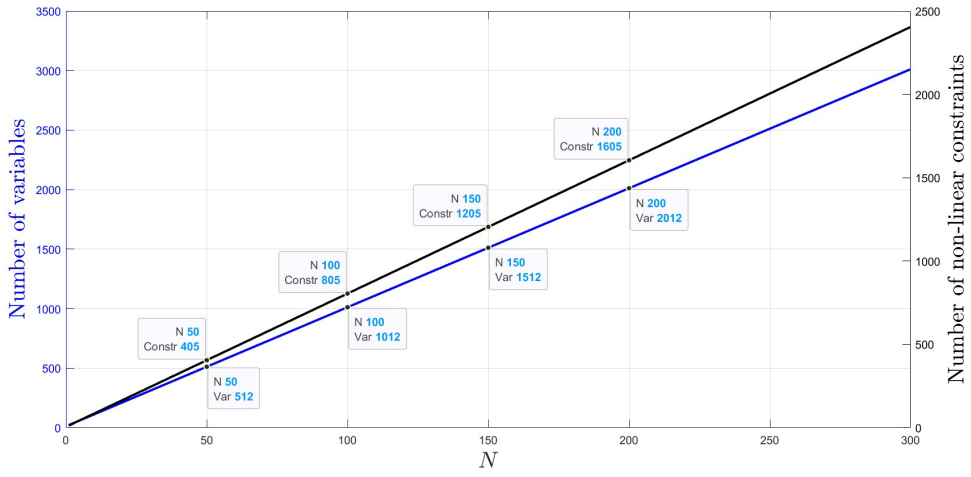

Figure 3.5: Variables and non-linear constraints number as function of N

Consequently, it is evident that the gradient vector, comprising $10N + 12$ elements, and the Jacobian matrix, consisting of $80N^2 + 146N + 60$ elements, necessitate the evaluation of an extensive number of finite differences at each iteration. This requirement, particularly concerning the jacobian matrix, makes the computation prohibitively expensive in terms of both memory usage and computational time.

Therefore, it is imperative to leverage the sparse structure of these matrices by conducting an ad-hoc analysis tailored to the specific problem and incorporating

this structure into the solver a priori. Through a detailed study, it is possible to identify the variables upon which each row of the Jacobian matrix depends (i.e., each non-linear constraint). Consequently, the solver can be configured to evaluate the finite differences exclusively for these non-zero elements at each iteration. The same approach applies to the gradient vector.

Beginning with the gradient of the objective function, it is evident that it depends solely on the variable representing the final mass of the spacecraft. Consequently, the gradient vector will contain only one non-zero component out of a total of $10N + 12$ components, corresponding to the variable $\underline{m}_N$.

It is necessary to determine the structure of the Jacobian matrix of constraints. The sparse structure of this matrix is illustrated in Figure 3.6, with N=5 used as an example for effective visualization. The thrust control constraints identify the constraints on the magnitude of the control (3.23). They are only depending on the components of the thrust control $\underline{\boldsymbol{u}}_i$ at each mesh point. The departure and arrival constraints identify the constraints on departure and arrival points (3.12),(3.13),(3.26),(3.27). Concerning the constraint on departure position, it is depending on t_0 and $\underline{\boldsymbol{r}}_0$ variables, while the departure velocity one is depending on t_0 and $\underline{\dot{\boldsymbol{r}}}_0$. On the other hand, the constraint on the arrival position is depending on t_0, Δt and $\underline{\boldsymbol{r}}_N$, while the arrival velocity one is determined by t_0, Δt and $\underline{\dot{\boldsymbol{r}}}_N$. Finally the Defect Constraints (3.28) for each interval $k \in [1, N]$ depend on Δt and on the state and control variables associated with the extreme nodes of the k-th interval, i.e. $\underline{\boldsymbol{s}}_{k-1}$, $\underline{\boldsymbol{u}}_{k-1}$, and $\underline{\boldsymbol{s}}_k$, $\underline{\boldsymbol{u}}_k$.

It is evident that the density of the Jacobian matrix decreases rapidly as the number of mesh nodes increases, becoming lower than 1 % for $N > 100$, thereby making the sparse optimization approach increasingly crucial and effective for the problem at hand, as illustrated in Figures 3.7 and 3.9.

Moreover, it should be noted that the matrix structure discussed thus far pertains to a low-thrust trajectory problem utilizing a two-body problem (2BP) model. If perturbations due to the presence of third bodies in the dynamics are to be considered, the structure can be readily adjusted by incorporating the t_0 dependence for all defect constraints. This adjustment is necessary because, at each point in the trajectory, the dynamics would be influenced by the positions of the perturbing bodies, which can be determined through ephemerides, which depend on the time and on the values of t_0 and Δt.

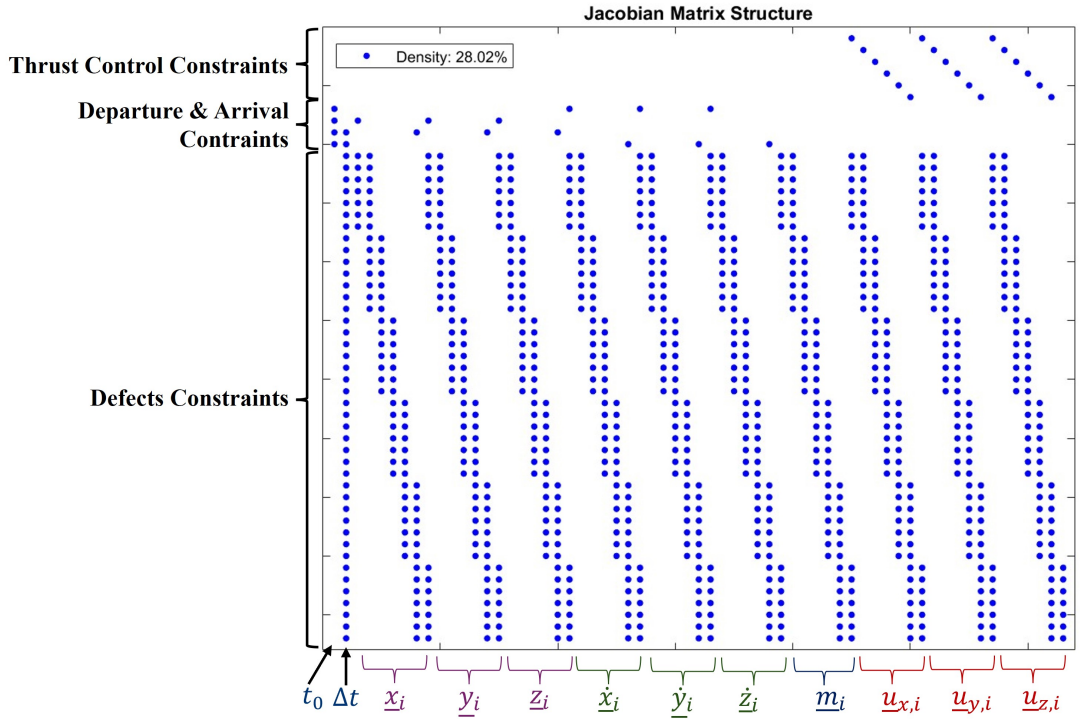

Figure 3.6: Jacobian matrix of constraints: sparsity pattern. The non-zero elements of the matrix are displayed in blue.

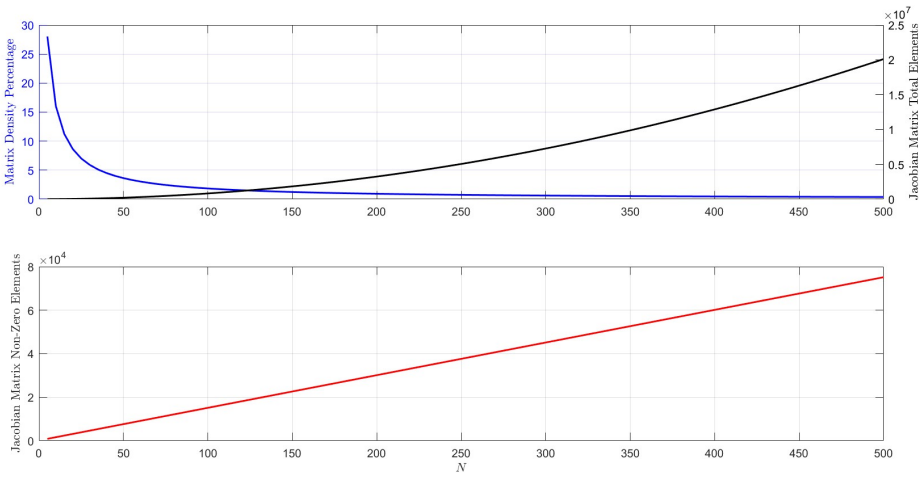

Figure 3.7: Plots representing sparsity trends as function of N (number of discretization intervals).

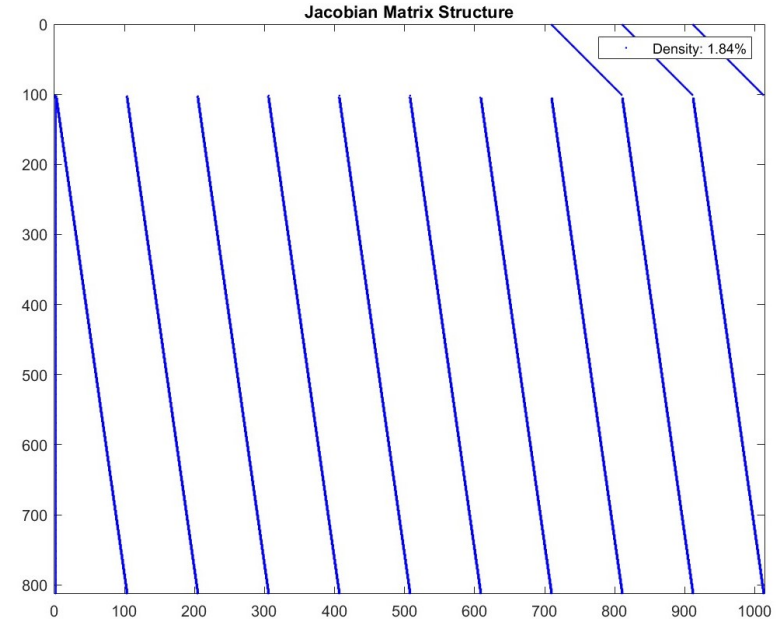

Figure 3.8: Jacobian matrix of constraints: structure visualization for N=100. The non-zero elements of the matrix are displayed in blue.

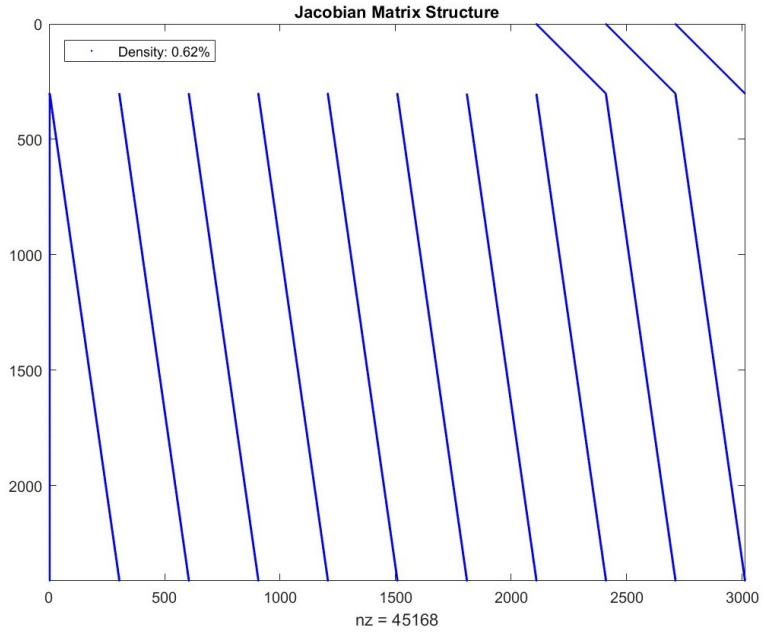

Figure 3.9: Jacobian matrix of constraints: structure visualization for N=200. The non-zero elements of the matrix are displayed in blue.

Chapter 4

Experimental Analysis and Applications in an Industrial Environment

This chapter presents the experimental analyses conducted within this thesis in an industrial setting. In Section 4.1, a comparative analysis of various solvers for simplified low-thrust trajectory optimization problems is introduced. This analysis aims to demonstrate the effectiveness of specific solvers for large-scale problems in comparison to generic commercial solvers, with particular emphasis on the critical role of sparsity implementation to solve the problem. Section 4.2 details the development of an operational tool for optimizing interplanetary low-thrust trajectories, followed by its validation in the context of Martian missions, referencing the analysis reported in Casanova-Álvarez, Navarro-Medina, and D.Tommasini, 2024[10]. The refinement post-processing of the optimal solution found by the optimization tool is then described, by focusing on the errors evaluation and the process necessary to enhance the accuracy of the solution, also accounting for third-body perturbations in the equations of motions.

4.1 Solver Performances Analysis

This section aims to compare the performance of two general-purpose solvers used in both academic and industrial settings for solving nonlinear problems, that specifically are *Matlab Optimization Toolbox* (see MathWorks, 2024[27]) and *We Optimize Really Huge Problems - WORHP* solver (see Büskens and Wassel, 2013[8]). The *Matlab Optimization Toolbox* provides functions for identifying parameters that minimize or maximize objectives while adhering to constraints. It includes solvers for different kinds of programming problems. Specifically, for NLP problems, the *fmincon* routine is applicable. Optimization problems can be defined using functions and matrices or by specifying variable expressions that reflect the underlying mathematics. Additionally, automatic differentiation of

objective and constraint functions can be utilized for faster and more accurate solutions.

Subsequently, the *WORHP* solver is analyzed. It is designed for tackling large-scale, sparse, nonlinear optimization problems involving up to millions of variables and constraints. *WORHP* is fundamentally built as a sparse SQP and IP method. It features efficient routines for computing sparse derivatives using finite difference techniques. Additionally, it utilizes reverse communication, providing an high level of interaction between the user and the solver. This solver was selected by the ESA as the European NLP solver due to its exceptional robustness and its philosophy of application-driven design and development.

The two solvers are employed to address a range of test cases, as detailed below, with a comparative analysis of their results and computational performances. This comparison aims to illustrate the necessity of selecting effective solvers for large-scale NLP problems in an industrial context, thereby ensuring significant resource savings.

4.1.1 Test Cases

A series of test cases were conducted to facilitate a comparative analysis of the solvers. Each test case is firstly solved by handling the dense problem with *fmincon* routine and *WORHP* solver without the implementation of sparsity structure. Then, the same cases are tested by implementing the sparse problem (see Section 3.2.3) within the *WORHP* solver. The analysis commenced with simplified interplanetary trajectory optimization problems and progressively introduced increasing levels of complexity by incorporating additional nonlinear constraints. Ultimately, the test cases evolved to reflect problems more representative of real-world scenarios. This subsection provides a detailed description of the various test cases and includes visualizations of the results obtained from some of the more significant tests. The comprehensive results are compiled in summary tables presented in subsection 4.1.2, accompanied by comments and conclusions. All test cases computed within this section assume the patched conics approach, by accounting for the 2BP with the Sun as central body.

Nuclear Electric Simple Raising between circular orbits (NE-SR)

The initial test case addresses the resolution of a simplified Earth-Mars interplanetary maneuvering problem. Specifically, it involves a straightforward transfer maneuver from Earth's orbit to Mars' orbit utilizing nuclear electric propulsion (NE) model. The orbits are approximated as circular, with the radii of Earth and Mars assumed to be constant at 1 AU and 1.5 AU, respectively.

Consequently, the trajectory optimization requires only that the S/C reaches the simplified Martian orbit, without any rendezvous conditions or constraints on the satellite's final velocity. Therefore, the constraints outlined in Eq. 3.27 are not considered. However, all other nonlinear constraints previously described in

the model are incorporated by reformulating the initial conditions specified in Eq. 3.24 and Eq. 3.25, such that $\boldsymbol{r}_0 = [1, 0, 0]\, AU$, $v_{escape,max} = 0.0\, km/s$, and $\boldsymbol{v}_{DP} = [0, v_{Earth}, 0]$, where $v_{Earth} = \sqrt{\mu_{Sun}/r_{Earth}} = 27.785\, km/s$. The arrival position condition expressed in Eq. 3.26 is reformulated such that $\|\boldsymbol{r}_N\| = 1.5\, AU$.

The propulsion system, based on the nuclear electric propulsion model, is characterized by a constant maximum available thrust level along the trajectory, denoted as $T_{max} = 250\, mN$. The specific impulse is also considered constant, with a value of $I_{sp} = 4010\, s$. The total initial mass of the spacecraft is set at $m_0 = 2500\, kg$. In this simplified scenario, the variable t_0 does not influence the optimization problem, as the starting point is defined as an independent parameter and there is no t_0-dependent constraint on the endpoint. Consequently, this variable is not considered in the initial test case. For the NE-SR1 test case, a constraint on Δt within the range [0,730.51] days was imposed, whereas in NE-SR2, the upper bound for the time of flight was reduced to 636.00 days.

A propagated trajectory with maximum tangential thrust acting throughout the transfer time was used as the initial guess for the optimization, as illustrated in Figure 4.1. The optimization results for the two solvers are presented in Figures 4.2, 4.3, 4.4, 4.5, 4.6, and 4.7, showing the optimized trajectories in 2D X-Y elliptical J2000 plane , the thrust profile and the S/C mass over the transfer elapsed days.

As expected, the optimization results in multi-revolution trajectories that saturate the ToF constraint, utilizing the maximum possible flight time to minimize consumption. Additionally, the solutions tend to concentrate the thrust arcs in the low-radius regions of the trajectory, where it is theoretically more advantageous to apply thrust. Furthermore, it is evident that decreasing the upper bound for the ToF leads to an increase in optimal consumption in NE-SR2 compared to NE-SR1.

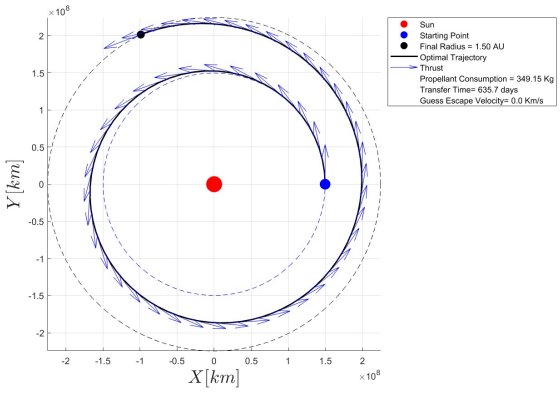

Figure 4.1: NE-SR1 and SR2 : Initial Tangential Guess

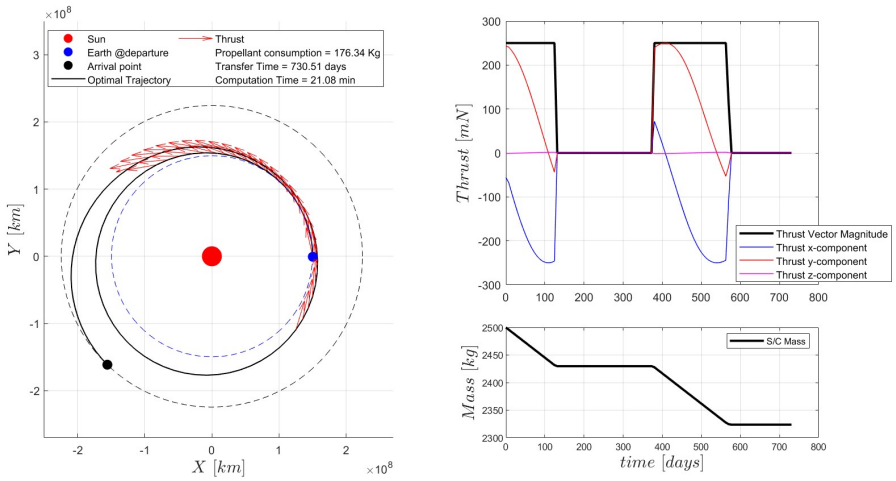

Figure 4.2: NE-SR1: *fmincon* Optimal Solution

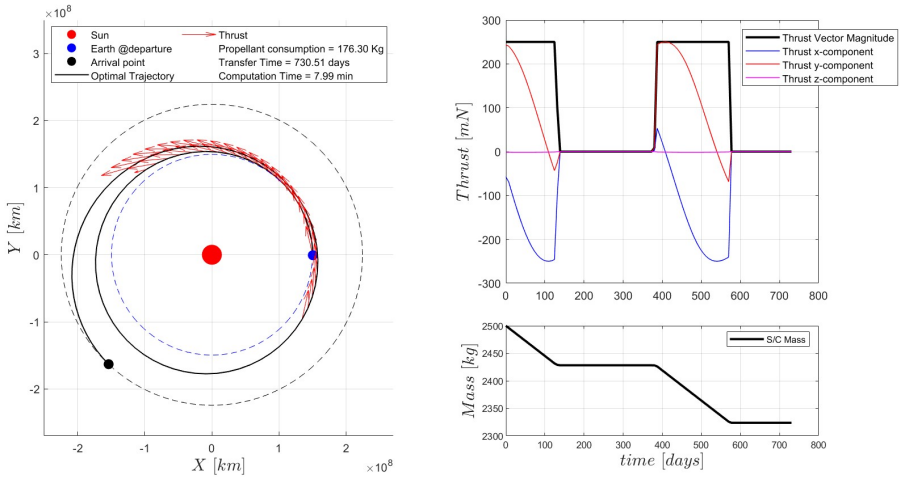

Figure 4.3: NE-SR1: *WORHP(dense)* Optimal Solution

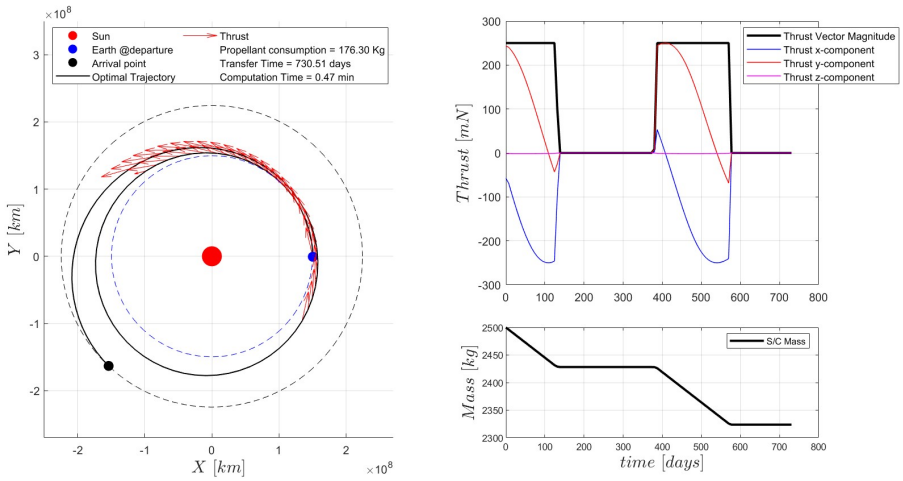

Figure 4.4: NE-SR1: *WORHP (sparse)* Optimal Solution

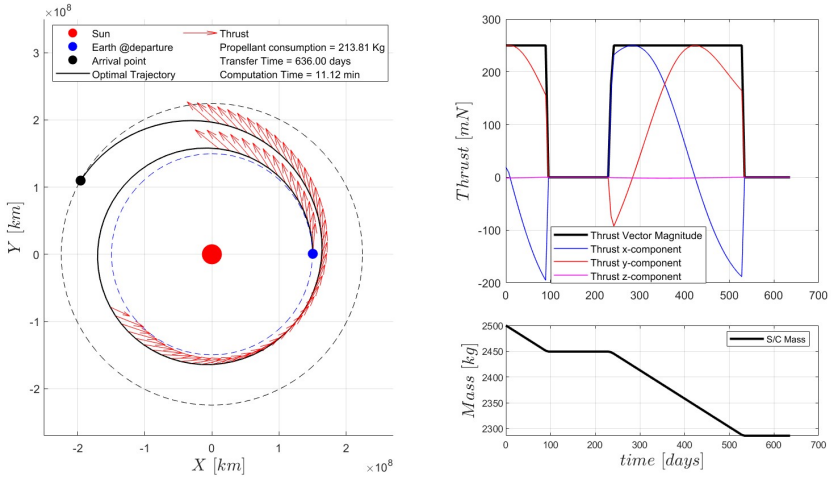

Figure 4.5: NE-SR2: *fmincon* Optimal Solution

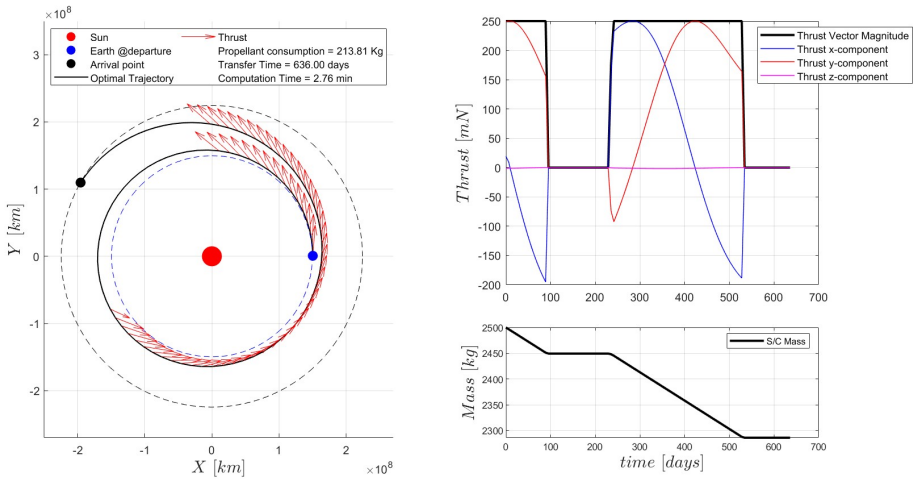

Figure 4.6: NE-SR2: *WORHP (dense)* Optimal Solution

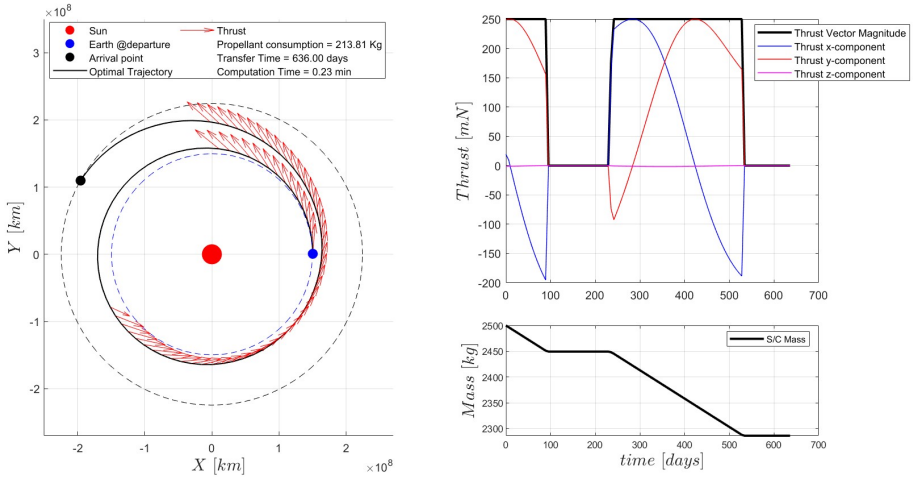

Figure 4.7: NE-SR2: *WORHP (sparse)* Optimal Solution

Nuclear electric raising between circular orbits with circular target velocity (NE-CR)

The objective of the following test case is to optimize the transfer trajectory from a circular Earth orbit to a circular Mars orbit, defined and approximated as in the previous test case. The optimization model is based on the same constraints expressed for the NE-SR problem, with the addition of the constraint that the spacecraft must reach Mars' orbit with a velocity that matches the circular velocity of that orbit, defined as a function of the mean anomaly θ as follows:

$$\boldsymbol{v}_{AP}(\theta) = \begin{cases} -v_{Mars}\, sin(\theta) \\ v_{Mars}\, cos(\theta) \\ 0 \end{cases} \tag{4.1}$$

Where $r_{Mars} = 1.50\, AU$, $v_{Mars} = \sqrt{\mu_{Sun}/r_{Mars}} = 24.32\, km/s$.

The thrust model and the initial mass of the spacecraft are identical to those in the previous test case. Similarly, the variable t_0 does not affect the optimal control problem and is therefore neglected. This test case examines nine specific scenarios. For all cases, an upper bound for the time of flight of 730.51 days is assumed.

In the first case (NE-CR1), the escape velocity from Earth is set to 0 km/s, and an initial guess solution propagated with tangential thrust is used, as shown in Figure 4.1. For all other cases (from NE-CR2 to NE-CR8), a maximum direct escape velocity from Earth of 3.0 km/s is imposed. The problem to be solved is thus the same for all these cases, but the difference lies in the choice of the initial guess solution.

For all cases, initial guesses propagated with maximum tangential thrust throughout the transfer are provided, but with different initial velocities of the S/C associated with different Earth escape velocities. In NE-CR2, the guess solution is characterized by an escape guess velocity of 0.0 km/s, which is incremented by 0.5 km/s for each subsequent case, up to an escape guess velocity of 3.0 km/s for NE-CR8. This approach allows for the observation of how different guess solutions can impact the results and performance of the optimizer for the same optimal control problem. Examples of different guesses are provided in Figures 4.8 and 4.9.

All results obtained by the solvers are presented in Tables 4.1 and 4.2, while a representative example of the outputs is shown in Figures 4.10, 4.11, and 4.12. As expected, the NE-CR1 case exhibits significantly higher propellant consumption compared to the other cases, as the ability to utilize a non-zero direct escape velocity from Earth allows for substantial savings in propellant mass. Furthermore, this parameter greatly influences the optimal ToF, as this constraint is saturated in the NE-CR1 case, whereas the other cases demonstrate significantly lower optimal transfer times.

Upon examining the results, it is evident that the choice of the initial guess significantly influences both the performance and the outcomes obtained. Specifically, each case identifies a different local optimum, resulting in variations in escape velocity and corresponding differences in ToF. Despite these differences, all cases exhibit a very similar trajectory geometry at a macroscopic level.

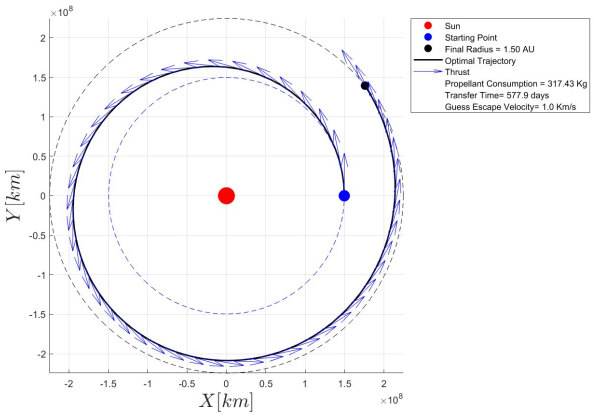

Figure 4.8: NE-CR4: Guess Solution

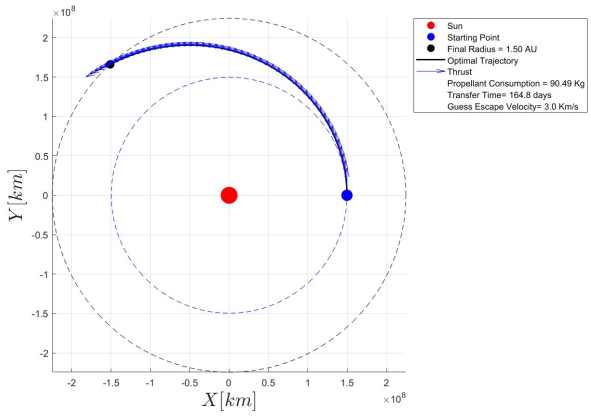

Figure 4.9: NE-CR8: Guess Solution

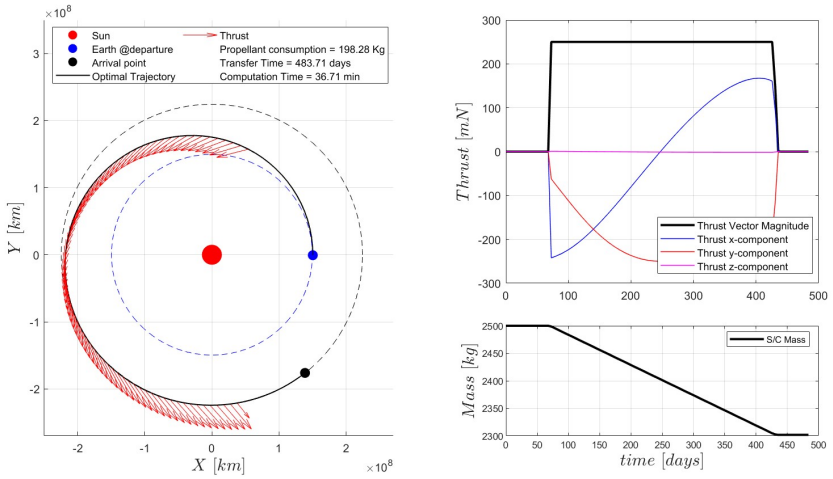

Figure 4.10: NE-CR8: *fmincon* Optimal Solution

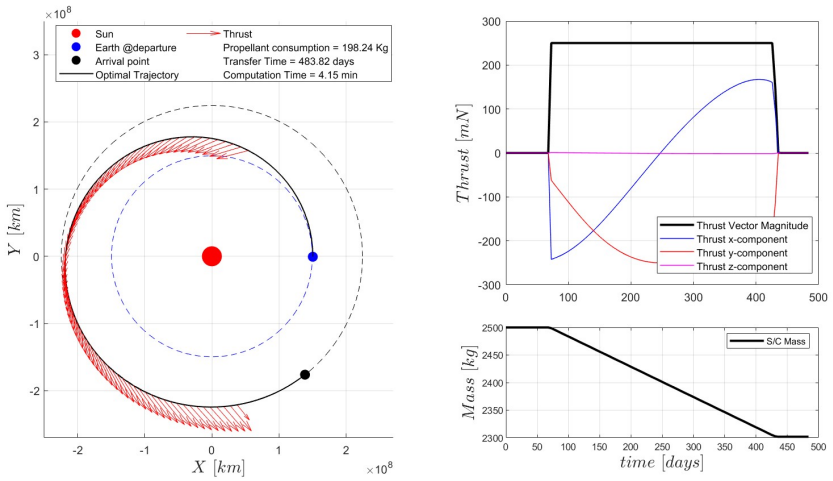

Figure 4.11: NE-CR8: *WORHP(dense)* Optimal Solution

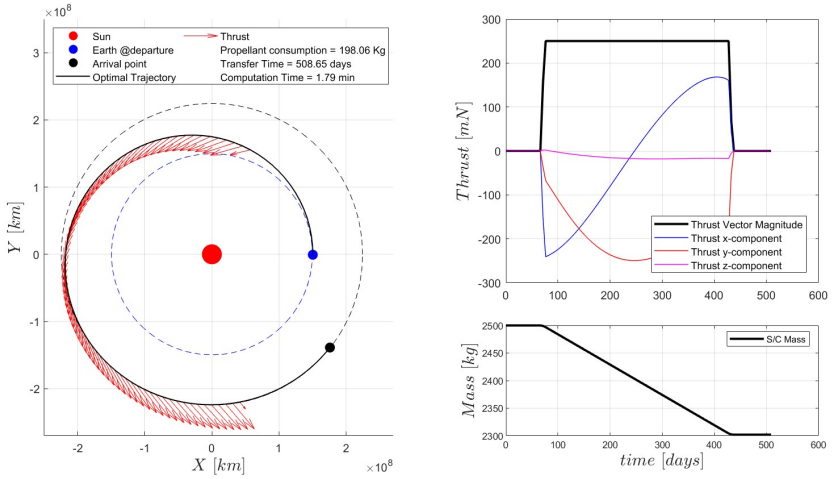

Figure 4.12: NE-CR8: *WORHP (sparse)* Optimal Solution

Nuclear electric rendezvous between circular orbits (NE-CRV)

The objective of the NE-CRV test cases is to optimize a simplified Earth-Mars rendezvous interplanetary trajectory, assuming coplanar circular orbits of the planets and defining simplified ephemerides by the user. This test case account for the same assumption on propulsion system features and S/C mass properties as the previous tests. This test case serves as a final intermediate step before addressing the rendezvous case with real orbits, which will be discussed subsequently. The optimization model is based on the general framework presented in Chapter 3 and utilizes arbitrarily defined simplified ephemerides as follows. For a generic planet, its position and velocity are functions of time such that

$$\theta(t) = \theta_0 + \omega t \ , \tag{4.2}$$

where $\theta(t)$ is the mean anomaly of the planet as function of time, θ_0 is the user-defined anomaly of the planet at $t = t_0$, and ω represents the mean angular velocity of the planet. Thus, it is possible to define the position and velocity vectors of the planet at time t as follows

$$\boldsymbol{r}(t) = \begin{cases} r\,cos(\theta(t)) \\ r\,sin(\theta(t)) \\ 0 \end{cases} \tag{4.3}$$

$$\boldsymbol{v}(t) = \begin{cases} -v\,sin(\theta(t)) \\ v\,cos(\theta(t)) \\ 0 \end{cases} \tag{4.4}$$

Where v represents the circular velocity magnitude of the planet along its simplified circular orbit with radius r.

In this case, unlike the previous test cases, the variable t_0 and the value of θ_0 assigned to the planets of interest become crucial. For the NE-CRV1 test case, $\theta_{0,Earth} = 0\,deg$, and $\theta_{0,Mars} = 0\,deg$ was assumed, and a tangential guess with maximum thrust and zero escape velocity was provided. The optimization results with these input parameters are shown in Figures 4.13, 4.14 and 4.15.

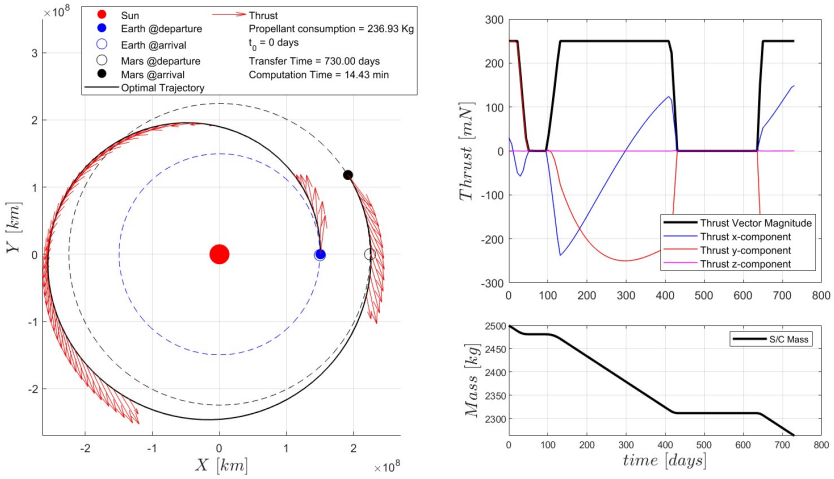

Figure 4.13: NE-CRV1: *fmincon* Optimal Solution

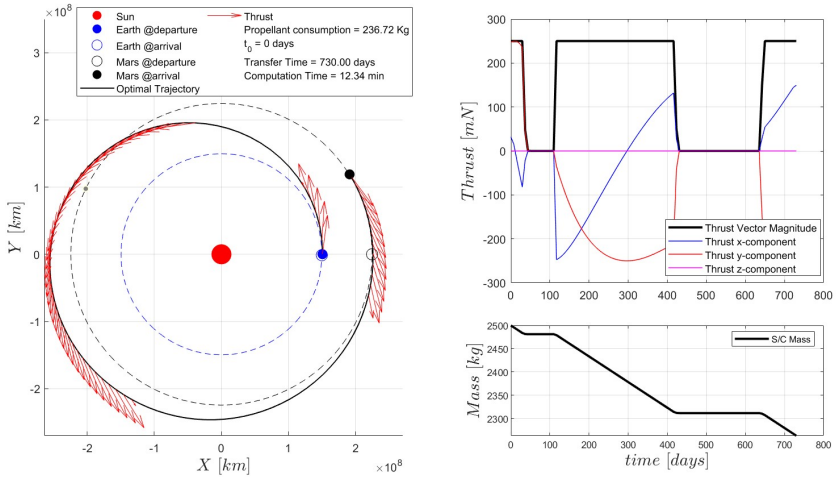

Figure 4.14: NE-CRV1: *WORHP(dense)* Optimal Solution

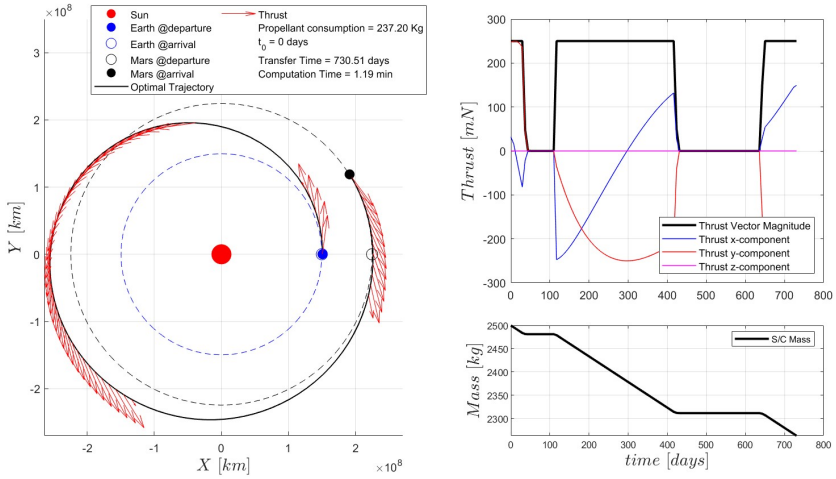

Figure 4.15: NE-CRV1: *WORHP (sparse)* Optimal Solution

It can be observed that the solver found an optimal solution for $t_0 = 0$, identifying a local minimum for this variable value. However, it is evident that this variable significantly influences the rendezvous problem compared to all other variables in the optimization problem. The solver, however, is unable to recognize the importance of this variable among the large number of variables it must handle. Therefore, it tends to find a local optimal solution around the guess value of this departure date variable.

Indeed, it can be confirmed that the solutions found in the NE-CR test cases could also satisfy this problem, provided the correct initial phasing of the planets determined to enable the rendezvous. The value of θ_0 required to establish the initial condition, ensuring that the solutions derived in the previous tests also satisfy this problem, can be determined through appropriate calculations. This

was specifically applied in the NE-CRV2 test case, setting $\theta_{0,Earth} = 0\,deg$, and $\theta_{0,Mars} = 48\,deg$. The optimization results can be seen in Figures 4.16, 4.17, and 4.18.

It can be observed that in this case, the solvers find another local optimum, different and better than the one previously computed, similar to the solutions of the previous test cases. This confirms the importance of the t_0 value and the necessity of a global search method for solutions based on the management of this variable. This will be further explored in Section 4.2.

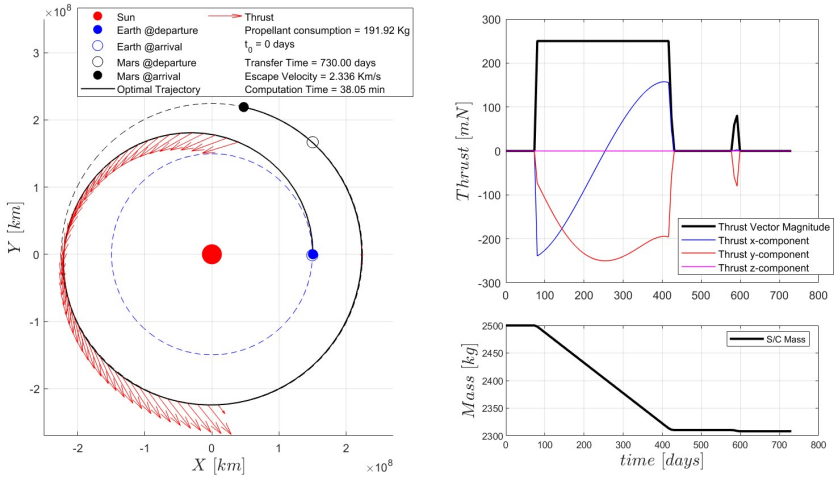

Figure 4.16: NE-CRV2: *fmincon* Optimal Solution

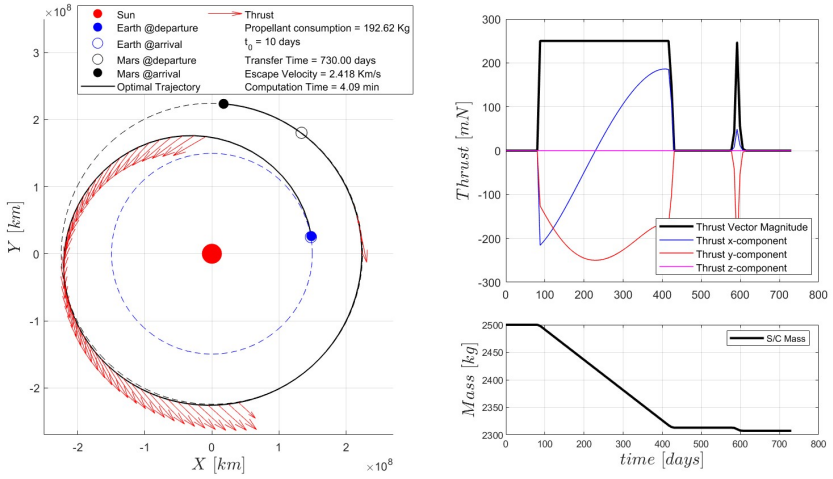

Figure 4.17: NE-CRV2: *WORHP(dense)* Optimal Solution

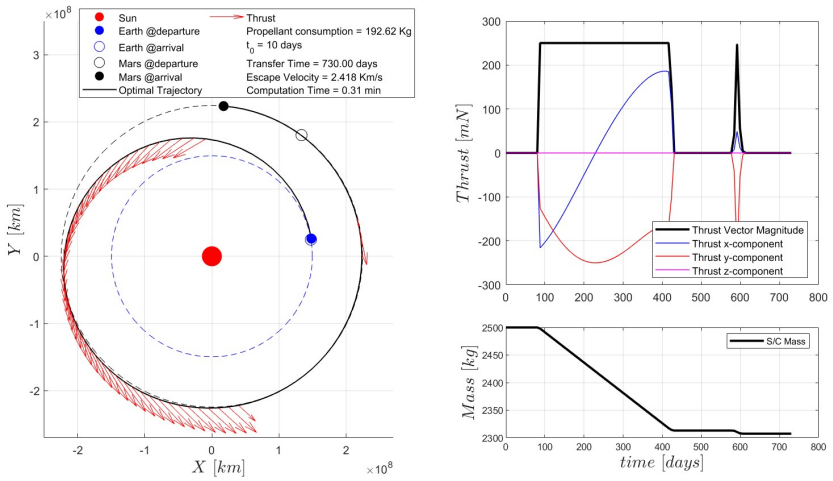

Figure 4.18: NE-CRV2: *WORHP (sparse)* Optimal Solution

Solar electric rendezvous between real orbits (SE-RRV)

This test case represents the real optimal control problem for low-thrust interplanetary rendezvous maneuvers. The ephemerides of the planets are calculated using the SPICE tool (see Section 3.1.6), and the propulsion system is modeled as solar electric propulsion (SE). Specifically, the profile of maximum available thrust as a function of the distance from the Sun is the one introduced in Section 3.1.2 and shown in Figure 3.1.

Given the critical importance of managing the departure date variable, a comprehensive global approach to identifying solutions is essential. This methodology is elaborated in detail in Section 5.1. From the solutions derived in the case study, an estimate for the variable t_0 was identified, pinpointing a local solution to the specific problem on 2026 OCT 06. The problem was subsequently addressed using a tangential-thrust guess, characterized by the aforementioned departure date. Constraints were applied to the ToF, ensuring it remained within the interval of [200, 550] days, and to the departure date t_0, allowing for a launch window of $\pm$ 1 day. The results are presented in Figures 4.19 and 4.20. It is noteworthy that the problem was resolved exclusively using the *WORHP* solver, as previous test cases (refer to Tables 4.1 and 4.2) demonstrated that the *Matlab's fmincon* solver was not computationally adequate for handling such a complex problem.

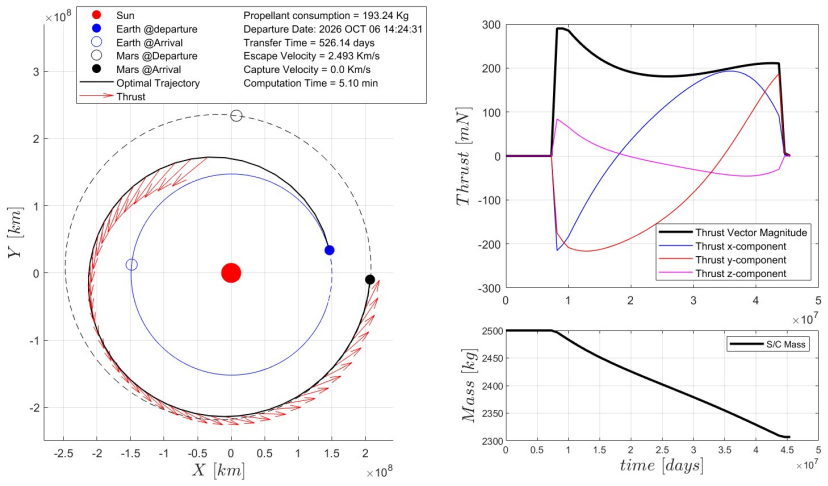

Figure 4.19: SE-RRV1: *WORHP(dense)* Optimal Solution

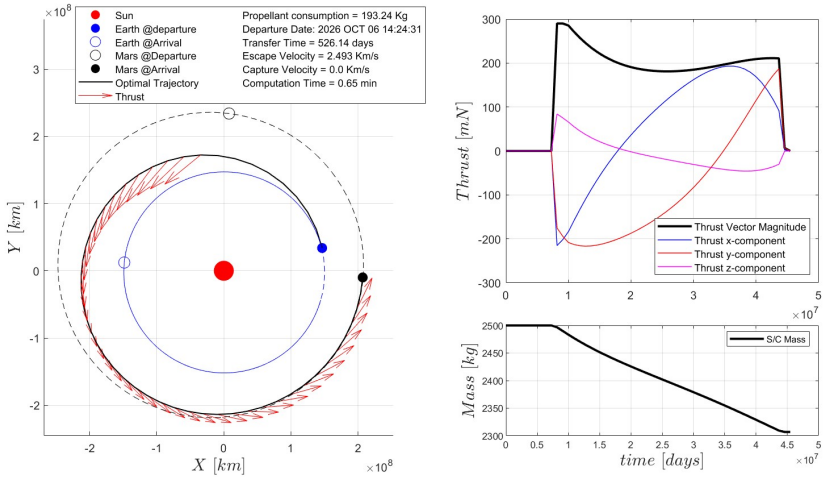

Figure 4.20: SE-RRV1: *WORHP (sparse)* Optimal Solution

4.1.2 Solver Outputs Comparison

Tables 4.1 and 4.2 provide a summary of the main outputs from the optimizations conducted in the previously described test cases. This summary includes both the key results and the computational performances of *Matlab's fmincon* and the *WORHP solver*, with and without the implementation of the sparse matrix structure.

Specifically, Table 4.1 presents a comparison between *Matlab's fmincon* and *WORHP*, both of which address the problem using a dense numerical approach. It is important to note that, to the extent possible, the same optimization parameters were set for both solvers. Among the most critical parameters are the

feasibility tolerance, set at 1×10^{-6}, and the optimality tolerance, which ensures compliance with the KKT conditions, also set at 1×10^{-6}. Furthermore, for each test case, both solvers were provided with the same initial guess solution, specifically those with tangential thrust.

The results indicate that both solvers converge to nearly identical solutions for each test case; however, there is a significant difference in computational performance. It is evident that in all cases, *WORHP* outperforms the *fmincon* routine, particularly when handling guess solutions that do not satisfy all the constraints of the control problem.

For instance, in the NE-SR1 and NE-SR2 cases, where the initial guess is already a feasible solution, *WORHP* requires at least half the computational time compared to *fmincon* to converge to the local optimum. Observing the results for other cases, where the initial solution does not meet all the optimization constraints, the performance discrepancy increases further. This behavior underscores *fmincon*'s difficulty in handling guesses, whereas *WORHP* demonstrates robust performance regardless of the initial guess. Specifically, in test cases NE-CR2 to NE-CR8, *fmincon* exhibits highly variable computational times, ranging from 13.93 minutes for NE-CR3 to 52.26 minutes for NE-CR6, when solving the same problem with different initial guesses. In contrast, *WORHP* maintains relatively stable performance across varying guesses, with times ranging from 3.79 minutes for NE-CR4 to 8.25 minutes for NE-CR2, achieving computational times up to 10 times faster than the other solver.

	MatLab's fmincon				**WORHP Dense**			
Cases	CPU time [min]	m_{prop} [kg]	ToF [days]	v_{escape} [km/s]	CPU time [min]	m_{prop} [kg]	ToF [days]	v_{escape} [km/s]
NE-SR1	21.075	176.34	730.51	0.00	7.99	176.30	730.51	0.00
NE-SR2	11.12	213.81	636.00	0.00	2.76	213.81	636.00	0.00
NE-CR1	12.00	302.55	730.51	0.00	7.03	302.65	730.51	0.00
NE-CR2	20.13	195.27	730.51	2.255	8.25	195.26	730.51	2.243
NE-CR3	13.93	195.27	730.51	2.256	4.36	195.27	730.51	2.259
NE-CR4	21.52	195.27	730.51	2.256	3.79	196.77	653.16	2.349
NE-CR5	24.87	197.04	588.08	2.310	3.98	197.07	586.38	2.312
NE-CR6	52.26	198.20	431.29	3.000	4.62	198.23	485.34	2.281
NE-CR7	45.67	197.65	539.31	3.310	4.09	197.67	539.36	2.311
NE-CR8	36.71	198.28	483.71	2.291	4.15	198.24	483.82	2.288
NE-CRV1	14.43	236.93	730	3.000	12.34	236.72	730	3.000
NE-CRV2	38.05	193.92	730	2.336	4.09	192.62	730	2.418

Table 4.1: Summary Table of comparison between *MatLab's fmincon solver* and *WORHP dense solver* (i.e. without sparsity implementation), reporting the main optimization outputs for each test case

However, it is important to note that even when using WORHP with a dense

structure, although the performance is significantly better than Matlab's nonlinear solver and typically on the order of a few minutes, it remains insufficient for this type of problem. This is because a global search for the optimal solution is required, necessitating the evaluation of hundreds, if not thousands, of local solutions. The significant advantage of such a solver lies in its ability to implement the sparse structure of the Jacobian matrix and gradient vector, which can further accelerate computational time. This potential was extensively leveraged in this thesis, as detailed in Section 3.2.3, and was tested across all previous cases, as illustrated in Table 4.2.

	WORHP Dense				**WORHP Sparse**			
Cases	CPU time [min]	m_{prop} [kg]	ToF [days]	v_{escape} [km/s]	CPU time [min]	m_{prop} [kg]	ToF [days]	v_{escape} [km/s]
NE-SR1	7.99	176.30	730.51	0.00	0.47	176.30	730.51	0.00
NE-SR2	2.76	213.81	636.00	0.00	0.23	213.81	636.00	0.00
NE-CR1	7.03	302.65	730.51	0.00	1.35	302.65	730.51	0.00
NE-CR2	8.25	195.26	730.51	2.243	1.57	195.26	730.51	2.243
NE-CR3	4.36	195.27	730.51	2.259	0.74	195.26	730.51	2.262
NE-CR4	3.79	196.77	653.16	2.349	0.79	196.77	653.16	2.349
NE-CR5	3.98	197.07	586.38	2.312	0.17	197.05	587.47	2.311
NE-CR6	4.62	198.23	485.34	2.281	0.16	198.23	486.34	2.251
NE-CR7	4.09	197.67	539.36	2.311	0.22	197.58	545.14	2.313
NE-CR8	4.15	198.24	483.82	2.288	1.79	198.06	508.65	3.000
NE-CRV1	12.34	236.72	730	3.000	1.19	237.2	730	3.000
NE-CRV2	4.09	192.62	730	2.418	0.31	192.62	730	2.418
SE-RRV1	5.10	193.24	526.14	2.493	0.65	193.24	526.14	2.493

Table 4.2: Summary Table of comparison between *WORHP dense solver* and *WORHP sparse solver* (i.e. with sparsity implementation), reporting the main optimization outputs for each test case

The results clearly demonstrate the effectiveness of the sparse approach. In most cases, computational times are significantly reduced to well below one minute, and for the most demanding cases, around one minute. This approach achieves performance improvements up to 10 times better than the dense approach with the same solver and up to 100 times faster than *Matlab's fmincon*, which also uses a dense approach. For instance, considering the worst-case scenario for the sparse structure's performance, the NE-CR8 case, the computational time is reduced to 1.79 minutes, compared to 4.15 minutes for *WORHP* with a dense structure and 36.71 minutes for *fmincon*. Conversely, in the NE-CR6 case, which presents the smallest computational time for sparse *WORHP* at 0.16 minutes, the dense *WORHP* takes 4.62 minutes, and *Matlab's fmincon* takes 52.26 minutes.

In conclusion, it can be asserted that the utilization of solvers specifically designed for large-scale nonlinear problems with a sparse structure is essential for drastically reducing computational time. This enables the execution of multiple local optimizations to thoroughly explore the solution space in search of the optimum. It is evident that such a process would be unsustainable in terms of computational timing when using dense solvers, particularly those not originally intended for large-scale problems.

4.2 Developed Optimization Tool

This section describes the purpose and operational structure of the optimization tool for low-thrust interplanetary trajectories, developed in industry within the mission analysis unit of Thales Alenia Space (Turin, Italy). The detailed purpose and architecture of this tool are described in Section 4.2.1. Its validity is subsequently assessed in Section 4.2.2 through comparison with feasibility studies of Earth-Mars electric propulsion missions, as documented in the literature by Casanova-Álvarez, Navarro-Medina, and D.Tommasini, 2024[10]. Section 4.2.3 presents the post-processing of the optimal solution found by the optimization tool in order to enhance its accuracy. The process of accuracy refinement is described, concerning refinement in collocation techniques and dynamics modeling, including perturbations due to third-bodies gravitational fields along the spacecraft interplanetary trajectory.

4.2.1 Tool Architecture

The objective of the developed tool is to determine the optimal interplanetary trajectory in terms of propellant consumption (and consequently ΔV) for a direct interplanetary transfer (without the possibility of fly-bys). The goal is to perform multiple local optimizations within a given launch window, starting from relatively simple guess solutions. This approach aims to identify the most promising specific launch windows and strive to converge towards the global optimal solution within the predefined domain. From an implementation perspective, the software is developed in the Matlab environment and interfaces with the WORHP solver for resolution, utilizing the sparse structure of the optimization problem.

The architecture of the tool, as depicted in the diagram in Figure 4.21, is based on a multi-start plus SQP approach (as explained in Section 2.4.1), inspecting the solution space at two levels. Specifically, for this problem, the critical variable in identifying local optima is t_0 (i.e. launch epoch or departure date). Therefore, the multi-start approach generates multiple guess solutions characterized by different t_0 values within the considered bounds.

At this point, the process advances to the second level of the search (low-level search), where the most promising regions of launch dates, identified from the

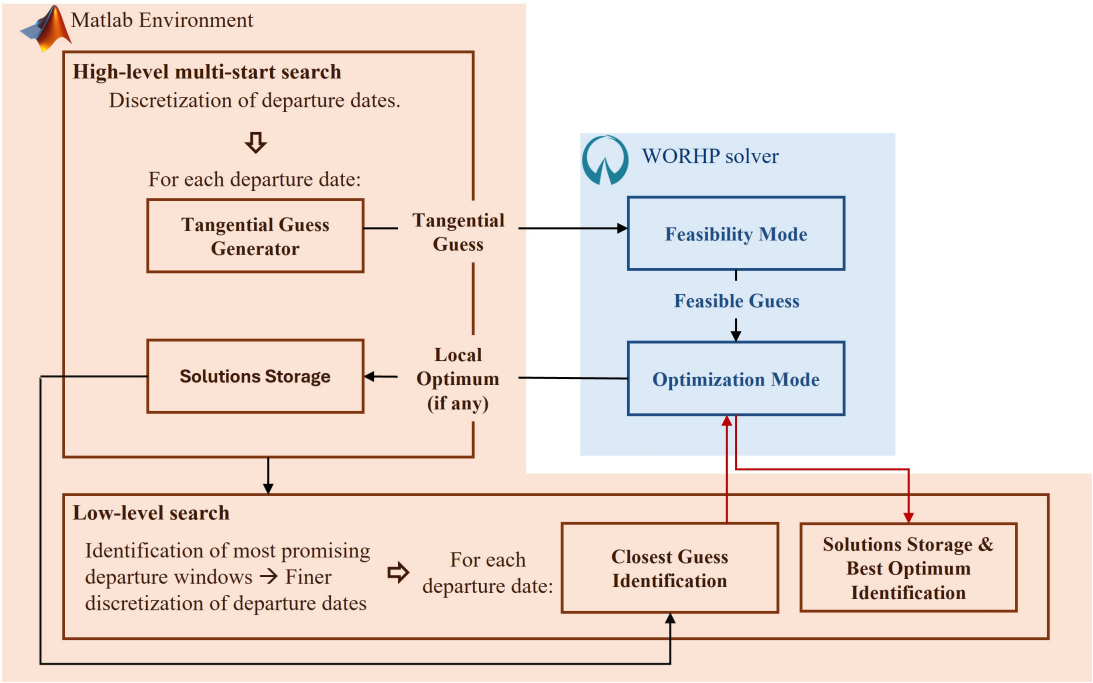

Figure 4.21: Optimization tool architecture diagram

previously found solutions, are refined. This involves increasing the discretization of the departure dates within the identified region and re-optimizing for each of them. The guess provided at this level is selected from the high-level solutions storage, identifying the one with the t_0 value closest to the target. The guess solution used at this stage is already feasible, as it is a solution to the problem. Therefore, the solver is used directly in optimization mode, yielding a new optimal solution within the refined interval of the launch date of interest. This process results in multiple new detailed solutions, allowing the identification of the best solution found, which is then designated as the global optimal solution within the domain of departure dates.

The entire process described above is conducted using direct collocation methods for trajectory optimization, initially with a relatively high level of approximation to benefit computationally. In the case studies within the thesis, collocation is initially modeled using the trapezoidal method with an equally spaced mesh of N=100 intervals, in order to approximate the equations of dynamics by accounting for the two-body problem. The global optimal solution identified in this manner is thus a high-level approximation that requires post-processing refinement to enhance its fidelity and precision in terms of dynamics. This refinement process is detailed and applied in Section 4.2.3.

In the first level of search (high-level multi-start search), the interval of launch epochs to be inspected is discretized on a relatively coarse mesh of various departure dates. For each descretized value in the grid, a tangential thrust guess is generated with the initial condition being the departure planet and an escape ve-

locity chosen by the user within the permissible v_{escape} range. This guess results from propagating the tangential control until a user-defined radius is reached, coinciding with the average orbit radius of the arrival planet. Generally, this guess, while feasible in terms of dynamics, does not satisfy the rendezvous constraint. Therefore, it is processed by the WORHP solver in feasibility mode, aiming solely to satisfy the constraints. The feasibility mode outputs a feasible guess for the problem, if found, which is then used as the initial guess for the solver in optimization mode. This process yields a local optimal solution within the sub-interval of launch epochs defined by the initial high-level mesh, which is then stored in a solutions storage.

4.2.2 Tool Validation

Earth-Mars Transfer with Constant Available Thrust Model

The optimization tool developed, as described in the previous section, is validated here by comparing the results with those proposed by Casanova-Álvarez, Navarro-Medina, and D.Tommasini, 2024[10]. The case study aims to optimize a direct Earth-Mars transfer using a constant thrust model, analogous to the NE model. The characteristics and constraints of the optimization are summarized in Table 4.3.

Optimization Inputs	**Value**	**Unit**
Launch Mass (m_0)	2000	[kg]
Max available propellant	500	[kg]
Max available thrust level (T_{max})	0.3 (constant)	[N]
Specific Impulse (I_{sp})	2029	[s]
Launch Epochs Range	[Jan 01, 2030; Jan 01, 2040]	
ToF Bounds	[200, 365]	[Days]
Escape velocity	2.94	[km/s]
Max arrival velocity	0.001	[km/s]

Table 4.3: Optimization Input Parameters and Constraints

The high-level multi-start search was conducted by discretizing the launch epochs range with a step of 4 days, generating 913 sub-intervals of departure windows. To decrease the overall computation time, Matlab's Parallel Computing Toolbox was utilized (see MathWorks, 2024[28]), parallelizing the calculations across 10 cores ($N_{cores} = 10$). Out of the 913 trials, 77 solutions converged, as depicted in the graph in Figure 4.22, which shows the propellant consumption of each solution and the corresponding departure date. This high-level search process took 2.21 hours to complete.

In Figure 4.22, it is evident that the region of solutions identified in the reference paper, specifically the launch window in 2031, is well delineated. However, other launch windows appear to be more promising in the subsequent years. Notably, the 2037 launch window presents the best solution among those found at this level. Following this initial analysis, the low-level search is conducted within the two aforementioned windows.

In particular, the optimization search within the launch window from January 1, 2031, to March 1, 2031, was refined using a new discretization step of 0.5 days. This approach resulted in 119 trials, of which 116 successfully converged, within 3.87 minutes of parallelized computation. It is interesting to note how the calculation of over 100 local solutions to the problem is remarkably fast in this case, due to the availability of good initial guesses from high-level research. The solutions obtained are summarized in Figure 4.23. A convex curve of optimal propellant consumption as a function of the departure date is easily identifiable, from which a local minimum within this launch window can be determined. The minimum propellant consumption is 368.388 kg, with the departure date identified as January 31, 2031 at 06:00. It is possible to compute the ΔV of this maneuver, defined as follows

$$\Delta V = \int_{t_0}^{t_0+\Delta t} \frac{\|\boldsymbol{T}(t)\|}{m(t)}\, dt \tag{4.5}$$

Thus, it can be integrated numerically with trapezoidal method and it yields $\Delta V = 4.051\, km/s$. The detailed results of this trajectory solution are shown in the following Figures 4.24, 4.25, and 4.26.

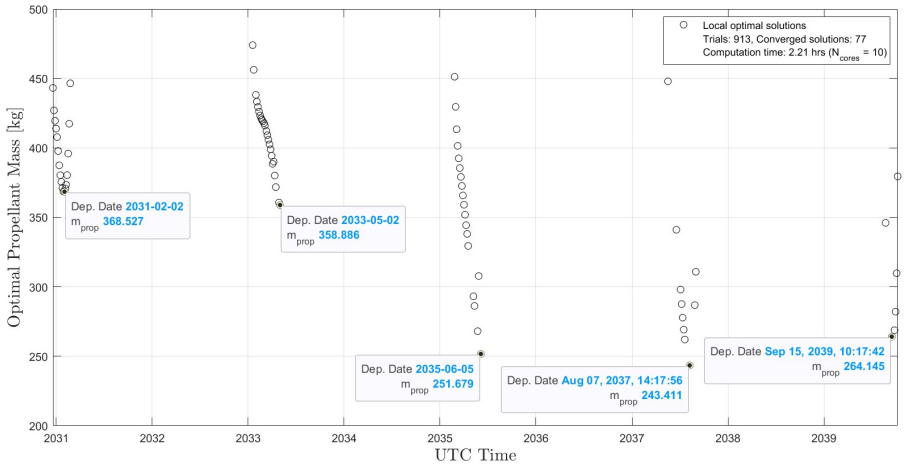

Figure 4.22: High level multi-start search summary plot. Details on the the local minima for each identified launch window.

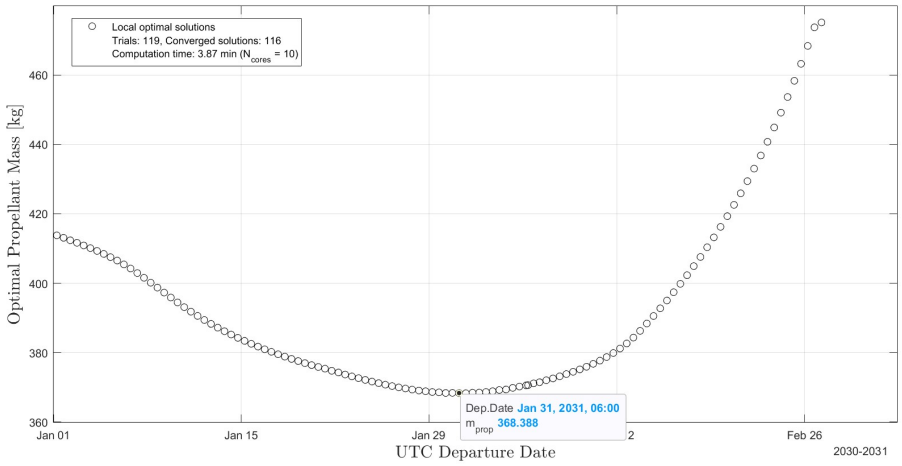

Figure 4.23: Low level search summary plot. Launch epochs window: Jan 01, 2031 - Mar 01, 2031. Detail on the minimum-propellant solution for the launch window.

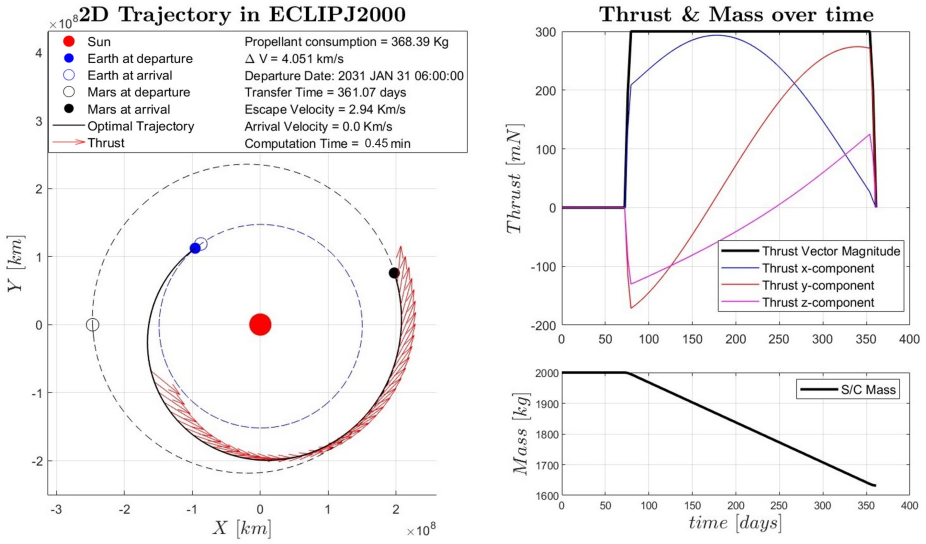

Figure 4.24: Optimal Solution of the 2031 launch window. On the left: 2D Trajectory in ECLIPJ2000 X-Y plane with its main features. On the right: thrust and mass over transfer elapsed time.

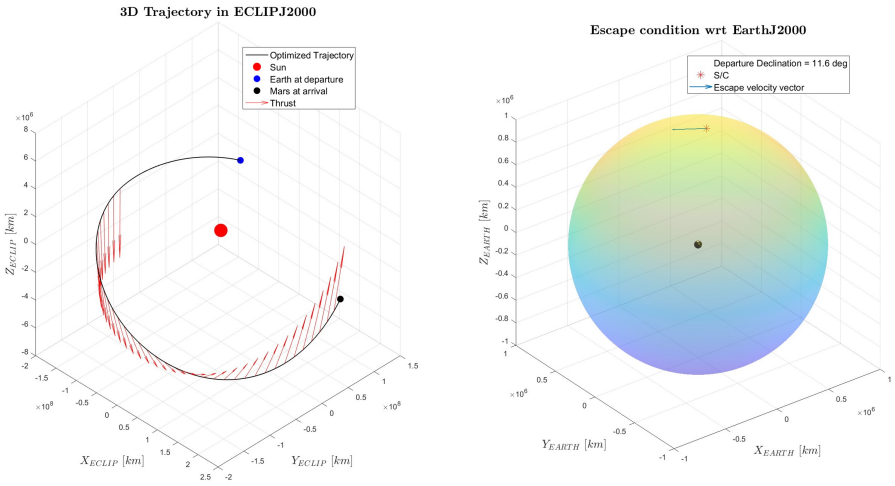

Figure 4.25: Optimal Solution of the 2031 launch window. On the left: 3D trajectory in ECLIPJ2000. On the right: departure condition wrt Earth J2000.

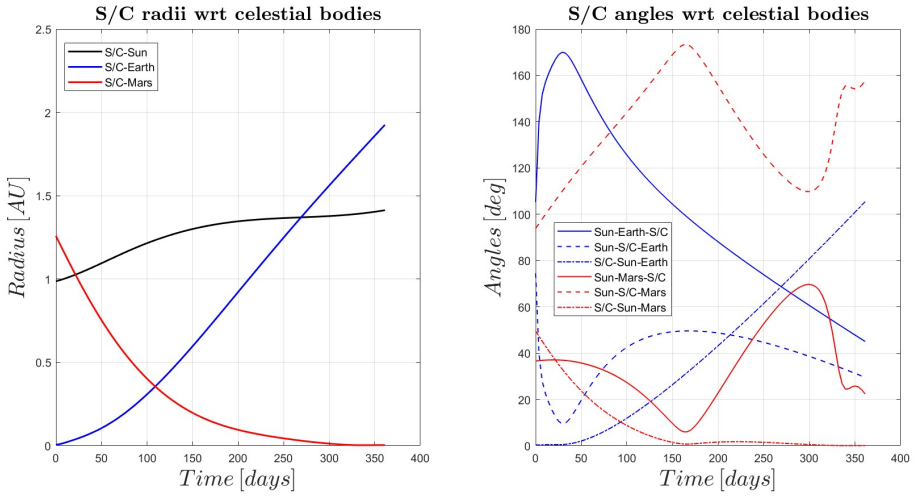

Figure 4.26: Optimal Solution of the 2031 launch window. On the left: S/C radii with respect to relevant celestial bodies. On the right: S/C angles with respect to relevant celestial bodies.

It is now interesting to analyze the low-level research on the launch opportunity of 2037, as high-level research indicated it to be the most promising within the 2030-2040 interval. The launch window was identified from May 1, 2037, to October 1, 2037, with the interval discretized using a time step of 0.5 days, generating 338 trial points. Among these trials, 239 solutions converged within 27.49 minutes of parallel computation on 10 cores. The solutions are summarized in the graph shown in Figure 4.27. Here too, a convex curve representing the

optimal propellant mass as a function of the departure date can be identified, with the minimum found for the launch epoch of August 7, 2037, at 06:00, with $\Delta V = 2.585\,km/s$ and a corresponding propellant consumption of 243.646 kg. The results of the optimized trajectory are detailed and visualized in Figures 4.28, 4.29, and 4.30.

The discussed results are reported and summarized in Table 4.4, comparing them with those presented in the aforementioned reference article. It is evident that the solution proposed in this thesis for the 2031 launch window is indeed validated by the one proposed in the paper. Additionally, this tool has enabled the identification of a more promising launch window in 2037, which offers further propellant savings compared to the 2031 window.

Thus, by exploiting this tool, it was possible to calculate over 400 local solutions for various departure dates, identifying five launch windows and exploring the most promising ones in greater detail. The best solution within the 2030-2040 interval was identified, with the entire process taking approximately 3 hours of computation. Once the best solution within the domain to be explored has been identified, a refinement process can be applied to the solution itself, increasing its level of accuracy as explained and demonstrated in the following Section 4.2.3.

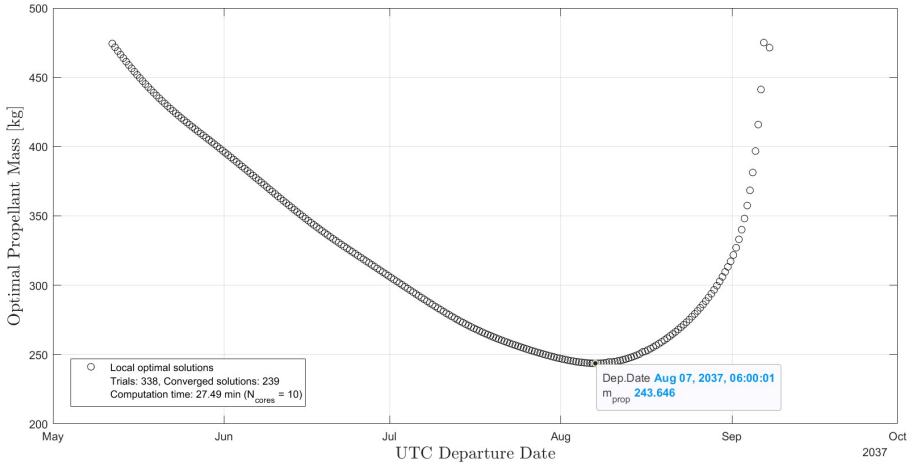

Figure 4.27: Low level search summary plot. Launch epochs window: May 01, 2037 - Oct 01, 2037. Detail on the minimum-propellant solution for the launch window.

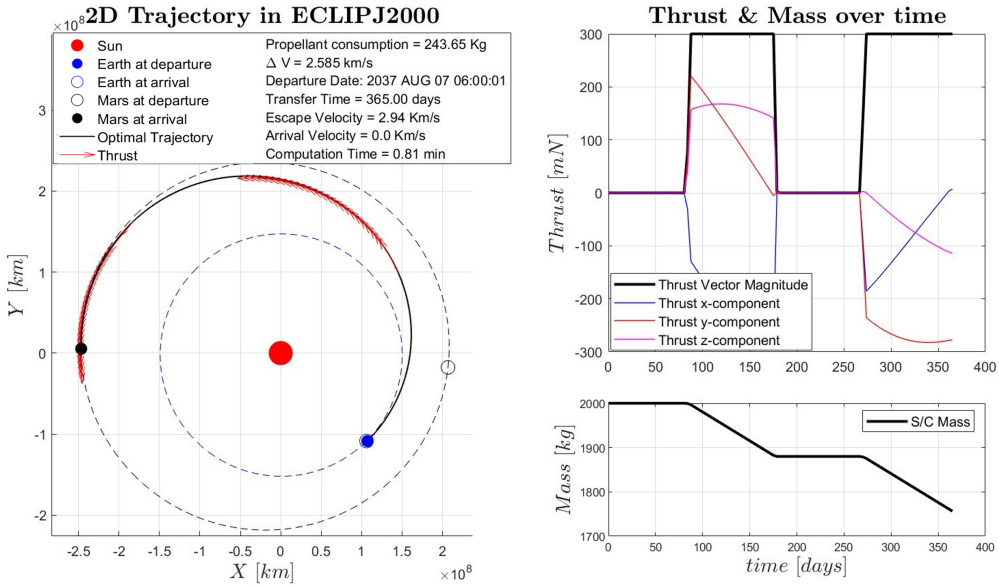

Figure 4.28: Optimal Solution of the 2037 launch window. On the left: 2D Trajectory in ECLIPJ2000 with main features. On the right: thrust and mass over transfer elapsed time.

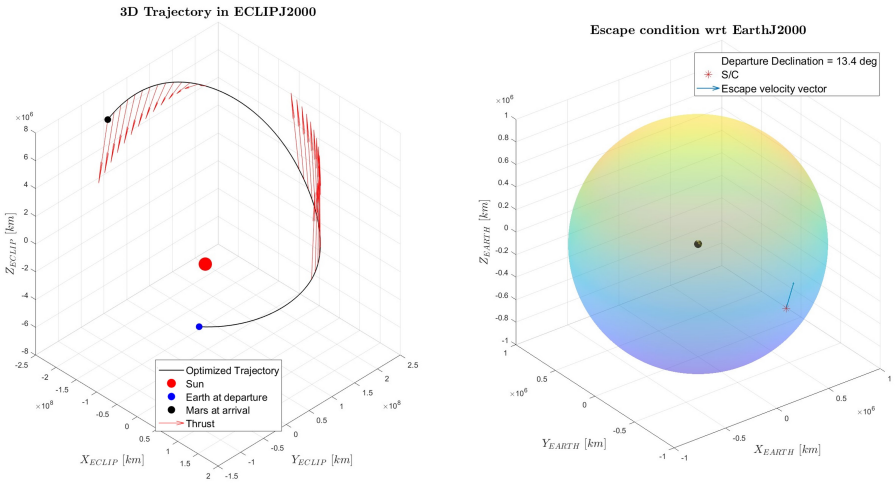

Figure 4.29: Optimal Solution of the 2037 launch window. On the left: 3D trajectory in ECLIPJ2000. On the right: departure condition wrt Earth J2000.

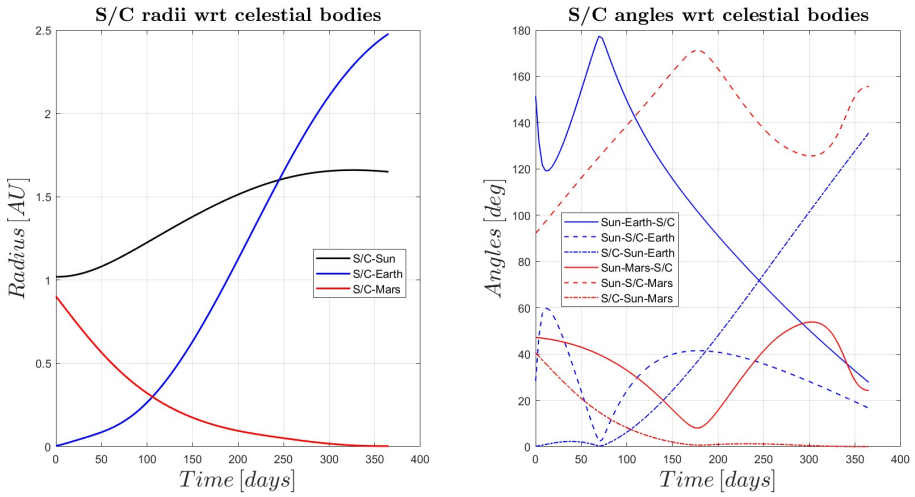

Figure 4.30: Optimal Solution of the 2037 launch window. On the left: S/C radii with respect to relevant celestial bodies. On the right: S/C angles with respect to relevant celestial bodies.

Optimization Outputs	**Paper**	**Thesis Tool: 2031 Solution**	**Thesis Tool: 2037 Solution (Best)**
Departure Date [UTC]	Jan 31, 2031 06:38	Jan 31, 2031 06:00	Aug 07, 2037 06:00
ToF [days]	362.94	361.07	365.00
Arrival Date [UTC]	Jan 29, 2032 05:14	Jan 27, 2032 07:34	Aug 07, 2038 06:00
Propellant Consumption [kg]	384	368.4	243.65
Mass at Mars SOI [kg]	1616	1631.6	1756.4

Table 4.4: Table of comparison between the results proposed by Casanova-Álvarez, Navarro-Medina, and D.Tommasini, 2024[10] and the results achieved by the developed optimization tool of this work

4.2.3 Solution Refinement

As elucidated in the previous section, the methodology for determining the global optimum employs trapezoidal collocation techniques with 100 equispaced nodes. These collocation parameters represent a compromise between computational cost and the precision of the derived optimal trajectory. At that stage of the process, prioritizing computational efficiency is crucial to facilitate the evaluation of multiple local optima, rather than concentrating solely on the precision of an individual solution.

Consequently, it is necessary to refine the identified optimal solution post hoc to enhance its dynamic accuracy. The error of the solution with the current

collocation properties can be evaluated through various methods documented in the literature. In this thesis, for the sake of simplicity and ease of visualization, a propagator of the equations of dynamics was selected for error calculation. Specifically, starting from an optimal solution obtained via direct collocation, the optimal control is propagated using a numerical propagator that accounts for the ODEs of the involved dynamics. The propagated trajectory, in terms of positions and velocities, is then compared with the trajectory approximated through collocation.

The objective is to develop a cascading process for refining the optimal solution, aiming to reduce errors below acceptable thresholds through step-by-step re-optimization. This involves incrementally increasing the number of nodes changing the collocation method and refining the dynamics model (by accounting for third-bodies perturbations) in order to achieve an high-fidelity trajectory. This process is applied to the 2031 and 2037 solutions previously presented, with the results of this refinement detailed in Tables 4.5 and 4.6.

Initially, since the dynamics are modeled as a 2BP in the previous optimization stage, the error is evaluated using a propagator (2B propagator) that considers only the dynamics of the two-body problem, with gravitational acceleration solely due to the central body (the Sun). As shown in the tables, the case 0, which corresponds to the output of the optimization tool with N=100 and the trapezoidal method, exhibits relatively high maximum errors in positions and velocities, on the order of 10^5 km for position and 10^2 m/s for velocity absolute errors. These maximum errors are located at the end of the trajectory, resulting from accumulation during propagation, and are not acceptable given the magnitudes involved. A positional error of 10^5 km would be comparable to the radius of Mars' sphere of influence (SOI), and is therefore excessively large. Given these observations, relative maximum error values of 10^{-6} for positions and 10^{-5} for velocities can be considered acceptable. These correspond to absolute values of less than 1000 km for positions and less than 1 m/s for velocities.

In Solution Refinement 1, the solution is re-optimized by doubling the number of mesh nodes (N=200) while maintaining the trapezoidal collocation method. This adjustment does not significantly impact the errors, reducing the relative errors in positions and velocities by one order of magnitude. In Solution Refinement 2, the number of mesh nodes remains the same as in Case 0, but the degree of the collocation polynomial is increased by employing the Hermite-Simpson method. This approach proves to be highly effective in reducing relative errors, achieving relative errors on the order of 10^{-7} for both positions and velocities in a computational time comparable to the refinement using the trapezoidal method with an increased number of nodes. It is thus evident that increasing the order of the collocation polynomial is much more effective than increasing the number of nodes.In Solution Refinement 3, the number of nodes is further doubled (N=200), while retaining the Hermite-Simpson method. This increase in the number of nodes results in a reduction of relative errors by approximately one order of magnitude.

At this point, it is interesting to examine the impact of gravitational perturbations due to the effect of third bodies on the interplanetary trajectory. While the patched conics approach is widely used for impulsive maneuvers and is generally acceptable, in this case, we aim to evaluate the effect of third bodies (i.e. Earth and Mars) on the interplanetary leg for low-thrust maneuvers. By examining the aforementioned summary tables, it is possible to observe the maximum errors wrt NB propagator for each Solution Refinement Case. Specifically, for the first four cases, which were optimized considering only the two-body dynamics, the non-negligible impact of third-body perturbations is evident. The relative maximum errors do not fall below 10^{-3}, implying absolute errors of 10^6 km in the final position and 10^2 m/s in the final velocity. These values are unacceptable for a trajectory that aims for a high level of accuracy. This indicates the necessity to further refine the solutions by re-optimizing with direct collocation, taking into account third-body perturbations in the equations of motion. Given the strong nonlinearity of the equations of motion with perturbations and the effectiveness of this method for refinement, all subsequent cases (from 4 to 6) are re-optimized using the Hermite-Simpson method. Specifically, in Case 4, only the Earth's perturbation is considered in the direct collocation, demonstrating that its contribution to the error is not particularly significant. In the following cases, the perturbation due to Mars is also included, which indeed proves to be significant in terms of error. In Case 6, satisfactory relative error values are achieved for both trajectories by using Hermite-Simpson with 400 equispaced nodes. The entire refinement process requires a total computational time of approximately 10 minutes for both trajectories.

The refined and accurate trajectories are shown in Figures 4.31, 4.32, 4.33, 4.34, 4.35, and 4.36. It is evident that third-body perturbations have impacted the solutions found, particularly the gravitational effect of Mars on the final part of the trajectory. Specifically, in the 2031 solution, these perturbations lead to a shift in the optimal launch date by two days and result in the saturation of the ToF constraint. The impact in terms of consumption is an additional 50 m/s of ΔV required, translating to approximately 4 kg more propellant. For the 2037 Solution, the perturbations do not affect the launch date but have a more significant impact on consumption, increasing the ΔV by 300 m/s and the propellant mass by 15 kg. This is likely because the optimizer could not take advantage of an increased ToF, as the constraint was already saturated in the two-body dynamics solution. In conclusion, it can be asserted that a following refinement process, which accounts for dynamic perturbations, is indeed necessary to achieve highly accurate trajectories.

Solution Refinement ID	0	1	2	3	4	5	6
Collocation Method	Trapezoidal	Trapezoidal	Hermite-Simpson	Hermite-Simpson	Hermite-Simpson	Hermite-Simpson	Hermite-Simpson
Mesh Points Number (N)	100	200	100	200	200	200	400
Dynamics	Central Body (2BP)	Central Body (2BP)	Central Body (2BP)	Central Body (2BP)	Central Body + Earth Perturbation	Central Body + Earth and Mars Perturbations	Central Body+ Earth and Mars Perturbations
Computation time [s]	27.21	63.45	76.23	119.96	191.06	268.37	504.08
Optimization Outputs							
Departure Date [UTC]	2031 JAN 31	2031 JAN 31	2031 JAN 31	2031 Jan 31	2031 JAN 30	2031 JAN 29	2031 JAN 28
ToF [Days]	361.07	362.63	360.89	360.77	361.65	364.99	365.00
$m_{propellant}$ [kg]	368.39	368.92	369.1	369.06	365.99	372.71	372.63
ΔV [km/s]	4.051	4.057	4.059	4.059	4.022	4.104	4.103
Max Errors wrt 2B propagator							
Position relative error	2.1e-3	5.3e-4	1.6e-7	1.2e-7			
Velocity relative error	1.4e-3	3.4e-4	1.1e-7	8.8e-8			
Position absolute error [km]	4.4e5	1.1e5	32.9	25.1			
Velocity absolute error [m/s]	35.2	8.9	2.6e-3	2.2e-3			
Max Errors wrt NB propagator							
Position relative error	4e-3	3.9e-3	5.2e-3	5.2e-3	3.2e-3	7.7e-6	6.3e-7
Velocity relative error	4.6e-3	6.3e-3	6.6e-3	6.6e-3	5e-3	3.5e-5	2.9e-6
Position absolute error [km]	7.5e5	1.03e6	1.1e6	1.1e6	6.8e5	1.6e3	134.26
Velocity absolute error [m/s]	118.6	161.1	168.9	169.3	282.9	0.9	7.7e-2

Table 4.5: Summary of the refinement process of the optimal solution found in the 2031 launch window

Solution Refinement ID	0	1	2	3	4	5	6
Collocation Method	Trapezoidal	Trapezoidal	Hermite-Simpson	Hermite-Simpson	Hermite-Simpson	Hermite-Simpson	Hermite-Simpson
Mesh Points Number (N)	100	200	100	200	200	200	400
Dynamics	Central Body (2BP)	Central Body (2BP)	Central Body (2BP)	Central Body (2BP)	Central Body + Earth Perturbation	Central Body + Earth and Mars Perturbations	Central Body+ Earth and Mars Perturbations
Computation time [s]	48.63	91.02	86.64	105.95	139.375	637.97	675.57
Optimization Outputs							
Departure Date [UTC]	2037 AUG 07	2037 AUG 07	2037 AUG 07	2037 AUG 07	2037 AUG 07	2037 AUG 07	2037 AUG 07
ToF [days]	365.00	365.00	365.00	365.00	365.00	365.00	365.00
m_{prop} [kg]	243.65	244.73	245.10	245.10	255.71	270.53	270.50
ΔV [km/s]	2.585	2.597	2.602	2.602	2.722	2.892	2.892
Max Errors wrt 2B propagator							
Position relative error	2.5e-3	6.3e-4	1.5e-7	3.3e-8			
Velocity relative error	1.8e-3	4.5e-4	1.1e-7	4.9e-9			
Position absolute error [km]	6.3e5	1.6e5	36.2	8.2			
Velocity absolute error [m/s]	40.4	10.1	2.44e-3	1.2e-4			
Max Errors wrt NB propagator							
Position relative error	2.12e-2	1.9e-2	1.8e-2	1.8e-2	2.6e-3	1.6e-6	3.9e-6
Velocity relative error	1.51e-2	1.4e-2	1.4e-2	1.3e-2	2.6e-2	9.9e-5	2.1e-5
Position absolute error [km]	5.1e6	4.6e6	4.5e6	4.5e6	6.5e5	3.9e3	976
Velocity absolute error [m/s]	339.9	307.4	296.3	296.6	595.3	2.2	0.47

Table 4.6: Summary table of the refinement process of the optimal solution found in the 2037 launch window

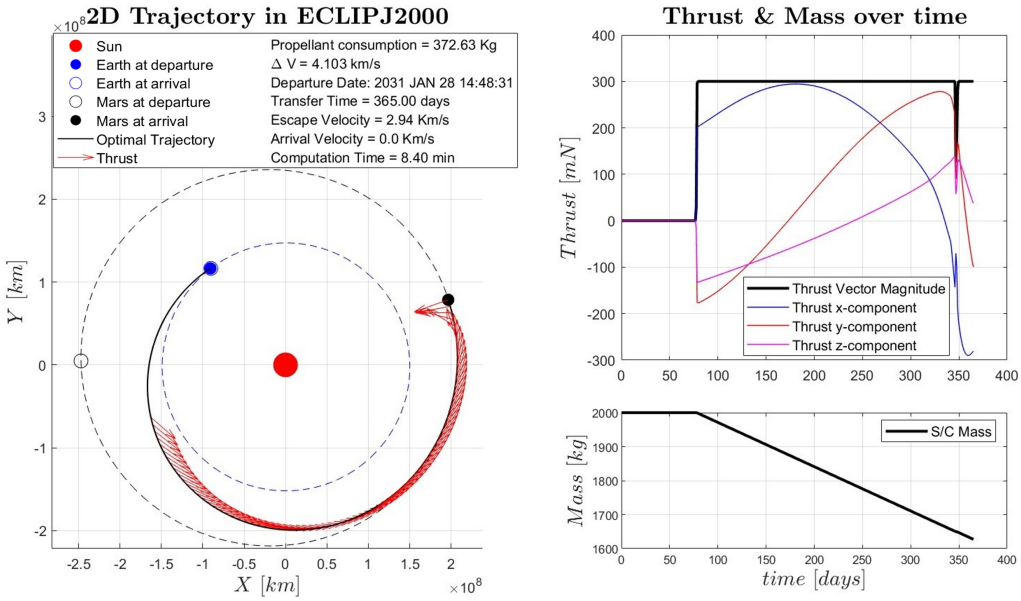

Figure 4.31: 2031 solution refined (with third-bodies perturbations). On the left: 2D Trajectory in ECLIPJ2000 with main features. On the right: thrust and mass over transfer elapsed time.

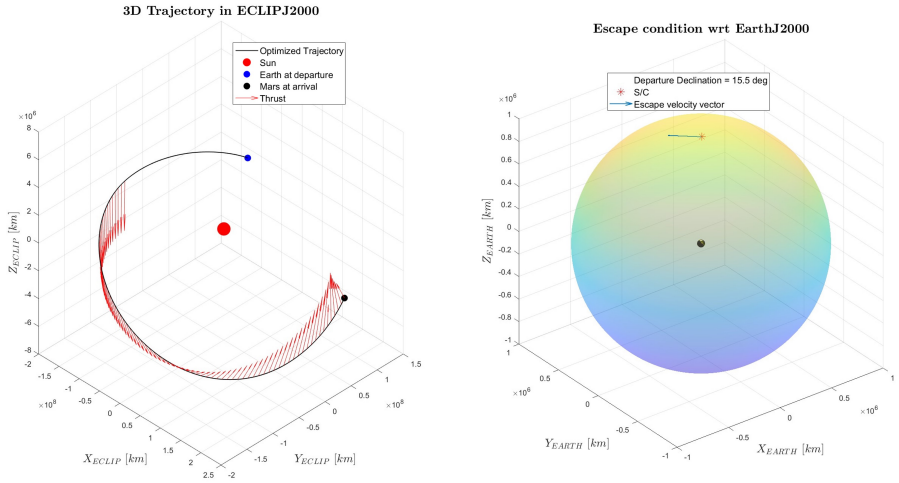

Figure 4.32: 2031 solution refined (with third-bodies perturbations). On the left: 3D trajectory in ECLIPJ2000. On the right: departure condition wrt Earth J2000.

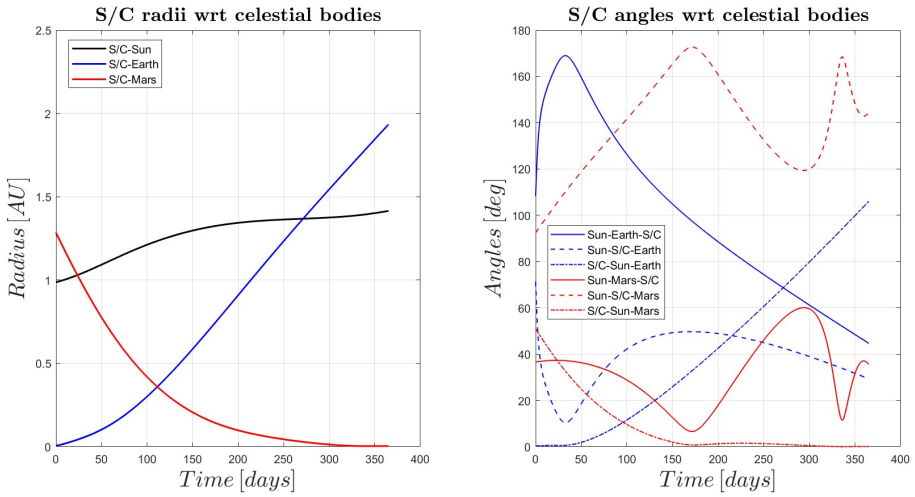

Figure 4.33: 2031 solution refined (with third-bodies perturbations). On the left: S/C radii with respect to relevant celestial bodies. On the right: S/C angles with respect to relevant celestial bodies.

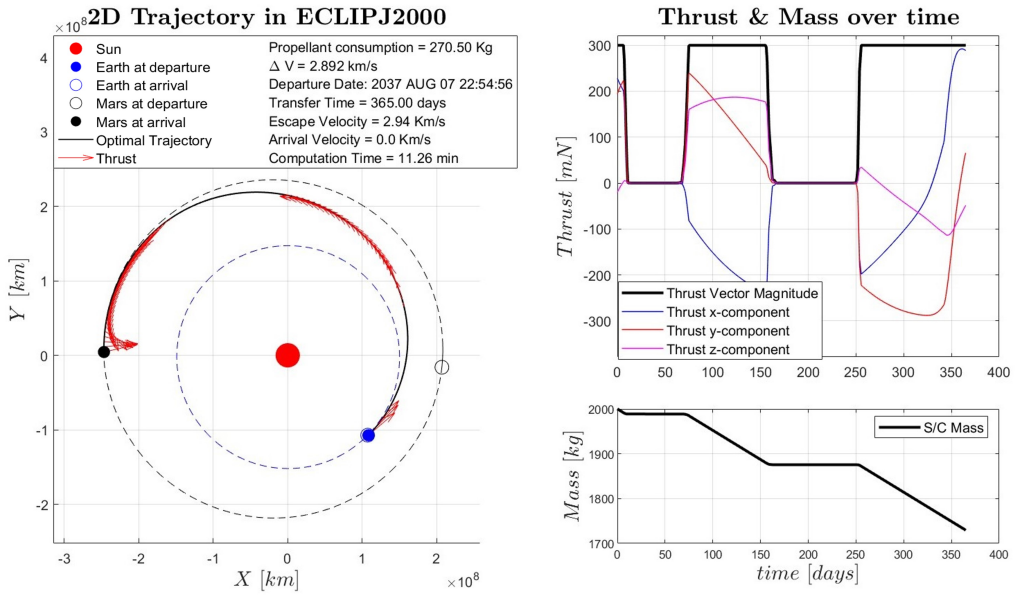

Figure 4.34: 2037 solution refined (with third-bodies perturbations). On the left: 2D Trajectory in ECLIPJ2000 with main features. On the right: thrust and mass over transfer elapsed time.

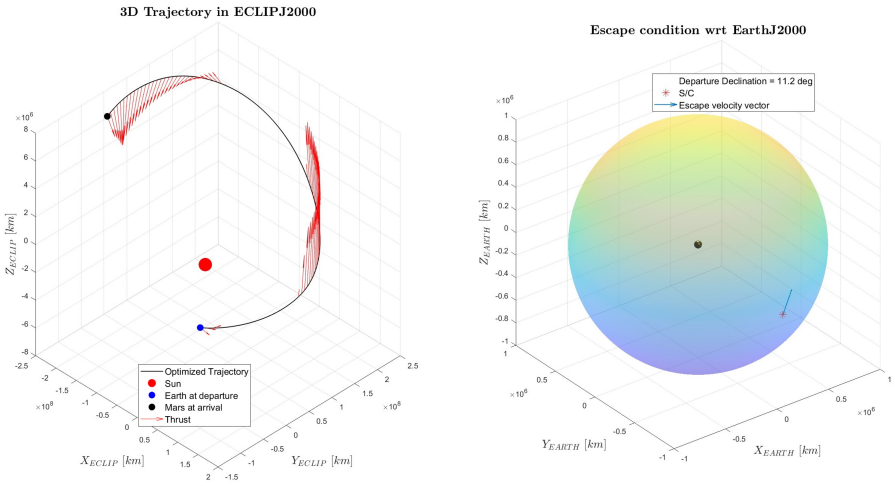

Figure 4.35: 2037 solution refined (with third-bodies perturbations). On the left: 3D trajectory in ECLIPJ2000. On the right: departure condition wrt Earth J2000.

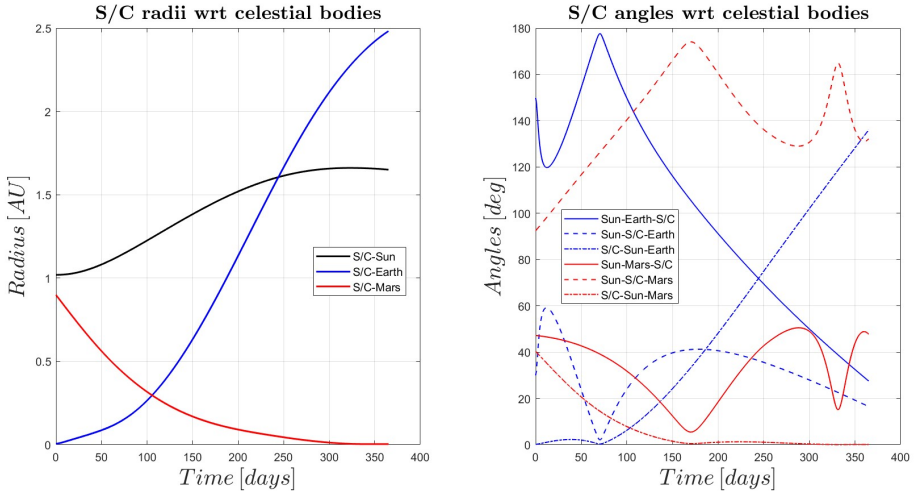

Figure 4.36: 2037 solution refined (with third-bodies perturbations). On the left: S/C radii with respect to relevant celestial bodies. On the right: S/C angles with respect to relevant celestial bodies.

Chapter 5

Case Studies

This chapter presents two primary case studies resolved using the previously developed and validated tool. Section 5.1 details the optimization process for an Earth-Mars transfer utilizing solar electric propulsion, characterized by a variable available thrust profile along the trajectory, dependent on the spacecraft's position relative to the Sun. Conversely, Section 5.2 provides a trajectory analysis for an Earth-Apophis transfer employing electric propulsion. Multiple trajectories were computed using the developed tool to visualize launch opportunities and identify optimal solutions concerning propellant consumption and time of flight.

For all the study cases presented in this chapter, an initial search for the optimal solution within the domain was conducted using a two-stage multi-start search, as described in the previous chapter. The initial search is performed using a relatively simple collocation model, employing the trapezoidal method and considering the dynamics of the 2BP. Once the optimal solutions for each study case are identified, they are further processed to reduce the approximation of the model. This involved re-optimizing these solutions by increasing the precision of the collocation model, increasing the number of grid nodes, employing the Hermite-Simpson method, and incorporating gravitational perturbations due to third bodies of interest into the dynamical equations. This approach ultimately yields a more realistic optimal solution to the problem.

5.1 Earth-Mars Transfer with Solar Electric Thrust Model

This section aims to evaluate the tool developed for addressing an interplanetary Earth-Mars transfer problem using a solar electric propulsion model. At the industrial level, it is essential to incorporate the capability to implement electric propulsion profiles with variable thrust along the trajectory, dependent on the spacecraft's position.

In many real-world case studies, particularly during the early stages of project development (phases 0/A/B), electric thruster performance is specified across

various operational ranges for different distances from the Sun. However, since performance does not change discontinuously but follows continuous operational curves as the spacecraft-Sun distance varies, interpolating these discrete values is necessary for a more accurate mission analysis. This approach is adopted in this work.

Specifically, a BepiColombo-based electric propulsion system, comprising two T6 ion thrusters, was modeled in this case, as documented in Sutherland, Stramaccioni, and Benkhoff, 2019[40] and previously referenced in Section 3.1.2. Specifically, the operating thrust conditions of the EP system provided to the mission analysis team are assumed to be as follows:

- T_{max}=290 mN until 1.25 AU;
- T_{max}=250 mN until 1.37 AU;
- T_{max} = 200 mN until 1.46 AU;
- T_{max}= 145 mN until 1.67 AU;

For the trajectory optimization, the continuous model approximated in Figure 3.1 has been utilized. The thrust profile, drawn in red in the aforementioned figure, maintains a constant thrust of 290 mN up to a distance of 1.22 AU from the Sun. Beyond this range, a second-order polynomial is applied. The coefficients of this polynomial have been calibrated to align with the average thrust level of the preceding stepwise thrust profile. This thrust model is expected to provide a highly accurate approximation of the actual transfers, particularly facilitating an adequate estimation of the transfer times. The variable power level of the electric propulsion engine influences both the thrust level and the specific impulse. However, this variation is expected to have a minor second-order effect on the transfer computations. Consequently, a fixed specific impulse (I_{sp}) value of 4010 seconds has been employed, regardless of the thrust level. All others S/C parameters and optimization input values and constraints are summarized in Table 5.1.

The launch epochs range was initially divided into a grid with a time step of 3 days for high-level interval research. For each time-grid point, a local SQP optimization is performed by using a number of grid point equal to 100 and the trapezoidal method for the direct collocation technique.

This multi-start search took 112.56 minutes to identify 29 convergent solutions and two possible launch windows, each spanning between 3 and 4 months. The first window is in 2026 (approximately from July 1, 2026, to November 1, 2026), and the second window is in 2028 (approximately from September 1, 2028, to December 1, 2028). The solutions found within this high-level search are summarized in Figure 5.1.

Based on prior high-level research, the launch window in 2026 is identified as the most promising. Consequently, we proceed to the low-level optimum search stage within the departure date window ranging from July 1, 2026, to

Optimization Input	**Value/Range**	**Unit**
Launch Mass (m_0)	2500	[kg]
Max available propellant	500	[kg]
Max available thrust level (T_{max})	Variable profile (see Fig. 3.1)	
Specific Impulse (I_{sp})	4010	[s]
Launch Epochs Range	[Jun 01, 2026 ; Jan 01, 2029]	
ToF Bounds	[200, 550]	[Days]
Escape velocity	[0, 3.00]	[km/s]
Arrival velocity	[0.000, 0.001]	[km/s]

Table 5.1: Optimization Input Parameters and Constraints

November 1, 2026. Utilizing a discretization step of 0.5 days, 246 trial points are generated. For each of these trial points, an optimization based on SQP algorithms is subsequently performed. The initial guess for this optimization is the solution closest to the trial point's launch date, selected and filtered from the solutions identified during the high-level search stage.

The results of this low-level search, which require a computational time of 33.82 minutes to identify 199 locally converged solutions, are illustrated in Figure 5.2. The time-dependent consumption curve of the various solutions exhibits a convex trend, enabling the identification of the optimal solution over the range of launch epochs. This optimal solution corresponds to a departure date of October 2, 2026, at 21:28, with a propellant consumption of 197.35 kg. The detailed outputs of this solution are summarized in Table 5.2 and depicted in detail in Figures 5.3, 5.4, and 5.5.

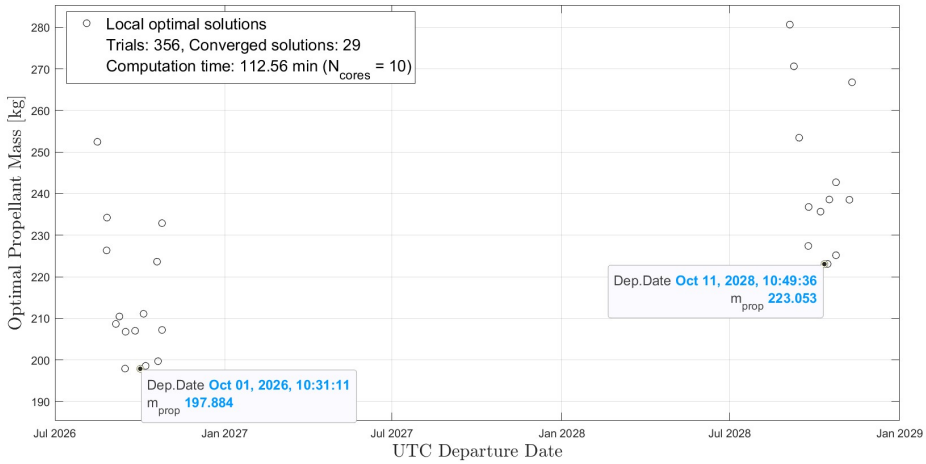

Figure 5.1: High level search summary plot. Details on the best solutions found for each launch window found

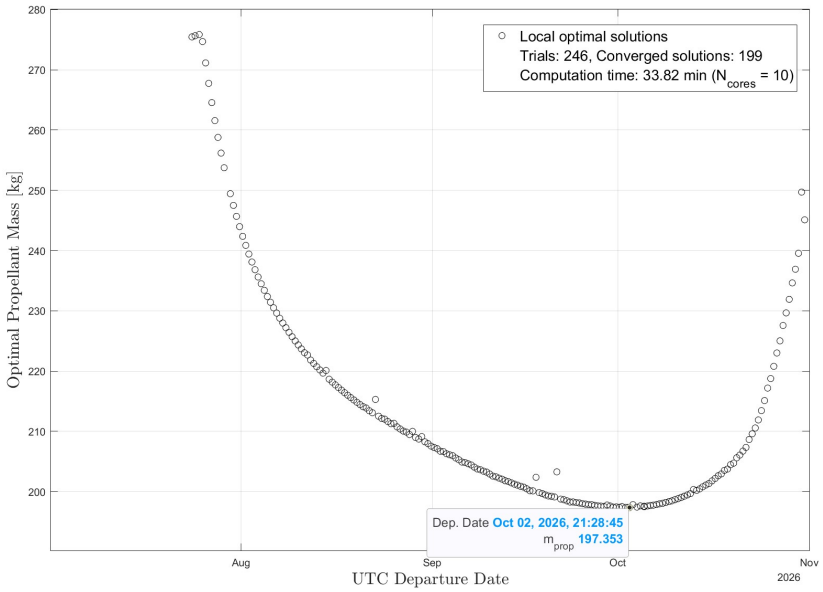

Figure 5.2: Low level search summary plot. Launch epochs window: Jul 20,2026 - Nov 01, 2026. Detail on the minimum-propellant solution for the launch window.

Optimization Outputs	**2026 Solution (Best)**
Departure Date [UTC]	Oct 02, 2026 21:28
ToF [days]	527.06
Arrival Date [UTC]	Mar 12, 2028 23:04
Propellant Consumption [kg]	197.35
ΔV [km/s]	3.234
Escape Velocity [km/s]	2.472
Arrival Velocity [km/s]	0.001
Mass at Mars SOI [kg]	2302.6

Table 5.2: Best solution (2026) outputs summary table

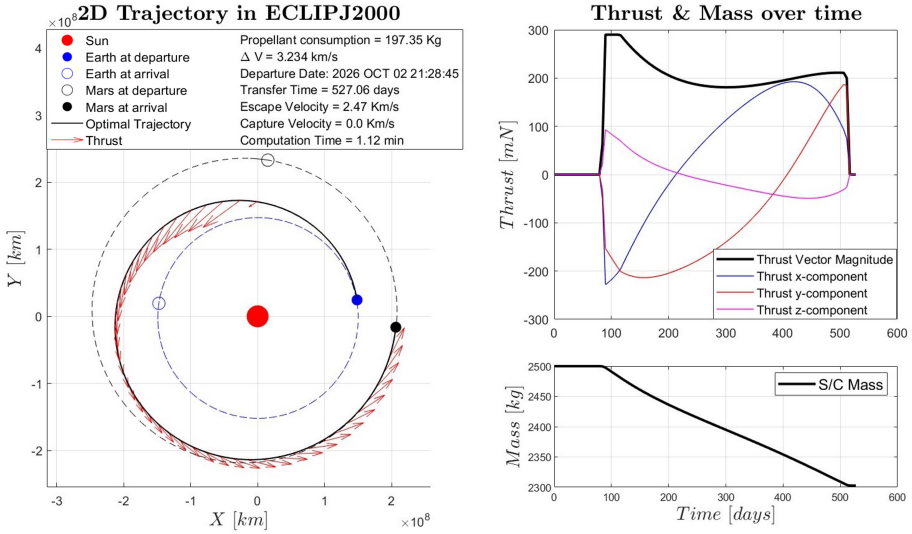

Figure 5.3: Optimal solution for the 2026 launch window. On the left: 2D Trajectory in ECLIPJ2000 x-y plane with main features. On the right: thrust and mass over transfer elapsed time.

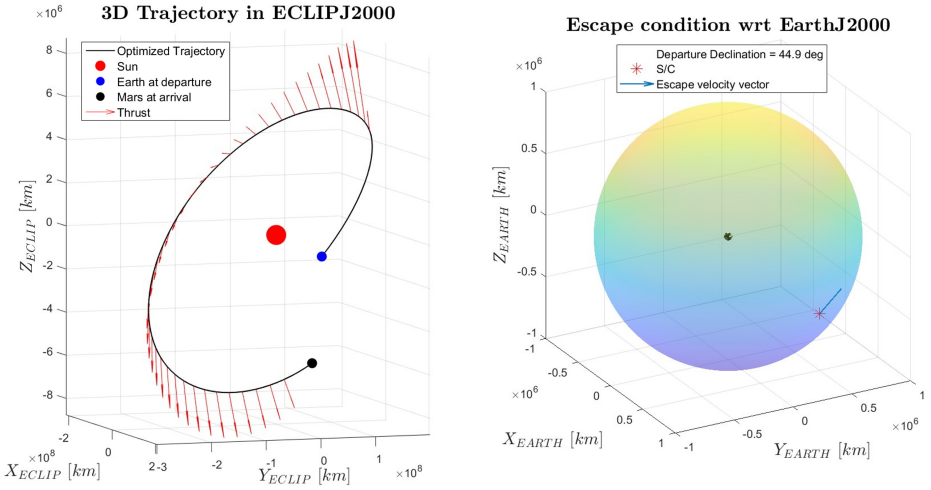

Figure 5.4: Optimal solution of the 2026 launch window. On the left: 3D Trajectory in ECLIPJ2000. On the right: departure condition wrt Earth J2000.

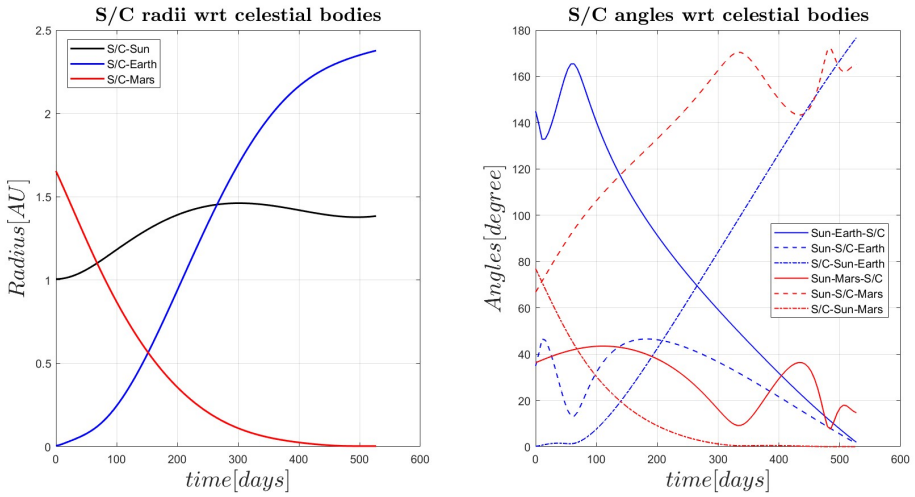

Figure 5.5: Optimal solution of the 2026 launch window. On the left: S/C radii with respect to relevant celestial bodies over elapsed time. On the right: S/C angles with respect to relevant celestial bodies over elapsed time.

In this case study, the refinement process of the optimal solution is performed with the goal of increasing its dynamical accuracy, following steps similar to those extensively described in Section 4.2.3. The main features and errors corresponding to each refinement step are summarized in Table 5.3. The objective is to generate a solution that satisfies relative errors of the order of 10^{-6} for positions (resulting in an absolute error of less than 1000 km) and relative errors of the order of 10^{-5} for velocities (resulting in an absolute error of less than 0.5 m/s), even by accounting for an higher level of dynamical fidelity that accounts for third-bodies perturbations along the trajectory.

It should be noted that, unlike previous cases, gravitational perturbations due to third bodies are introduced only after increasing the refinement of the placement mesh to 400 nodes using the Hermite-Simpson method. This adjustment was necessary because attempting to introduce these perturbations and re-optimize with 200 nodes resulted in the solver being unable to find a solution that met the constraints, even with slight relaxations. This increased complexity is likely due to the additional nonlinearity introduced by the non-constant thrust profile, making the problem more challenging than the previous one involving nuclear electric propulsion model. However, the process concludes with the achievement of a satisfactory final solution in about 6 minutes of computational time. Although the constraint on the arrival velocity is not strictly satisfied, requiring relaxation to 10 m/s to achieve convergence, this relaxation is considered acceptable for the interplanetary transfer problem under consideration.

The refined optimal trajectory is illustrated in Figures 5.6, 5.7, and 5.8, with its main characteristics and optimization outputs summarized in Table 5.3. It is evident that considering the N-body problem (NBP) influences the optimal solution found with the simplified 2BP.

In the 2BP solution, there is a small final coasting phase of a few days before reaching Mars' SOI. However, in the NBP solution, this coasting phase is replaced by a propelled arc, where the thrust acceleration balances the minor perturbation due to Mars' proximity. This results in an increase in the propellant mass used, which rises to 199.36 kg (an increase of approximately 2 kg compared to the 2BP), and consequently, an increase in ΔV to 3.268 km/s (an increase of 34 m/s compared to the 2BP).

On the other hand, there is no significant impact on the optimal launch date. It is also noted that the optimal escape velocity from Earth has increased, and the total time of flight (ToF) for the interplanetary leg has decreased.

Solution Refinement ID	0	1	2	3	4	5
Collocation Method	Trapezoidal	Hermite-Simpson	Hermite-Simpson	Hermite-Simpson	Hermite-Simpson	Hermite-Simpson
Mesh Intervals Number (N)	100	100	200	400	400	400
Dynamics	Central Body (2BP)	Central Body (2BP)	Central Body (2BP)	Central Body (2BP)	Central Body + Earth Perturbations	Central Body+ Earth and Mars Perturbations
Computation time [s]	67.35	81.27	92.57	130.42	185.28	382.82
Optimization Outputs						
Departure Date [UTC]	2026 OCT 02	2026 OCT 02	2026 OCT 02	2026 OCT 02	2026 OCT 02	2026 OCT 02
ToF [Days]	527.06	527.11	527.00	527.01	526.97	518.45
m_{prop} [kg]	197.35	198.83	198.79	198.78	197.67	199.36
ΔV [km/s]	3.234	3.260	3.259	3.258	3.239	3.268
Escape Velocity [km/s]	2.472	2.482	2.482	2.482	2.624	2.614
Arrival Velocity [km/s]	0.001	0.001	0.001	0.001	0.001	0.010
Max Errors wrt 2B propagator						
Position relative error	5.3e-3	1.2e-5	5.0e-7	1.9e-7		
Velocity relative error	3.9e-3	7.7e-6	1.2e-6	3.4e-7		
Position absolute error [km]	1.1e6	2.4e3	104.1	41.8		
Velocity absolute error [m/s]	97.7	2.03e-1	3.2e-2	9.1e-3		
Max Errors wrt NB propagator						
Position relative error	2.1e-2	1.8e-2	1.8e-2	1.8e-2	8.8e-3	1.1e-6
Velocity relative error	1.5e-2	1.3e-2	1.3e-2	1.3e-2	1e-2	1.3e-5
Position absolute error [km]	4.5e6	3.7e6	3.7e6	3.7e6	1.8e6	871.3
Velocity absolute error [m/s]	351.2	314.5	314.0	314.3	268.2	2.3e-1

Table 5.3: Summary table of the refinement process to the optimal solution found

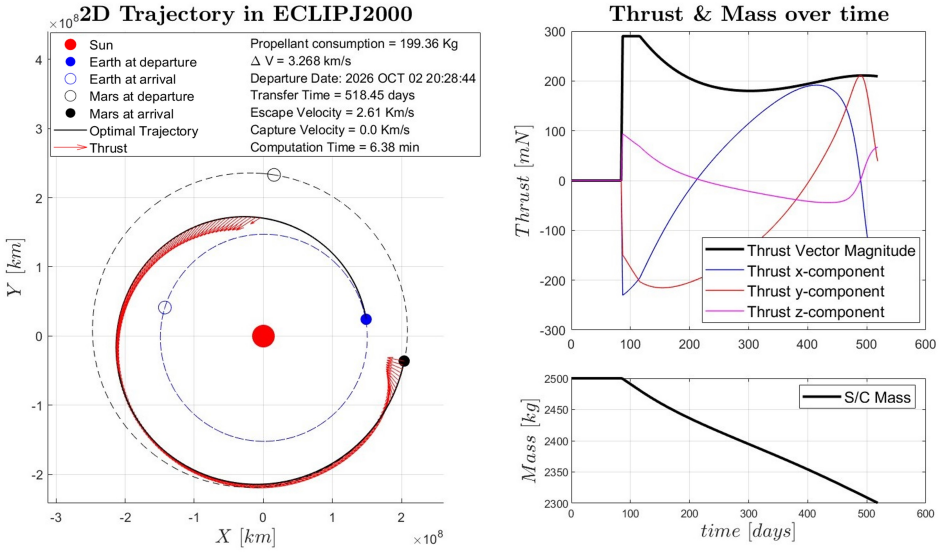

Figure 5.6: Optimal solution refined (with third-bodies perturbations). On the left: 2D Trajectory in ECLIPJ2000 x-y plane with main features. On the right: thrust and mass over transfer elapsed time.

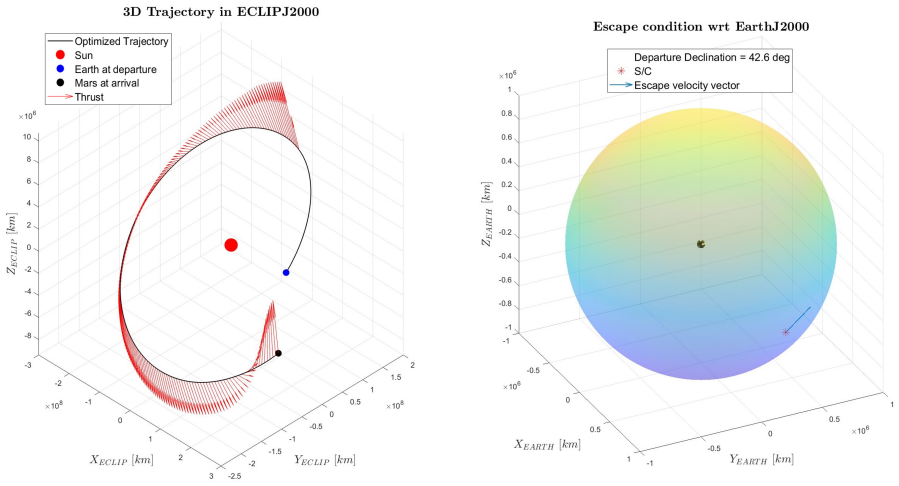

Figure 5.7: Optimal solution refined (with third-bodies perturbations). On the left: 3D Trajectory in ECLIPJ2000. On the right: departure condition wrt Earth J2000.

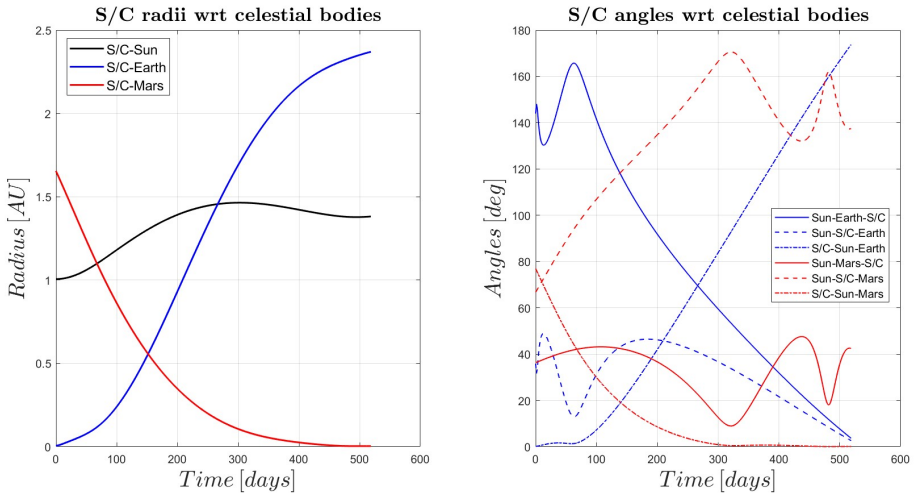

Figure 5.8: Optimal solution refined (with third-bodies perturbations). On the left: S/C radii with respect to relevant celestial bodies over elapsed time. On the right: S/C angles with respect to relevant celestial bodies over elapsed time.

5.2 Earth-Apophis Transfer

This section presents a preliminary mission analysis to evaluate the feasibility and potential benefits of an Earth-Apophis transfer utilizing electric propulsion. Apophis, an asteroid with an approximate diameter of 370 meters, orbits the Sun and is expected to make a close approach to Earth, passing at a distance of approximately 31,000 kilometers from the Earth's surface on April 13, 2029. This event offers a unique opportunity to plan an observational mission to study the physical and dynamic effects of the Earth-Apophis gravitational interaction, thereby providing insights into the behavior and properties of asteroids.To this end, ESA has proposed the RAMSES (Rapid Apophis Mission for Security and Safety) mission, which aims to reach Apophis prior to its close encounter with Earth.

The work presented within this section focuses on the reachability analysis of Apophis, investigating potential trajectories from Earth to the asteroid using a low-thrust spacecraft, with the goal of arriving before the fly-by. The objective is to utilize and test the tool developed in this thesis to generate a pork chop-like plot of low-thrust trajectories. This involves calculating and storing a large number of potential solutions, and subsequently identifying the optimal ones in terms of propellant consumption and ToF.

To achieve this, it is necessary to make minor modifications to the architecture of the previously presented tool. In this case study, a parametric search of solutions is conducted by varying departure dates and times of flight within certain bounds. Therefore, not only the departure dates but also the times of flight will be discretized, requiring the solver to find a local solution within the sub-interval of departure dates and ToF identified by the corresponding grid point. Additionally, in this specific case, identifying an appropriate tangential thrust guess is not straightforward. Since Apophis is not a planet in the solar system, it is challenging to determine an average radius of its orbit around the Sun to use as a target for the tangential thrust guess. To address this, considering the asteroid's relative proximity to Earth's orbit during the reference period, a simple guess is employed: the spacecraft's trajectory with zero escape velocity and zero thrust along the path. Thus, the guess provided is the same as Earth's orbit propagated over the ToF corresponding to the grid point, starting from the departure date of that point.

Regarding the input parameters for optimization, such as the spacecraft characteristics, the properties of the electric propulsion system, and the bounds on the main variables involved, these are based on the proposals in the referenced article by Morelli et al., 2024[31] and are summarized in Table 5.4. In particular, the selected values for the maximum thrust-to-initial mass ratio $\frac{T_{\max}}{m_0}$, and the specific impulse I_{sp} are 1.2×10^{-4} m/s^2 and 1500 s, respectively. These values are compatible with state-of-the-art Hall-effect thrusters (see Dannenmayer and Mazouffre, 2009[12]). Consequently, the maximum available thrust is $T_{max} = 60$ mN, and it is considered constant along the trajectory.

Optimization Inputs	**Value/Range**	**Unit**
Launch Mass (m_0)	500	[kg]
Max available propellant	75	[kg]
Max available thrust level (T_{max})	0.060 (constant)	[N]
Specific Impulse (I_{sp})	1500	[s]
Launch Epochs Range	[Nov 01, 2026; Mar 01, 2028]	
ToF Bounds	[250, 800]	[Days]
Escape velocity	[0, 4.00]	[km/s]
Arrival velocity	[0.000, 0.001]	[km/s]

Table 5.4: Optimization Input Parameters and Constraints assumptions

At this stage, a parametric search for local solutions is conducted on the mesh points. Initially, a step size of 4 days is used to discretize the departure dates, and a step size of 50 days is used to discretize the ToFs, generating 1464 trial points from which 451 local solutions are found. From these solutions, the mesh refinement of the most critical and promising areas is performed by locally reducing the discretization steps of ToFs and launch epochs, generating new trial points. For each trial point, the nearest high-level solution is used as a guess. The closest solution is identified as the high-level solution with the smaller distance from the new trial point of the mesh, in terms of ToF and launch epoch. Thus, it is selected by evaluating the solution that meets the requirement as follows:

$$min(\frac{|\Delta t_{trial} - \Delta t_{sol}|}{\Delta t_{max} - \Delta t_{min}} + \frac{|t_{0,trial} - t_{0,sol}|}{t_{0,max} - t_{0,min}}) \tag{5.1}$$

Where $t_{0,trial}$ and Δt_{trial} are the departure date (in days past j2000) and the ToF corresponding to the trial point we want to evaluate, $t_{0,sol}$ and Δt_{sol} are the departure date and the ToF relative to the considered high-level solution already available, and $t_{0,max}$, $t_{0,min}$, Δt_{max}, and Δt_{min} represents the bounds on launch epochs and ToFs of the optimal research, useful to nondimensionalize the quantities involved.

The entire process concludes with the evaluation of optimal local solutions across 7995 starting points, resulting in 3781 convergent solutions within a computation time of 26.66 hours. These results are presented in the form of a porkchop-like plot in Figure 5.9. The x-axis represents the departure dates, while the y-axis denotes the maneuver time. The contour plot levels indicate the propellant consumption values associated with each solution found, following a color scale depicted in the color bar on the right. In this porkchop plot, it is possible to identify the most promising region in terms of propellant consumption and times of flight.

Specifically, the plot allows for the identification of Solution A, which represents the minimum propellant consumption, and Solution B, which corresponds

to the minimum ToF required to reach Apophis. The main outcomes of these solutions are summarized in Table 5.5 and they are described in details in the dedicated Section 5.2.2 and Section 5.2.1, with the detailed process of accuracy refinement applied on each solution. It is important to note that both solutions have arrival dates preceding the predicted fly-by date of Apophis, making them suitable for the mission.

This parametric approach to generating local solutions within the ranges of launch epochs and transfer times is computationally demanding, as it requires evaluating a large number of solutions. However, it is highly beneficial because it allows for the creation of a manageable and modifiable storage of possible solutions without the need for re-optimization. This approach enables the application or modification of constraints on the solution domain as needed. A clear example is the requirement for the arrival date at the asteroid, which was not considered a constraint during the optimization. If this constraint needs to be applied or modified, the solution domain can be easily managed by identifying the new optimal solution within the available storage for the updated constraint, as presented in Section 5.2.3.

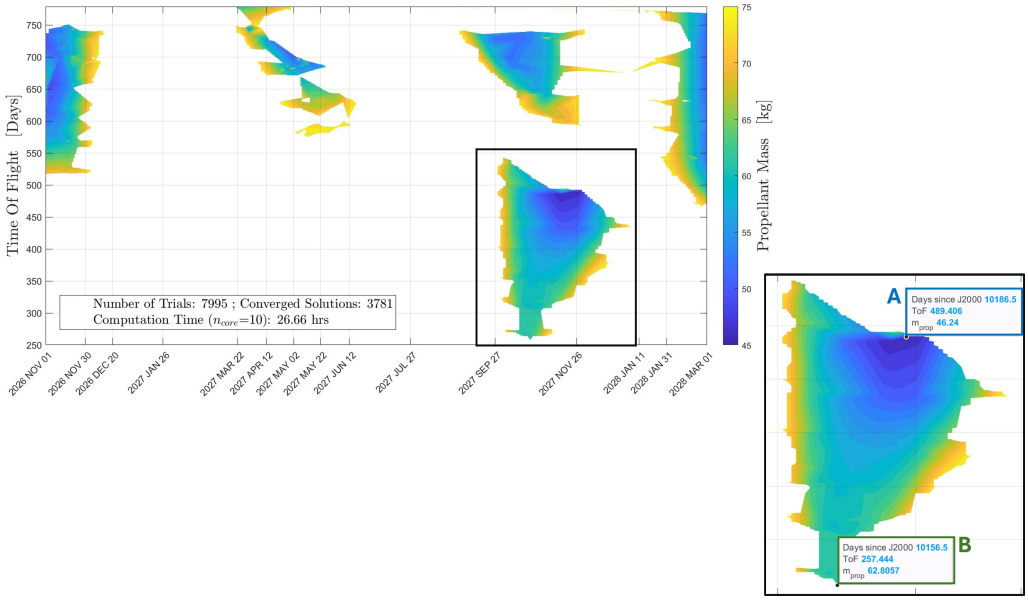

Figure 5.9: Porkchop plot of the Earth-Apophis transfer opportunities within the 2026-2028 launch window. Detail on the most promising solutions area with the identification of 2 best solutions. The A solution represents the optimal-consumption trajectory, while the B solution represents the optimal-ToF trajectory.

Optimization Outputs	Solution A: Minimum Propellant	Solution B: Minimum ToF
Departure Date [UTC]	Nov 21, 2027 00:00	Oct 23, 2027 13:08
ToF [days]	489.41	264.62
Arrival Date [UTC]	Mar 20, 2029 14:35	Jul 14, 2028 03:58
ΔV [km/s]	1.428	1.975
Propellant Consumption [kg]	46.24	65.09
Escape velocity [km/s]	4.00	4.00
Arrival velocity [km/s]	0.001	0.001

Table 5.5: Summary table reporting the main features of the two proposed solutions for the Earth-Apophis transfer

5.2.1 Minimum Propellant Consumption Solution

The objective of this case study is to determine the optimal trajectory with a focus on minimizing propellant consumption. In this section, we present Solution A, which identifies the trajectory that achieves the lowest propellant usage. Figure 5.10 displays the trajectory projected onto the ecliptic plane, with the legend highlighting the key characteristics of the transfer. Additionally, the thrust and mass profiles are shown as functions of elapsed time. Figure 5.11 presents the three-dimensional trajectory in the ecliptic reference system, emphasizing the propelled arc in red. It also depicts the escape condition from the Earth's sphere of influence in the EarthJ2000 equatorial reference system, highlighting the required departure declination. Figure 5.12 illustrates the trends of positions and angles relative to the main celestial reference bodies (Earth, Apophis, and Sun).

Subsequently, the refinement process of the identified solution is implemented, similar to the previously discussed case studies, to enhance its accuracy. The step-by-step results of this process are presented in Table 5.2.1. This table details the solution's main characteristics, computation time for each step, the main attributes of the newly identified solution, and the discrepancies between the solution and the reference propagator (2B or NB) for each refinement step. It is important to note that the primary perturbation affecting the trajectory is due to the Earth, as the spacecraft remains in close proximity to Earth on the way to Apophis, making the influence of other planets (such as Mars) negligible.

By increasing the number of mesh segments to 400, employing the Hermite-Simpson method, and incorporating Earth's perturbation into the dynamic equations, a new optimal solution can be obtained in about 12 minutes (specifically 743.4 seconds). This solution exhibits satisfactory relative errors compared to

the NB propagator, with maximum absolute position error of 150 km and velocity errors less than 0.4 m/s. The results of this high-accuracy optimal solution are depicted in Figures 5.13, 5.14, and 5.15.

In this case, the impact of Earth's gravitational perturbation on propellant consumption is less pronounced compared to the previously discussed scenarios. This is because the spacecraft velocity at the beginning of the trajectory, where Earth's gravitational influence is strongest, is relatively high. Consequently, the spacecraft spends less time in this region, minimizing the perturbation's effect on the trajectory. However, this perturbation does cause a shift in the optimal departure date from Earth to November 18, 2027, and a reduction of approximately two days in the time of flight, that lead to an arrival date set to March 20, 2029. The ΔV increases by about 43 m/s, yielding an increase in propellant consumption by about 1.33 kg.

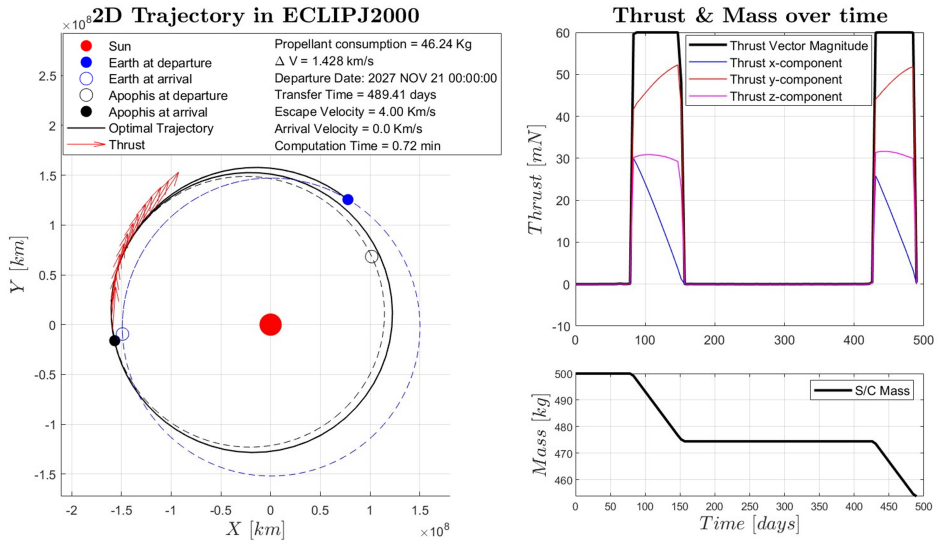

Figure 5.10: Optimal Solution for Earth-Apophis transfer with minimum propellant consumption. On the left: 2D Trajectory Visualization with main features. On the right: thrust and mass over transfer elapsed time

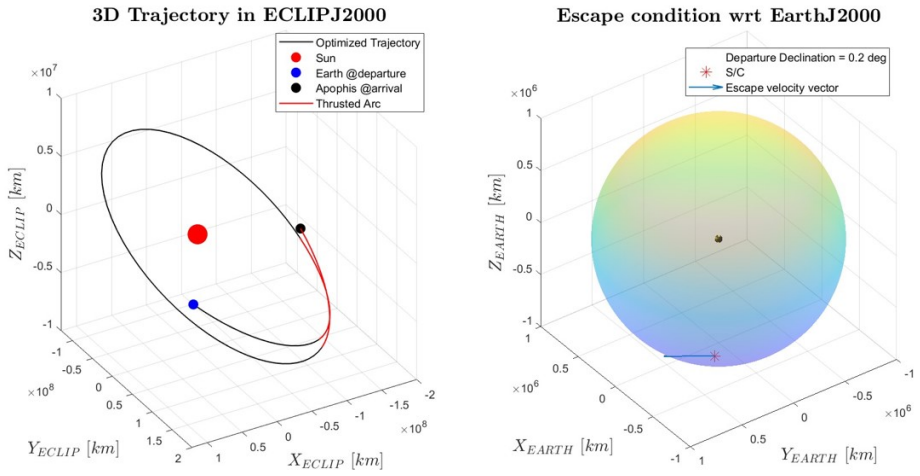

Figure 5.11: Optimal Solution for Earth-Apophis transfer with minimum propellant consumption. On the left: 3D trajectory visualization. On the right: departure condition wrt Earth J2000

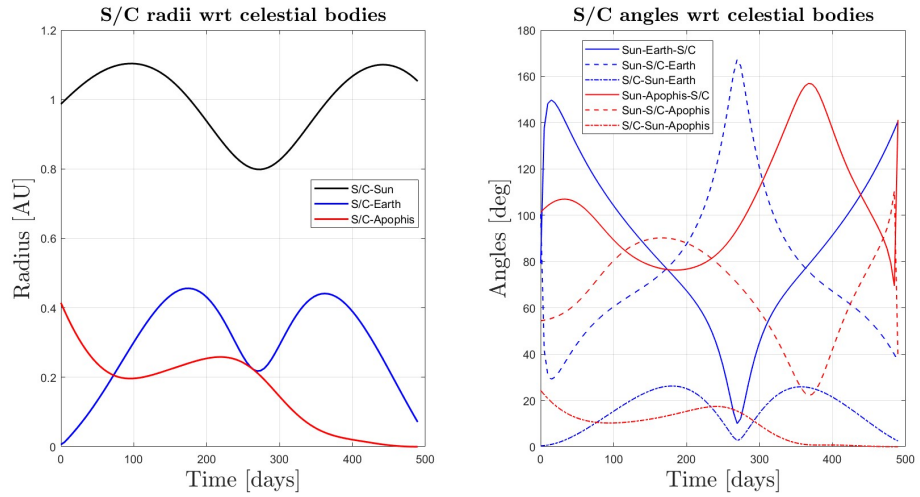

Figure 5.12: Optimal Solution for Earth-Apophis transfer with minimum propellant consumption. On the left: S/C radii with respect to relevant celestial bodies. On the right: S/C angles with respect to relevant celestial bodies

Solution Refinement ID	0	1	2	3	4
Collocation Method	Trapezoidal	Hermite-Simpson	Hermite-Simpson	Hermite-Simpson	Hermite-Simpson
Mesh Intervals Number (N)	100	100	200	200	400
Dynamics	Central Body (2BP)	Central Body (2BP)	Central Body (2BP)	Central Body + Earth Perturbation	Central Body + Earth Perturbation
Computation time [s]	43.4	55.6	102.72	595.7	743.4
Optimization Outputs					
Departure Date [UTC]	2027 NOV 21	2027 NOV 21	2027 NOV 20	2027 NOV 19	2027 NOV 18
ToF [days]	489.41	489.34	490.17	486.71	487.7
$m_{propellant}$ [kg]	46.24	46.58	46.58	47.66	47.57
ΔV [km/s]	1.428	1.441	1.444	1.473	1.471
Errors wrt 2B propagator					
Position relative error	4.9e-3	2.7e-6	1.58e-7		
Position absolute error [km]	6.51e5	363.85	21.63		
Velocity relative error	4.2e-3	2.47e-6	1.56e-7		
Velocity absolute error [m/s]	132.4	7.9e-2	4.9e-3		
Errors wrt NB propagator					
Position relative error	1.5e-2	1.1e-2	1.2e-2	1.7e-5	9.7e-7
Position absolute error [km]	2.4e6	1.8e6	1.8e6	2.08e3	153.4
Velocity relative error	1.5e-2	1.1e-2	1.1e-2	1.4e-5	1.3e-6
Velocity absolute error [m/s]	404.8	297.9	301.5	5.2e-1	3.4e-2

Table 5.6: Summary table of the refinement process to the minimum propellant solution

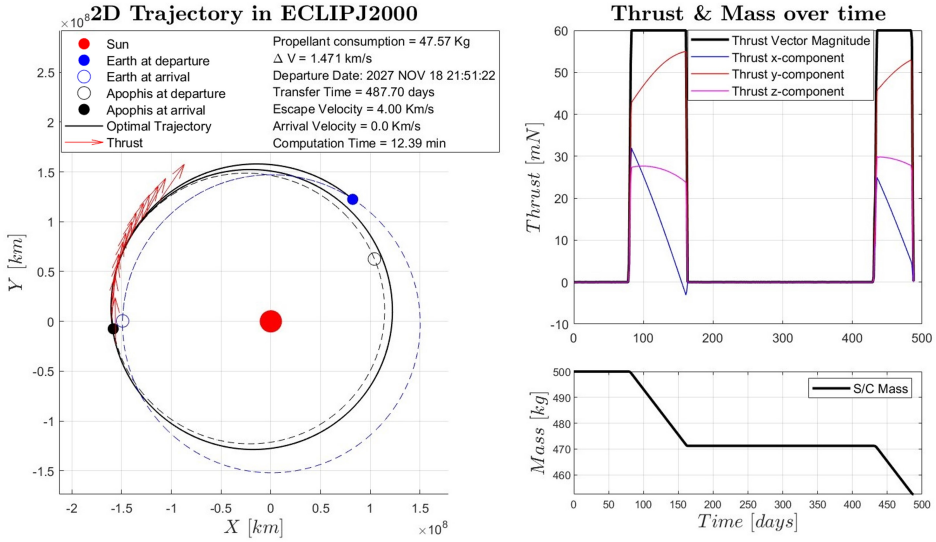

Figure 5.13: Refined optimal solution for Earth-Apophis transfer with minimum propellant consumption (accounting for Earth's third-body perturbation). On the left: 2D Trajectory Visualization with main features. On the right: thrust and mass over transfer elapsed time

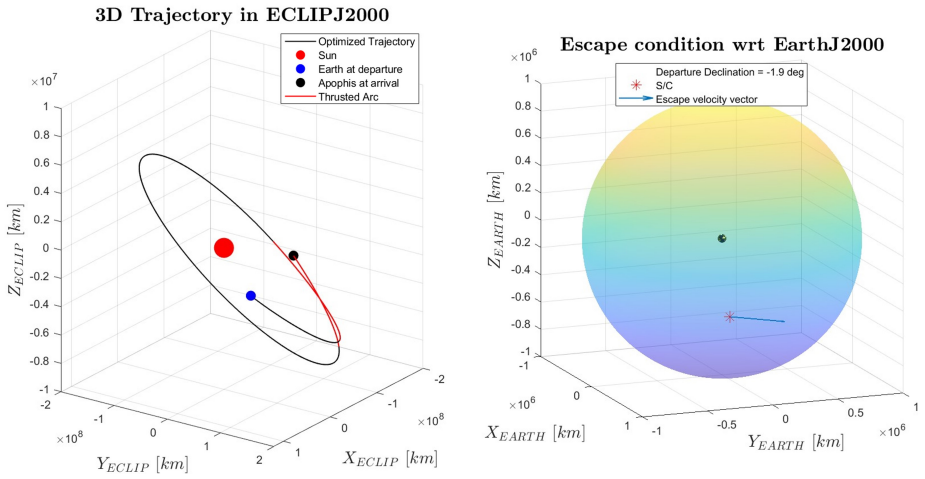

Figure 5.14: Refined optimal solution for Earth-Apophis transfer with minimum propellant consumption (accounting for Earth's third-body perturbation). On the left: 3D trajectory visualization. On the right: departure condition wrt Earth J2000

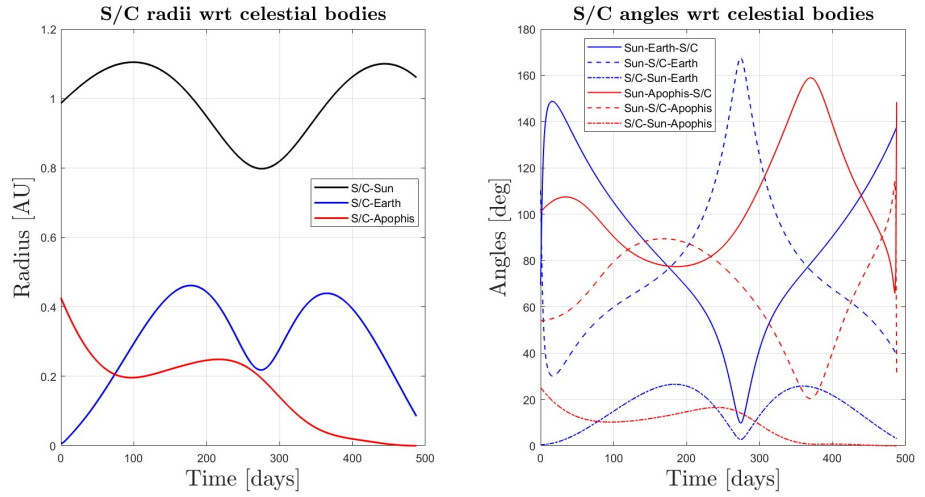

Figure 5.15: Refined optimal solution for Earth-Apophis transfer with minimum propellant consumption (accounting for Earth's third-body perturbation). On the left: S/C radii with respect to relevant celestial bodies. On the right: S/C angles with respect to relevant celestial bodies

5.2.2 Best ToF Solution

In this section, we present an alternative solution, namely the solution identified with the shortest maneuvering time (Solution B). The 2BP solution, which is the output of the developed optimization tool, is shown in Figure 5.16, Figure 5.17, and Figure 5.18.

Subsequently, the refinement process of the identified solution is implemented, similar to the previously discussed case studies, to enhance its accuracy. The step-by-step results of this process are summarized in Table 5.7. By increasing the number of mesh segments to 400, employing the Hermite-Simpson method, and incorporating Earth's perturbation into the dynamic equations, a new optimal solution can be obtained in less than 10 minutes (specifically 531.28 seconds). This solution exhibits satisfactory relative errors compared to the NB propagator, with absolute position errors less than 60 km and velocity errors less than 0.15 m/s.

The detailed representations of the new high-accuracy optimal solution are presented in Figures 5.19, 5.20, 5.21. The results indicate that, in this case as well, the impact of third-body perturbations on the trajectory is not entirely negligible, adversely affecting performance in terms of ΔV and propellant consumption. It is evident that this perturbation leads to a new optimal solution, resulting in an extension of the propelled arc of the trajectory by approximately 10 additional days. Consequently, there is an increase of about 77 m/s in the required ΔV and an associated increase in the necessary propellant by approximately 2.3 kg.

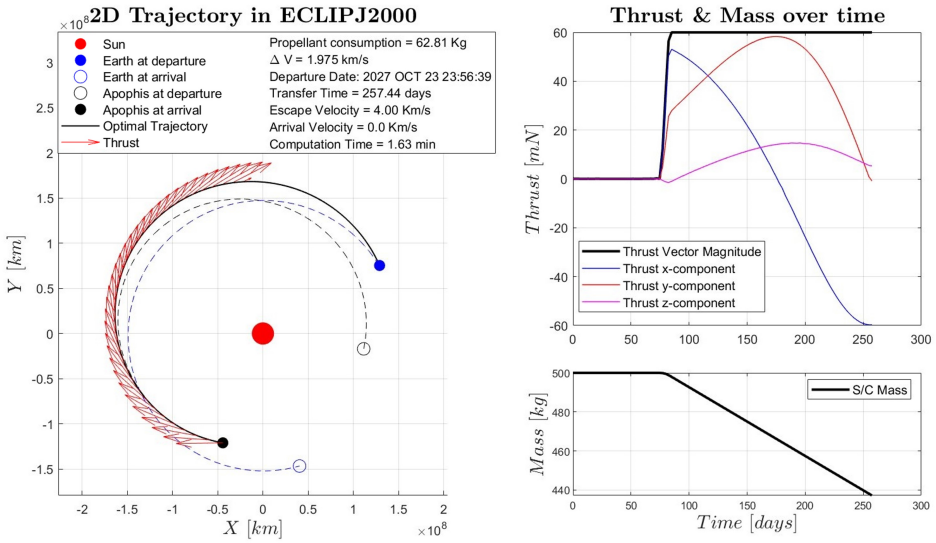

Figure 5.16: Optimal solution for Earth-Apophis transfer with minimum ToF. On the left: 2D Trajectory Visualization with main features. On the right: thrust and mass over transfer elapsed time.

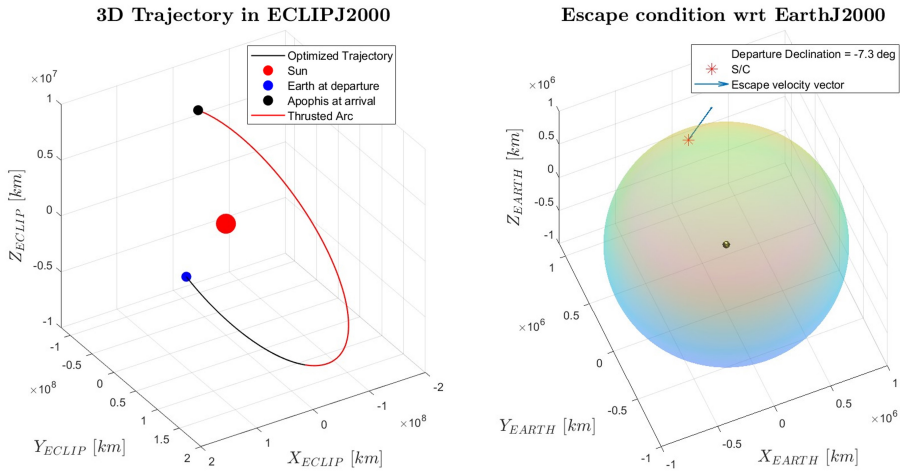

Figure 5.17: Optimal solution for Earth-Apophis transfer with minimum ToF. On the left: 3D trajectory visualization. On the right: departure condition wrt Earth J2000.

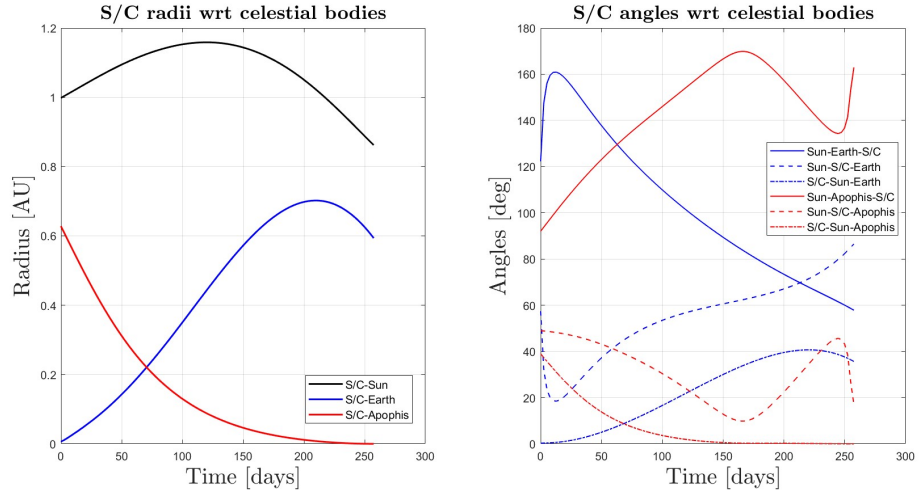

Figure 5.18: Optimal solution for Earth-Apophis transfer with minimum ToF. On the left: S/C radii with respect to relevant celestial bodies. On the right: S/C angles with respect to relevant celestial bodies.

Solution Refinement ID	0	1	2	3	4
Collocation Method	Trapezoidal	Hermite-Simpson	Hermite-Simpson	Hermite-Simpson	Hermite-Simpson
Mesh Intervals Number (N)	100	100	200	200	400
Dynamics	Central Body (2BP)	Central Body (2BP)	Central Body (2BP)	Central Body + Earth Perturbation	Central Body + Earth Perturbation
Computation time [s]	97.65	152.25	192.94	310.29	531.28
Optimization Outputs					
Departure Date [UTC]	2027 OCT 23	2027 OCT 22	2027 OCT 21	2027 OCT 22	2027 OCT 23
ToF [days]	257.44	257.75	258.76	262.58	264.62
$m_{propellant}$ [kg]	62.81	62.71	62.18	65.22	65.10
ΔV [km/s]	1.975	1.973	1.956	2.063	2.052
Errors wrt 2B propagator					
Position relative error	6.9e-4	1.9e-8	5.5e-9		
Position absolute error [km]	9.94e4	2.68	0.7		
Velocity relative error	7.5e-4	2.4e-8	4.2e-9		
Velocity absolute error [m/s]	24.3	7.74e-4	1.4e-4		
Errors wrt NB propagator					
Position relative error	1.5e-2	1.5e-2	1.6e-2	5.5e-6	4.0e-7
Position absolute error [km]	2.1e6	2.1e6	2.4e6	796	59.9
Velocity relative error	1.2e-2	1.2e-2	1.4e-2	4.6e-6	3.7e-7
Velocity absolute error [m/s]	371.6	370.5	417.7	1.39e-1	1.1e-2

Table 5.7: Summary table of the refinement process to the minimum ToF solution

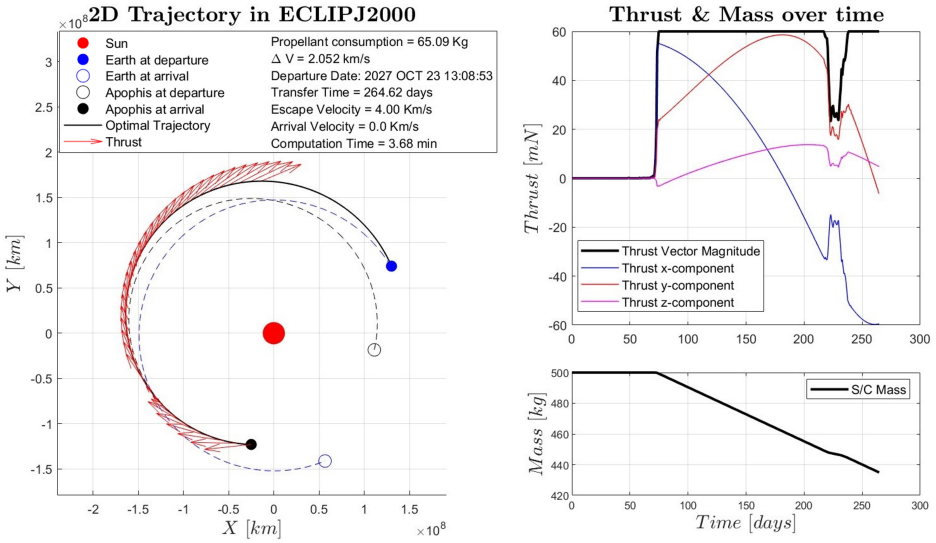

Figure 5.19: Refined optimal solution for Earth-Apophis transfer with minimum ToF (accounting for Earth's third-body perturbation). On the left: 2D Trajectory Visualization with main features. On the right: thrust and mass over transfer elapsed time.

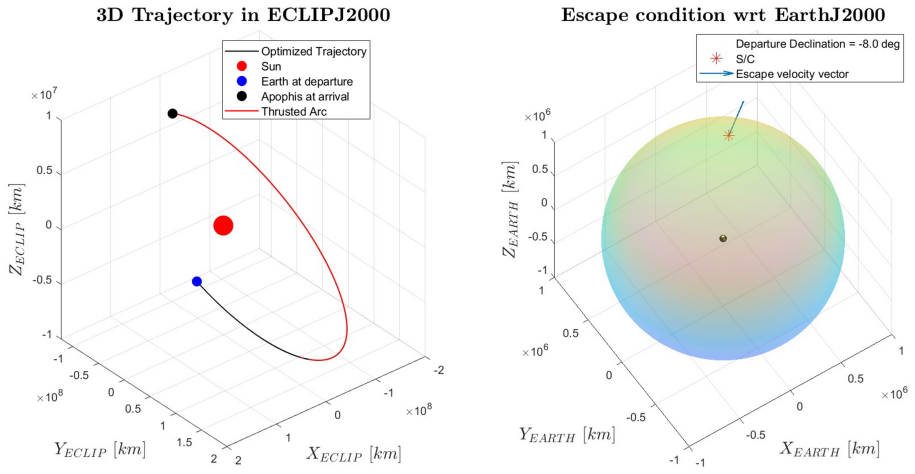

Figure 5.20: Refined optimal solution for Earth-Apophis transfer with minimum ToF (accounting for Earth's third-body perturbation). On the left: 3D trajectory visualization. On the right: departure condition wrt Earth J2000.

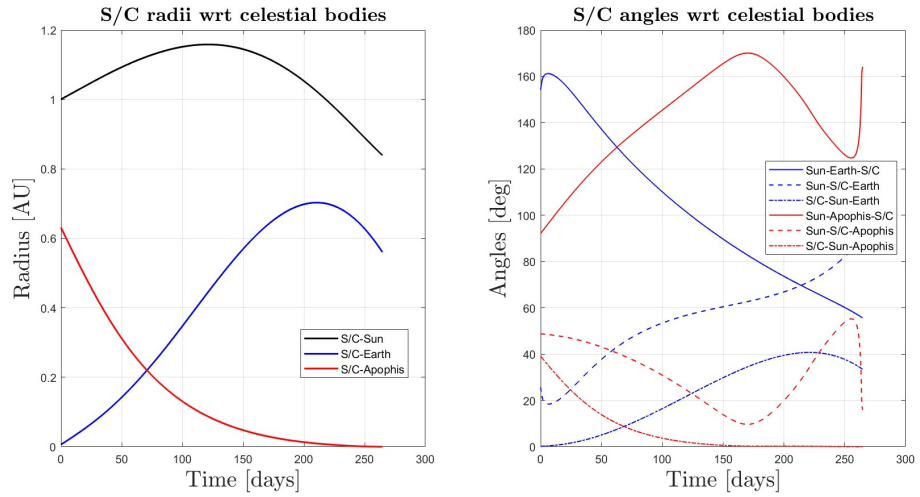

Figure 5.21: Refined optimal solution for Earth-Apophis transfer with minimum ToF (accounting for Earth's third-body perturbation). On the left: S/C radii with respect to relevant celestial bodies. On the right: S/C angles with respect to relevant celestial bodies.

5.2.3 Arrival Date Assessments

Given the objective of this preliminary transfer study, the focus is on solutions that optimize propellant consumption rather than time of flight. Solution A is characterized by an arrival date at Apophis of March 20, 2029. Since the fly-by of Apophis around Earth is expected on April 13, 2029, it can be theoretically

assumed that Solution A is suitable for this mission. The satellite would be able to reach the asteroid before the fly-by, thus enabling observation of the phenomenon.

However, it is important to consider potential requirements regarding the arrival date due to possible commissioning operations of the systems and payloads. These operations might be necessary to ensure that the satellite is in nominal observation condition upon arrival, requiring a certain period of time, which would necessitate a more stringent constraint on the arrival date. Based on the solutions previously computed by the optimization tool and depicted in the porkchop plot, these assumptions can be easily implemented retrospectively. It is possible to draw lines on the porkchop plot representing the limiting condition of ToF as a function of the launch epoch for a given arrival date constraint. These lines would indicate the subspace of solutions that remain valid while respecting this constraint. Specifically, all solutions above this line would not meet the constraint, whereas the solutions below it would satisfy the requirement.

The aforementioned description is illustrated in Figure 5.22, which depicts the previously presented porkchop plot with superimposed constraint lines for various arrival date limits. Specifically, three representative lines are shown: the first (in red) represents the limiting condition of an arrival date coinciding with the predicted fly-by date, the second (in magenta) represents the condition of arrival by March 13, 2029 (i.e., one month before the fly-by), and the last represents the condition of arrival by February 13, 2029 (i.e., two months before the fly-by). It can be observed that for each imposed arrival date constraint, a new optimal solution in terms of propellant consumption can be identified that also meets the constraint. Specifically, the Figure 5.22 indicates the three optimal solutions for the three cases: solution A meets the limiting condition of arrival by the fly-by date and corresponds to the previously discussed global minimum consumption solution. Solution A2 represents the new optimal solution for the condition of arrival within one month before the fly-by, while solution A3 represents the minimum-propellant solution for the condition of arrival within two months before the fly-by. In solution A2, the time of flight decreases by approximately 7 days, accompanied by a minor increase in propellant consumption of 0.56 kg. Conversely, solution A3 exhibits a substantial reduction in ToF by 55.45 days, with a corresponding increase in propellant consumption of approximately 4.6 kg, representing an increase of about 10 %. Therefore, it can be inferred that meeting the arrival date requirements is feasible without significantly compromising the required propellant consumption.

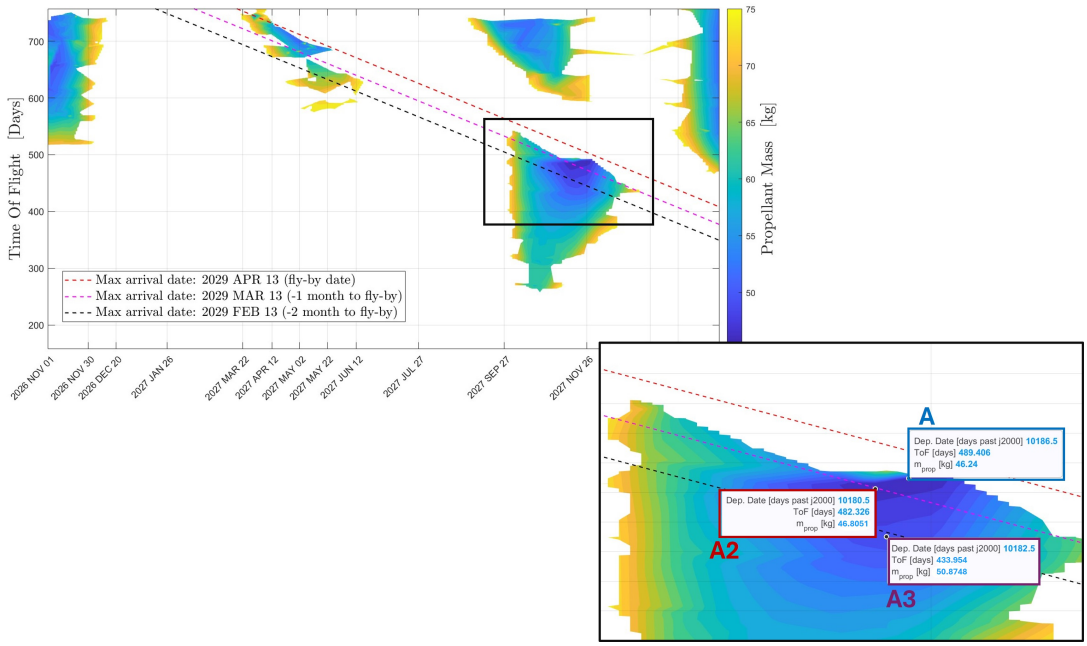

Figure 5.22: Porkchop plot of the Earth-Apophis transfer opportunities with the limit arrival date conditions superimposed (dashed lines). Detail the 3 minimum-propellant solutions A, A2, A3 found for each limit arrival date.

5.3 Future Developments

This section aims to summarize the future prospects opened by this thesis work, with the objective of enhancing or completing it. As demonstrated in the test cases presented in Chapter 4 and the study cases in Chapter 5, the optimization process using direct collocation is significantly influenced by the initial guesses employed as starting solutions for the SQP solvers, both in computational terms and in the local results obtained. This work utilized very simple guess generation methods to produce useful results within the available time frame. However, this approach often led to issues related to convergence and computation time for each solution. Future improvements could involve studying and implementing more efficient guess generation methods, which would provide the solver with an initial solution for the rendezvous problem that is at least very close to feasibility.

To this end, one could initially consider using shape based methods for generating guesses. These methods are based on the generation of geometrically approximated trajectories that closely adhere to the constraints of the real problem and provide a good estimate of the objective function. Various state-of-the-art approximation methods have been proposed and implemented, based on models using Chebyshev polynomials (see Patel et al., 2009[35]), logarithmic spirals (see Roa, 2018[37]), or Fourier series (see Taheri and Abdelkhalik, 2012[41]).

Another alternative could be to use the Sims-Flanagan model for an initial

approximate direct optimization of the problem, whose solution can then be used as a guess for the more accurate problem with direct collocation. This model (see Sims and Flanagan, 1997[38]) proposes a direct optimization method in which the low-thrust trajectory is discretized into a finite number of segments (arcs), denoted by N, where the continuous thrust is approximated by a corresponding impulsive $\Delta \boldsymbol{V}$. These $\Delta \boldsymbol{V}_i$ (corresponding to the i-th segment) will be the variables of the optimization problem and their magnitudes must not exceed the magnitude of ΔV_{max} associated with the maximum available thrust level, which can be calculated as follows

$$\Delta V_{max} = (T_{max}/m)(t_f - t_0)/N \tag{5.2}$$

where T_{max} id the maximum thrust of the low-thrust engine, m is the mass of the S/C at the beginning of the leg, t_0 is the departure epoch and t_f is the arrival epoch of the leg. The S/C mass is then propagated using the rocket equation:

$$m_{i+1} = m_i \, e^{-\Delta V_i / g_0 I_{sp}} \tag{5.3}$$

where i represents the i-th segment of the leg, g_0 is the standard gravity, and I_{sp} is the specific impulse of the low-thrust engine. The trajectory is propagated forward from the departure point (e.g., the departure planet) and backward from the arrival point (e.g, the arrival planet) to a matchpoint, usually halfway through a leg. At that matchpoint the forward propagated S/C state and the backward proagated S/C state should meet in order to achieve a feasible solution of the problem (or at least they should respect a tolerance on the mismatch in position, velocity, and mass). The concept of this model is visualized in Figure 5.23.

Thus, this problem can be transcripted into a NLP problem where the objective is to maximize the final S/C mass while meeting the constraints on maximum ΔV and state mismatch. To properly interface that approach with direct collocation employed in the tool developed within this thesis work, it is possible to set the number of segments in the Sims-Flanagan model equal to the number of mesh intervals of the collocation model. Then, it is possible to extrapolate the guess for the collocation variables from the results of the Sims-Flanagan, in particular by finding a way to transform the impulsive $\Delta \boldsymbol{V}_i$ vector of each segment in control vectors on each mesh point of the collocation grid ($\boldsymbol{u}_k$, for the k-th grid point). The first step is to formulate an expression of the thrust control vector given an impulse in a specific point i in which the impulse is applied, that could be expressed starting from Eq. 5.2 as

$$\boldsymbol{u}_i = \frac{m_i \, \Delta \boldsymbol{V}_i}{T_{max} \, \Delta t_i} \tag{5.4}$$

where Δt_i is the ToF relative to the i-th segment of the Sims-Flanagan model. It is important to notice that this impulse is applied at the midpoint of the segment in the Sims-Flanagan model, while in the collocation model the control

vector is defined in the grid points, that match with the vertices of the segment. A way to compute the control vector value for the k-th mesh point of the collocation grid can be to compute the average value of the control vectors associated to the previous and following segments of the Sims-Flanagan model. Since for the first collocation mesh point there is not a previous segment, the control of the previous segment can be set equal to zero. The same approach can be used for the last collocation point, by setting to zero the control of the next segment.

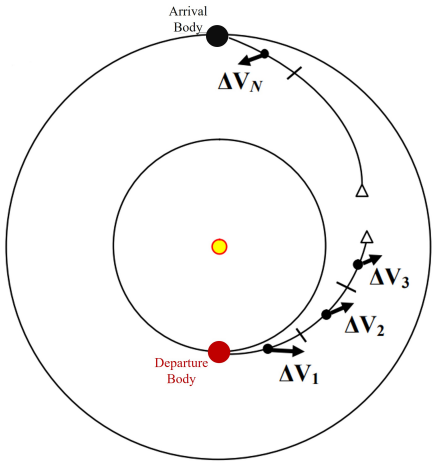

Figure 5.23: Sims-Flanagan model concept visualization. The triangle represents the matchpoint between the forward-propagated segment (from the departure point) and the backward-propagated segment (from the arrival point).

The generation methods for guesses described above are therefore essential for enhancing the performance of the optimization tool presented in this thesis. The primary goal is to reduce the computation time for each trial within the multi-start process, thereby significantly decreasing the total computational time required to find the global optimum. Additionally, these methods are crucial for addressing more complex low-thrust interplanetary trajectory optimization problems, such as those involving the possibility of interplanetary flybys.

A potential future enhancement of the tool involves extending the optimization of low-thrust interplanetary trajectories to include scenarios where planetary swing-bys are possible. These swing-bys can be modeled as instantaneous changes in the spacecraft's heliocentric velocity, resulting from the appropriate alignment of the satellite and planet positions at specific points along the trajectory. For a comprehensive understanding of the modeling and implementation of this phenomenon within the optimal control framework, readers are referred to the literature (see Kostantinov and M.Thein, 2017[25], Pesenti, 2024[36]). In this context, the trajectory would be segmented into a finite number of legs between planets, for which optimal solutions would be determined, alongside managing the optimal fly-by conditions with the desired planets. It is thus evident that addressing this problem necessitates the generation of accurate initial

guesses for the various legs to ensure convergence to optimal solutions within acceptable time frames.

Furthermore, extending the optimization to such interplanetary trajectories would necessitate a more sophisticated global solution search management compared to the multi-start approach presented in this thesis. Specifically, it would be essential to manage not only the launch dates and overall transfer times within the external global search loop but also the sequence of fly-bys to be executed. This management might require the implementation of more advanced and efficient algorithms than the multi-start method, such as heuristics or metaheuristics, for instance genetic algorithms, as utilized in the open-source tool MOLTO-IT (see Morante et al., 2018[30]).

Concluding Remarks

In conclusion, this thesis successfully achieved its intended objectives by addressing and resolving the challenges associated with the optimal control of interplanetary low-thrust trajectories. This accomplishment was facilitated by a thorough analysis of methodologies for managing and solving optimal control problems, with the selection of transcription via direct collocation as the foundational technique for the entire work. This choice, combined with an experimental analysis of implementation methodologies and algorithms for solving such complex problems, led to the successful development of a numerical tool capable of producing satisfactory results. The tool was tested on various case studies, and the consistency of the results was verified against the existing literature on the subjects.

Furthermore, the subsequent step involved the successful development and testing of the refinement process for the optimal solutions found accounting for the two-body problem. This advancement enabled the incorporation of third-body perturbations into the system dynamics, allowing for the assessment of their impact on the interplanetary leg. Consequently, the optimal trajectories were re-optimized, incorporating refinements in collocation discretization and methods employed, to account for these perturbations, ultimately yielding high-accuracy solutions.

This thesis work serves as a solid foundation for potential future studies and developments. Among them, the primary possible advancements were introduced, as the enhancement of the numerical tool performance through the generation of more effective and faster initial guess solutions, and the extension of the optimization problem to low-thrust interplanetary trajectories which include the possibility of multiple gravity assists.

Bibliography

[1] R.R. Bate, D.D. Mueller, and J.E. White. *Fundamentals of Astrodynamics*. New York: Dover Publications, 1971.

[2] V.M. Becerra. "Practical Direct Collocation Methods for Computational Optimal Control". In: *Modeling and optimization in space engineering*. Ed. by G. Fasano and J. Pintér. Springer, 2013, pp. 33–59.

[3] R. Bellman. *Dynamic Programming*. Courier Corporation, 2013.

[4] J.T. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear programming*. Philadelphia: Society for Industrial and Applied Mathematics, 2010.

[5] J.T. Betts and W.P. Huffman. "Application of Sparse Nonlinear Programming to Trajectory Optimization". In: *Journal of Guidance* 15.1 (1992), pp. 198–206.

[6] Britannica. *Interplanetary Exploration*. 2024. URL: `https://www.britannica.com/science/space-exploration` Accessed on Aug. 26, 2024.

[7] A.E. Bryson and Y.C. Ho. *Applied Optimal Control*. New York: Blaisdell, 1969.

[8] C. Büskens and D. Wassel. "The ESA NLP Solver WORHP". In: *Modeling and Optimization in Space Engineering*. Ed. by G. Fasano and J. Pintér. Springer, 2013, pp. 85–109.

[9] L. Casalino and G. Colasurdo. "Indirect Methods for the Optimization of Spacecraft Trajectories". In: *Modeling and optimization in space engineering*. Ed. by G. Fasano and J. Pintér. Springer, 2013, pp. 141–158.

[10] M. Casanova-Álvarez, F. Navarro-Medina, and D.Tommasini. "Feasibility study of a Solar Electric Propulsion mission to Mars". In: *Acta Astronautica* 216 (2024), pp. 129–142.

[11] H.D. Curtis. *Orbital mechanics for engineering students: Revised Reprint*. Butterworth-Heinemann, 2020.

[12] K. Dannenmayer and S. Mazouffre. "Elementary scaling laws for the design of low and high power Hall effect thrusters". In: *Progress in Propulsion Physics, EDP Sciences* (2009), pp. 601–616.

[13] ESA Science & Technology. *BepiColombo*. 2024. URL: `https://www.esa.int/Science_Exploration/Space_Science/BepiColombo` Accessed on Aug. 26, 2024.

[14] ESA Science & Technology. *SMART-1, Electric Spacecraft Propulsion*. 2019. URL: `https://sci.esa.int/s/8ZkRa1A` Accessed on Aug. 26, 2024.

[15] ESA Space & Safety. *Introducing Ramses, ESA's mission to asteroid Apophis*. 2024. URL: `https://www.esa.int/Space_Safety/Planetary_Defence/Introducing_Ramses_ESA_s_mission_to_asteroid_Apophis` Accessed on Aug. 26, 2024.

[16] G. Fasano and J.D. Pintér. *Modeling and Optimization in Space Engineering*. Springer, 2014.

[17] G. Genta and P.E. Maffione. "Optimal Low-thrust Trajectories for Nuclear and Solar Electric Propulsion". In: *Acta Astronautica* (2015).

[18] P.E. Gill, W. Murray, and M.A. Saunders. "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization". In: *SIAM Review* 47.1 (2005), pp. 99–131.

[19] C. Hofmann, A.C. Morelli, and F. Topputo. "On the performance of discretization and trust-region methods for on-board convex low-thrust trajectory optimization". In: *AIAA Scitech 2022 Forum*. 2022.

[20] D. Izzo. "Pygmo and pykep: Open source tools for massively parallel optimization in astrodynamics (the case of interplanetary trajectory optimization)". In: *Proceedings of the Fifth International Conference on Astrodynamics Tools and Techniques, ICATT*. 2012.

[21] R.G. Jahn. *Physics of Electric Propulsion*. New York: McGraw-Hill, 1968.

[22] M. Kim. "Continuous Low-Thrust Trajectory Optimization: Techniques and Applications". PhD Thesis. Blacksburg, Virginia: Virginia Polytechnique Institute and State University, 2005.

[23] C.A. Kluever. “Optimal Low-Thrust Interplanetary Trajectories by Direct Method Techniques”. In: *The Journal of the Astronautical Sciences* 45.3 (1997), pp. 247–262.

[24] M. Knauer and C. Büskens. “Understanding concepts of optimization and optimal control with WORHP Lab”. In: *Proceedings of the 6th International Conference on Astrodynamics Tools and Techniques, ICATT*. 2016.

[25] M.S. Kostantinov and M.Thein. “Method of interplanetary trajectory optimization for the spacecraft with low thrust and swing-bys”. In: *Acta Astronautica* 136 (2017), pp. 297–311.

[26] M.Minoux. *Mathematical Programming: Theory and Algorithms*. New York: John Wiley & Sons, 1986.

[27] MathWorks. *Optimization Toolbox Documentation*. 2024. URL: `https://it.mathworks.com/help/optim/` Accessed on Aug. 26, 2024.

[28] MathWorks. *Parallel Computing Toolbox Documentation*. 2024. URL: `https://it.mathworks.com/help/parallel-computing/` Accessed on Aug. 26, 2024.

[29] D. Morante, M.S. Rivo, and M. Soler. “A Survey on Low-Thrust Trajectory Optimization Approaches”. In: *Aerospace* 8.88 (2021).

[30] D. Morante et al. “MOLTO-IT A Multi-Objective Low-Thrust Optimizer for Interplanetary Trajectories”. In: *Proceedings of the International Conference on Astrodynamics Tools and Techniques, ICATT*. 2018.

[31] A.C. Morelli et al. “Initial Trajectory Assessment of a low-thrust option for the RAMSES Mission to (99942) Apophis”. In: *Advances in Space Research* (2024).

[32] NASA Science. *Hayabusa2*. 2024. URL: `https://science.nasa.gov/mission/hayabusa-2/` Accessed on Aug. 26, 2024.

[33] NASA’s Navigation and Ancillary Information Facility (NAIF). *About SPICE*. 2024. URL: `https://naif.jpl.nasa.gov/naif/aboutspice.html` Accessed on Aug. 26, 2024.

[34] NASA’s Navigation and Ancillary Information Facility (NAIF). *An Overview of Reference Frames and Coordinate Systems in the SPICE Context*. 2023. URL: `https://naif.jpl.nasa.gov/pub/naif/toolkit_docs/Tutorials/pdf/individual_docs/17_frames_and_coordinate_systems.pdf` Accessed on Aug. 26, 2024.

[35] P.R. Patel et al. “An algorithm for generating feasible low thrust interplanetary trajectories”. In: *Proceedings of the International Electric Propulsion Conference, IEPC*. 2009.

[36] G. Pesenti. “Optimization of interplanetary transfers with flybys and deep-space maneuvers”. Master’s Thesis. Turin, Italy: Politecnico di Torino, 2024.

[37] Javier Roa. “New Analytic Solution with Continuous Thrust: Generalized Logarithmic Spirals”. In: *Journal of Guidance, Control, and Dynamics* 39.10 (2018), pp. 2336–2351.

[38] J.A. Sims and S.N. Flanagan. *Preliminary design of low-thrust interplanetary missions*. Tech. rep. Jet Propulsion Lab., CA United States: NASA, 1997. URL: `https://ntrs.nasa.gov/api/citations/20000057422/downloads/20000057422.pdf` Accessed on Aug. 26, 2024.

[39] Steinbeis-Forschungszentrum Optimierung, Steuerung und Regelung. *WORHP: Large-scale Sparse Nonlinear Optimisation*. 2024. URL: `https://worhp.de/` Accessed on Aug. 26, 2024.

[40] O. Sutherland, D. Stramaccioni, and J. Benkhoff. “BepiColombo: ESA’s Interplanetary Electric Propulsion Mission to Mercury”. In: *Proceedings of the International Electric Propulsion Conference, IEPC*. 2019.

[41] E. Taheri and O. Abdelkhalik. “Shape Based Approximation of Constrained Low-Thrust Space Trajectories using Fourier Series”. In: *Journal of Spacecraft and Rockets* 49.3 (2012).

[42] F. Topputo and G. Aguiar. “A Technique for Designing Earth-Mars Low-Thrust Transfers Culminating in Ballistic Capture”. In: *Proceedings of the International Conference on Astrodynamics Tools and Techniques, ICATT*. 2018.

[43] F. Topputo and C. Zhang. “Survey of Direct Transcription for Low-Thrust Space Trajectory Optimization with Applications”. In: *Abstract and Applied Analysis*. Vol. 2014. 1. Wiley Online Library. 2014, p. 851720.

[44] D.A. Vallado. *Fundamentals of Astrodynamics and Applications*. Microcosm Press and Kluwer Academic Publishers, 2001.

[45] S.N. Williams and V.L. Coverstone-Carroll. “Benefits of solar electric propulsion for the next generation of planetary exploration mission”. In: *The Journal of the Astronautical Sciences* 45.2 (1997), pp. 143–160.