

Master of Aerospace Engineering Research Project

**Optimal control of Trajectory of reusable launcher in
OpenMDAO/dymos**

S3 Project Report

Authors: Alberto Fossà, Giuliana Elena MICELI

Due date of report: 27 March 2020
Actual submission date: 25 March 2020

Begin of the project: 29 January 2019

Duration: 14 Months

Tutors: L. BEAUREGARD, J. MORLIER, S. LIZY-DESTREZ

Contents

1	Introduction	1
2	Context and Key Issues	2
3	Work done during Semester 2	3
3.1	2D Model	3
3.1.1	State of the Art	3
3.1.2	Analytical formulation	3
3.1.3	Numerical solution	4
3.2	3D model	5
3.2.1	State of the Art	5
3.2.2	Analytical Formulation	5
3.3	Results obtained in semester 2	7
3.3.1	2D Simulations results	7
3.3.2	3D Simulations results	12
4	Work done during Semester 3	14
4.1	Low Lunar Orbit to Moon surface transfers	14
4.2	Low Lunar Orbit to Highly Elliptical Orbit transfers	14
4.3	Surrogate Models	15
4.4	Documentation	15
5	Investigation Methods	16
5.1	Equations of Motion	16
5.2	Optimal Control Problem	17
5.3	Low Lunar Orbit to Moon Surface	18
5.4	Low Lunar Orbit to Highly Elliptical Orbit	19
5.4.1	Single Phase Trajectory	19
5.4.2	Three Phases Trajectory	20
5.4.3	Finite Escape Burn	21
5.5	Surrogate Model	23
5.5.1	Surrogate Modeling Toolbox	24
5.5.2	OpenMDAO Metamodels	24
5.6	Implemented Python Package	24
5.6.1	Code Architecture	25
5.6.2	Documentation	25
6	Results and Analysis	26
6.1	Low Lunar Orbit to Moon Surface	26
6.1.1	Single-Phase Descent with Variable Thrust	27
6.1.2	Single-Phase Descent with Variable Thrust and Vertical Landing	28

6.2	Low Lunar Orbit to Highly Elliptical Orbit	29
6.2.1	Single-Phase Trajectory	29
6.2.2	Three-Phases Trajectory	30
6.2.3	Finite Escape Burn	31
6.3	Surrogate Models	32
6.3.1	Moon surface to Low Lunar Orbit	33
6.3.2	Low Lunar Orbit to Highly Elliptical Orbit	33
7	Conclusions and Perspectives	34
Bibliography		36
A Surrogate Models		38
B UML Diagrams		39

Acronyms

BC Boundary Condition.

CR3BP Circular Restricted Three-Body Problem.

EOM Equations of Motion.

HEO Highly Elliptical Orbit.

HTML Hypertext Markup Language.

IC Initial Condition.

IDW Inverse-Distance Weighting.

IPOPT Interior Point Optimizer.

ISP Specific Impulse.

ISS International Space Station.

IVP Initial Value Problem.

LaTOM Launcher Trajectory Optimization Module.

LEO Low Earth Orbit.

LHS Latin Hypercube Sampling.

LLO Low Lunar Orbit.

LOP-G Lunar Orbital Platform - Gateway.

LS Least-Squares approximation.

MDA Multidisciplinary Analysis.

MDO Multidisciplinary Optimization.

NLP Non Linear Programming Problem.

NRHO Near Rectilinear Halo Orbit.

ODE Ordinary Differential Equation.

OOP Object-Oriented Programming.

QP Second-order Polynomial approximation.

R2BP Restricted Two-Body Problem.

RBF Radial Basis Functions.

RMTB Regularized Minimal-energy Tensor-product splines (B-splines).

RMTC Regularized Minimal-energy Tensor-product splines (Cubic Hermite splines).

SMT Surrogate Modeling Toolbox.

SNOPT Sparse Non-Linear Optimizer.

SRP Solar Radiation Pressure.

TOF Time of Flight.

TWR Thrust over initial Weight Ratio.

UML Unified Modeling Language.

Declaration of Authenticity

This assignment is entirely our own work. Quotations from literature are properly indicated with appropriated references in the text. All literature used in this piece of work is indicated in the bibliography placed at the end. We confirm that no sources have been used other than those stated.

We understand that plagiarism (copy without mentioning the reference) is a serious examinations offense that may result in disciplinary action being taken.

Date: 25 March 2020

Signatures:

Abstract

The challenges of the future human space exploration era are currently one of the most debated topics in the space technology field. The main focus is on Moon missions for the development of a human outpost on the lunar surface dedicated to scientific and business services.

In this context, the international community is working on the development of the Lunar Orbital Platform-Gateway (LOP-G) foreseen in an orbit around the Earth-Moon Lagrangian point L_2 to be used both as scientific laboratory and intermediate service station for lunar and interplanetary missions. LOP-G to Moon transportation systems have to be conceived to bring crews and goods from the Moon surface to the station and vice-versa. The design must optimize their architecture in terms of costs and allows for vehicle reusability.

The aim of this study is to propose a trajectory optimization package for a reusable lunar lander. Being the fuel management one of the major concerns, the tool leverages optimal control techniques to minimize the propellant consumption. To achieve the aforementioned objectives OpenMDAO and dymos, two Python-based libraries, are employed to carry out the numerical simulations while orbital mechanics and optimal control theory are at the bottom of the developed models.

Keywords: Trajectory Optimization, Optimal Control Theory, Lunar missions, MDO, Python

1 Introduction

In the context of a renewed interest in human space exploration, this study aims to implement a trajectory optimization tool for Earth-Moon missions. In particular, given the realization of a future international space station, the Lunar Orbital Platform Gateway (LOP-G), orbiting around the Lagrangian point L_2 of the Earth-Moon system, the main interest is to perform minimum-fuel transfers from the future lunar outpost, most probably located near the Moon South Pole, to the LOP-G. A further description of the project context can be found in chapter 2.

Some studies have already been published, especially regarding two-dimensional transfers from a Low Lunar Orbit (LLO) to the Moon surface, which is indeed one of the phases for a lunar mission. These papers have been taken as a reference during the first part of the project, as described in chapter 3 section 3.1.1. Firstly, the authors' results have been emulated to test the correctness of the implemented algorithms and new simulations have been then carried out to find optimal trajectories from the Moon surface to the LLO and back. The analytical formulation and the numerical implementation are described in details in sections 3.1 and 5.3. The two-dimensional problem has been then extended to a three-dimensional one starting from other studies present in literature. The new computations require a different mathematical model and a new optimal control problem formulation as explained in section 3.2. All simulated scenarios for 2D and 3D trajectories, considering constant and variable thrust, as well as path-free and path constrained transfers, are discussed in section 3.3 and 6.1.

Starting from LLO to Moon surface trajectories, the work has been extended to analyze the second part of a Moon to LOP-G transfer which starts from the LLO and reaches the NRHO where the station is located. Due to the complex dynamical model required to describe such transfers, a first planar approximation has been evaluated. The target orbit has been replaced with an Highly Elliptical Orbit (HEO) with similar characteristics of a 9 : 2 lunar synodic resonant L_2 NRHO and considered coplanar with the departure LLO. This topic is extensively discussed in section 5.4 and the results coming from the optimization are presented in section 6.2.

Finally, to easily integrate the optimization package in a multidisciplinary optimization tool, the previous simulations have been carried out for a given range of I_{sp} and thrust over weight ratios (twr) to explore the different results given by all possible combinations of these two design variables. The obtained surrogate models will constitute the trajectory module that together with other disciplines will be used to search for the best architecture for a Moon transportation vehicle. All the libraries and techniques employed to build these models are described in section 5.5 while the resulting models are presented in section 6.3.

The package has been written in Python using the OpenMDAO and dymos libraries to transcribe the optimal control problem and the IPOPT and SNOPT gradient-based optimizers to solve the resulting Non Linear Programming Problem. The detailed code architecture can be found in section 5.6 together with an introduction to the code documentation.

Considering the adopted strategy and the obtained results, but also the current status of the study, some conclusions and future perspectives are finally drawn in chapter 7.

2 Context and Key Issues

The interest for Moon missions and human space exploration has seen a huge increase in the last few years. The resulting demand in tools and techniques to achieve this goal, coming from both space agencies and private companies, has brought to a huge development in every field of science and more investigations are still going on. Apollo missions are the main starting point for these studies. Indeed, Apollo 11 was able to bring the first astronaut on the Moon surface in 1969 and to perform the first soft landing/depart of the history. After the program has been stopped, only Low Earth Orbit (LEO) missions have still seen a human presence: the Mir space station, the International Space Station (ISS) and the Tiangong space laboratory are a few example of them.

In the new era of Human Space Exploration, the presence on the Moon and on other colonizable Solar System bodies will consist on actual human villages for both scientific and economic purposes. *In-situ* resources exploitation is an example of activities with a double objective: propellant production for vehicles refueling but also extraction of valuable materials to be shipped back on Earth.

The first step in this direction is the Lunar Orbital Platform Gateway (LOP-G), a space station which is currently being developed and that will replace the ISS at its end of life. The LOP-G will not orbit around the Earth, but it will be placed in a Near Rectilinear Halo Orbit (NRHO) around the Lagrangian Point L_2 , in order to serve both as scientific laboratory and in-orbit outpost to facilitate future human interplanetary transfers.

Differently from the past, the current exploration missions considers a whole new transportation system concepts which includes reusability and modularity. For what concerns the last one, a structure composed of combinable modules allows to satisfy the requirements imposed by different transfer characteristics. Indeed, since a direct transfer from the Moon to the LOP-G is not always feasible, it will be split into two main phases: ascent from the Moon surface to a Low Lunar Orbit (LLO) and transfer from LLO to NRHO. Each of them presents unique challenges which can be better handled designing vehicles with specific characteristics. *Isp*, thrust/weight ratio (*twr*) and payload capacity being three of the main design parameters. On the other hand, reusability is the main requirement for recurrent interplanetary transfers and long lasting mission with human crew.

In this scenario, the propellant mass carried on board of such reusable launcher should be carefully handled in order to maximize the payload mass and account for additional reserve. Therefore, the trajectory design must be optimized in order reduce the fuel consumption while traveling from the Moon to the LLO, then from the LLO to the LOP-G and back.

Trajectory optimization is one of the disciplines involved in the design of a space vehicle such as a reusable launcher. For a complete optimal architecture structure, propulsion system and possibly other disciplines are also taken into account. In this context the trajectory block is plugged into a multidisciplinary optimization tool to determine the best possible design for a reusable launcher commuting in the cislunar realm.

3 Work done during Semester 2

The second semester has been dedicated to an in-depth study of State-of-the-Art methodologies for the modellization and simulation of optimal, continuous-thrust transfer trajectories and the implementation of suitable Python routines for the computation of fuel-optimal, two-dimensional and three-dimensional ascent and descent trajectories from the Moon surface to a circular Low Lunar Orbit (LLO).

3.1 2D Model

3.1.1 State of the Art

For what concerns trajectory models in lunar missions, Wilhite [1], Ramanan [2] and Remesh [3] conduct different analysis regarding the descent scenario. The first one gives a comparison between the flight data from Apollo missions performed until 1972 and an optimized solution that takes into account the safety requirements for a crewed vehicle. The last two authors study different solutions for an automatic probe soft lunar landing with different initial conditions and path constraints. Starting from a 100 km circular parking orbit, the descent is performed with a first deorbit burn to inject into a ballistic Hohmann transfer followed by a powered descent to allow a soft landing.

The ascent has been investigated by Zhang [4] who proposes an *on/off* control scheme to regulate the thrust magnitude thus finding the most fuel-efficient transfer for different final orbits. Also Ma [5] proposes a solution for the ascent phase using a constant thrust ascent trajectory with an initial constraint to perform a safe vertical takeoff. This is achieved imposing a zero tangential velocity for a predefined time or altitude interval, in this case 10 s or 500 m . Indeed, the natural behavior of the optimizer would lead the launcher to depart almost horizontally following the body surface until the necessary speed to reach the target orbit is achieved. This solution would not respect the safety constraints related to the presence of geographical features on the surface and a human crew on-board.

3.1.2 Analytical formulation

The problem has been formulated as a continuous-time optimal control problem in which the underlying dynamics describes the motion of a unitary mass spacecraft subject to the gravitational pull of the Moon modeled as a perfect spherical and homogeneous body and the acceleration due to its own thrust. The Equations of Motion (EOMs) are expressed in polar coordinates as given by equations 3.1.

$$\begin{aligned}
\dot{r} &= u \\
\dot{\theta} &= \frac{v}{r} \\
\dot{u} &= -\frac{\mu}{r^2} + \frac{v^2}{r} + \frac{T}{m} \sin \alpha \\
\dot{v} &= -\frac{uv}{r} + \frac{T}{m} \cos \alpha \\
\dot{m} &= -\frac{T}{Isp g_0}
\end{aligned} \tag{3.1}$$

Where r is the distance from the central body, θ the spawn angle, u and v the radial and tangential velocities, m the spacecraft mass, T the thrust magnitude, α the thrust direction, I_{sp} the specific impulse, g_0 the standard gravitational acceleration and μ the Moon standard gravitational parameter.

Every scenario is formulated as an optimal control problem where the objective is to minimize the propellant consumption or equivalently maximize the final spacecraft mass. Appropriate boundary and path constraints guarantee the spacecraft departs from a predetermined state and inject on a specified orbit or lands smoothly on the Moon surface without violating any physical law or bound imposed on the engine performances.

The mathematical formulation is given in equations 3.2 and 3.3 [6].

Minimize:

$$J = \phi(t_f, \mathbf{x}_f) + \int_{t_0}^{t_f} L(t, \mathbf{x}, \mathbf{u}) dt \tag{3.2}$$

Subject to:

$$\begin{aligned}
\dot{\mathbf{x}} &= \mathbf{f}(t, \mathbf{x}, \mathbf{u}) \\
\mathbf{x}(t_0) &= \mathbf{x}_0 \\
\mathbf{u} &\in U \\
\Psi(t_f, \mathbf{x}_f) &= 0 \\
S(\mathbf{x}) &\geq 0
\end{aligned} \tag{3.3}$$

Where \mathbf{x} are the states variables, \mathbf{u} the controls, \mathbf{x}_0 the initial conditions (ICs) at time t_0 , Ψ the terminal constraints, S the path constraints, $\mathbf{f}(t, \mathbf{x}, \mathbf{u})$ the dynamics and J the objective function. The aim is to find the optimal control profile $\mathbf{u}(t)$ and the corresponding state history $\mathbf{x}(t)$ that lead to a minimum for J .

Specific formulations of the problem have been derived for both ascent and descent trajectories as detailed in section 5.3.

3.1.3 Numerical solution

A direct method is employed in the solution of the optimal control problem. In particular, the infinite-dimensional problem is discretized through a High-Order Gauss-Lobatto [7] or Radau Pseudospectral [8] transcription method into a finite-dimensional Non Linear Programming Problem (NLP) and finally solved with gradient-based optimizers such as IPOPT [9, 10] and SNOPT [11] that uses differential information on the design variables to find the optimum of the objective function. Their main advantages are a fast convergence when starting from a good initial guess and a small error on the obtained solution. On the other side, gradient-free algorithms are based

on natural selection and converge to the optimal solution evolving an initial population composed by several states and controls profiles. Contrarily to gradient-based methods, gradient-free algorithms do not require any initial guess but convergence is achieved after a higher number of iterations and the final accuracy is usually lower.

An initial guess has to be provided for every single case and its complexity depends on the constraints imposed on the trajectory. For the first simulations a linear interpolation of the boundary conditions is accurate enough to find the optimal solution, while in the most complex scenarios a more precise guess has to be provided to obtain a feasible solution. The last cases arise for example when the constraint on the minimum safe altitude is enforced to obtain an ascent trajectory that avoids the Moon geographical features.

In this context an accurate initial guess is computed solving a simplified ascent trajectory divided in three separate phases for which a semi-analytic solution can be easily obtained. The corresponding initial approximation is thus obtained as follows:

- powered phase at constant radius $r \equiv R$ and constant thrust $T \equiv T_{max}$ to acquire the tangential velocity at the periapsis of an Hohmann transfer whose apoapsis altitude coincides with the target LLO altitude
- coasting phase (Hohmann transfer) from the lunar surface up to the target LLO
- impulsive burn at the apoapsis of the previous maneuver to inject the spacecraft in the desired circular parking orbit

The employed transcription methods are provided by the Python package dymos [12] while the interface between Python code and NLP solvers is guaranteed by the open-source libraries OpenMDAO [13] and pyOptSparse [14].

3.2 3D model

3.2.1 State of the Art

The three-dimensional problem has been analyzed in other papers by some of the authors already seen in section 3.2.1. Ma [15, 16] studies the optimal control problem for ascent trajectories given the presence of geographical features modeled as simple cones. In its studies the impact of a path constraint to impose a minimum safe distance from the mountainous landscape is analyzed with respect to the global fuel consumption.

3.2.2 Analytical Formulation

The spacecraft dynamics under the keplerian two-body assumption is described in Cartesian coordinates respect to an inertial reference frame whose origin coincides with the center of the Moon.

Following the example presented by Benson [17], the Equations of Motion (EOMs) for the three-dimensional scenario are presented below:

$$\dot{\mathbf{r}} = \mathbf{v} \quad (3.4)$$

$$\dot{\mathbf{v}} = -\frac{\mu}{r^3}\mathbf{r} + \frac{T}{m}\hat{\mathbf{u}} \quad (3.5)$$

$$\dot{m} = -\frac{T}{Isp g_0} \quad (3.6)$$

$$\begin{aligned}\mathbf{r} &= [x, y, z] \\ \mathbf{v} &= [v_x, v_y, v_z] \\ \hat{\mathbf{u}} &= [u_x, u_y, u_z]\end{aligned}\tag{3.7}$$

Where \mathbf{r} and \mathbf{v} are the spacecraft position and velocity vectors, r the distance from the central body, T the thrust magnitude and $\hat{\mathbf{u}}$ a unit vector that represents the thrust direction.

The resulting optimal control problem for the three-dimensional ascent trajectory is then formulated as follows:

Minimize:

$$J = -m(t_f) \quad (\text{objective function})\tag{3.8}$$

Subject to:

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{u}) \quad (\text{EOMs})\tag{3.9}$$

$$\mathbf{x} = [\mathbf{r}, \mathbf{v}, m] \quad (\text{state variables})\tag{3.10}$$

$$\mathbf{u} = [T, \hat{\mathbf{u}}] \quad (\text{control variables})\tag{3.11}$$

where 3.9 groups together equations 3.4 to 3.6. The initial conditions $t_0, \mathbf{x}(t_0)$ are specified as:

$$\begin{cases} t_0 = 0 \\ \mathbf{r}(t_0) = \mathbf{r}_0 \\ \mathbf{v}(t_0) = \mathbf{0} \\ m(t_0) = m_0 \end{cases}\tag{3.12}$$

With \mathbf{r}_0 initial position on the Moon surface and m_0 initial spacecraft mass. The final boundaries $\Psi(t_f, \mathbf{x}_f)$ are given by:

$$\begin{cases} t_f = \text{free} \\ \mathbf{h}(t_f) - \mathbf{h}_{tgt} = \mathbf{0} \\ \mathbf{e}(t_f) - \mathbf{e}_{tgt} = \mathbf{0} \\ m(t_f) = \text{free} \end{cases}\tag{3.13}$$

Finally, the path constraints are specified as follows:

$$\begin{cases} \|\mathbf{r}(t)\| > R \\ m(t) > 0 \\ T_{min} < T < T_{max} \\ \|\hat{\mathbf{u}}\| = 1 \end{cases}\tag{3.14}$$

The numerical solution is finally obtained as described in subsection 3.1.3.

3.3 Results obtained in semester 2

The results obtained during the second semester are summarized in the following paragraphs. Where not specified, the initial guess required to solve the associated NLP problem is obtained from a linear interpolation of the boundary conditions as previously stated while the time of flight is estimated on a case-by-case basis. Moreover, the numerical results are compared with the ones present in the literature when possible and validated using the *simulate* method. This method is aimed to reproduce the obtained solution through an explicit integration step. More in details, a Runge-Kutta integration scheme is employed to solve the Initial Value Problem (IVP) formulated starting from the specified EOMs, imposed Initial Conditions and previously computed optimal controls profiles. If the error between IVP and optimal solutions is below the specified threshold, the obtained transfer trajectory is said to be feasible.

All the solutions are computed setting the following numerical values for the different constant parameters:

Constant	Units	Value
g_0	m/s	9.80665
μ	m^3/s^2	$4.9028 \cdot 10^{12}$
R	m	$1737.4 \cdot 10^3$

Table 3.1: Constant parameters

3.3.1 2D Simulations results

Different scenarios are investigated in the two-dimensional model. The results can be split in ascent and descent trajectories, where for the first set the following cases have been studied:

- Constant thrust
- Variable thrust
- Variable thrust with safe altitude constraint
- Constant thrust with vertical take off

In the descent case only two cases have been treated during the second semester:

- Constant thrust
- Constant thrust with vertical landing

Ascent Trajectories

The first three ascent trajectories have been computed for the Isp , twr , initial spacecraft mass m_0 and final orbit altitude H given in table 3.2:

	Units	Value
Isp	s	450.0
twr	-	2.1
m_0	kg	1.0
H	m	$86.87 \cdot 10^3$

Table 3.2: Isp , twr , initial mass and target orbit altitude for the 2D ascent trajectories

Firstly, the constant thrust trajectory is computed imposing that:

$$T_{min} = T_{max} = T \quad (3.15)$$

In this case a Gauss-Lobatto transcription method with 10 segments and a third-order interpolating polynomial is used to convert the continuous-time optimal control problem into the corresponding NLP problem finally solved with the IPOPT routine. The results obtained with an initial estimated time of flight $tof = 500$ s are given in table 3.3.

	Units	Value
tof	s	476.13
m_p/m_0	-	0.3680

Table 3.3: Time of flight and propellant fraction for the 2D ascent trajectory with constant thrust

Secondly, the constant thrust assumption is removed and is substituted with the following inequality constraints:

$$0 \leq T \leq T_{max} \quad (3.16)$$

Where T_{max} is computed from the twr given in table 3.2 and the time of flight initial guess is set equal to $tof = 4000$ s.

The optimal transfer trajectory is obtained computing the solution of the NLP problem that arises from a Gauss-Lobatto transcription with 150 segments and a third-order interpolating polynomial using the IPOPT solver. The results are reported in table 3.4. As it can be deducted from them, the thrust variation allows a reduction of fuel consumption with the drawback of an important increase in the time of flight. In case no specific constraints are imposed on the mission duration, this compromise can be accepted.

	Units	Value
tof	s	3697.56
m_p/m_0	-	0.3364

Table 3.4: Time of flight and propellant fraction for the 2D ascent trajectory with variable thrust

As demonstrated in the previous results and discussed by Ma [5], if the thrust is let vary from a minimum to a maximum value, the spacecraft clears the ground with a really shallow angle to minimize the fuel consumption, thus posing several safety concerns due to the presence of lunar highlands around the launch site. Consequently, a successful mission can be only guaranteed if an appropriate constraint is added to specify a minimum safe distance between the spacecraft and the ground throughout the whole ascent phase. This is done using an analytical formula representing the shape of a simplified geographical feature on the Moon surface and adding an inequality constraint on the position to impose a minimum altitude throughout the whole phase as described in section 5.3.

The numerical results are obtained applying a Gauss-Lobatto transcription method with 200 segments and a third-order interpolating polynomial and solving the resulting NLP problem with the optimizer IPOPT. In this case the NLP solver is unable to converge starting from an initial guess obtained as a simple linear interpolation of the boundaries, thus requiring the semi-analytic solution described in section 3.1.3 to be computed and set as starting point for the subsequent iterations. The time series of both states and controls variables are depicted in figure 3.1, while the

time of flight and the propellant mass resulting from the optimization are listed in table 3.5. The numerical values for the asymptotic safe altitude h and the constraint slope s are also provided in the same table.

	Units	Value
tof	s	3367.77
m_p/m_0	-	0.3550
h	m	5000
s	-	100

Table 3.5: Time of flight, propellant fraction, minimum safe altitude and slope for the 2D ascent trajectory with variable thrust and constrained minimum altitude

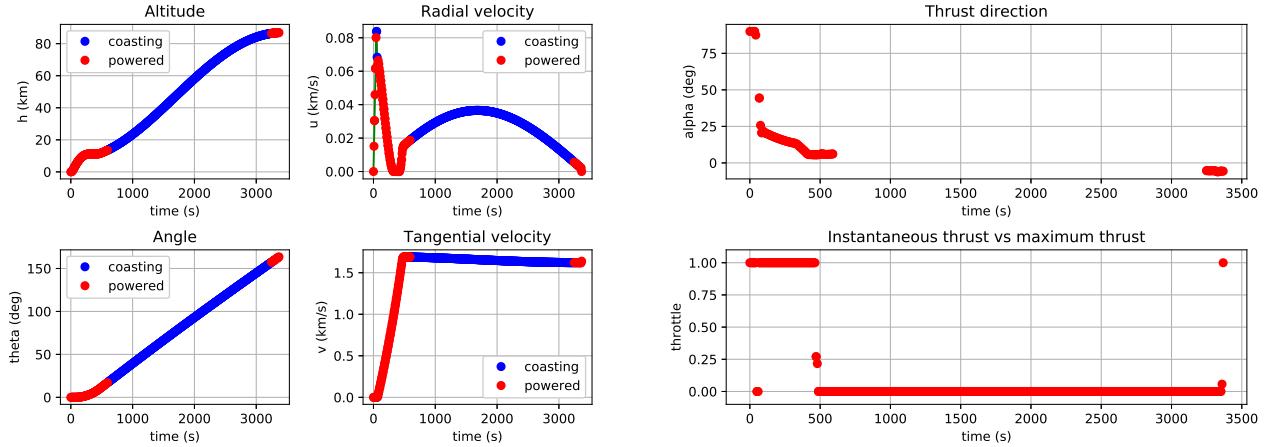


Figure 3.1: States and controls profiles for the 2D ascent trajectory with variable thrust and constrained minimum altitude

Finally, the approach used by Ma [5] was tested as well to compare the previous result to cope with the safety constraints to the one present in the literature. In this simulation an appropriate constraint is applied to ensure a vertical take-off. The thrust direction is in fact forced to remain perpendicular to the Moon surface for a certain amount of time while a second solution is obtained imposing the same limitation up to a predetermined altitude.

In this case, different values of target orbit altitude H , initial spacecraft mass m_0 , twr and Isp have been chosen in order to be comparable with the paper results. These values can be found in table 3.6. The numerical solutions are then computed after transcribing the optimal control problem into the corresponding NLP formulation applying a Gauss-Lobatto transcription method with 3 and 15 segments and a third-order interpolating polynomial for the first and second phase respectively. The obtained NLP is then handily solved with the IPOPT routine.

	Units	Value
H	km	51.44
m_0	kg	4869
Isp	s	309
twr	-	1.95

Table 3.6: Target orbit altitude, initial spacecraft mass, Isp and twr for the 2D ascent trajectory with initial vertical constraint

Tables 3.7 and 3.8 show that the propellant consumption is lower in the first case since the vertical constraint is relaxed earlier than in the second solution for which 25.09 s are required to achieve a 500 m altitude.

	Units	Value
tof	s	448.31
m_f/m_0	—	0.4686

Table 3.7: Time of flight and propellant fraction for the 2D ascent trajectory with initial vertical constraint for $t < 10\text{ s}$

	Units	Value
tof	s	451.06
m_f/m_0	—	0.4714

Table 3.8: Time of flight and propellant fraction for the 2D ascent trajectory with initial vertical constraint for $r < R + 500\text{ m}$

In conclusion, the previous results show that using a constant thrust propulsion system reduces the time of flight of almost 8 times with respect to the use of a throttleable engine. On the other side, the introduction of a variable thrust magnitude allows to further reduce the required fuel fraction. Then, taking into account the practical feasibility of the mission, safety constraints have to be taken into account to assure the safeness of a human crew. A first solution to this problem is to impose a path constraint to set a minimum safe altitude and avoid possible geographical features. The resulting propellant fraction is higher than a constraint-free trajectory, but it represents a more flexible solution compared to the one proposed by Ma [5] which imposes a vertical take off for a fixed amount of time or altitude interval.

Descent Trajectories

The results coming from the two-dimensional descent trajectory are presented in the following paragraphs. Only two cases have been analyzed in the second semester, the constant thrust descent and the constrained vertical landing transfers. For both simulations the Isp , twr , departing LLO altitude, initial spacecraft mass m_0 and periapsis altitude h_p are presented in table 3.9:

	Units	Value
Isp	s	400
twr	-	0.9
H	km	100
m_0	kg	1.0
h_p	km	15

Table 3.9: Isp , twr , LLO altitude, initial spacecraft mass and periapsis altitude for the 2D descent trajectories

In the constant thrust trajectory the simulation is split into three main steps: a deorbit burn followed by an Hohmann transfer from the LLO to a lower periapsis altitude and a final powered descent. The optimization is performed only on the third phase from the periapsis of the ballistic arc until the Moon surface. The optimal transfer trajectory is obtained computing the solution of the NLP problem that arises from a Gauss-Lobatto transcription with 20 segments and a 3th-order interpolating polynomial using the SNOPT solver. The time of flight initial guess is set equal to $tof = 1000\text{ s}$. The numerical results are reported in table 3.10.

	Units	Value
tof	s	1121.64
m_p/m_0	-	0.4180

Table 3.10: Time of flight and propellant fraction for the 2D descent trajectory without vertical constraints

The descent with constraint vertical landing consist into a two-phases powered descent trajectory at constant thrust divided as follows:

- Hohmann transfer from the initial circular LLO at 100 km altitude to a lower periapsis altitude h_p
- Attitude-free powered descent from $h_p = 15 \text{ km}$ to 4 km altitude
- Vertical powered descent from 4 km altitude to the Moon surface

The corresponding optimal control problem is transcribed into a NLP using a Gauss-Lobatto transcription with third-order interpolating polynomials and 100 and 20 segments for the first and second phases respectively. The NLP is then solved using the gradient-based solver IPOPT. In this case the results reported in table 3.11 have been computed with $Isp = 400 \text{ s}$ and $twr = 0.9$.

	Units	Value
tof	s	1158.81614
m_p/m_0	-	0.4318

Table 3.11: Time of flight and propellant fraction for the 2D descent trajectory with vertical constraints in thrust

The corresponding states and controls time series are depicted in figure 3.2a and 3.2b respectively.

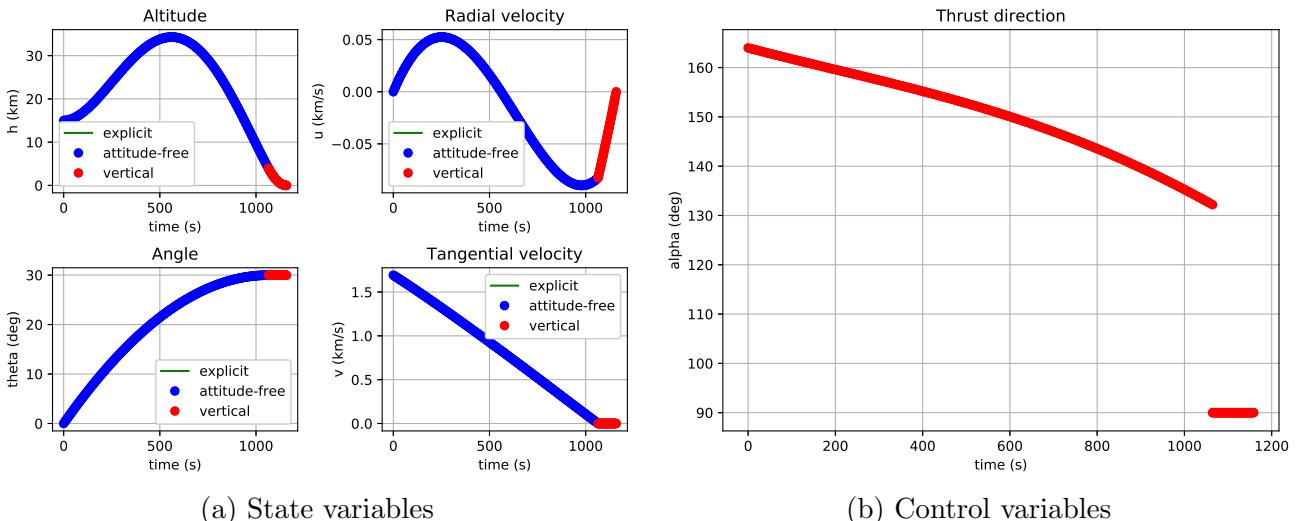


Figure 3.2: Time series for the two-phases descent trajectory with constraint vertical landing

From the comparison of the two simulations it is seen that the overall propellant fraction is greater when the vertical constraint is applied and grows from 0.4180 to 0.4318. Notwithstanding the higher value of m_p/m_0 , the increased fuel consumption is necessary to ensure a soft and harmless landing.

3.3.2 3D Simulations results

Three-dimensional ascent trajectories are sought combining together the dynamic model presented in section 3.2.2 with a constant thrust assumption $T_{min} = T = T_{max}$.

Two cases have been analyzed:

- Trajectory to a polar Low Lunar Orbit
- Trajectory to an High Elliptical Orbit with $i = 60^\circ$

Both solution assume the Moon South Pole as the designed launch site.

The numerical values of Isp , twr and initial spacecraft mass m_0 used in the subsequent simulations are summarized in table 3.12.

	Units	Value
Isp	s	450.0
twr	-	2.1
m_0	kg	1.0

Table 3.12: Isp , twr and initial spacecraft mass for the 2D ascent trajectories

The first trajectory simulates the departure from the Moon South Pole to a circular polar LLO. The required initial guess is computed as a linear interpolation of the boundaries imposed on the state variables \mathbf{r} , \mathbf{v} and the control variable \hat{u} , thus requiring to provide both the initial values and an estimation of the final ones at the beginning of the simulation.

The optimal solution is obtained applying a Gauss-Lobatto transcription method with 10 segments and a third-order interpolating polynomial while the corresponding NLP is handily solved by the IPOPT routine. The obtained results in terms of the required time of flight and propellant fraction are reported in table 3.13.

	Units	Value
tof	s	476.29
m_p/m_0	-	0.3681

Table 3.13: Time of flight and propellant fraction for the 3D ascent trajectory to a polar LLO

In this solution the selected values for Isp , twr and m_0 are equal to the ones used for the two-dimensional transfers in section 3.3.1. As expected, the results are very similar to those obtained for the 2D ascent trajectory at constant thrust since also in this case an in-plane maneuver at T_{max} is performed, thus corroborating both results.

In the second simulation an highly elliptical orbit is targeted such that the Moon South Pole, the designed launch site, does not belong to the orbit plane. Consequently, the spacecraft is forced to perform both in-plane and out-of-plane maneuvers during the transfer.

Also in this case the initial guess on the states is computed as a linear interpolation between the initial and final spacecraft state vectors as described for the previous simulation. The continuous time optimal control problem is then transcribed applying a Gauss-Lobatto transcription method with 10 segments and a third-order interpolating polynomial while the resulting NLP problem is solved using the IPOPT routines.

The numerical results reported in table 3.14 demonstrate that a huge amount of propellant is required when the spacecraft is asked to perform an important out-of-plane maneuver. On the other side, looking at figure 3.3 the optimal solution is achieved letting the launcher reaching

	Units	Value
tof	s	876.73
m_p/m_0	-	0.6776

Table 3.14: Time of flight and propellant fraction for the 3D ascent trajectory to an highly elliptical orbit

the final velocity to insert the transfer trajectory with a very shallow angle with respect to the ground, as already pointed out in section 3.3.1. Therefore, if no other constraints are added, the vehicle travels along the surface until it reaches the correct speed to leave the vicinity of the Moon before performing the final injection maneuver in the farthest point from the surface so that the amount of propellant needed is the minimum possible. Even if performed in the most optimal point, the aforementioned plane change maneuver still requires much more propellant than the coplanar counterpart.

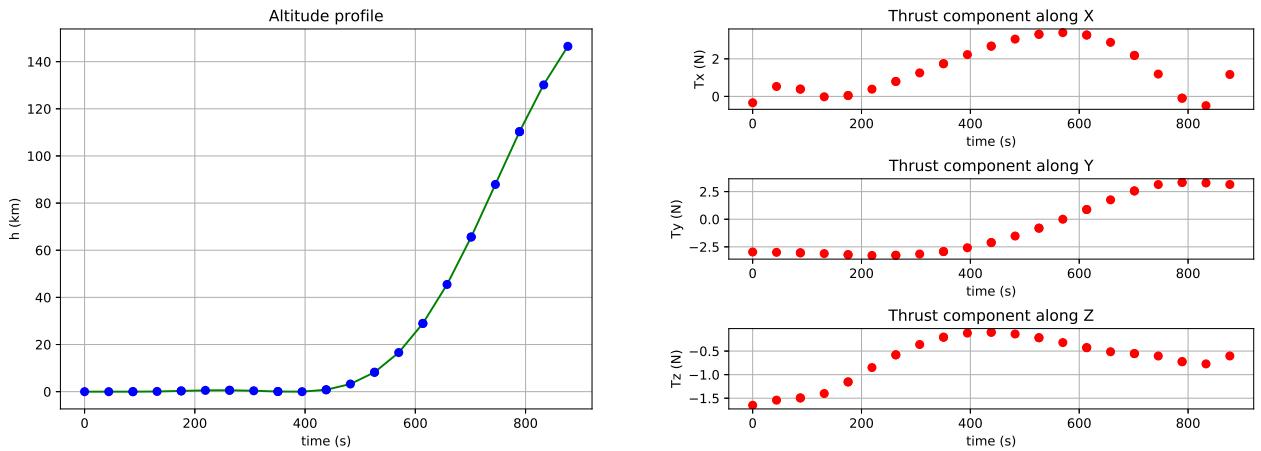


Figure 3.3: Altitude and thrust profiles for the 3D ascent trajectory to an highly elliptical orbit

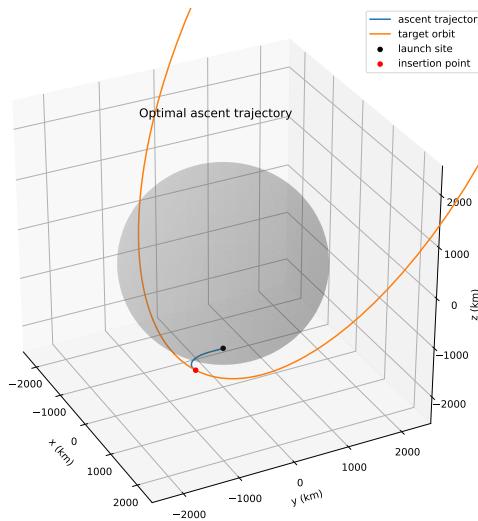


Figure 3.4: 3D ascent trajectory to an highly elliptical orbit

4 Work done during Semester 3

The second part of the year has been dedicated to further develop the two-dimensional descent trajectories from the Low Lunar Orbit (LLO) to the Moon surface. The variable thrust and vertical landing cases are added to the scenarios already studied in chapter 3. Going further, the second part of the trajectory to reach the LOP-G from the Moon surface, namely the transfer from the parking LLO to the Near Rectilinear Halo Orbit (NRHO), has been investigated. Finally, within the context of Multidisciplinary Optimization (MDO), the results coming from the described models have been translated into surrogate models. In this way, several solutions with all possible combination of *Isp* and *twr* have been computed for each ascent and descent trajectory to create a complete database to be called from the outer optimization loop. Along with the structure and propulsion disciplines, this will allow an optimal design of a reusable space launcher.

4.1 Low Lunar Orbit to Moon surface transfers

The work done during the first part of semester 3 focused on the implementation of all remaining descent cases for the LLO to Moon surface transfers.

The previously developed models consider a descent phase at constant thrust with an optional constrained vertical landing. Removing the first hypothesis, a variable thrust trajectory is obtained. This modification brings to a possible decrease in the propellant consumption which is one of the main requirements for the mission, causing on the other side an increase in the corresponding time of flight as observed for the ascent case. The implemented models and simulations, as well as the time of flight and propellant mass resulting from the subsequent optimization, are described in section 5.3 and 6.1.1.

Going further, a safety constraint is added to the variable thrust transfer in order to satisfy the requirements for a crewed mission in a realistic scenario. The same implementation as for the ascent trajectory with variable thrust and constrained minimum altitude is employed to model the Moon's geographical features and impose a path constraint throughout the descent phase. The analytical formula are described in section 5.3. Similarly to previous simulations, the problem is then transcribed into a NLP and solved with gradient-based optimizers. The corresponding results are presented in section 6.1.2.

4.2 Low Lunar Orbit to Highly Elliptical Orbit transfers

The complete Moon to LOP-G transfer consists into transferring the spacecraft from the lunar surface to the NRHO where the space station will orbit. In this sense, the whole trajectory can be divided into two main phases:

- Departure from the Moon surface to reach the parking orbit at 100 km
- LLO to NRHO transfer to catch the Lunar Orbital Platform-Gateway

For the first part, different techniques have already been analyzed in the previous chapters while the second transfer adds new challenges to be tackled.

The main one is due to the different dynamical models used to describe the first part of the trajectory and the NRHO. Indeed, this last kind of orbit is only defined in the Circular Restricted Three Body Problem (CR3BP) which considers the influence of two massive bodies (Earth and Moon in this case) that move in a circular orbit about their common barycenter on a third body (the spacecraft) whose negligible mass does not affect the motion of the former primaries [18]. The underlying dynamics is commonly studied in the so-called synodic frame, a rotating frame centered at the barycenter of the primaries with the x axis going from the first to the second primary and the z one aligned with the primaries' angular momentum vector. In this frame NRHOs appear as simple closed orbits about one of the first two Lagrangian points, while their geometry is much more difficult to visualize in an inertial frame in which keplerian orbits are defined. A proper handling of these two different frames is thus needed to easily characterize both departure and arrival spacecraft states. Moreover, a more complex dynamical model would also require additional adjustments in the implemented optimization framework.

Due to the several complexities imposed by the CR3BP framework, it has been decided to compute first an approximation of the LLO to NRHO transfer in the well-known restricted two-body problem scenario. In this sense the transfer is computed considering departure and arrival orbits as fixed and coplanar thus modeling the NRHO as frozen in an inertial frame. In addition to that, since NRHOs are not confined within a plane, the approximation takes into account an Highly Elliptical Orbit (HEO) with similar characteristics as of those of the former one.

This low fidelity model allows a first computation of the required fuel and time of flight to reach an hypothetical NRHO without taking into account a rendezvous with the LOP-G and the time evolution of the orbit itself, thus adopting a sub-optimal approach in the selection of the departure and arrival states. Nevertheless, considering these solutions being used for a preliminary design of a reusable launcher, the approximation is acceptable.

The results of the LLO to HEO transfers are analyzed in sections 5.4 and 6.2.

4.3 Surrogate Models

As a piece of a bigger optimization process, part of the third semester has been dedicated to the development of surrogate models that would allow the trajectory discipline being integrated in the multidisciplinary design and optimization tool for a complete lunar lander.

Two different libraries have been employed for each developed scenario while a final comparison between the results coming from the two models allows to select the most accurate one. A detailed description of these models and the respective results are given in sections 5.5 and 6.3.

4.4 Documentation

Given the multiplicity of functions implemented in the developed code and the complexity of its structure, a complete documentation of the implemented package [19] is provided to ease its usage.

Even if the redacted documentation tries to be as much complete as possible, a previous knowledge of the Python programming language, the OpenMDAO and dymos libraries as well as some fundamentals of orbital mechanics are expected for a correct and productive use of the aforementioned package.

5 Investigation Methods

The investigation methods and tools employed throughout this work are extensively discussed in this chapter to provide the reader a comprehensive overview of dynamical models, optimal control problem formulations ad numerical algorithms implemented in the developed Python package to reproduce the results detailed in chapter 6.

5.1 Equations of Motion

All the analysis have been conducted in the restricted two-body problem framework assuming a unitary mass spacecraft and a perfectly spherical and homogeneous central attracting body representing the Moon. Those two masses constitute an isolated physical system in which all perturbations due to the Moon non-uniform gravitational field, solar radiation pressure and third-body effect caused by the presence of the Earth, the Sun and all other celestial bodies in the Solar System are neglected.

Under those assumptions, the Equations of Motion (EOMs) that describe the variation of the spacecraft state due to the gravitational pull of the Moon and the acceleration produced by its own engines are written in polar coordinates centered at the Moon center as follows:

$$\begin{aligned}\dot{r} &= u \\ \dot{\theta} &= \frac{v}{r} \\ \dot{u} &= -\frac{\mu}{r^2} + \frac{v^2}{r} + \frac{T}{m} \sin \alpha \\ \dot{v} &= -\frac{uv}{r} + \frac{T}{m} \cos \alpha \\ \dot{m} &= -\frac{T}{Isp g_0}\end{aligned}\tag{5.1}$$

With r radial distance from the Moon center, θ spawn angle, u radial velocity, v tangential velocity, m spacecraft mass, T thrust magnitude, α thrust direction, I_{sp} specific impulse, μ central body standard gravitational parameter and g_0 standard sea-level gravity. It has to be noticed here that, depending on the problem, the thrust magnitude T can be constant, variable or even null depending on the different flight phases. Moreover, the initial condition on the spawn angle θ is arbitrary and only the difference $\Delta\theta = \theta_f - \theta_0$ between its final and initial values has a physical interpretation.

5.2 Optimal Control Problem

The main goal of this work is to determine the most fuel-efficient transfer trajectories in the cislunar space. For that purpose, the optimal control problem in trajectory optimization is stated as follows [6]:

Minimize:

$$J = -m(t_f) \quad (5.2)$$

Subject to:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(t, \mathbf{x}, \mathbf{u}) \\ \mathbf{x}(t_0) &= \mathbf{x}_0 \\ \mathbf{u} &\in U \\ \Psi(t_f, \mathbf{x}_f) &= \mathbf{0} \\ S(\mathbf{x}) &\geq 0 \end{aligned} \quad (5.3)$$

With $\mathbf{x} = [r, \theta, u, v, m]$ state variables, $\mathbf{u} = [T, \alpha]$ control variables, $\mathbf{f}(t, \mathbf{x}, \mathbf{u})$ dynamics given by 5.1, \mathbf{x}_0 Initial Conditions (ICs), Ψ terminal constraints and S path constraints. The aim is to determine the optimal control profile $\mathbf{u}(t)$ and the corresponding state history $\mathbf{x}(t)$ that lead to a minimum of the objective function J . As seen from 5.2, the objective is to minimize the opposite of the final spacecraft mass $-m(t_f)$ or equivalently minimize its propellant consumption.

Two main approaches have been developed to solve an optimal control problem as stated above: direct methods and indirect ones. The difference between the two has already been detailed in the semester 2 outcomes while a quick overview is provided by Shirazi et al. [20]. This work adopts only direct methods for which the solution of the optimal control problem is split into multiple steps:

- Discretization of the state and control variables on a finite set of points determined by the selected transcription method, number of segment and order of the corresponding interpolating polynomial [21]. Gauss-Lobatto [7] and Radau-Pseudospectral [8] being the ones employed in this work.
- Transcription of the continuous-time optimal control problem into a Non-Linear Programming Problem (NLP) constituted by a finite set of decision variables and constraints.
- Solution of the obtained NLP using gradient-free or gradient-based optimizers.
- Interpolation of the discrete time solution according to the previously selected transcription method.

Already available libraries written in Python, C/C++ and FORTRAN are used in the developed software package to provide the transcription methods and gradient-based solvers required to carry out this work:

- OpenMDAO [13] and dymos [12] to transcribe the continuous-time problem and derive the corresponding NLP
- pyOptSparse [14] to interface the aforementioned libraries with high-performance optimizers
- IPOPT [9, 10] and SNOPT [11] as gradient-based solvers to iteratively compute the NLP solution

A more detailed overview of the different functionalities and interfaces between them is provided in section 5.6.1.

5.3 Low Lunar Orbit to Moon Surface

A first model has been developed to analyze descent trajectories from a Low Lunar Orbit (LLO) to the Moon surface thus improving the preliminary studies conducted in semester 2 and widely inspired by the work of Ramanan [2] and Remesh [3]. Rather than splitting the transfer trajectory into three subsequent phases (Hohmann transfer, free-attitude powered descent at constant thrust, vertical touchdown at constant thrust), this new approach is based on a single powered phase trajectory with variable thrust magnitude and direction that covers the entire descent from the initial circular LLO to the Moon surface where the spacecraft is imposed to be at rest. Its implementation is based on the corresponding ascent case already modeled in the previous semester from which the corresponding path constraints and techniques to generate a suitable initial guess have been taken and adapted to the newly implemented scenario.

The optimal control problem is formulated with the formalism presented in section 5.2 defining appropriate boundary constraints on the spacecraft states that guarantee the last leaves the specified LLO and lands safely on the Moon surface. Their mathematical description is given below:

$$\begin{aligned} \left\{ \begin{array}{l} t_0 = 0 \\ r(t_0) = R + H \\ \theta(t_0) = \text{free} \\ u(t_0) = 0 \\ v(t_0) = \sqrt{\mu/(R + H)} \\ m(t_0) = m_0 \end{array} \right. & \quad (5.4) \\ \left\{ \begin{array}{l} t_f = \text{free} \\ r(t_f) = R \\ \theta(t_f) = \theta_f \\ u(t_f) = 0 \\ v(t_f) = 0 \\ m(t_f) = \text{free} \end{array} \right. & \quad (5.5) \end{aligned}$$

With R Moon radius, H initial orbit altitude, m_0 initial spacecraft mass and θ_f final spawn angle. Differently from the ascent case, the initial angle θ_0 is left free while a terminal boundary constraint has been imposed on the same state variable. This formulation allows in fact an easier handling of the path constraint imposed on the minimum altitude to be maintained by the spacecraft above the Moon surface as described below. The corresponding numerical value θ_f is completely arbitrary and determined on a case by case basis depending on the computed initial guess.

Three path constraints are then enforced on the radial distance $r(t)$, spacecraft mass $m(t)$ and thrust magnitude $T(t)$ as follows:

$$\left\{ \begin{array}{l} r(t) > R \\ m(t) > m_{dry} \\ T_{min} < T(t) < T_{max} \end{array} \right. \quad (5.6)$$

The first of 5.6 guarantees that the spacecraft remains always above the Moon surface, the second one that only the propellant carried on-board is burnt while the last enforces the thrust magnitude being limited between its minimum and maximum values specified for each case of interest.

Finally, an additional constraint on the state variables r, θ is optionally included in the problem formulation to impose a minimum safe altitude from the Moon surface while far from the landing site followed by a predetermined descent slope in proximity of the final touchdown location. Similarly to the vertical takeoff described in the ascent case, this constraint forces the spacecraft to perform a final vertical landing thus avoiding possible geographical features located around the designated landing site and satisfy the safety requirements imposed by the mission.

The aforementioned constraint is expressed as follows:

$$r(t) > R + \frac{h_{min} \cdot R \cdot \theta(t)}{R \cdot \theta(t) + h_{min}/s} \quad (5.7)$$

With h_{min} asymptotic minimum safe altitude far from the landing site and s constraint slope. Due to the different geometry, s assumes only negative values while the sharpness of the landing profile is increased for $s \rightarrow -\infty$.

5.4 Low Lunar Orbit to Highly Elliptical Orbit

Different models have been then developed to provide a first estimate of the propellant fraction required to boost the spacecraft from a circular Low Lunar Orbit to an Highly Elliptical Orbit (HEO) with a periapsis close to the Moon surface and an apoapsis very far from the former. The main aim is to obtain a first approximate solution for a LLO to Near Rectilinear Halo Orbit (NRHO) transfer in a lower fidelity model represented by the Restricted Two-Body Problem (R2BP). Indeed, the target HEO is chosen as the best planar approximation of an NRHO, a three dimensional periodic orbit defined in the Circular Restricted Three-Body Problem (CR3BP) framework where the gravitational pull of the Earth is also taken into account [18].

Three models have been implemented and tested so far:

- A single phase trajectory with open departure state on the initial LLO and fixed arrival state at the HEO apoapsis
- A three phases trajectory with open departure state on the initial LLO and open arrival state close to but not necessarily coinciding with the HEO apoapsis
- A single phase trajectory to inject the spacecraft into a ballistic transfer arc followed by an impulsive burn carried out at the HEO apoapsis for the final injection

These models are presented in more detail in sections 5.4.1 to 5.4.3 below.

5.4.1 Single Phase Trajectory

The first model consists into a transfer trajectory modeled as a single powered phase with variable thrust magnitude. Appropriate boundary constraints are imposed on the initial and final spacecraft states as follows:

$$\begin{cases} t_0 = 0 \\ r(t_0) = R + H \\ \theta(t_0) = \text{free} \\ u(t_0) = 0 \\ v(t_0) = v_{LLO} \\ m(t_0) = m_0 \end{cases} \quad (5.8)$$

$$\begin{cases} t_f = \text{free} \\ r(t_f) = r_{a,HEO} \\ \theta(t_f) = \theta_f \\ u(t_f) = 0 \\ v(t_f) = v_{a,HEO} \\ m(t_f) = \text{free} \end{cases} \quad (5.9)$$

With v_{LLO} initial circular velocity on the departure LLO and $r_{a,HEO}, v_{a,HEO}$ apoapsis radius and tangential velocity on the target HEO. The optimal control problem is then given by equations 5.2 and 5.3 after replacing \mathbf{x}_0, Ψ with 5.8 and 5.9 respectively. Path constraints 5.6 are also applied to avoid the spacecraft crossing the Moon surface and expelling more mass than the initial propellant carried on board.

The initial guess required by the gradient-based solvers used throughout this work is obtained in a similar manner as already done for the ascent and descent trajectories from the Moon surface to a LLO modeled as single phase powered transfers with variable thrust. For relatively high thrust over initial weight ratios (twr), the solution is assumed close to an Hohmann transfer from an initial radius $r_{LLO} = R + H$ to a final one equal to the HEO apoapsis radius $r_{a,HEO}$.

As a consequence, the corresponding initial guess is obtained as follows:

- An initial powered phase at constant radius r_{LLO} and maximum thrust T_{max} to accelerate the spacecraft from the initial circular velocity v_{LLO} to the periapsis velocity of the aforementioned Hohmann transfer $v_{p,Hohmann}$
- A ballistic arc corresponding to the Hohmann transfer from r_{LLO} to $r_{a,HEO}$
- A final impulsive burn to accelerate from the apoapsis velocity on the Hohmann transfer arc $v_{a,Hohmann}$ to the corresponding tangential velocity on the target HEO $v_{a,HEO}$

Starting from the provided guess, the initial solution is then converged to the true optimum presented in chapter 6.

5.4.2 Three Phases Trajectory

To improve the results obtained with the previous model, and in particular relax the fixed injection state at the HEO apoapsis, a more sophisticated model has been developed as described below. The whole transfer is now split into three subsequent phases as follows:

- Powered phase at constant maximum thrust to accelerate the spacecraft from v_{LLO} to the velocity at the beginning of a ballistic arc determined by the solution of the problem itself
- Coasting phase along the predetermined ballistic arc
- Powered phase to accelerate the spacecraft from the velocity at the end of the ballistic arc to the velocity at the injection point on the target HEO. Differently as before, this last point does not necessarily coincide with the HEO apoapsis and its precise location is determined during the iterative solution as for the intermediate ballistic arc.

Appropriate constraints are then imposed at each phase transition to guarantee the continuity of the spacecraft states r, θ, u, v, m across subsequent phases. At the same location the spacecraft engines are supposed to be instantaneously turned off and on thus introducing a discontinuity in the corresponding thrust magnitude from T_{max} to zero and vice-versa.

Initial conditions on the spacecraft states are then imposed as follows:

$$\begin{cases} t_0 = 0 \\ r(t_0) = R + H \\ \theta(t_0) = \theta_0 \\ u(t_0) = 0 \\ v(t_0) = v_{LLO} \\ m(t_0) = m_0 \end{cases} \quad (5.10)$$

Differently as before, the final spacecraft states are left free but additional boundary constraints are imposed to guarantee the former being injected in the target HEO. These constraints relates the three state variables r, u, v to the orbit semi-major axis a and specific angular momentum h which are imposed to be equal to the HEO ones a_{HEO}, h_{HEO} . Mathematically speaking, the aforementioned constraints are expressed as:

$$\begin{cases} \frac{\mu r(t_f)}{2\mu - r(t_f) \cdot [u^2(t_f) + v^2(t_f)]} - a_{HEO} = 0 \\ r(t_f) \cdot v(t_f) - h_{HEO} = 0 \end{cases} \quad (5.11)$$

Equations 5.11 are easily derived from the energy equation and the definition of specific angular momentum as follows:

$$\frac{u^2 + v^2}{2} - \frac{\mu}{r} = -\frac{\mu}{2a} \implies a = \frac{\mu r}{2\mu - r \cdot (u^2 + v^2)} \quad (5.12)$$

$$\mathbf{h} = \mathbf{r} \times \mathbf{v} \implies \|\mathbf{h}\| = h = r \cdot v$$

With r radius, u, v radial and tangential velocities, a semi-major axis, \mathbf{h} specific angular momentum vector, \mathbf{r} position vector and \mathbf{v} velocity vector on each point along the considered orbit.

The optimal control problem is defined once again through equations 5.2 and 5.3 substituting the two boundary conditions \mathbf{x}_0 and Ψ with equations 5.10 and 5.11 respectively. Once the problem is described, the corresponding initial guess is obtained in a similar manner as presented in section 5.4.1 assuming an optimal transfer close to the well-known Hohmann maneuver. However, a key difference is introduced in this case replacing the final impulsive Δv with a finite burn carried out at constant radius as already done for the first powered phase. To summarize, the first approximation is thus obtained as follows:

- Powered phase at constant radius r_{LLO} to accelerate the spacecraft from v_{LLO} to $v_{p,Hohmann}$, where the last represents the periapsis velocity of an Hohmann transfer between r_{LLO} and $r_{a,HEO}$
- Ballistic arc from the vicinity of the Moon to $r_{a,HEO}$
- Powered phase at constant radius r_{HEO} to accelerate the spacecraft from $v_{a,Hohmann}$ to $v_{a,HEO}$, the last being the tangential velocity at the apoapsis of the target HEO

Once the problem is defined and the initial guess has been computed, the true optimum is found solving the corresponding NLP as described in chapter 6.

5.4.3 Finite Escape Burn

After the implementation and validation of the two models presented in sections 5.4.1 and 5.4.2, it has been verified for the *twr* of interests that the optimal transfer differs only slightly from the theoretical Hohmann maneuver, and in particular the final insertion burn is negligible with respect to the initial one carried out in close proximity of the Moon. Taking into account the aforementioned observations, a third simplified model has been then developed to drastically reduce the computational effort required for the NLP solution without affecting too much the accuracy of the corresponding solution.

In particular, the ballistic arc and final insertion burn that constitute the second and third phases of the previously described model have been completely removed thus rewriting the optimal control problem for a single powered phase trajectory at constant thrust corresponding to the first of 5.4.2. More precisely, the spacecraft is required to leave the initial LLO and inject into a ballistic transfer arc whose apoapsis radius coincides with $r_{a,HEO}$, thus guaranteeing that the targeted orbit is finally reached by the former. Due to the negligible magnitude of the Δv to be carried out for the final insertion, the final burn is assumed to be impulsive and thus computed analytically once the optimal solution for the initial thrusting arc is obtained. Similarly as before, the spacecraft states along the ballistic arc are also propagated analytically once r_f, u_f, v_f at the end of the powered phase are known from the solution of the corresponding NLP.

As a consequence, initial conditions 5.10 still applies while an appropriate boundary constraint is enforced on the final spacecraft states r_f, u_f, v_f to guarantee the former being injected into a ballistic transfer arc whose apoapsis coincides with the specified $r_{a,HEO}$. This condition can be written in the form $\Psi(r, u, v, r_a) = 0$ as follows.

Firstly, semi-major axis and eccentricity along the ballistic arc are expressed in function of r, u, v as given by equations 5.13 below:

$$\begin{cases} a = \frac{\mu r}{2\mu - r(u^2 + v^2)} \\ e = \frac{1}{\mu} \cdot \sqrt{(rv^2 - \mu)^2 + (ruv)^2} \end{cases} \quad (5.13)$$

Secondly, the definition of apoapsis radius is considered:

$$r_a = a \cdot (1 + e) \quad (5.14)$$

After substituting 5.13 into 5.14 the following relationship between r, u, v, r_a is obtained:

$$r_a = \frac{r[\mu + \sqrt{(rv^2 - \mu)^2 + (ruv)^2}]}{2\mu - r(u^2 + v^2)} \quad (5.15)$$

Which is easily rewritten in the form $\Psi(r, u, v, r_a) = 0$ as:

$$\Psi(r, u, v, r_a) = r_a - \frac{r[\mu + \sqrt{(rv^2 - \mu)^2 + (ruv)^2}]}{2\mu - r(u^2 + v^2)} = 0 \quad (5.16)$$

After some manipulation equation 5.16 is finally reduced to 5.17 and implemented in the developed software:

$$[2\mu - r(u^2 + v^2)] \cdot \left\{ [2\mu - r(u^2 + v^2)]r_a^2 + r^3v^2 - 2\mu rr_a \right\} = 0 \quad (5.17)$$

The optimal control problem is thus defined taking into account 5.2 and 5.3 together with boundary conditions 5.10 and 5.17 while the required initial guess is computed as before assuming the spacecraft accelerates from v_{LLO} to $\|\mathbf{v}_f\| = (v_f^2 + u_f^2)^{1/2}$ during a powered phase at constant radius equal to r_{LLO} . This first approximate solution is then converged to the optimal transfer through the iterative solution of the corresponding NLP.

Once the optimal solution is known, the keplerian parameters along the ballistic arc a_{tr}, e_{tr} are computed through equations 5.13 while the corresponding apoapsis velocity $v_{a,tr}$ is obtained as:

$$v_{a,tr} = \sqrt{\frac{\mu(1 - e_{tr})}{a_{tr}(1 + e_{tr})}} \quad (5.18)$$

The additional propellant mass $m_{prop,a}$ required for the final insertion burn is finally computed from the Tsiolkovsky rocket equation as described below:

$$m_{prop,a} = m_{tr} \left(1 - e^{-\frac{\Delta v_a}{I_{sp} \cdot g_0}} \right) \quad (5.19)$$

With m_{tr} spacecraft wet mass along the ballistic arc and $\Delta v_a = v_{a,HEO} - v_{a,tr}$ impulsive Δv required for the final insertion.

Continuation Methods

With an extensive test campaign it has been verified that the optimal control problem formulated above can be easily solved employing a direct transcription with few nodes only for relatively high thrust over initial weight ratios typically greater or very close to one. At the same time, for lower twr values a very high number of segments is required for the NLP solver to converge, thus vanishing the interests of the simplified model mainly linked to the fast convergence behavior. As a consequence, a continuation method has been implemented to obtain the optimal solution for low twr solving first the NLP for higher twr and then employing the previously obtained solution rather than the aforementioned procedure as new initial guess. The effectiveness of this strategy has already been proved by many authors while dealing with low-thrust transfer trajectories and its advantages have been confirmed by the results obtained in this work.

Assuming initial LLO, final HEO, boundary conditions and engines specific impulse being fixed, the implemented algorithm is summarized as follows:

1. Choose a set of thrust/weight ratios twr_k , $k = 0, 1, \dots, n$ such that $twr_{k+1} < twr_k \forall k = 0, 1, \dots, n - 1$
2. Compute the solution for the highest value twr_0 starting from the initial guess described above
3. Continue the family of solutions reducing progressively the thrust/weight ratio and computing the solution for twr_{k+1} starting from the results obtained for $twr_k \forall k = 0, 1, \dots, n - 1$

It has to be noticed here that no other hypothesis rather than a strictly monotonically decreasing distribution of twr have been introduced in the presented algorithm and thus subsequent twr values might not result in a uniform grid with $twr_k - twr_{k+1} \equiv constant \forall k = 0, 1, \dots, n - 1$. In fact, two different distributions have been considered for the numerical application:

- Linear distribution: $twr_{k+1} = twr_k - \Delta twr \forall k$
- Logarithmic distribution: $\ln twr_{k+1} = \ln twr_k - \Delta \ln twr \forall k$

This algorithm is then employed to generate the surrogate model presented in section 6.8.

5.5 Surrogate Model

Given the multidisciplinary context in which the conducted work fits into, it is interesting to analyze the dependency of the computed optimal solutions on the thrust over weight ratio and specific impulse that characterize the spacecraft engines. If their values are treated as fixed parameters during the trajectory optimization, the same quantities are in fact treated as free design variables once the former discipline is included in the overall MDO for the complete launcher. As a consequence, the optimal control problems stated above have to be solved once for each iteration of the MDO (also known as MDA) thus introducing non-negligible computational bottlenecks that considerably increase the CPU time required to converge the overall model. To overcome those issues, surrogate models can be computed and stored before hands and substituted to the actual discipline when needed to speed up the MDO.

For the specific case of interest, a surrogate model (also known as meta model in OpenMDAO) consist in a high number of solutions of the same optimal control problem computed for different values of thrust/weight ratios and specific impulses coupled with an interpolation method capable to extract an approximate solution for an arbitrary tuple (twr, Isp) from the only knowledge of the previously stored set of data. It is then readily seen that even if less accurate, this method allows to dramatically speed up the convergence of the overall MDO while guaranteeing an accuracy mainly linked to the grid refinement and the adopted interpolation scheme.

Different packages available in the Python community already provide those capabilities and two of them have been included in the developed software: the Surrogate Modeling Toolbox [22] developed in collaboration between University of Michigan, NASA, ONERA and ISAE-SUPAERO and the *MetaModelStructuredComp* provided by OpenMDAO. Both of them are described in more details in sections 5.5.1 and 5.5.2 respectively.

5.5.1 Surrogate Modeling Toolbox

The Surrogate Modeling Toolbox (SMT) is a Python package providing both sampling and surrogate modeling methods. As suggested by their name, sampling schemes are used to extract a finite set of samples in the input space (twr, Isp) in which the solution of the corresponding NLP is sought while surrogate modeling methods allow to interpolate the obtained results in the neighborhood of the previously selected points.

Among the implemented schemes, Latin Hypercube Sampling (LHS) and Full-factorial sampling have been employed in this work. The first algorithm optimizes the distribution across the input space of a given set of points guaranteeing that $twr_i \neq twr_j$ and $Isp_i \neq Isp_j \forall i, j = 0, 1, \dots, n - 1$ where n is the total number of samples. In other words, each straight line parallel to the twr or Isp axes will cross at most one point included in the aforementioned set. On the other side, the second algorithm provides a regular grid in which the same coordinates along one axis are repeated for each value along the perpendicular one (the same twr values appears for each Isp value and vice-versa).

Regarding the surrogate modeling methods, SMT implements a complete suite of common methods available in the literature for which a comprehensive description is available in [22] as well as in the provided software documentation [23].

5.5.2 OpenMDAO Metamodels

Among many features related to Multidisciplinary Design and Optimization, OpenMDAO implements two different classes with similar capabilities as the ones of SMT: *MetaModelUnStructuredComp* and *MetaModelStructuredComp*. The main advantage of these components is the straightforward integration within the OpenMDAO optimization framework since they can be easily added as additional MDO disciplines for the problem of interest. On the other side, they provide only surrogate modeling methods thus requiring the sampling grid and corresponding training data as inputs.

MetaModelUnStructuredComp accepts any kind of grid (for example LHS) and implements Kriging interpolation, N-Dimensional interpolation and second order response surface equations as available surrogate modeling methods. On the other hand, *MetaModelStructuredComp* ensures a smooth interpolation of data provided on a regular, structured grid such as the one obtained with Full-factorial sampling. It implements polynomial splines of various order from linear (least accurate) to quintic (most accurate) as summarized in table 5.1.

5.6 Implemented Python Package

To ease the maintenance and reusability of the implemented software, the last has been developed as a Python package using Object-Oriented Programming (OOP) techniques compliant with the recommended Python standards. These guidelines have been applied to the code structure (organization of the software package into sub-packages and modules), code style (*CamelCase* convention for class names, inline comments) and documentation (docstring standards) as described in the following sections. The developed package has been named *LaTOM*, for *Launcher Trajectory Optimization Module*.

Method	Order	Description
slinear	1	Basic linear interpolation
lagrange2	2	Second order Lagrange polynomial
lagrange3	3	Third order Lagrange polynomial
akima	3	Interpolation using Akima splines
cubic	3	Cubic spline, with continuity of derivatives between segments
scipy_slinear	1	Scipy linear interpolation. Same as slinear, though slower
scipy_cubic	3	Scipy cubic interpolation. More accurate than cubic, but slower
scipy_quintic	5	Scipy quintic interpolation. Most accurate, but slowest

Table 5.1: Interpolation methods available in *MetaModelStructuredComp*

5.6.1 Code Architecture

As recommended by the Python guidelines, *LaTOM* package is organized in several sub-packages, each of them constituted by different modules. The top-level package *latom* contains nine sub-packages corresponding to the main components of the developed code:

1. *analyzer*: provides the user interfaces to define an optimal control problem, setup and solve the corresponding NLP and visualize the obtained results
2. *data*: data storage with precomputed solutions for surrogate models and computationally expensive NLPs
3. *guess*: routines to generate an appropriate initial guess for each defined NLP
4. *nlp*: classes and methods to transcribe a continuous-time optimal control problem into a finite-dimensional NLP and interface the corresponding gradient-based solvers
5. *odes*: defines the Ordinary Differential Equations and analytical derivatives for the implemented dynamical models. All implemented classes directly inherit from *ExplicitComponent* and *Group* defined within OpenMDAO
6. *plots*: routines to display the NLP solutions based on the well-known *Matplotlib* library
7. *reader*: uses the OpenMDAO *CaseReader* class and methods to load and visualize a stored solution
8. *surrogate*: computes surrogate models for the different cases of interest using both SMT and OpenMDAO *MetaModelStructuredComp*
9. *utils*: a series of utility classes and methods to define physical constants, spacecraft characteristics and keplerian orbits properties

The UML diagrams of *analyzer*, *guess*, *nlp* and *odes* sub-packages are provided in appendix B while a comprehensive description of all implemented classes and methods is provided in the software documentation introduced in section 5.6.2.

5.6.2 Documentation

The source code has been extensively commented following the NumPy docstring standard used in many scientific Python packages such as NumPy, SciPy and Matplotlib. The same convention is also adopted by OpenMDAO and dymos thus becoming the obvious choice for the developed package since the last is highly reliant on the aforementioned libraries. Once the docstrings have been properly included in the source code, the corresponding package documentation in HTML format has been built using Sphinx [24], a Python Documentation Generator tool, and made available on the GitHub Pages of the project repository [19]. Installation instructions for both *LaTOM* and its dependencies are also included in the same web page.

6 Results and Analysis

In this chapter an extensive dissertation is carried out to present and analyze the different results obtained throughout this work. Optimal solutions corresponding to the implemented models are discussed in detail to highlight the main achievements of the project as well as propose new guidelines for further developments as described in chapter 7.

All numerical simulations have been carried out using data in table 6.1. The former defines both physical constants (g_0, R, μ, g) and scaling parameters (l_c, t_c, a_c, v_c) used to obtain all subsequent results.

Parameter	Units	Value
g_0	m/s^2	9.80665
μ	m^3/s^2	4902800066163.796
R, l_c	m	1737400.0
g, a_c	m/s^2	1.6242188593883116
t_c	s	1034.2550119009243
v_c	m/s	1679.8564957463636

Table 6.1: Constant parameters and characteristic quantities

Regarding the first set of data, g_0 is the standard sea-level gravity, R and μ the equatorial radius and standard gravitational parameter for the Moon and g the acceleration of gravity on the Moon surface. On the other side, the scaling parameters are selected such that $R = 1.0$ and $\mu = 1.0$ in non-dimensional units. As a consequence, the following characteristic quantities are derived:

- l_c : characteristic length equal to the equatorial radius of the Moon R
- t_c : characteristic time equal to $T/2\pi$ with T orbital period on a keplerian path about the Moon with semi-major axis equal to R
- a_c : characteristic acceleration equal to the gravitational acceleration on the Moon surface g
- v_c : characteristic velocity equal to the tangential velocity in a circular orbit about the Moon with semi-major axis equal to R

The aforementioned quantities are used within each defined NLP to properly scale the corresponding input parameters (characteristics of the initial and target orbits, spacecraft properties, boundary conditions, path constraints) thus improving the convergence behavior of the selected solver and increase the accuracy of the obtained solution while reducing the required computational time.

6.1 Low Lunar Orbit to Moon Surface

A first analysis is conducted for a two-dimensional descent trajectory from a circular Low Lunar Orbit at 100 km altitude to the Moon surface. Solutions obtained with the newly developed model

presented in section 5.3 are compared with the ones corresponding to a simpler model implemented during the second semester to highlight the benefits of a more flexible control scheme in the resulting optimal transfer. All numerical simulations have been carried out with the spacecraft characteristics summarized in table 6.2 and boundary conditions reported in table 6.3.

Parameter	Units	Value
m_0	kg	1.0
Isp	s	400.0
twr	—	0.9

Table 6.2: Spacecraft characteristics for LLO to Moon surface transfers

Initial Conditions			Final Conditions		
r_0	km	1837.4	r_f	m	1737.4
u_0	m/s	0.0	u_f	m/s	0.0
v_0	m/s	1633.5041	v_f	m/s	0.0

Table 6.3: Initial and final boundary conditions for LLO to Moon surface transfers

6.1.1 Single-Phase Descent with Variable Thrust

A first simulation is carried out for a two-dimensional decent trajectory with variable thrust using the model detailed in section 5.3 without imposing the additional constraint expressed by equation 5.7. A Gauss-Lobatto transcription with 200 segments and third-order interpolating polynomials is chosen to discretize the continuous states and controls variables while the corresponding NLP is solved using SNOPT. The corresponding optimal solution requires 4516.7708 s time of flight and a propellant fraction equal to 0.4197 to land the spacecraft from the initial LLO. The resulting states and controls profiles are depicted in figures 6.1a and 6.1b respectively.

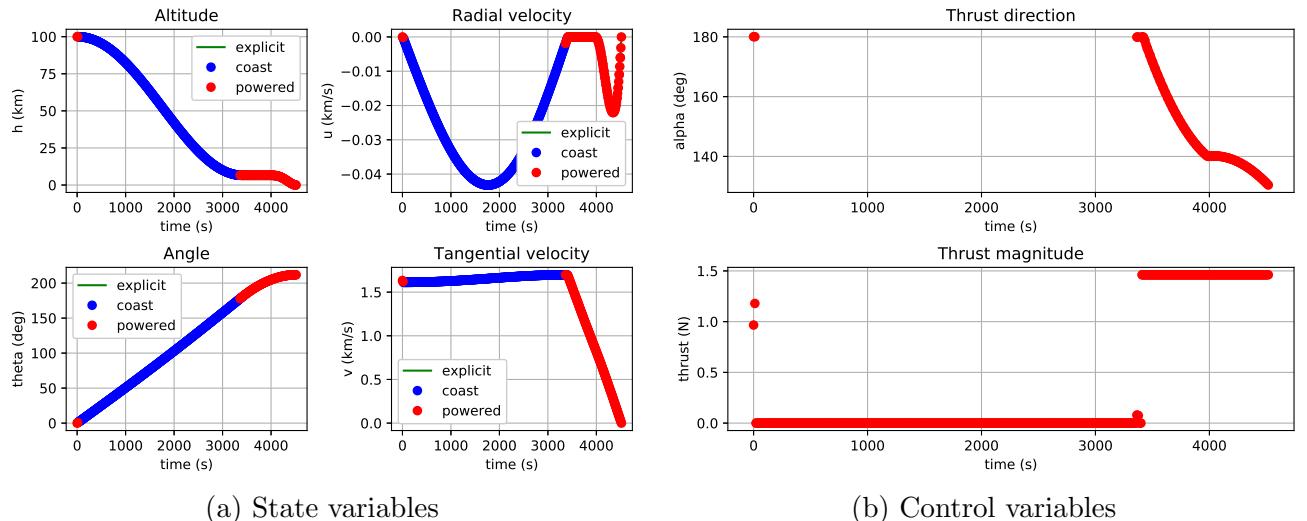


Figure 6.1: Time series for the single-phase descent trajectory with variable thrust

As expected from the theoretical background, the optimal control profile results in a *bang-bang* control scheme in which the thrust magnitude is either zero or maximum. More precisely, the spacecraft performs a small burn at the beginning of the transfer to lower its energy thus entering in a ballistic descent phase followed by a final burn required for soft landing on the Moon surface. If compared with results summarized in table 3.11, it is also possible to conclude that a more flexible control scheme limited only by the imposed boundary conditions allows to save an additional 1.2% of propellant fraction with respect to the former solution.

6.1.2 Single-Phase Descent with Variable Thrust and Vertical Landing

As easily seen in figure 6.3a, the solution presented in the previous section requires the spacecraft to fly at very low altitudes and perform the final touch down approaching the designed landing site with a very narrow angle with respect to the Moon surface. This behavior poses several safety concerns due to the possible presence of geographical features and obstacles around the landing site thus requiring additional constraints in problem formulation to guarantee a safe and vertical touchdown. For that purpose, the path constraint expressed by equation 5.7 is included in the problem formulation and the previously obtained solution recomputed taking into account the newly introduced limitations. In the specific case of interest a minimum safe altitude $h_{min} = 5 \text{ km}$ and constraint slope $s = -5.0$ are selected while the same transcription and NLP solver as 6.1.1 are employed to discretize and solve the optimal control problem. The obtained solution requires 4426.9527 s time of flight and a propellant fraction of 0.4267, slightly higher than the previous one but lower than the two-phases descent described in 3.3.1. The corresponding states and controls time series are depicted in figures 6.2a and 6.2b while figures 6.3b and 6.3a show the altitude profiles for the single-phase descent trajectories with and without constraint 5.7.

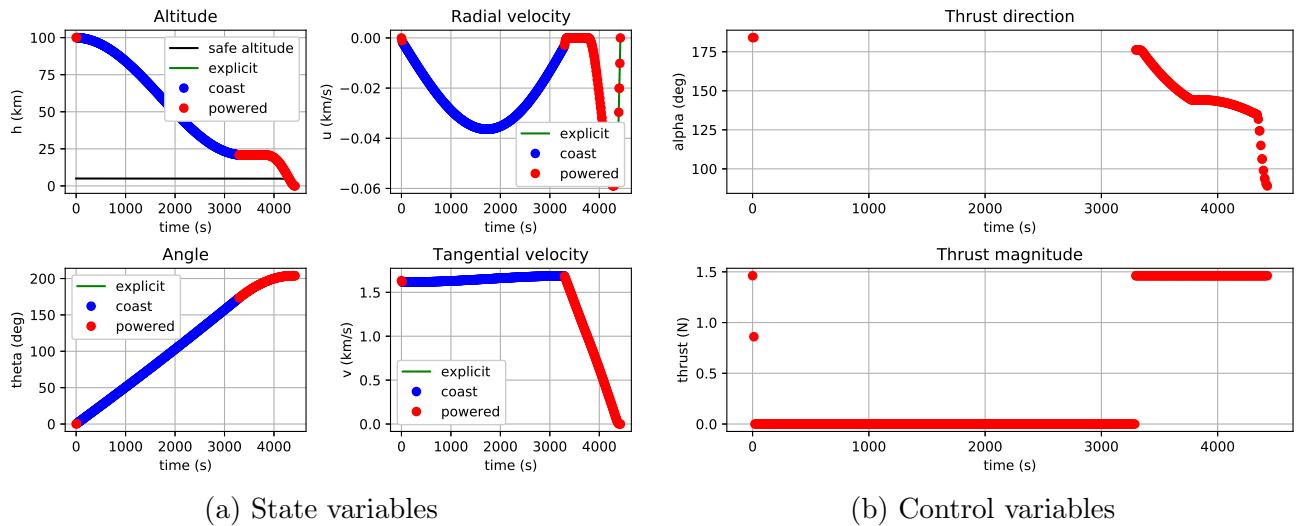


Figure 6.2: Time series for the single-phase descent trajectory with variable thrust and vertical landing

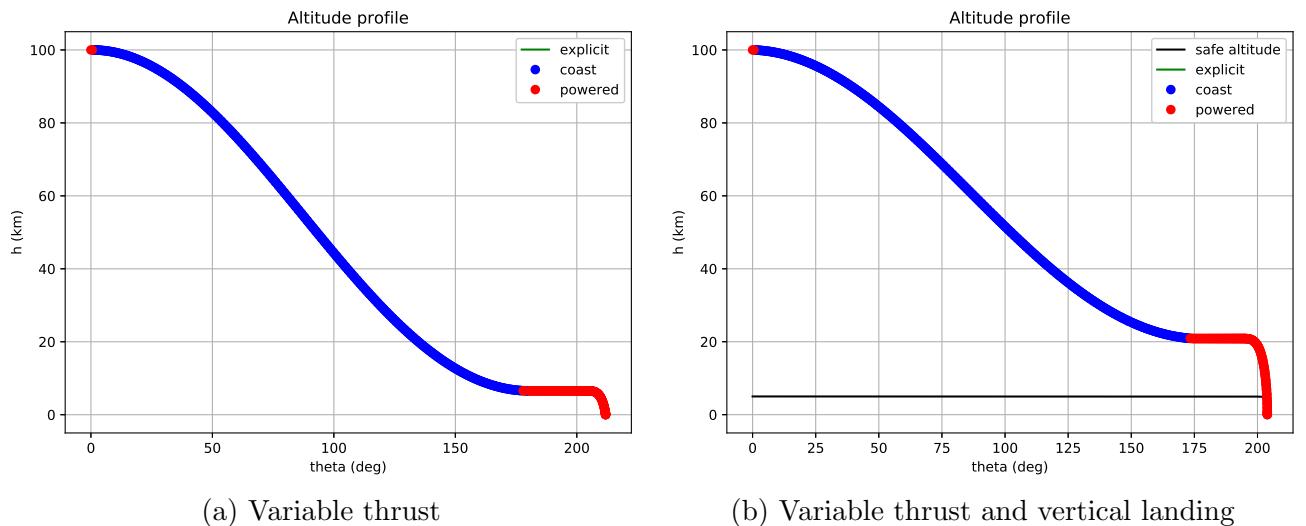


Figure 6.3: Altitude profiles for the single-phase descent trajectories with variable thrust and optional vertical landing

As seen from figure 6.3b, in the last case the spacecraft hovers at about 20 km altitude while approaching the designed landing site before performing a strong maneuver suddenly changing its attitude to orient the thrust plume toward the Moon surface and complete the final descent. To conclude, this last solution brings together the main advantages of the two former models guaranteeing a safe vertical touchdown while optimizing the propellant consumption thanks to a more flexible control scheme as the one introduced in 5.3.

6.2 Low Lunar Orbit to Highly Elliptical Orbit

In this section a Low Lunar Orbit (LLO) to Highly Elliptical Orbit (HEO) transfer is analyzed employing the three different models introduced in section 5.4 to highlight the main advantages and drawbacks of the proposed approaches. The numerical simulations have been carried out with the spacecraft characteristics summarized in table 6.4 while the orbital parameters for the departure LLO and target HEO are given in table 6.5. The last is characterized by a 6.56 days period almost equal to the one of a 9 : 2 lunar synodic resonant L_2 NRHO.

Parameter	Units	Value
m_0	kg	1.0
Isp	s	450.0
twr	—	2.1

Table 6.4: Spacecraft characteristics for LLO to HEO transfers

Parameter	Units	LLO	HEO
a	km	1837.4	34188.694246
e	—	0.0	0.907864

Table 6.5: Initial and final boundary conditions for LLO to HEO transfers

6.2.1 Single-Phase Trajectory

A first solution is computed with the single-phase model described in section 5.4.1 using a Gauss-Lobatto transcription with 400 segments and third-order interpolating polynomials. The corresponding NLP is then solved using SNOPT while the resulting states and controls profiles are depicted in figures 6.4a and 6.4b respectively.

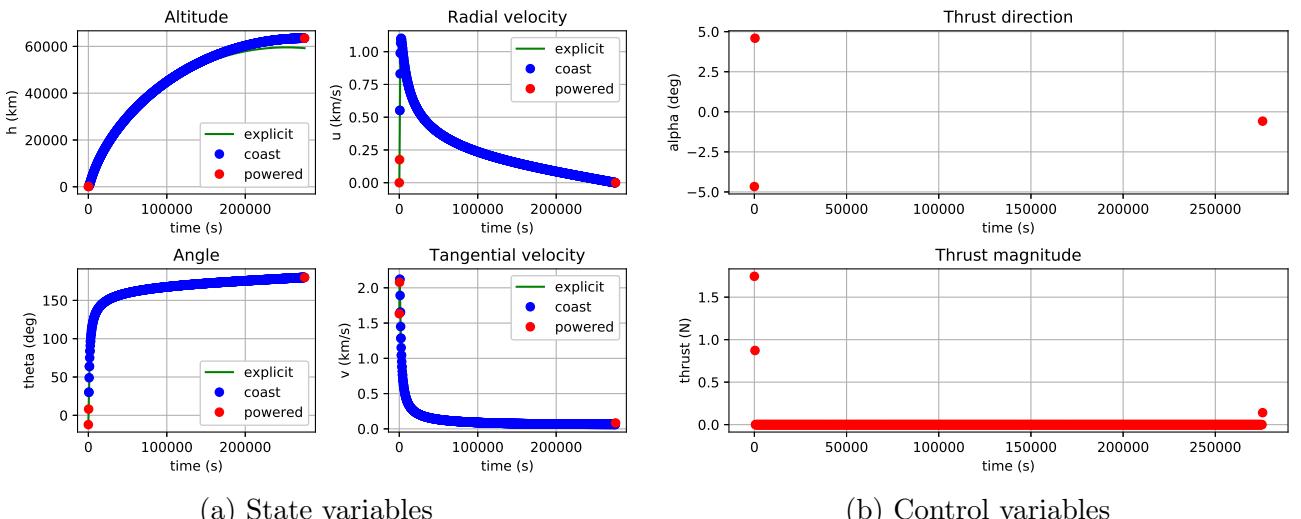


Figure 6.4: Time series for the single-phase LLO to HEO transfer trajectory

The optimal transfer requires 3.1904 *days* time of flight and a propellant fraction of 0.1399 to move the spacecraft from the initial LLO to the target HEO. It has to be recalled here that a fixed insertion point is imposed at the HEO apoapsis. The obtained optimal control profile is once again a *bang-bang* scheme with two finite burns to leave the initial LLO and inject in the target HEO respectively. However, due to the long coasting phase between the two aforementioned burns, the powered phases are covered with just a couple of nodes thus invalidating the reliability of the obtained solution. As a demonstration, an explicit integration of the ODEs starting from the imposed initial conditions and computed optimal control profile leads to a final position error of 4253.1278 *km*, unacceptable even for a first preliminary result. To overcome this issue, a three-phases trajectory as the one described in section 6.2.2 has to be preferred to guarantee an adequate number of nodes being reserved for each modeled burn.

6.2.2 Three-Phases Trajectory

Starting from the optimal control problem formulation detailed in section 5.4.2, a second solution for an LLO to HEO transfer is computed as three-phases trajectory with two finite burns separated by an intermediate coasting arc. A Gauss-Lobatto transcription with third-order polynomials and 200, 1200 and 100 segments is chosen to discretize the states and controls variables while the corresponding NLP is solved with the gradient-based solver IPOPT. An high number of segments is required here to overcome the issues highlighted in section 6.2.1 thus obtaining an highly accurate solution at the expenses of a much higher computational effort and required CPU time. However, the obtained solutions presents a final position error lower than 5 *km* thus justifying the previous choices. The optimal transfer is characterized by a slightly longer time of flight equal to 3.1907 *days* and a somewhat smaller propellant fraction of 0.1397. The corresponding states and controls time series are shown in figures 6.5a and 6.5b while 6.6 depicts the computed optimal transfer in a reference frame centered at the Moon center. Similarly as before, the obtained trajectory is characterized by two short burns separated by a long coasting phase during which the spacecraft engines are turned off and no control is applied.

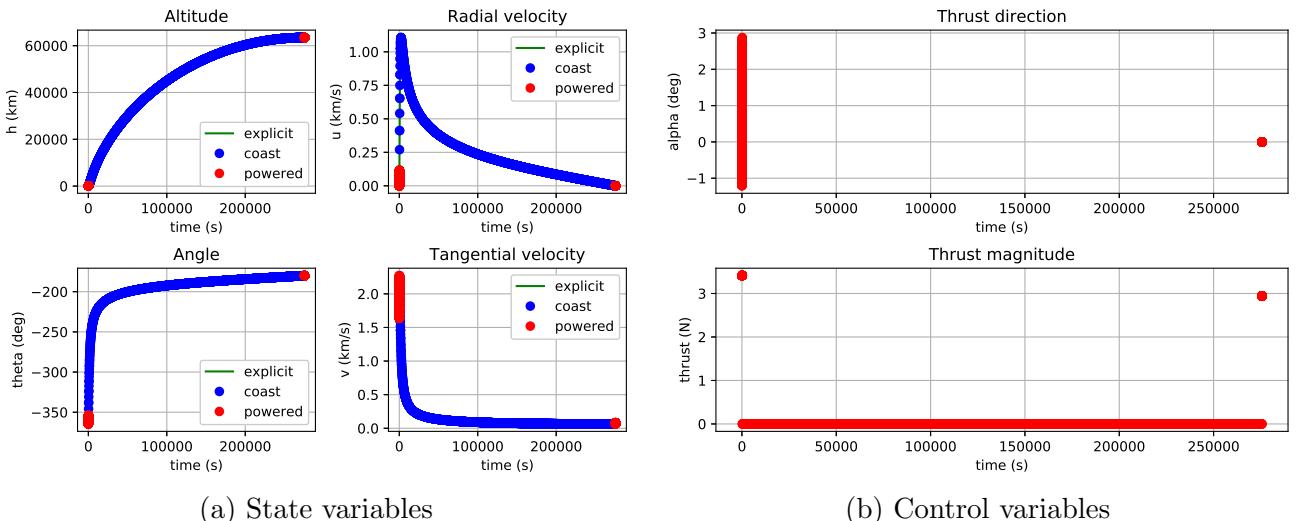


Figure 6.5: Time series for the three-phases LLO to HEO transfer trajectory

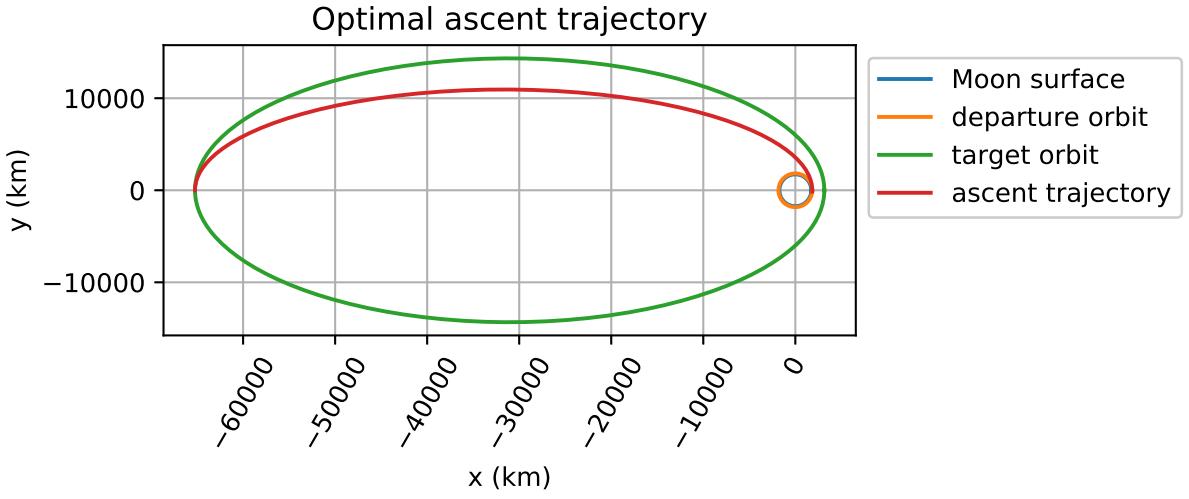


Figure 6.6: Three-phases LLO to HEO transfer trajectory

6.2.3 Finite Escape Burn

Even if extremely accurate, due to the required computational effort the previously described solution is not adapted to be included in the MDO of the complete launcher or to be used as base solution during the extrapolation of a surrogate model. As a consequence, the third model described in section 5.4.3 is tested in this section as promising compromise between high accuracy and low computational effort. The optimal control problem is discretized using a Gauss-Lobatto transcription with third-order polynomials and 100 segments in time. The resulting NLP is then quickly solved with IPOPT and the corresponding optimal transfer depicted in figures 6.7a and 6.7b for the states and controls variables respectively.

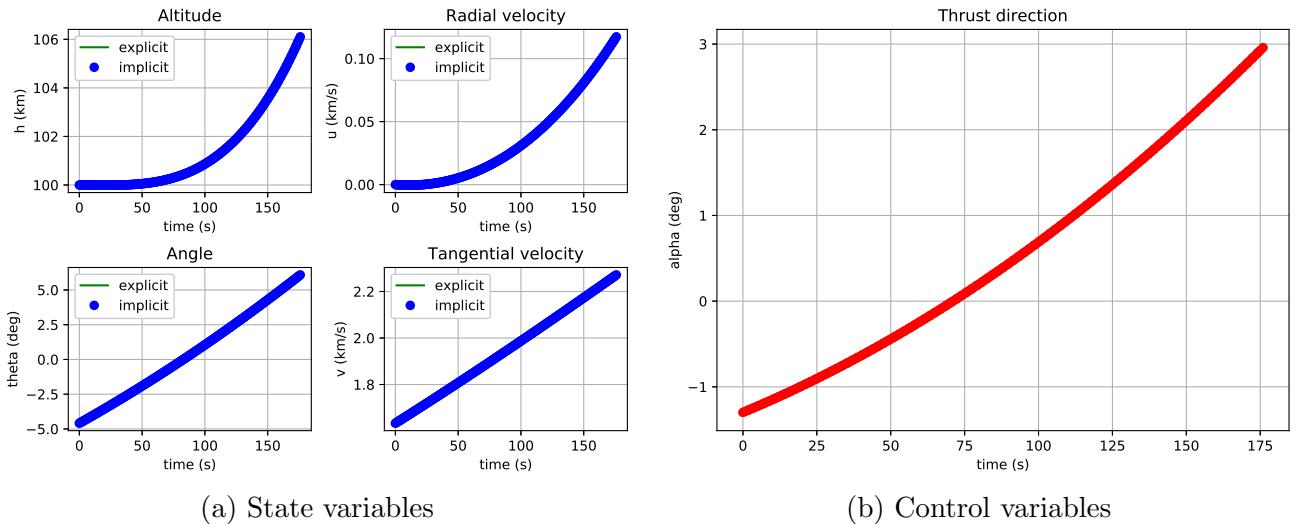


Figure 6.7: Time series for the single-phase finite escape burn

Taking into account the coasting phase and final insertion burn, the overall transfer requires 3.1898 *days* time of flight and a propellant fraction of 0.1397. The last results differs only by $3.6368 \cdot 10^{-7}$ with respect to the one obtained in section 6.2.2 thus validating the proposed approach as an accurate but low computationally expensive model for a two-dimensional LLO to HEO transfer. Tanks to its characteristics it will be employed in section 6.3 to generate the surrogate models for the previously discussed scenario.

6.3 Surrogate Models

In this section different surrogate models based both on SMT and OpenMDAO *MetaModelStructuredComp* are analyzed to determine the most suitable surrogate modeling methods that lead to the highest accuracy for the interpolated solutions. All models have been obtained with a full-factorial sampling scheme made of 50×50 samples or a Latin Hypercube Sampling (LHS) with 50 different values in the intervals $[250.0 \text{ s}, 500.0 \text{ s}]$ and $[1.0, 4.0]$ for the *Isp* and thrust/weight ratio respectively. A circular LLO at 100 km altitude and an initial spacecraft mass of 1.0 kg have also been taken into account for both ascent and descent trajectories to and from the Moon surface.

Firstly, the ascent trajectory at constant thrust is chosen as test case scenario to perform a systematic comparison of all surrogate modeling methods and interpolation schemes implemented in both SMT and OpenMDAO *MetaModelStructuredComp*. A set of random tuples (*Isp*, *twr*) that do not belong to the initial distribution used to obtain the surrogate models is firstly generated and the corresponding optimal control problems solved to retrieve the real propellant consumption for all of them. The aforementioned surrogate models are then employed to estimate the same quantities using different available algorithms as shown in table 6.6. The minimum and maximum absolute errors between predicted and computed solutions is finally sought to determine the most suited algorithms to be used in the subsequent work.

Interpolation method	Minimum error	Maximum error
SMT - Latin Hypercube Sampling		
Kriging	0.0000000417477165	0.0004977662749238
LS	0.0000481695237877	0.0338920743371021
QP	0.0002337918004142	0.0152680154660705
SMT - Full-Factorial Sampling		
IDW	0.0001776634182886	0.0259634110130311
LS	0.0010438074863512	0.0309666353064399
QP	0.0012622520427849	0.0133466878438001
RBF	0.0003813727263005	0.5776342499405575
RMTB	0.0002622093505307	0.0064871293030750
RMTC	0.0000371017377981	0.0064318185390693
OpenMDAO - <i>MetaModelStructuredComp</i>		
slinear	0.0000007209869139	0.0002748574866944
lagrange2	0.0000000269045953	0.0000449304391342
lagrange3	0.0000000049192525	0.0000130518774502
akima	0.0000000165195178	0.0000410551874606
cubic	0.0000000027830413	0.0001277253500638
scipy_slinear	0.0000007209869139	0.0002748574866943
scipy_cubic	0.0000000032624012	0.0000103325990789
scipy_quintic	0.0000000034542169	0.0000014038601861

Table 6.6: Comparison between surrogate modeling methods and interpolation schemes for an ascent trajectory at constant thrust

Among the surrogate modeling methods available in SMT, Kriging exhibits the best performances with a maximum error of about $5 \cdot 10^{-4}$ between the predicted and computed solutions. However, the aforementioned algorithm cannot be employed on a set of data obtained with a full-factorial sampling scheme thus causing the LHS distribution to be preferred within SMT. Indeed, the smallest maximum error obtained with a structured grid is about $1.3 \cdot 10^{-2}$ when a second-order polynomial interpolation is applied.

Regarding the OpenMDAO interpolation scheme, the accuracy of the predicted data is directly correlated with the order of the interpolating polynomial as already mentioned in section 5.5.2. According to table 5.1, the highest accuracy prediction is obtained with a fifth-order interpolating polynomial implemented by the *scipy_quintic* method which guarantees a maximum error as small as $1.4 \cdot 10^{-6}$. However, high quality results are also obtained with third-order interpolants such as *lagrange3* and *scipy_cubic* that exhibit a maximum error in the order of 10^{-5} . Depending on the context, higher order interpolants should be used if high accuracy predictions are needed while lower order polynomials are best suited for fastest results in lower fidelity models.

6.3.1 Moon surface to Low Lunar Orbit

After the first comparison carried out for a simple ascent trajectory at constant thrust, surrogate models and OpenMDAO meta-models are also computed in the following cases:

- Ascent trajectory with variable thrust
- Ascent trajectory with variable thrust and vertical takeoff
- Descent trajectory with constant thrust
- Descent trajectory with variable thrust
- Descent trajectory with variable thrust and vertical landing

The corresponding response surfaces are depicted in figure A.1 together with the aforementioned test case scenario.

6.3.2 Low Lunar Orbit to Highly Elliptical Orbit

A surrogate model is finally computed for the Low Lunar Orbit to Highly Elliptical Orbit transfer described in section 5.4.3. The initial and target orbit are taken from table 6.3 while the *Isp* and *twr* ranges are set equal to $[250.0 \text{ s}, 500.0 \text{ s}]$ and $[0.1, 4.0]$ respectively. 26 equally spaced *Isp* values are selected within the aforementioned interval (constant step $\Delta Isp = 10 \text{ s}$) and the corresponding solutions computed across the entire *twr* range using the continuation methods detailed in 5.4.3. For the last parameter a uniform distribution with 157 samples and $\Delta twr = 0.025$ is chosen thus resulting in 26×157 NLPs to be solved. Once all solutions are computed, the obtained data are interpolated using a fifth-order interpolating polynomial available in OpenMDAO *MetaModelStructuredComp* finally resulting in the response surface depicted in figure 6.8 below.

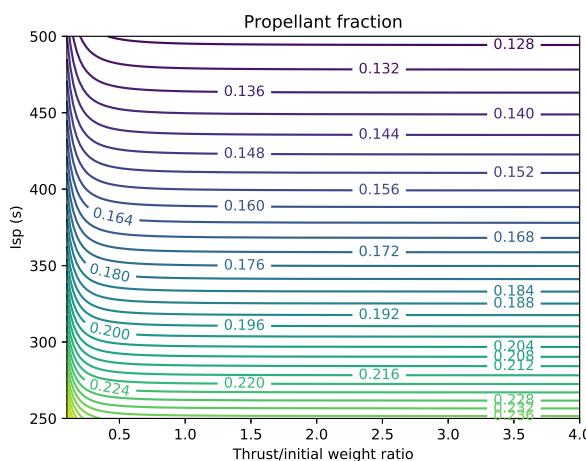


Figure 6.8: Surrogate model for Low Lunar Orbit to Highly Elliptical Orbit transfer

7 Conclusions and Perspectives

The work is aimed to determine the most fuel-efficient transfer trajectories for a reusable lunar lander operating between the future lunar Gateway and the Moon surface. The project fits into the Multidisciplinary Design and Optimization (MDO) of the aforementioned launcher whose main goal is to perform a concurrent design of all spacecraft subsystems and functions such as structure, engines, and mission plan.

Concerning the mission design and trajectory optimization, the complex task is addressed splitting the overall transfer from the vicinity of the orbital platform to the Moon surface and back into a series of smaller and easy tractable problems. To start with, semester 2 have been dedicated to the study of optimal transfer trajectories in the vicinity of the Moon, namely ascent and descent paths from the Moon surface to a circular Low Lunar Orbit (LLO) and back. After a deep literature review aimed to determine the most suitable investigation methods and tools, several test cases have been implemented to reproduce former solutions published by different authors and propose a number of advancements aimed to improve the flexibility and reliability of the implemented tools.

Firstly, two-dimensional ascent trajectories with both constant and variable thrust magnitude are analyzed to assess the impact of the engines throttling capabilities on the transfer performance and duration. If a constant thrust guarantees a minimum-time transfer, introducing throttling capabilities in the developed model allows to determine a *bang-bang* control scheme corresponding to the most fuel efficient transfer between the two specified endpoints. Safety concerns due to the presence of geographical features in the vicinity of the designed launch site have been finally addressed introducing an additional path constraint that guarantees an initial vertical takeoff and a minimum safe altitude across the entire ascent phase.

Moving further, simple models have been implemented to reproduce the results published by Ramanan [2] and Remesh [3] for a descent trajectory with optional vertical touchdown. Several assumptions are retained here such as a fixed control scheme and a constant thrust magnitude thus resulting in a poorly optimized transfer. These models have been then greatly improved during the third semester with the introduction of throttling capabilities and appropriate path constraints to guarantee both a flexible control scheme and a vertical touch-down. A simple three-dimensional model has been finally developed to analyze transfers from the Moon South Pole to several lunar orbits with different inclinations and shapes.

Differently as before, semester 3 has been mostly dedicated to the integration of the developed algorithms within the multidisciplinary optimization framework dedicated to the complete launcher model. As a consequence, the implemented tools have been organized into a Python package easily maintainable and callable from external routines while several surrogate models have been computed and made available to speed-up the analysis. Regarding the software package, it has been organized into several sub-packages and modules as described in section 5.6.1 and an extensive documentation has been generated as explained in section 5.6.2.

On the other side, surrogate models are needed to substitute the trajectory discipline with a simpler analytical model based on stored results and different interpolation schemes thus greatly reducing

the computational effort allocated to the most expensive block included in the overall launcher model. The underlying idea is to precompute and store an high number of solutions for a given case of interest, for example an ascent trajectory from the Moon surface to a 100 km LLO, each of them differing from the others in terms of spacecraft characteristics, namely specific impulse (I_{sp}) and thrust over initial weight ratio (twr). Within the trajectory block the aforementioned parameters are in fact considered as fixed input values while in overall MDO they are treated as additional design variables to be fully optimized. As a consequence, each iteration during the optimization of the complete model (MDA) would require the solution of a new optimal control problem as described in the previous chapters thus introducing a non-negligible computational bottleneck for the outer optimization loop. On the other side, replacing the aforementioned trajectory discipline with a precomputed surrogate model will lead to a much faster prediction of the transfer characteristics thus improving the overall algorithm performance. As described in section 6.3, these models have been made available for all ascent and descent trajectories described above as well as for LLO to Highly Elliptical Orbit (HEO) transfers introduced in the next paragraph.

The last task addressed in semester 3 is the modeling and analysis of LLO to HEO transfers as a first step towards the study of more complex LLO to Near Rectilinear Halo Orbit (NRHO) trajectories. Different two-dimensional models have been implemented and tested as described in sections 5.4 and 6.2 to determine the most fuel-efficient transfer trajectory from a circular LLO to a HEO corresponding to the best planar approximation of a $9 : 2$ lunar synodic resonant L_2 NRHO. The obtained results are characterized by an initial powered phase to boost the spacecraft from the departure LLO to an intermediate transfer orbit followed by a coasting phase lasting for some days and a final insertion burn in close proximity of the HEO apoapsis. If a two-vehicles scenario is considered to perform the round-trip from the lunar Gateway to the Moon surface and back, a tug is likely to be used to transfer the lander from the platform to a circular LLO and wait there to bring the last back to the Gateway. This architecture might employ a low-thrust propulsion system for the tug thus driving the interest towards optimal LLO to HEO transfers with very low twr values characteristic of the aforementioned technology. As a consequence, a continuation method is implemented to extrapolate those solutions starting from higher twr values as described in section 5.4.3. A surrogate model is finally assembled for a twr range between 0.1 and 4.0 as detailed in section 6.3.2.

Starting from the last achievements made in semester 2, further developments will be initially focused on the implementation and validation of a more flexible three-dimensional model to be adapted for the study of NRHO to Moon surface transfers in higher fidelity models such as the Circular Restricted Three-Body Problem (CR3BP). The analysis of the perturbed keplerian dynamics in the vicinity of the Moon could also be envisioned to assess the impact of the non-uniform gravity field of the Moon, solar radiation pressure (SRP) and third-body effects on the spacecraft dynamics while traveling from an LLO to the Moon surface and back.

Taking into account the LLO to HEO transfers analyzed so far, the next logical step would be the replacement of the keplerian two-body dynamics with the CR3BP framework to study actual LLO to NRHO transfers in a higher fidelity model taking into account the gravitational pull of the Earth and the three-dimensional extension of the baseline orbit designed for the future lunar Gateway. Moving further, additional perturbations due to SRP and fourth-body effects might also be introduced as a first transition toward a full-ephemeris model to be employed at a later stage in the context of a more detailed mission design.

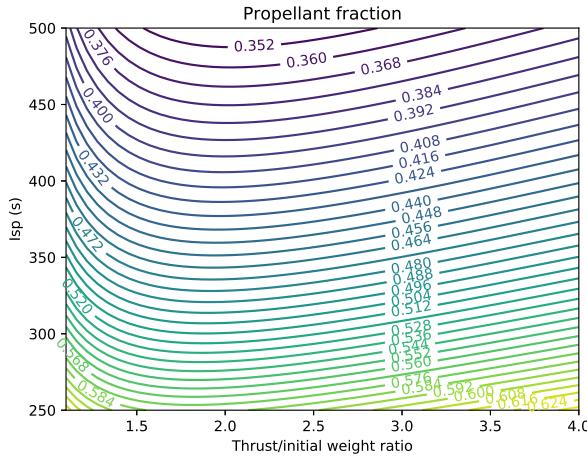
An extensive test campaign should be finally performed to validate the implemented package with both unit testing and comparison with already published results to guarantee the correctness of the implemented algorithms and reliability of the corresponding results. Compatibility with other pieces of software should also be maximized to ease the inclusion of the developed package in the multidisciplinary optimization tool dedicated to the complete launcher model.

Bibliography

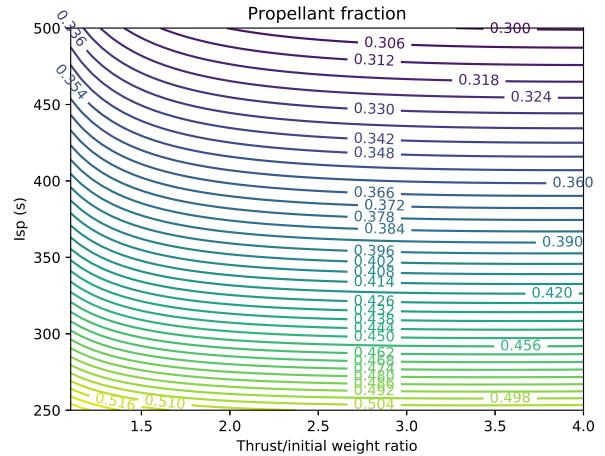
- [1] Alan Wilhite et al. “Lunar Module Descent Mission Design”. In: *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*. Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, Aug. 2008. DOI: 10.2514/6.2008-6939. URL: <https://arc.aiaa.org/doi/10.2514/6.2008-6939>.
- [2] R. V. Ramanan and Madan Lal. “Analysis of optimal strategies for soft landing on the Moon from lunar parking orbits”. en. In: *Journal of Earth System Science* 114.6 (Dec. 2005), pp. 807–813. ISSN: 0973-774X. DOI: 10.1007/BF02715967. URL: <https://doi.org/10.1007/BF02715967>.
- [3] N. Remesh, R. V. Ramanan, and V. R. Lalithambika. “Fuel Optimum Lunar Soft Landing Trajectory Design Using Different Solution Schemes”. en. In: *International Review of Aerospace Engineering (IREASE)* 9.5 (Oct. 2016), pp. 131–143–143. ISSN: 2533-2279. DOI: 10.15866/irease.v9i5.10119. URL: [https://www.praiseworthyprize.org/jsm/index.php?journal=irease&page=article&op=view&path\[\]](https://www.praiseworthyprize.org/jsm/index.php?journal=irease&page=article&op=view&path[])=19460.
- [4] Zhiguo Zhang, Shengping Gong, and Junfeng Li. “The fuel-optimal trajectory for finite-thrust LUNAR ASCENT”. In: *Aerospace Science and Technology* 39 (Dec. 2014), pp. 675–684. ISSN: 1270-9638. DOI: 10.1016/j.ast.2014.06.011. URL: <http://www.sciencedirect.com/science/article/pii/S1270963814001308>.
- [5] Lin Ma et al. “A unified trajectory optimization framework for lunar ascent”. In: *Advances in Engineering Software* 94 (Apr. 2016), pp. 32–45. ISSN: 0965-9978. DOI: 10.1016/j.advengsoft.2016.01.002. URL: <http://www.sciencedirect.com/science/article/pii/S0965997816300102>.
- [6] James M. Longuski, Jose J. Guzmán, and John E. Prussing. *Optimal Control with Aerospace Applications*. Space Technology Library. New York: Springer-Verlag, 2014. ISBN: 978-1-4614-8944-3. URL: <https://www.springer.com/fr/book/9781461489443>.
- [7] Albert L. Herman and Bruce A. Conway. “Direct optimization using collocation based on high-order Gauss-Lobatto quadrature rules”. In: *Journal of Guidance, Control, and Dynamics* 19.3 (May 1996), pp. 592–599. DOI: 10.2514/3.21662. URL: <https://arc.aiaa.org/doi/10.2514/3.21662>.
- [8] Divya Garg et al. “Direct Trajectory Optimization and Costate Estimation of General Optimal Control Problems Using a Radau Pseudospectral Method”. In: *AIAA Guidance, Navigation, and Control Conference*. Chicago, Illinois, USA: American Institute of Aeronautics and Astronautics, Aug. 2009, p. 29. DOI: 10.2514/6.2009-5989. URL: <https://arc.aiaa.org/doi/abs/10.2514/6.2009-5989>.
- [9] Andreas Wächter and Lorenz T. Biegler. “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming”. In: *Mathematical Programming* 106.1 (Mar. 2006), pp. 25–57. ISSN: 0025-5610, 1436-4646. DOI: 10.1007/s10107-004-0559-y. URL: <https://link.springer.com/article/10.1007/s10107-004-0559-y>.
- [10] *HSL, A collection of Fortran codes for large scale scientific computation*. URL: <http://www.hsl.rl.ac.uk/>.
- [11] P. Gill, W. Murray, and M. Saunders. “SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization”. In: *SIAM Review* 47.1 (Jan. 2005), pp. 99–131. ISSN: 0036-1445.

- DOI: 10.1137/S0036144504446096. URL: <https://pubs.siam.org/doi/10.1137/S0036144504446096>.
- [12] Eric S. Hendricks, Robert D. Falck, and Justin S. Gray. “Simultaneous Propulsion System and Trajectory Optimization”. In: *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. American Institute of Aeronautics and Astronautics, 2017. DOI: 10.2514/6.2017-4435. URL: <https://arc.aiaa.org/doi/abs/10.2514/6.2017-4435>.
 - [13] Justin S. Gray et al. “OpenMDAO: an open-source framework for multidisciplinary design, analysis, and optimization”. In: *Structural and Multidisciplinary Optimization* 59.4 (Apr. 2019), pp. 1075–1104. ISSN: 1615-1488. DOI: 10.1007/s00158-019-02211-z. URL: <https://doi.org/10.1007/s00158-019-02211-z>.
 - [14] Ruben E. Perez, Peter W. Jansen, and Joaquim R. R. A. Martins. “pyOpt: a Python-based object-oriented framework for nonlinear constrained optimization”. In: *Structural and Multidisciplinary Optimization* 45.1 (Jan. 2012), pp. 101–118. ISSN: 1615-1488. DOI: 10.1007/s00158-011-0666-3. URL: <https://doi.org/10.1007/s00158-011-0666-3>.
 - [15] Lin Ma et al. “Three-Dimensional Trajectory Optimization for Lunar Ascent Using Gauss Pseudospectral Method”. In: *AIAA Guidance, Navigation, and Control Conference*. AIAA SciTech Forum. American Institute of Aeronautics and Astronautics, Jan. 2016. DOI: 10.2514/6.2016-1372. URL: <https://arc.aiaa.org/doi/10.2514/6.2016-1372>.
 - [16] Lin Ma et al. “Trajectory optimization for lunar rover performing vertical takeoff vertical landing maneuvers in the presence of terrain”. In: *Acta Astronautica* 146 (May 2018), pp. 289–299. ISSN: 0094-5765. DOI: 10.1016/j.actaastro.2018.03.013. URL: <http://www.sciencedirect.com/science/article/pii/S0094576516314114>.
 - [17] David Benson. “A Gauss pseudospectral transcription for optimal control”. Thesis. Massachusetts Institute of Technology, 2005. URL: <https://dspace.mit.edu/handle/1721.1/28919>.
 - [18] Wang Sang Koon et al. *Dynamical Systems, the Three-Body Problem and Space Mission Design*. Apr. 2011. URL: http://www.cds.caltech.edu/~koon/book/KoLoMaRo_DMissionBook_2011-04-25.pdf.
 - [19] *LaTOM 1.0 documentation*. URL: <https://mid2supaero.github.io/LaTOM/>.
 - [20] Abolfazl Shirazi, Josu Ceberio, and Jose A. Lozano. “Spacecraft trajectory optimization: A review of models, objectives, approaches and solutions”. In: *Progress in Aerospace Sciences* 102 (Oct. 2018), pp. 76–98. ISSN: 0376-0421. DOI: 10.1016/j.paerosci.2018.07.007. URL: <http://www.sciencedirect.com/science/article/pii/S0376042118300198>.
 - [21] F. Topputo and C. Zhang. “Survey of Direct Transcription for Low-Thrust Space Trajectory Optimization with Applications”. In: *Abstract and Applied Analysis* (2014). DOI: 10.1155/2014/851720. URL: <https://www.hindawi.com/journals/aaa/2014/851720/>.
 - [22] Mohamed Amine Bouhlel et al. “A Python surrogate modeling framework with derivatives”. In: *Advances in Engineering Software* (July 2019), p. 102662. ISSN: 09659978. DOI: 10.1016/j.advengsoft.2019.03.005. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0965997818309360>.
 - [23] *SMT: Surrogate Modelling Toolbox - SMT 0.4.2*. URL: <https://smt.readthedocs.io/en/latest/>.
 - [24] *Overview — Sphinx 4.0.0+*. URL: <https://www.sphinx-doc.org/en/master/>.

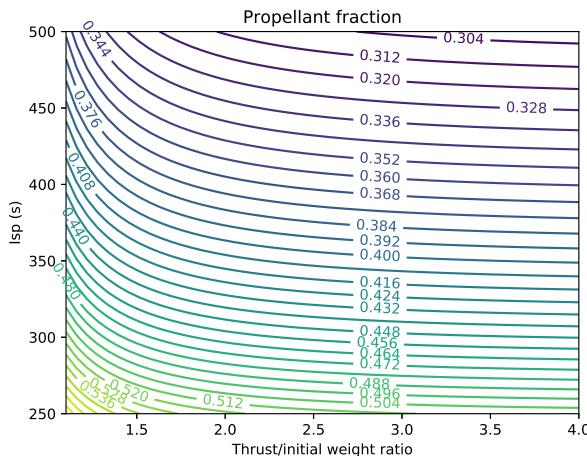
A Surrogate Models



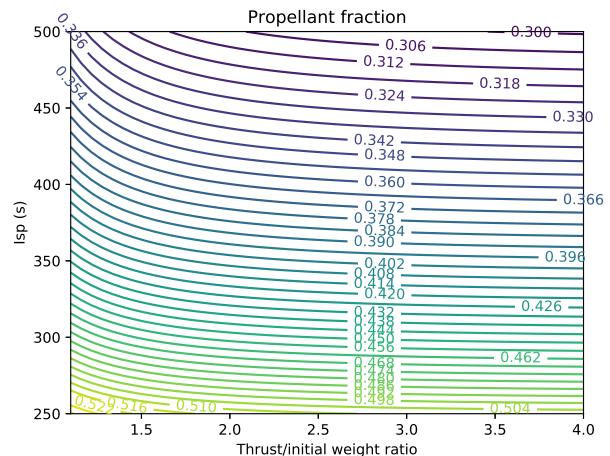
(a) Ascent with constant thrust



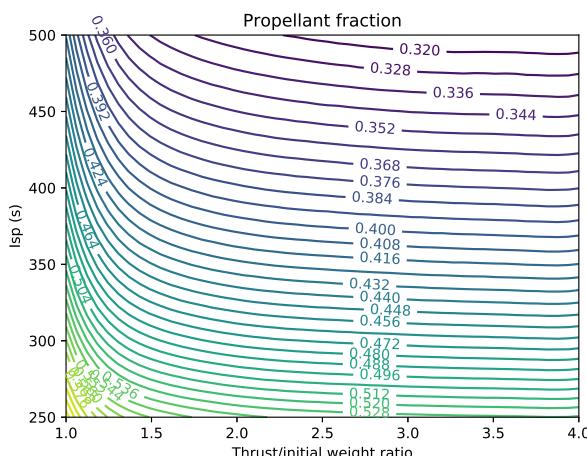
(b) Descent with constant thrust



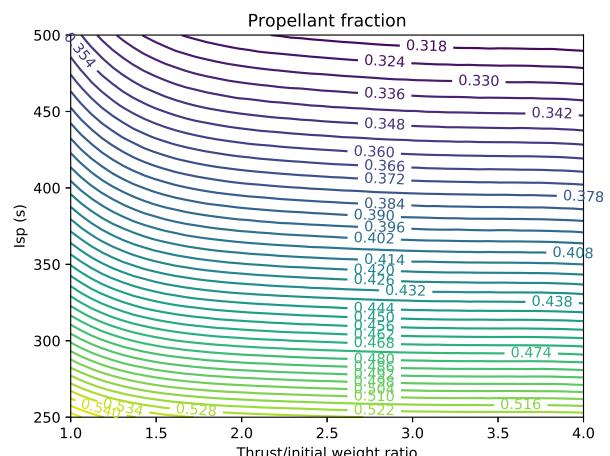
(c) Ascent with variable thrust



(d) Descent with variable thrust



(e) Ascent with vertical takeoff



(f) Descent with vertical landing

Figure A.1: Surrogate models for Moon surface to Low Lunar Orbit transfers

B UML Diagrams

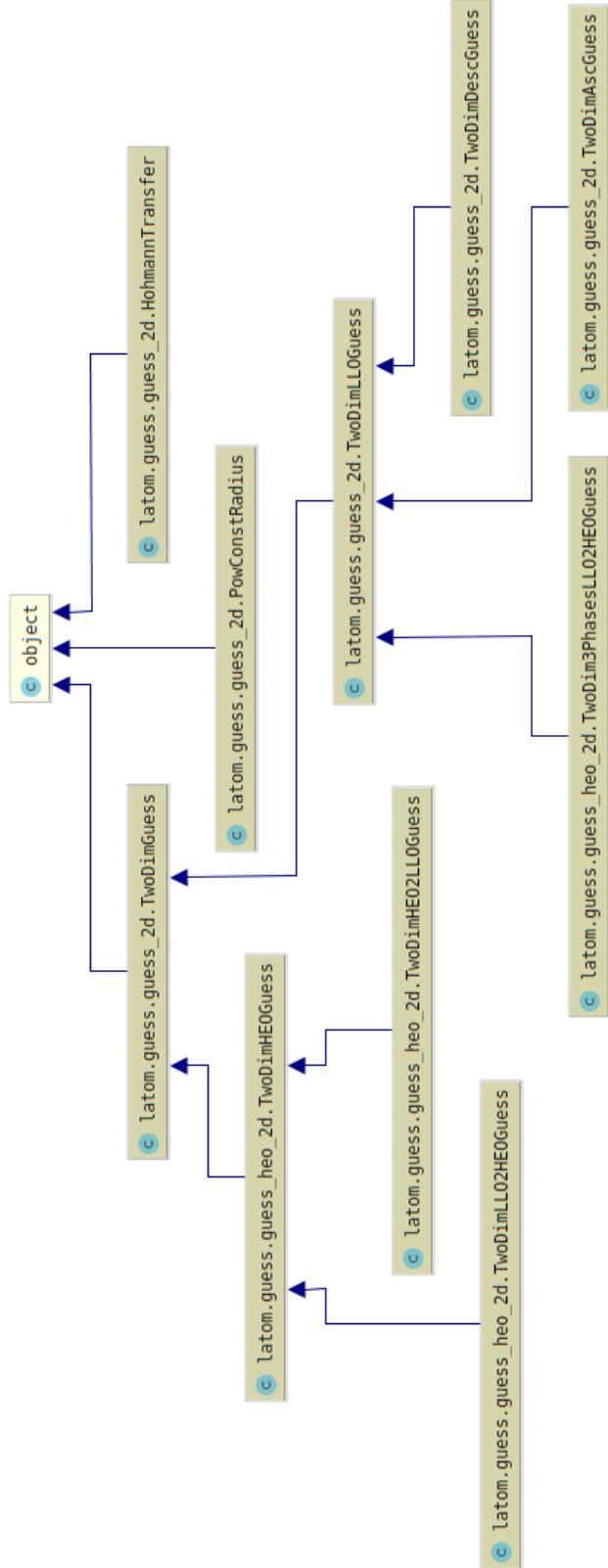


Figure B.1: UML diagram for `Guess` sub-package

Powered by yFiles

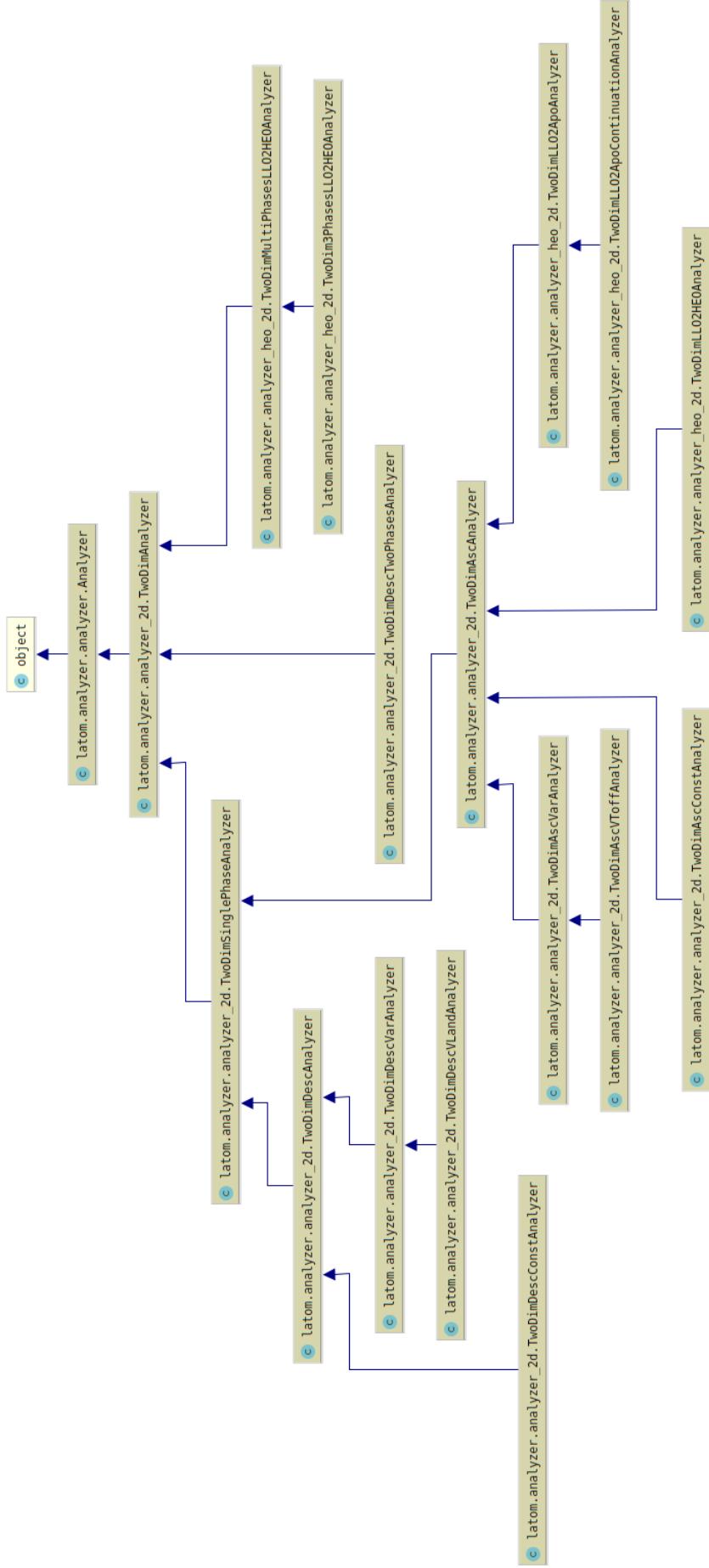


Figure B.2: UML diagram for *Analyzer* sub-package

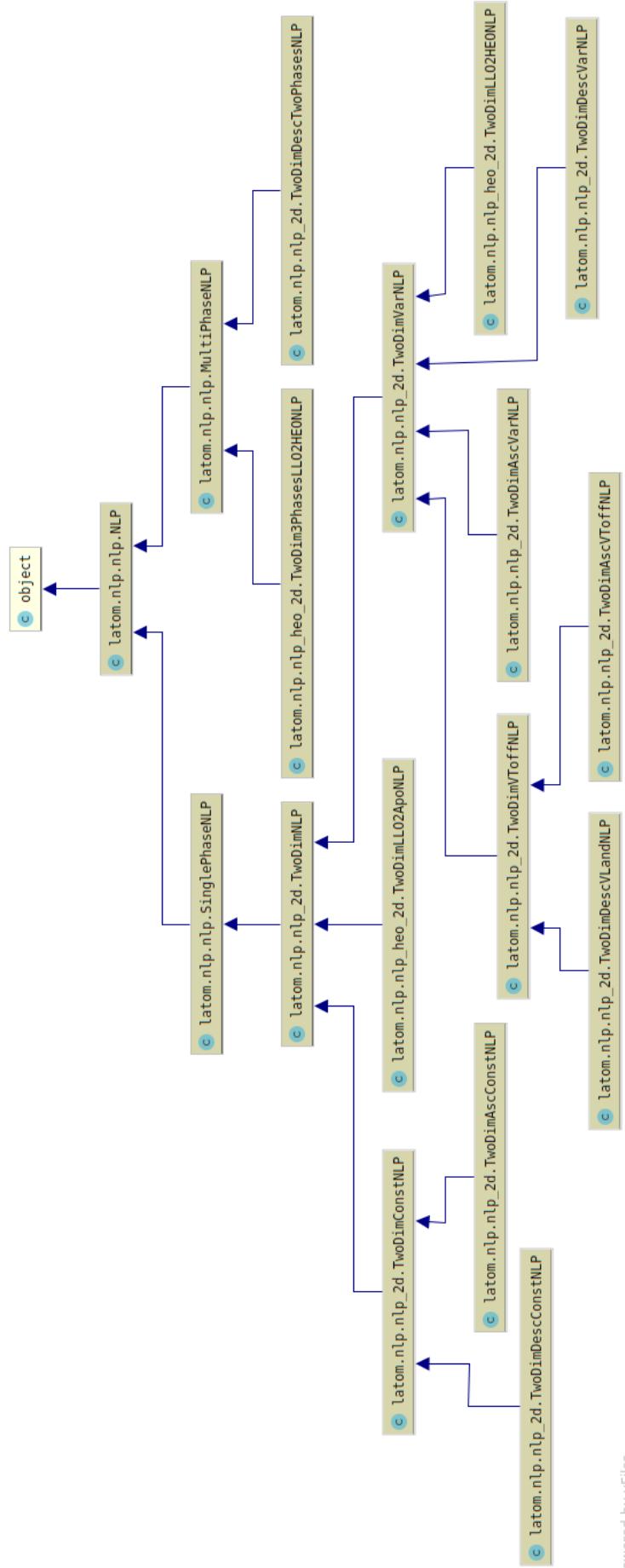


Figure B.3: UML diagram for *NLP* sub-package

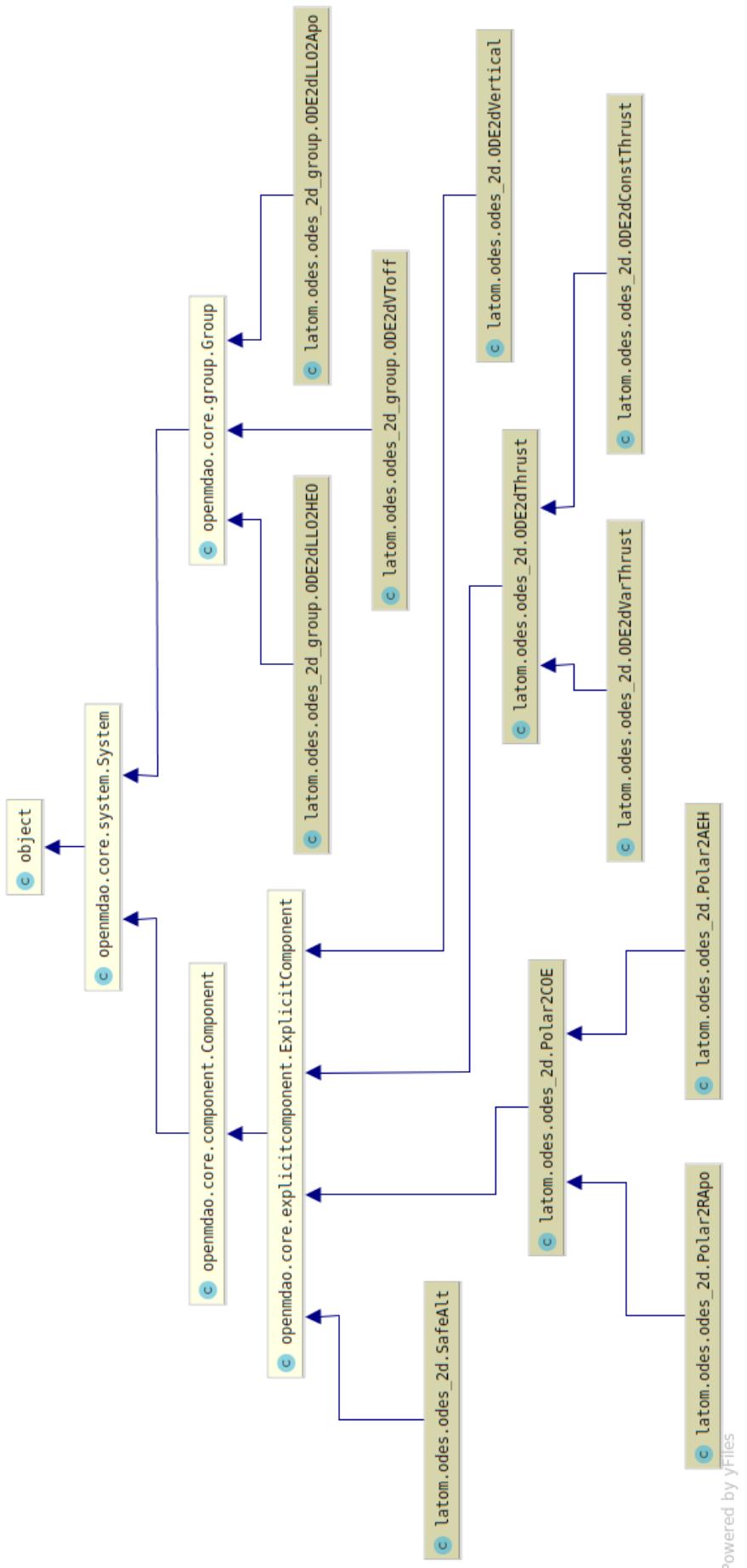


Figure B.4: UML diagram for *ODEs* sub-package