UNIVERSIDADE DA BEIRA INTERIOR
Engenharia

Earth-Mars trajectories with lunar gravity assist study using the self-adaptive Levenberg-Marquardt Algorithm

(Versão corrigida após defesa)

Flávio Daniel Dias Rosa

Dissertação para obtenção do Grau de Mestre em

Engenharia Aeronáutica

(Ciclo de estudos integrado)

Orientador: Prof. Doutor Kouamana Bousson

Covilhã, Fevereiro de 2020

À minha família, à minha namorada e aos meus amigos agradeço profundamente por estarem ao meu lado e me apoiarem incondicionalmente.

Ao Prof. Doutor Bousson agradeço por me dar a possibilidade de trabalhar num tema que me motiva desde sempre.

Ao Rui Oliveira agradeço pela ajuda prestada na realização deste trabalho.

Resumo

Com o aumento do interesse em missões interplanetárias, novas trajetórias e métodos devem ser estudados e analisados de maneira a diminuir os custos e aumentar a capacidade de transportar instrumentação científica.

Neste trabalho, é realizado um estudo numérico de trajetórias interplanetárias entre a Terra e Marte, utilizando a Lua para efetuar uma manobra de assistência gravitacional, com os objetivos de diminuir a energia necessária para a transferência interplanetária e testar e analisar o uso do algoritmo *self-adaptive Levenberg-Marquardt* como corretor diferencial para o desenho de missões espaciais.

Os resultados obtidos são comparados com valores de transferência direta alcançados com os mesmos métodos e com os valores estimados para as próximas oportunidades de transferência interplanetária entre Terra e Marte. São obtidos resultados com o problema de dois corpos de astrodinâmica e verificados e validados com o *software* aberto GMAT desenvolvido pela NASA para uma abordagem mais realista. O algoritmo *self-adaptive Levenberg-Marquardt* desenvolvido para este trabalho na linguagem de programação *Python* 3.6 é testado e utilizado como corretor diferencial para obter as trajetórias para o problema de dois corpos.

Os resultados demonstram que o algoritmo *self-adaptive Levenberg-Marquardt* é adequado para planear missões, que a assistência gravitacional lunar pode ser executada em todas as situações estudadas e que apenas em poucas ocorrências não é viável. Das 4 oportunidades de lançamento analisadas apenas em uma situação a assistência gravitacional lunar não diminuiu a energia de lançamento.

Os resultados indicam que a energia necessária para efetuar futuras missões a Marte ou a outros corpos do sistema solar pode ser reduzida e consequentemente a massa de carga útil nestas missões pode ser aumentada. A possível introdução de um novo método de cálculo para desenhar missões espaciais também é demonstrado através dos resultados obtidos.

Palavras-chave

Transferência interplanetária; assistência gravitacional; *self-adaptive Levenberg-Marquardt;* Marte

Abstract

As the interest in interplanetary missions is rising, new trajectories and methods should be studied and analyzed to decrease the costs and increase the capacity of transporting scientific instruments and payload to Mars.

In this work, a numerical study of interplanetary trajectories between Earth and Mars is performed, using the Moon to carry out a lunar gravity assist manoeuvre, with the objective of decreasing the launch energy for the interplanetary transfer and analyze the use of the self-adaptive Levenberg-Marquardt algorithm as a differential corrector for space mission design.

The obtained results are compared with the values of the direct transfer achieved with the same methods and with the estimated values for the next interplanetary transfer Windows between Earth and Mars. The results are obtained with the astrodynamics two body problem simplistic model and verified and validated with the open source NASA's software GMAT for a more realistic approach. The self-adaptive Levenberg-Marquardt algorithm developed for this work in the programming language Python 3.6 is tested and used as a differential corrector to obtain the trajectories for the two-body problem.

The results demonstrate that the self-adaptive Levenberg-Marquardt algorithm is adequate to design space missions, a lunar gravity assist can be executed in all situations studied and only in a few cases is not viable. Of the four launch windows analyzed only in one situation the lunar gravity assist does not diminish the launch energy.

The results show that the energy needed to perform future Mars missions or missions to other Solar System bodies can be reduced and consequently the payload mass can be increased. The possible introduction of a new calculation method for space mission design is also shown due to the observed results.

Keywords

Interplanetary transfer; gravity assist; self-adaptive Levenberg-Marquardt; Mars

Contents

List of figures

List of tables

List of Acronyms

MAVEN	Mars Atmosphere and Volatile Evolution
NASA	National Aeronautics and Space Administration
ESA	European Space Agency
SLM	Self-adaptive Levenberg-Marquardt algorithm
MER-A	Mars Exploration Rover - A
GMAT	General Mission Analysis Tool
SOI	Sphere of Influence
MJD	Modified Julian Date
LM	Levenberg-Marquardt algorithm
SDO	Solar Dynamics Observatory
SOHO	Solar and Heliospheric Observatory
ACE	Advanced Composition Explorer
TESS	Transiting Exoplanet Survey Satellite
LRO	Lunar Reconnaissance Orbiter
SSM	Single shooting method
LGA	Lunar gravity assist
TCM	Trajectory Correction Manoeuvres

Nomenclature

a	Semi-major axis of orbit	$[km]$
b	Semi-minor axis of orbit	$[km]$
B	Parabolic anomaly	$[rad]$
$B \cdot T$	B-plane parameter	$[km]$
$B \cdot R$	B-plane parameter	$[km]$
c_2, c_3	Universal variables	$[-]$
C_3	Characteristic energy	$[km^2 / s^2]$
d	SLM corrector parameter	$[-]$
e	Eccentricity	$[-]$
E	Eccentric anomaly	$[rad]$
F	General mathematical function	$[-]$
h	Specific angular momentum	$[km^2 / s]$
H	Hyperbolic anomaly	$[rad]$
i	Orbital inclination	$[rad]$
I	Identity matrix	$[-]$
J	Jacobian matrix	$[-]$
m	Mass	$[kg]$
M	Mean anomaly	$[rad]$
p	Semi parameter	$[km]$
q	SLM algorithm function	$[-]$
r	Position	$[km]$
R	Rotation matrix	$[-]$
R	B-plane axis	$[-]$
r_p	Periapsis radius	$[km]$
S	B-plane axis	$[-]$
t	Time	$[s]$
T	B-plane axis	$[-]$
v	Velocity	$[km/s]$
X	*State vector*	$[-]$

Greek letters

α	$1/a$	$[km]$
χ	Universal variable	$[-]$
ψ	Universal variable	$[-]$
δ	SLM parameter	$[-]$

φ	SLM parameter	$[-]$
φ	Spherical coordinate	$[rad]$
μ	Gravitational parameter	$[km^3/s^2]$
υ	True anomaly	$[rad]$
ξ	Specific orbit energy	$[km^2/s^2]$
ω	Argument of periapsis	$[rad]$
θ	Spherical coordinate	$[rad]$
Δ	Variation	$[-]$
Ω	*Right ascension of the ascending node*	$[rad]$

Subscripts

0	Injection
i	Initial
f	Final
k	k^{th} iteration
ref	reference state
opt	Optimal

Chapter 1

1 Introduction

1.1 Motivation

In less than a century, space exploration evolved, from barely existent, with exploration with nothing but telescopes and other instruments, to a billion euro worth business sector with dozens of governmental agencies and private corporations investing in space technology and exploration. In competition or working together, all these entities made unbelievable advancements for humanity and in technology that today is available to anyone on Earth.
It is important to look forward for the space race that is in progress, because we are searching for the limits of the human body, spirit and our place in the Universe. This search can unite nations, promote cooperation between countries and organizations, create a healthy competition for research and technologies. We can contribute to the advancement of humanity, although we have numerous problems as a species, still struggling to find the best way to coexist with each other and to live as a sustainable civilization in our home planet. Despite these inconveniences, space exploration must never stop. Knowledge and scientific advancements are never enough and there is no argument to not support the study of the Solar System, and the Universe around us, which no doubt will help the progress of Humanity.

1.2 Problem overview

Mars is the second planet closest to Earth, in distance and size, and has two known small moons: *Phobos* and Deimos. It is the main target for interplanetary missions: the United States of America, the Soviet Union, ESA and India performed unmanned missions. Although most of the attempted missions failed, some of them were successful. The completed missions consisted of orbiters, landers or just simple flybys. To this day, there is one active rover operating in the surface of Mars, *Curiosity,* one lander, *InSight* and six orbiters scanning the planet: *Mars Odyssey*, *Mars Express*, *Mars Reconnaissance Orbiter*, *Mars Orbiter Mission*, *MAVEN*, and the *Trace Gas Orbiter*. Currently there are planned missions for the next five years from NASA, ESA and China to keep exploring and reveal new data about the celestial body and its system.
However, there is still a lot to know about this planet; we can claim that it is the one that we have more knowledge about, besides Earth. Because of this and other reasons, it is the most probable candidate for a manned mission to another celestial body other than the Moon and probably one of the first to be colonized by humanity in the future.

Generally, to reach Mars a direct transfer is used, which consists of burning while in the parking orbit around the Earth. The parking orbit can be a low Earth orbit and can vary from 400 km to 1500 km of altitude, with any inclination, depending on the next manoeuvre to perform. For a Mars direct transfer, this manoeuvre puts the spacecraft on escape velocity, in order to achieve a solar orbit that intercepts Mars orbit. The duration of the journey usually is no less than 180 days and not more than 450 days. To perform the most efficient trajectory, the time of flight may vary depending on the position, inclination and proximity of both. Some manoeuvres can be performed during the cruise phase to correct anomalies in the trajectory. Finally, the orbit insertion burn is performed near the target body, to ensure that the spacecraft enters in orbit, preventing the vehicle from continuing to outer space. Depending on the type of mission and its objective, the vehicle can perform more burns to achieve the desired orbit, or to achieve the trajectory to enter the atmosphere and initiate the descent to the surface landing site. On some cases, together with the retrograde burn to slow down the spacecraft, aerobraking can be used. This technique consists of using the outer layers of the atmosphere, so that atmospheric drag helps reducing the spacecraft's velocity and thus saving fuel.

As Mars exploration interest keeps growing in worldwide space agencies, the need for cheaper and alternative means of getting to the red planet are important. Some means can be used to increase the payload capacity and allow a vaster scientific payload.

1.3 Space Missions

From the photographs of the surface collected by orbiters and other missions, it is plausible that Mars once had liquid water on its surface, mainly because of geologic formations that reminds us of dried river beds, similar of what we find on Earth, and other formations that point to water flowing as their origin. Because of this and other data, scientists believe that we can soon find evidence that simple life forms may have existed once on the Martian surface. Its surface temperature is relatively similar to Earth's comparing to all bodies in the solar system, and it is relatively close to Earth, with one transfer window approximately every two years, or 780 days. It is known that there are large quantities of solid water on both poles, which can be used to supply a colony, as well as systems of caves that can provide future shelters from cosmic radiation.

Gravity assist manoeuvres are widely used since the first decades of deep space exploration, mainly for space missions to the outer or inner bodies of the Solar system and are an efficient method to reduce fuel mass and increase the payload capacity, although a gravity assist manoeuvre can extend the mission time significantly . This technique consists of using another celestial body in order to increase or decrease the orbit energy relative to the main celestial body. The spacecraft performs a ballistic flyby and departures from the assisting body's sphere of influence with the same velocity magnitude but in a different direction, changing its orbit relative to the main body.

This method was used for the first time in an interplanetary mission in 1974, by the Mariner 10, the spacecraft performed a gravity assist, using Venus as the assisting body to reach Mercury. After this mission, several other missions also took advantage of gravity assist manoeuvres, the most notorious being the Voyagers 1 and 2, launched in the 1970's, and are still operating on their journey to interstellar space. Both missions were launched to take advantage of a rare planet alignment that allowed Voyager 1 to perform a flyby in Jupiter and Saturn, studying and photographing the planets and their moons. Although Voyager 2 was launched before, its trajectory was longer and slower, as it would reach Jupiter later. The spacecraft continued its planned mission and performed also a flyby to Saturn. Due to Voyager 1 flight path, that aimed to get close with Saturn's moon Titan, the spacecraft orbit inclination changed considerably, and prevented any further encounters with other objects. However, Voyager 2 managed to perform a Uranus flyby with operating instrumentation, as well as a Neptune encounter. Both missions prove the potential of gravity assist missions, performing missions that otherwise could be unlikely to happen with the existing technology [1].

More recently, in 2004 ESA's Rosetta mission was launched, with the purpose of studying and attempt to land on the Comet 67P/Churyumov-Gerasimenko, with a high elliptic orbit, which apoapsis is beyond Jupiter's orbit, and periapsis between Mars and Earth's orbit. This kind of orbit was unreachable with a direct trajectory. So, the spacecraft was launched into a heliocentric orbit, that would encounter Earth one year after its departure, using Earth as a gravity assisting body to increase the orbit energy. After the first Earth flyby, the spacecraft headed to Mars, where it also performed a flyby, so that its trajectory would encounter Earth again in 2007, two years after the first flyby. The last gravity assist was in 2009, also using Earth's gravity to increase velocity and change the inclination, to finally reach the comet in 2014. The mission was successful, and it was the first to land on a comet [2].

As this work aims to find new alternative more effective trajectories to Mars for the near future, namely trajectories with Lunar gravity assist, it is important to analyse the next launch windows, in order to find the most efficient launch dates and times of flight, as well as a favourable Moon position to perform the gravity assist effectively. With that purpose, the Pork Chop analysis is needed. The Pork Chop diagrams shows characteristic energy C_3, that is needed to obtain an interplanetary transfer, depending on the launch dates and arrival dates. The curves represent the constant values of C_3, where the centre represents the minimum required for the interplanetary transfer [3].

For the numerical calculations a Differential Corrector was used [4]. A Differential Corrector is a numerical solver for boundary value problems, it works as an equation system solver. It converges the function value to the defined target, or as close as possible to it, varying the chosen parameters to find a solution. The general method used as a differential corrector is the Newton-Raphson method. In this work, a new method is used as Differential Corrector: the self-adaptive Levenberg-Marquardt method.

The Levenberg-Marquardt method has its origins in Newton-Gauss method. This method added the damping parameter μ_k to the previous method. This is a positive multiplier and was introduced in this method with the purpose of reducing the impact of the singularity of the Jacobi Matrix $\boldsymbol{J}_k$. The method and the algorithm will be described in more detail in the work ahead.

In order to test the self-adaptive Levenberg-Marquardt (SLM) as a Differential Corrector, this method was applied to a previous studied Lunar gravity assist trajectory in a similar work [5]. In this work it was proven that for the MER-A mission it was possible to decrease the launch energy due to a Moon flyby. So, this mission was also examined to verify the capability of the SLM as a potential Differential Corrector.

The launch windows analysed span from 2020 to 2026. In these six years, there are four possible launch opportunities, one every two years. Each one of those has two possible types of trajectories: a short one and a long one. These types of trajectories depend on the angle between the departure point and the arrival point relative to the Sun. If the angle is greater than 180º, it is type II trajectory (long), if the angle is less than 180º it is a type I trajectory (short). Although the type II requires more time of flight, it is more effective, so in this work, it was the general type of trajectory chosen. According to a previous study performed for the ExoMars mission [6], a lunar gravity assist can increase the mass of the vehicle injected to Mars between 9% and 10.5%.

1.4 Objectives

The main objective of this dissertation is to test and use the self-adaptive Levenberg-Marquardt algorithm as a potential differential corrector to design the trajectory, by applying the algorithm to try to replicate the results of a previously studied lunar gravity assist transfer to Mars.

The method is then applied to several future launch windows to find a way to reduce the launch energy and consequently reduce the fuel mass or increase the payload capacity of future missions to Mars, by taking advantage of the Moon's gravity and its favourable positions to perform a gravity assist and decrease the initial orbit energy to reach its target.

1.5 Dissertation Structure and Software

The work is divided in 4 chapters:

Chapter 1: Introduction – A small motivation, and short description of the importance of space exploration and Mars missions, objectives of the work, as well as a brief explanation of some concepts and methods used in the work ahead.

Chapter 2: Models and Algorithms – In this chapter the algorithms and numerical methods to describe the astrodynamics analysis are explained.

Chapter 3: Results – The results are presented and discussed, comparing values from the two-body problem and GMAT to the pork chop analysis and the ones obtained for the direct transfers.

Chapter 4: Conclusions – Presentation of the final conclusions of the work, possible errors and recommendations for future work.

The software used to perform this analysis was the open source programming language Python 3.6 and NASA's General Mission Analysis Tool (GMAT) 2018 version.

Chapter 2

2 Models and Algorithms

In this chapter, the astrodynamics model used for this analysis are described. Note that the two-body propagation was mainly used for this analysis, being GMAT used to conclude and obtain the result, since it describes more accurately the motion of all bodies in the Solar System.

For an adequate use of the two-body problem, the trajectory is divided in four sections: from Earth injection burn to the Moon's sphere of influence (SOI), the flyby until the escape from Moon's SOI, the flight until the Earth's SOI, and the journey between Earth's SOI and Mars's SOI.

The sphere of influence of a celestial body is delimited by a sphere of constant radius, and inside this sphere the prevalent force is the body's gravity, disregarding the forces of all other bodies. This method is not the most accurate, although is the less complex for the initial calculations, and gives a good first guess that can be worked on. The vectors are identified with the general symbol $\vec{a}$ and $\hat{a}$ is used for unit vectors. The norm of the vector is represented by the same letter without symbols a.

2.1 Keplerian and Cartesian coordinates

The satellite state representations are essential to define the orbit and the motion of an object in space, there are several types of coordinates systems, but in this work only three of those are used.

The Keplerian coordinate system defines the orbit of the spacecraft through orbital elements: *a*, the semi-major axis, *e*, the eccentricity, *i*, inclination, Ω, the right ascension of the ascending node, ω, the argument of periapsis and υ, the true anomaly. For hyperbolic orbits, $a<0$ negative and $e>1$, for parabolic orbits $a=\infty$ and $e=1$ and for elliptic orbits $a>0$ and $0<e<1$.

The Cartesian coordinate system is inertial, the orbit is given also by six elements, the IJK components and the v_i, v_j, v_k. The I axis is pointed towards the vernal equinox, J axis is 90º east in the equatorial plane and K axis extends towards the north pole of Earth.

It is important to transform the coordinates, according to the calculations that best suit the models used, so the transformation equations are also represented in [7].

Cartesian to Keplerian $(\vec{r},\vec{v} \Rightarrow a,e,i,\Omega,\omega,\upsilon)$

The angular momentum vector $\vec{h}$ is needed, and it is calculated with the general cross-product equation:

$$\vec{h} = \vec{r} \times \vec{v} \tag{2.1}$$

The vector pointing to the node is found doing the cross product of the angular momentum $\vec{h}$ and the unit vector $\hat{K} = [0 \quad 0 \quad 1]$:

$$\vec{n} = \hat{K} \times \vec{h} \tag{2.2}$$

The eccentricity vector is then calculated using the cartesian vectors, their norms and the gravitational parameter μ of the orbiting body:

$$\vec{e} = \frac{\left(v^2 - \frac{\mu}{r}\right)\vec{r} - (\vec{r} \cdot \vec{v})\vec{v}}{\mu} \tag{2.3}$$

To calculate the semimajor axis a the specific mechanical energy ξ is needed:

$$\xi = \frac{v^2}{2} - \frac{\mu}{r} \tag{2.4}$$

It is important to verify if the orbit is a parabola, calculating the eccentricity vector norm.

If $|\vec{e}| \neq 1$ then:

$$a = -\frac{\mu}{2\xi} \tag{2.5}$$

$$p = a\left(1 - e^2\right) \tag{2.6}$$

If $|\vec{e}| = 1$ then:

$$p = \frac{h^2}{\mu} \tag{2.7}$$

$$a = \infty \tag{2.8}$$

And to calculate the other orbital parameters:

$$\cos(i) = \frac{h_K}{|\vec{h}|} \tag{2.9}$$

$$\cos(\Omega) = \frac{n_I}{|\vec{n}|} \quad \begin{Bmatrix} \Omega & if\,(n_J \geq 0) \\ \Omega = 2\pi - \Omega & if\,(n_J < 0) \end{Bmatrix} \tag{2.10}$$

$$\cos(\omega) = \frac{\vec{n} \cdot \vec{e}}{|\vec{n}||\vec{e}|} \quad \begin{Bmatrix} \omega & if\,(e_K \geq 0) \\ \omega = 2\pi - \omega & if\,(e_K < 0) \end{Bmatrix} \tag{2.11}$$

$$\cos(\upsilon) = \frac{\vec{e} \cdot \vec{r}}{|\vec{e}||\vec{r}|} \quad \begin{Bmatrix} \upsilon & if\,(\vec{r} \cdot \vec{v} \geq 0) \\ \upsilon = 2\pi - \upsilon & if\,(\vec{r} \cdot \vec{v} < 0) \end{Bmatrix} \tag{2.12}$$

In the equations that have a condition statement, the mentioned parameters have to be verified, as the answer can be between 180^o and 360^o.

Keplerian to Cartesian $(a, e, i, \Omega, \omega, \upsilon \Rightarrow \vec{r}, \vec{v})$ **[7]**

This method consists of determining the position and velocity vectors in perifocal coordinate system (PQW), and then using a rotation matrix to convert to the geocentric equatorial system. For this, the semiparameter is used because the semimajor axis in a parabola is infinite.

$$p = a(1-e^2) \tag{2.13}$$

$$\vec{r}_{PQW} = \begin{bmatrix} \frac{p\cos(\upsilon)}{1+e\cos(\upsilon)} \\ \frac{p\sin(\upsilon)}{1+e\cos(\upsilon)} \\ 0 \end{bmatrix} \tag{2.14}$$

$$\vec{v}_{PQW} = \begin{bmatrix} -\sqrt{\frac{\mu}{p}}\sin(\upsilon) \\ \sqrt{\frac{\mu}{p}}(e+\cos(\upsilon)) \\ 0 \end{bmatrix} \tag{2.15}$$

$$R = \begin{bmatrix} \cos(\Omega)\cos(\omega)-\sin(\Omega)\sin(\omega)\cos(i) & -\cos(\Omega)\sin(\omega)-\sin(\Omega)\cos(\omega)\cos(i) & \sin(\Omega)\sin(i) \\ \sin(\Omega)\cos(\omega)+\cos(\Omega)\sin(\omega)\cos(i) & -\sin(\Omega)\sin(\omega)+\cos(\Omega)\cos(\omega)\cos(i) & -\cos(\Omega)\sin(i) \\ \sin(\omega)\sin(i) & \cos(\omega)\sin(i) & \cos(i) \end{bmatrix} \tag{2.16}$$

To obtain the final vectors in the geocentric equatorial reference frame, the obtained cartesian vectors are rotated through the rotation matrix R for this case [7][8].

$$\vec{r} = R \cdot \vec{r}_{PQW} \tag{2.18}$$

$$\vec{v} = R \cdot \vec{v}_{PQW} \tag{2.19}$$

2.2 Epoch format

The epoch format chosen is the Modified Julian Date, as it is compatible with GMAT and simplifies the conversions and calculations.

The conversion from Gregorian Date to Julian Date and to Modified Julian Date is shown below [7]:

$$JD = 367\,y - \text{int}\left(\frac{7\left(y + \text{int}\left(\frac{m+9}{12} \right) \right)}{4} \right) + \text{int}\left(\frac{275\,m}{9} \right) + d + 1721013\ .5 \tag{2.20}$$

$$MJD = (JD - 2430000\ ) + p_{day} \tag{2.21}$$

$$p_{day} = \frac{\frac{s}{60} + mn}{60} + h \over 24 \tag{2.22}$$

Where y is year, m is month, d is day, h is hour, mn is minutes and s is seconds, while the term p_{day} is the fraction of day.

2.3 Patched conic trajectories

The patched conic approximation is a method to simplify ballistic interplanetary trajectories. The method consists of an interplanetary transfer be approximated by several arcs, and in each one of these the motion of the spacecraft is described by the two-body problem, considering only the influence of the main body on the vehicle.

In this case, with the application of the patched conic method, the trajectory will be divided in 5 phases:

- The arc from Earth to the Moon's SOI, considering a hyperbolic trajectory relative to Earth;
- The arc that describes the Moon's flyby, considering a hyperbolic trajectory relative to the Moon;
- The arc from the Moon's SOI to the end of Earth's SOI, also hyperbolic;
- The arc of the interplanetary transfer, between Earth and Mars, described by an elliptic orbit relative to the Sun;
- Finally, the Mars arrival, described by a hyperbola relative to Mars.

The dimension of the celestial bodies SOI and other important data are present in Appendix A, Table A.1.

2.4 Kepler propagation

For the Earth's and Sun's sections of the trajectory, the two-body Kepler propagation is used. This method predicts with a reasonable accuracy the position and velocity vectors on any moment of the orbit, given the initial position, velocity vectors and the time between the initial position and the position to be calculated.

The universal variables method of Kepler propagation is one of the methods to solve Kepler's problem, but unlike other methods mentioned in [7], this method allows to solve the problem in all possible conic sections so, in this case it is the best option. The classical formulas involving the different anomalies E, B and H have some problems when dealing with nearly parabolic orbits.

The used algorithm is described below:

Kepler Propagation $(\vec{r}_0, \vec{v}_0, \Delta t \Rightarrow \vec{r}, \vec{v})$**:**

$$\xi = \frac{v_0^2}{2} - \frac{\mu}{r_0} \tag{2.23}$$

$$\alpha = \frac{-v_0^2}{\mu} + \frac{2}{r_0} \tag{2.24}$$

where α is the inverse of the semimajor axis to verify the geometry of the orbit and χ is an universal variable that is defined to replace time as the independent variable.

For circle or ellipse $(\alpha > 0.000001)$: $\chi_0 \approx \sqrt{\mu}(\Delta t)\alpha$

For parabola $(|\alpha| < 0.000001)$:

$$\vec{h} = \vec{r}_0 \times \vec{v}_0 \tag{2.25}$$

$$p = \frac{h^2}{\mu} \tag{2.26}$$

Equations 2.26 and 2.27 represent the Baker's equation for parabolic orbits [7] and the calculation of the variable χ_0, where s is the period of the orbit.

$$\cot(2s) = 3\sqrt{\frac{\mu}{p^3}}(\Delta t) \tag{2.27}$$

$$\tan^3(w) = \tan(s) \tag{2.28}$$

$$\chi_0 \approx \sqrt{p}\,2\cot(2w) \tag{2.29}$$

For Hyperbola: $(\alpha < -0.000001)$:

$$a = \frac{1}{\alpha} \tag{2.30}$$

$$\chi_0 \approx sign(\Delta t)\sqrt{-a}\,\ln\left\{\frac{-2\mu\alpha(\Delta t)}{\vec{r}_0 \cdot \vec{v}_0 + sign(\Delta t)\sqrt{-\mu a}(1 - r_0\alpha)}\right\} \tag{2.31}$$

$$\psi = \chi_n^2\alpha \tag{2.32}$$

ψ is a convenient variable introduced to relate χ and the semimajor axis.

Compute $c_2 c_3(\psi \Rightarrow c_2, c_3)$

$$r = \chi_n^2 c_2 + \frac{\vec{r}_0 \cdot \vec{v}_0}{\sqrt{\mu}}\chi_n(1 - \psi c_3) + r_0(1 - \psi c_2) \tag{2.33}$$

$$\chi_{n+1} = \chi_n + \frac{\sqrt{\mu}\Delta t - \chi_n^3 c_3 - \frac{\vec{r}_0 \cdot \vec{v}_0}{\sqrt{\mu}}\chi_n^2 c_2 - r_0\chi_n(1 - \psi c_3)}{r} \tag{2.34}$$

χ_n assumes the value of χ_{n+1} before the loop restarts.

A loop sequence using the Newtown-Raphson is required until $|\chi_n - \chi_{n-1}| < 1 \times 10^{-6}$.

Finally, the f and g functions calculation:

$$f = 1 - \frac{\chi_n^2}{r_0} c_2 \tag{2.35}$$

$$\dot{f} = \frac{\sqrt{\mu}}{r \cdot r_0} \chi_n (\psi c_3 - 1) \tag{2.36}$$

$$g = \Delta t - \frac{\chi_n^3}{\sqrt{\mu}} c_3 \tag{2.37}$$

$$\dot{g} = 1 - \frac{\chi_n^2}{r} c_2 \tag{2.38}$$

Finally, the f and g functions are applied to the position and velocity vectors:

$$\vec{r} = f \cdot \vec{r}_0 + g \cdot \vec{v}_0 \tag{2.39}$$

$$\vec{v} = \dot{f} \cdot \vec{r}_0 + \dot{g} \cdot \vec{v}_0 \tag{2.40}$$

This method uses the c_2 and c_3 values, the universal variables and it is known as the Sundman transformation, which relate the time and orbit's properties, their calculation is explained:

If $\psi > 1 \times 10^{-6}$:

$$c_2 = \frac{1 - \cos(\sqrt{\psi})}{\psi} \tag{2.41}$$

$$c_3 = \frac{\sqrt{\psi} - \sin(\sqrt{\psi})}{\sqrt{\psi^3}} \tag{2.42}$$

If $\psi < -1 \times 10^{-6}$:

$$c_2 = \frac{1 - \cosh(\sqrt{-\psi})}{\psi} \tag{2.43}$$

$$c_3 = \frac{\sinh(\sqrt{-\psi}) - \sqrt{-\psi}}{\sqrt{(-\psi)^3}} \tag{2.44}$$

In case the $-1 \times 10^{-6} < \psi < 1 \times 10^{-6}$:

$$c_2 = \frac{1}{2} \tag{2.45}$$

$$c_3 = \frac{1}{6} \tag{2.46}$$

The previous algorithm is presented in [7], as well as the c_2 and c_3 values, the universal variables.

2.5 Earth, Moon and Mars coordinates

The position and velocities of the celestial bodies were based on data from GMAT, as it is possible to retrieve report files with the planets or any other celestial bodies data. The report files consisted in the MJD time, and cartesian state vectors in each 1/1000 of a day approximately. If the files were used as described, the amount of time and computation needed to search for the state vector several times per iteration would be immense. So instead of this method, the report file was reduced to one state vector per day, this state vector is used to calculate the position and velocity of the body, using the Kepler Propagation, according to the Δt between the time in the report file for that day, and the time of the state vector to determine.

Kepler Propagation $(\vec{r}_{ref}, \vec{v}_{ref}, t - t_{ref} \Rightarrow \vec{r}, \vec{v})$

$\vec{r}_{ref}, \vec{v}_{ref}, t_{ref}$ are the position vector, velocity vector and time reference given by the GMAT report file.

2.6 Lambert's problem

The Lambert's problem is one of the most important tools for initial orbit determination, as it calculates an initial velocity vector to connect any given points in space, given the time of flight but the orbit is unknown. It is specially known for interplanetary mission design and likewise widely used as a tool to construct the useful pork chop plots.

The method used for this work to solve this problem, is the Lambert's problem function from the library *pykep*, which is a scientific library for Python, developed by the European Space Agency for astrodynamics research [9].

Lambert's Problem $(\vec{r}_0, \vec{r}_f, \Delta t \Rightarrow \vec{v}_0, \vec{v}_f)$

This function is used to estimate initial velocity vectors that connect the initial starting points of the Moon's SOI, during the pruning phase, and Mars position in the arrival epoch. It is also used to calculate the first velocity vector, in the Earth's parking orbit, that connects this point and the point in the B-plane given by the pruning phase.

2.7 Gravity assist model

The gravity assist model is used to describe the motion of the vehicle inside the Moon's SOI, more specifically to know the position and velocity vectors after the flyby, as well as the distance of closest approach to the Moon and the time of flight of the hyperbolic passage.

In relation to the Moon, the excess velocity magnitude is the same, before and after the flyby, but the change of direction causes the initial orbit around Earth to be different. If the spacecraft passes behind the Moon, relative to its direction of motion, the vehicle will

increase its orbit's energy in the Earth reference. However if it passes in front of the Moon, the opposite will happen, resulting in an orbit with less energy.

The spacecraft's state vector in the Moon reference frame are:

$$\vec{r}_m = \vec{r}_E - \vec{r}_{Moon} \tag{2.47}$$

$$\vec{v}_m = \vec{v}_E - \vec{v}_{Moon} \tag{2.48}$$

$\vec{r}_E, \vec{v}_E$ are the position and velocity vectors in Earth's reference frame, and $\vec{r}_{Moon}, \vec{v}_{Moon}$ are Moon's position and velocity, also in Earth's reference frame.

To calculate the state vectors after the flyby, it is necessary to calculate the Keplerian elements of the hyperbolic orbit in the Moon's reference frame.

Cartesian to Keplerian $(\vec{r}_{mi}, \vec{v}_{mi} \Rightarrow a, e, i, \Omega, \omega, \upsilon_i)$

From the hyperbolic flight path, it is known that the true anomaly at the end of the flyby is

$$\upsilon_f = 2\pi - \upsilon_i \tag{2.49}$$

so, the rotation matrix (2.15) can be used to obtain the final state vectors.

Keplerian to Cartesian $(a, e, i, \Omega, \omega, \upsilon_f \Rightarrow \vec{r}_{mf}, \vec{v}_{mf})$

It is also vital to calculate the time of flight in the Moon's SOI.

From the Kepler equation that describes the mean motion [7], the time since periapsis and the eccentric anomaly can be related:

$$M = E - e\sin(E) = \sqrt{\mu a^3}\,(t - T) \tag{2.50}$$

The expression $(t - T)$ represents the change in time.

Equation 2.49 can be arranged to define the time of flight between the two positions:

$$t_{flight} = \frac{(E - E_0) - e(\sin E - \sin E_0)}{\sqrt{\mu a^3}} \tag{2.51}$$

In this case, the time from the initial position on the Moon's SOI of the hyperbolic passage to periapsis and from periapsis to the final position is the same, equation 2.50 can be:

$$t_{flight} = 2\sqrt{\frac{-a^3}{\mu}}\,(e\sinh(H) - H) \tag{2.52}$$

The hyperbolic anomaly H is determined by [7]:

$$r = a(1 - e\cosh(H)) \tag{2.53}$$

After the state vectors calculation in the Moon's reference frame, it is necessary, to transform it into the Earth's reference frame again.

$$\vec{r}_{Ef} = \vec{r}_{mf} + \vec{r}_{Moon} \tag{2.54}$$

$$\vec{v}_{Ef} = \vec{v}_{mf} + \vec{v}_{Moon} \tag{2.55}$$

2.8 B-Plane

The B-plane or Body plane is a planar coordinate system that helps targeting a celestial body. It can be used to aim for a specific point to enter in orbit of a celestial body, for example to enter a polar orbit when arriving at the planet or moon [10]. Generally, the targeting is made in mid-course or in a correction manoeuvre, but it can also be made in the injection burn. The B-plane can be seen as an attached target to the body, and it is always perpendicular to the incoming asymptote of the spacecraft approaching the body.

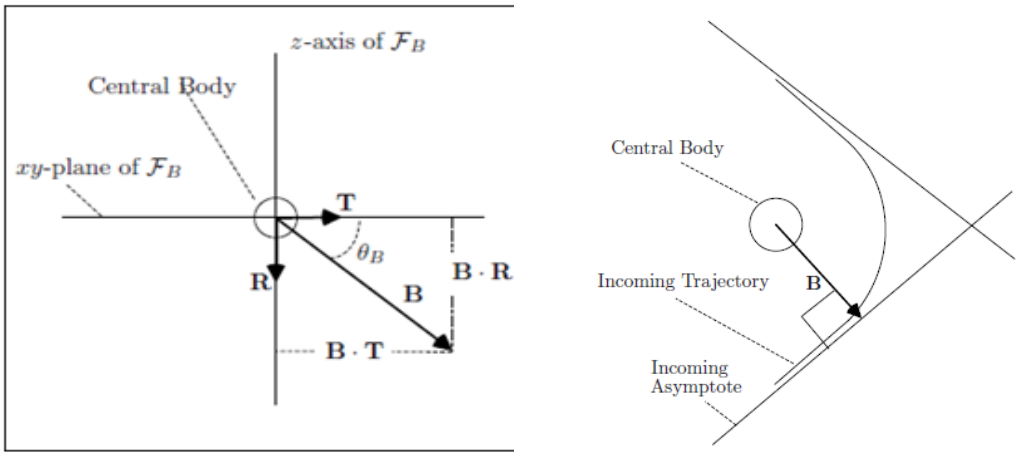


Figure2.1-B-plane Coordinate System seen from a viewpoint perpendicular to the B-plane, and seen from a perpendicular viewpoint to the orbit plane

In figure 2.1, in the left, $B \cdot T$ and $B \cdot R$ represent the point where the vehicle pierces the B-plane, while in the right, both asymptotes of the hyperbolic trajectory that the spacecraft performs are represented.

It is possible to obtain the $B \cdot T$ and $B \cdot R$ coordinates from the $\vec{r}$ and $\vec{v}$ coordinates [7][11]:

$$\hat{h} = \frac{\vec{r} \times \vec{v}}{|\vec{r} \times \vec{v}|} \tag{2.56}$$

The unit vector normal to the orbital plane is written in Equation 2.56. The eccentricity vector is obtained in equation 2.56:

$$\vec{e} = \frac{\left(v^2 - \frac{\mu}{r}\right)\vec{r} - (\vec{r} \cdot \vec{v})\vec{v}}{\mu} \tag{2.57}$$

The semimajor and semiminor axis are determined, where v_∞ is the excess velocity at infinity:

$$a = -\frac{\mu}{v_\infty^2} \tag{2.58}$$

$$b = -a\sqrt{e^2 - 1} \tag{2.59}$$

It is possible to use the eccentricity norm to calculate the hyperbolic asymptote angle φ_s in equation 2.59:

$$\varphi_s = \cos^{-1}\left(\frac{1}{e}\right) \tag{2.60}$$

The unit vector $\hat{S}$ and $\hat{T}$ are calculated in, where $\hat{K} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$:

$$\hat{S} = \frac{\vec{e}}{e}\cos(\varphi_s) + \frac{\hat{h} \times \vec{e}}{\left|\hat{h} \times \vec{e}\right|}\sin(\varphi_s) \tag{2.61}$$

$$\hat{T} = \frac{\hat{S} \times \hat{K}}{\left|\hat{S} \times \hat{K}\right|} \tag{2.62}$$

The $\hat{R}$ vector can be calculated with the cross product:

$$\hat{R} = \hat{S} \times \hat{T} \tag{2.63}$$

Finally, the $\vec{B}$ vector, the B_T and B_R parameters are given by:

$$\vec{B} = b(\hat{S} \times \hat{h}) \tag{2.64}$$

$$B_T = \vec{B} \cdot \hat{T} \tag{2.65}$$

$$B_R = \vec{B} \cdot \hat{R} \tag{2.66}$$

2.9 Spherical coordinates

In parts of this work, namely in the pruning phase, it is important to find the positions in the Moon's SOI that can be connected to Mars with a proper trajectory. The best way to do it is to make a grid search, using spherical coordinates. So, the coordinate transformation between spherical and cartesian coordinates is presented:

Spherical to Cartesian $(r, \theta, \varphi \Rightarrow r_I, r_J, r_K)$

$$r_I = r\sin(\theta)\cos(\varphi) \tag{2.67}$$

$$r_J = r\sin(\theta)\sin(\varphi) \tag{2.68}$$

$$r_K = r\cos(\theta) \tag{2.69}$$

Where r is the radial distance, θ is the polar angle, or the angle between the point and $\hat{K}$, φ is the azimuthal angle, or the angle between the point and $\hat{I}$. r_I, r_J, r_K are the $\vec{r}$ components.

2.10 Self-adaptive Levenberg-Marquardt algorithm

The self-adaptive Levenberg-Marquardt (SLM) algorithm is an iterative method used to find the minimum of a system of non-linear equations or to find its solution as in [12].

$$F_n(x_1, x_2, x_{,3} \ldots, x_i) = 0 \tag{2.70}$$

This method has been used for various applications through numerous fields of study, but in this work it is used as differential corrector, in order to minimize the difference between the target values and the calculated trajectories values.

Over the last two decades, the LM method was improved several times, including the addition of new parameters, mainly to improve the updating of the damping parameter μ_k.

The first changes in the method were introduced in [13], choosing the LM parameter as $\mu_k = \|F_k\|^2$. Although this value could be either too large and not converge quickly or could be too small and lose its role according to [12].

Following this modification [14] proposed $\mu_k = \|F_k\|$ and concluded it has a quadratic convergence under the local error bound condition. So, the next modification to the value of this parameter came with [15], who added a positive constant $\delta \in [0,2]$: $\mu_k = \|F_k\|^\delta$. In this case the constant causes the method to converge to the solution super linearly when $\delta \in [0,2]$ and quadratically for $\delta = 2$.

However, the last LM parameter on which [12] was based, brings another variable to determine the size of the step: $\mu_k = \alpha_k \|F_k\|$. The parameter α_k is also updated along the iteration process according to the value of r_k, the ratio between the actual reduction and the predicted reduction.

In the used algorithm, the chosen definition for the LM parameter is:

$$\mu_k = \alpha_k \|F_k\|^\delta \tag{2.71}$$

$$\alpha_{k+1} = \alpha_k q_k(r_k) \tag{2.72}$$

where a $q(r_k)$ continuous nonnegative function appears, and $r = r_k$. Now, α_k is changed at each iteration, depending on the ratio between reductions. So r_k controls the step of the convergence by updating constantly α_k and consequently the LM parameter.

q(r) can be given by:

$$q(r) = \max\left\{\frac{1}{4}, 1 - 2(2r-1)^3\right\} \tag{2.73}$$

We consider the solution to be:

$$F_n(\boldsymbol{X}) = F_n(\boldsymbol{X}_k) + \boldsymbol{d}_k \tag{2.74}$$

where $\boldsymbol{d}_k$ is a correction factor calculated in:

$$\boldsymbol{d}_k = -(\boldsymbol{J}_k^T \boldsymbol{J}_k + \mu_k \boldsymbol{I})^{-1} \boldsymbol{J}_k^T F_k \tag{2.75}$$

$\boldsymbol{J}_k$ being the Jacobian Matrix of the system, μ_k is the damping parameter, $\boldsymbol{I}$ the identity matrix and $F_k = F(\boldsymbol{X}_k)$.

The damping parameter μ_k is a positive multiplier and was introduced in this method with the purpose of reducing the impact of the singularity of the Jacobian Matrix $\boldsymbol{J}_k$. The parameter α_k is also updated along the iteration process according to the value of r_k, the ratio between the actual reduction and the predicted reduction.

The algorithm is briefly described below:

SLM: $(F_n(\boldsymbol{X}), \boldsymbol{X}_I, F_{objective} \Rightarrow \boldsymbol{X}_k, F_n(\boldsymbol{X}_k) - F_{objective})$:

The iteration process starts with the calculation of the step $\boldsymbol{d}_k$, through equation 2.74.

The next phase is to find the reduction ratio and predict the next x_k values:

$$r_k = \frac{Ared_k}{Pred_k} \tag{2.76}$$

$$Pred_k = \varphi_k(0) - \varphi_k(d_k) \tag{2.77}$$

$$Ared_k = \|F_k\|^2 - \|F(x_k + d_k)\|^2 \tag{2.78}$$

$$\varphi(0) = \|F_k\|^2 \tag{2.79}$$

$$\varphi(d_k) = \|F_k + J_k d_k\|^2 \tag{2.80}$$

According to the value of r_k, a condition is established for the value of x_{k+1}:

$$x_{k+1} = \begin{cases} x_k + d_k \; if \quad r_k > p_0, \\ x_k, \quad r_k \le p_0 \end{cases} \tag{2.81}$$

$p_0 = 0.0001$ is a value to check if the reduction is acceptable. The last step is to update the parameter α_k:

$$\alpha_{k+1} = \max\{m, \alpha_k q(r_k)\} \tag{2.82}$$

Finally, repeat until $\left\|J_k^T F_k\right\| = 0$ or $k = 100$ or until the desired objectives are obtained. In this algorithm the m constant [12] is the minimum limit for the LM parameter in order to avoid a larger step than desired near the solution.

2.11 Differential corrector application

In this subsection, it is described how the differential corrector is used to reach the set objectives.

2.11.1 Pruning phase

The Self-Adaptive Levenberg-Marquardt algorithm is first used in the pruning phase, where the objective is to find the first trajectories that connect points on the surface of the Moon's SOI and Mars. So, in this case, the trajectory starts with a Kepler Propagation from the end of the Moon's SOI until the end of Earth's SOI, followed by a Kepler Propagation from this point to Mars. The objective function gives the arrival point depending on the departure position $\vec{r}_i$ and initial velocity $\vec{v}_i$, as well as the departure and arrival times t_i, t_f, respectively.

$$F_{pruning}(\vec{r}_i, \vec{v}_i, t_i, t_f) \Rightarrow F_{pruning}(\vec{v}_i)$$

$$F_{pruning}(\vec{v}_i) = \begin{bmatrix} R_I \\ R_J \\ R_K \end{bmatrix} \tag{2.83}$$

R_I, R_J, R_K are the cartesian IJK coordinates for the spacecraft position at t_f relative to the Sun.

Although the function is dependent on all the parameters, the only variable is the velocity $\vec{v}_i$. As this function defines the motion of the spacecraft according to the two-body problem, and its position in t_f, the Mars position at t_f is our objective value.

$$(F_{pruning}(\vec{v}), \vec{v}_i, R_{Mars} \Rightarrow \vec{v}_{opt}, F_{pruning}(\vec{v}_{opt}) - R_{Mars})$$

$$R_{Mars} = \begin{bmatrix} R_I^{ref} \\ R_J^{ref} \\ R_K^{ref} \end{bmatrix} \tag{2.84}$$

R_I^{ref}, R_J^{ref}, R_K^{ref} are the Mars IJK coordinates for the position of the planet in t_f relative to the Sun.

It is necessary to calculate the Jacobian matrix of the system:

$$J_k = \begin{bmatrix} \frac{\partial F(\vec{v})_I}{\partial v_I} & \frac{\partial F(\vec{v})_I}{\partial v_J} & \frac{\partial F(\vec{v})_I}{\partial v_K} \\ \frac{\partial F(\vec{v})_J}{\partial v_I} & \frac{\partial F(\vec{v})_J}{\partial v_J} & \frac{\partial F(\vec{v})_J}{\partial v_K} \\ \frac{\partial F(\vec{v})_K}{\partial v_I} & \frac{\partial F(\vec{v})_K}{\partial v_J} & \frac{\partial F(\vec{v})_K}{\partial v_K} \end{bmatrix} \tag{2.85}$$

The partial derivatives are calculated with the forward finite difference:

$$\frac{\partial F(\vec{v})}{\partial v} = \frac{F(v + \Delta v) - F(v)}{\Delta v} \tag{2.86}$$

where $\Delta v = 10^{-10}$ is the perturbation, which was chosen due to the brief study performed in [5].

The pruning analysis is performed in two or more stages, as in first stage, a search for the region in the Moon's SOI that best suits the chosen criteria. This starts with the intervals for spherical coordinates $0<\theta<\pi$, $0<\varphi<2\pi$ and the value of r is the Moon SOI radius. Each of these intervals was divided in 40, resulting in a grid with 1600 points each 1/5 of a day over the course of 5 days. After this preliminary search, to select the suitable values, a backwards propagation is needed to calculate the radius of periapsis, the B-plane parameters, the time of flight and the eccentricity of the orbit relative to Earth before the flyby. So, to select the values for the next pruning phase, the requirements are the minimum periapsis radius, while above the Moon's equatorial radius $(\approx 1750\,km)$ and below $5000km$, and the minimum eccentricity before the flyby, as well as the requirement of $B\cdot T>0$, with the objective of only selecting the trajectories that passed behind the Moon according to the Moon's velocity vector.

In the second phase, the process is repeated with a smaller grid, with an interval of $0.4rad$ for both θ and φ angles, surrounding the values that gave the best results in the previous stage. The time window is also reduced to 2 days and the analysis is made each 1/10 of a day.

In the third and final stage, the grid is reduced to an interval of $0.2rad$, the same way as in the previous stage, and the time is reduced to one day with a 1/50 of a day analysis step.

The results of this analysis consist of the position coordinates, the velocity vector and the time in the surface of the Moon's SOI, as well as the radius of periapsis and eccentricity values. The trajectories with the smallest periapsis radius are analyzed, which can represent the trajectories where the Moon contributes more to the angle change, and consequently to greater energy orbit. The trajectories where the eccentricity relative to Earth before the flyby values are smaller are also studied, which represents the orbits that depart Earth with less energy.

2.11.2 Single shooting method

In order to find the complete trajectory that connects the departure point and Mars with a Lunar gravity assist, the single shooting method was used. According to this method, the self-adaptive Levenberg-Marquardt algorithm was used to target simultaneously the B-plane parameters given by the pruning analysis and Mars position in t_f . This method is implemented in a similar way as in the pruning phase, however in this phase the whole trajectory is determined.

The trajectory starts with a Kepler propagation from the parking orbit until the Moon's SOI, followed by the hyperbolic passage while in the Moon's SOI, and similarly to the pruning, a Kepler propagation until the Earth's SOI limit and another until Mars arrival.

The function that describes the path is:

$$F_{final}(\vec{r}_i,\vec{v}_i,t_i,t_f)\Rightarrow F_{final}(\vec{v}_i)$$

$$F_{final} = \begin{bmatrix} R_I \\ R_J \\ R_K \\ B \cdot T \\ B \cdot R \\ e \end{bmatrix} \tag{2.87}$$

R_I , R_J , R_K are the cartesian IJK coordinates for the spacecraft position at t_f relative to the Sun. $B \cdot T$ and $B \cdot R$ are the B-plane parameters calculated from the initial velocity vector and e is the orbit's eccentricity [5].

Likewise, the function variable is just the initial velocity vector $\vec{v}_i$

$$(F_{final}(\vec{v}), \vec{v}_i, F_{goal} \Rightarrow \vec{v}_{opt}, F_{final}(\vec{v}_{opt}) - F_{goal})$$

$$F_{goal} = \begin{bmatrix} R_I^{ref} \\ R_J^{ref} \\ R_K^{ref} \\ B \cdot T^{ref} \\ B \cdot R^{ref} \\ e^{ref} \end{bmatrix} \tag{2.88}$$

The matrix F_{goal} is formed by the reference values for the algorithm to solve to, minimizing the difference between the function outcome and the objective. R_I^{ref} , R_J^{ref} , R_K^{ref} are the Mars IJK coordinates for the position of the planet in t_f relative to the Sun, while $B \cdot T^{ref}$ and $B \cdot R^{ref}$ are the B-plane reference coordinates to target the gravity assist, and e^{ref} is the reference eccentricity relative to the Earth given by the pruning phase.

As this function yields six values from three variables, the Jacobian matrix for this problem is:

$$J_k = \begin{bmatrix} \frac{\partial F(\vec{v})_I}{\partial v_I} & \frac{\partial F(\vec{v})_I}{\partial v_J} & \frac{\partial F(\vec{v})_I}{\partial v_K} \\ \frac{\partial F(\vec{v})_J}{\partial v_I} & \frac{\partial F(\vec{v})_J}{\partial v_J} & \frac{\partial F(\vec{v})_J}{\partial v_K} \\ \frac{\partial F(\vec{v})_K}{\partial v_I} & \frac{\partial F(\vec{v})_K}{\partial v_J} & \frac{\partial F(\vec{v})_K}{\partial v_K} \\ \frac{\partial B \cdot T(\vec{v})}{\partial v_I} & \frac{\partial B \cdot T(\vec{v})}{\partial v_J} & \frac{\partial B \cdot T(\vec{v})}{\partial v_K} \\ \frac{\partial B \cdot R(\vec{v})}{\partial v_I} & \frac{\partial B \cdot R(\vec{v})}{\partial v_J} & \frac{\partial B \cdot R(\vec{v})}{\partial v_K} \\ \frac{\partial e(\vec{v})}{\partial v_I} & \frac{\partial e(\vec{v})}{\partial v_J} & \frac{\partial e(\vec{v})}{\partial v_K} \end{bmatrix} \tag{2.89}$$

Chapter 3

3 Results

In this chapter, the results, graphs and tables obtained from the simulations are presented and discussed. The values obtained are compared with the previous work [5] in case of the MER-A mission, and the others are compared to the pork chop plots, results from [3] and to results obtained with the single shooting method for a direct transfer.

The most efficient epoch dates for departure and arrival, as well as the type of transfer are from [3] results.

Table 3.1: Direct Launch Windows with results from [3] and [5]

Departure (UTCGregorian) dd-mm-yy	**Arrival (UTCGregorian) dd-mm-yy**	**Departure (MJD)**	**Arrival (MJD)**	**Orbit type**	**Direct Characteristic Energy** $(C_3)[km^2/s^2]$
10-06-2003	04-01-2004	22801	23009	Short	8.94481
23-08-2020	06-10-2021	29085	29494	Long	16.5038
15-09-2022	04-10-2023	29837	30222	Long	13.7934
05-10-2024	15-09-2025	30589	30934	Long	11.1894
30-10-2026	21-08-2027	31344	31629	Long	9.1371

The table (3.1) describes the direct transfers that can be used for an interplanetary mission to Mars, including departure and arrival epochs, the orbit type and approximate characteristic energy C_3.

The purpose of the gravity assist manoeuvre is to reduce the characteristic energy C_3 and obtain a trajectory that accomplishes, with a small deviation, the departure and arrival dates. It is unlikely that the Moon is in the correct position to perform the gravity assist at the most efficient time to do the transfer manoeuvre. So, in the worst-case scenario, the injection epoch can be up to 15 days after or prior to the ideal epoch for interplanetary transfer, which can cause a significant reduction of the orbit energy gain, or even question the viability of the gravity assist. According to [3], the estimates of the direct characteristic energy are calculated with Lambert's problem. Similarly to the two-body problem it only considers the gravitational influence of one body at a time.

In order to approximate the numerical study as close to reality as possible, the General Mission Analysis Tool (GMAT) is used. GMAT is an open source software developed by a NASA's team, but also private and public contributors and it is the only available open source software for mission design, analysis, optimization and navigation. The first released version was in 2007 and since then it has been continuously improved with several accessible

versions. It was validated and verified in [16]. The latest version released is the R2018a, the one applied in this work. This tool is supporting some real-world missions such as the Solar Dynamics Observatory (SDO), the Solar and Heliospheric Observatory (SOHO), the Advanced Composition Explorer (ACE), the Transiting Exoplanet Survey Satellite (TESS) and the Lunar Reconnaissance Orbiter (LRO). GMAT's syntax is based on MATLAB, and it can be used either by user interface or script language. Besides that, there are two interfaces with Python and MATLAB that can be used to run user build functions.

In this work 5 launch windows are studied: the first case of study is the MER-A mission, that was launched on 10th of June of 2003. This mission was studied previously with a Lunar gravity assist manoeuvre in [5], and it was proven that a Moon flyby would decrease the characteristic energy significantly. So, to test the algorithm that is used in this work, this case is studied once more to replicate or to improve the results. The remaining cases studied are all future possible launch windows that are scheduled to happen in the next 10 years. So, in these cases, there is the option of choosing the most favourable launch and arrival epochs, which clearly improve the chances of performing a successful gravity assist manoeuvre.

In this chapter, the graphs and steps of only two of the analysed trajectories are displayed and discussed, while for the remaining, only the results are presented. The launch windows studied with the steps presented in detail start in around October of 2024 and October of 2026, and are identified by case 1 and case 2, respectively.

3.1 Direct single shooting and GMAT values

In order to compare the results from the gravity assist, a brief study of the Mars direct transfers is performed. Using the departure and arrival epochs from [3], the self-adaptive Levenberg-Marquardt algorithm is applied the same way as in the single shooting method, only considering the departure position as the same position that is described in the work ahead to perform the gravity assist. The parts of the function that describe the motion considering the Moon are also discarded.

The results from the direct trajectory obtained from the single shooting method and the GMAT simulation are presented in table 3.2.

Table 3.2: Direct transfer results

DEPARTURE (MJD)	**[3]** $(C_3)[km^2/s^2]$	**SSM** **DIRECT** $(C_3)[km^2/s^2]$	**GMAT** **DIRECT** $(C_3)[km^2/s^2]$
29085	16.5038	15.5614	16.3724
29837	13.7934	12.8764	14.2287
30589	11.1894	10.2156	11.0333
31344	9.1371	8.2732	9.2671

3.2 Pruning phase - case 1

In case 1, the first stage of pruning started 11 days after the ideal injection date $t_i = 30600$, because the Moon is not at the adequate position to perform the gravity assist.

In Figure 3.1, the velocity magnitudes from the pruned trajectories represented. From this plot it is possible to observe that the minimum velocity magnitude after the flyby orbits are closer to the ideal epoch given by the pork chop plots. Although by analysing the eccentricity e values in Figure 3.3 and radius of the flyby periapsis in figure 3.4 it is explicit that the

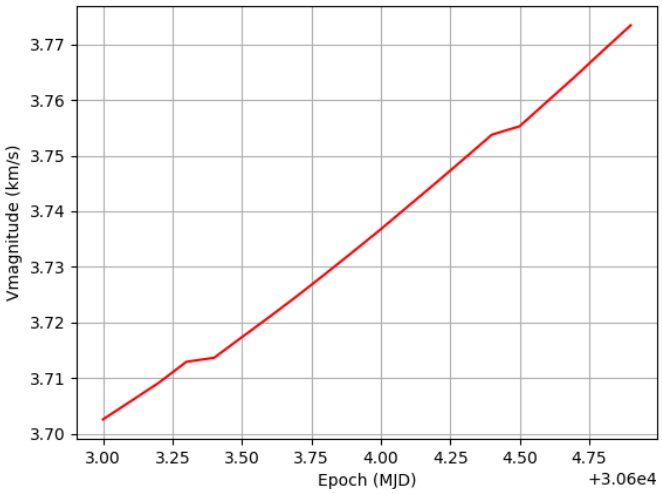


Figure 3.1: Velocity magnitude of the pruning 2nd stage – case 1

lower energy initial orbits are when $t_i = 30604$.

In Figure 3.2, the spherical coordinates angles that represent the point of departure from the Moon's SOI represented. Each one of the dots represents a trajectory that can be studied. The angle intervals used in the third pruning stage are $1.2 < \theta < 1.4$ and $1.2 < \varphi < 1.4$. The time interval is $30604 < t_i < 30605$ .The plot represents the 3rd stage of the pruning and there are less dots than expected, which means there are several trajectories per dot represented.

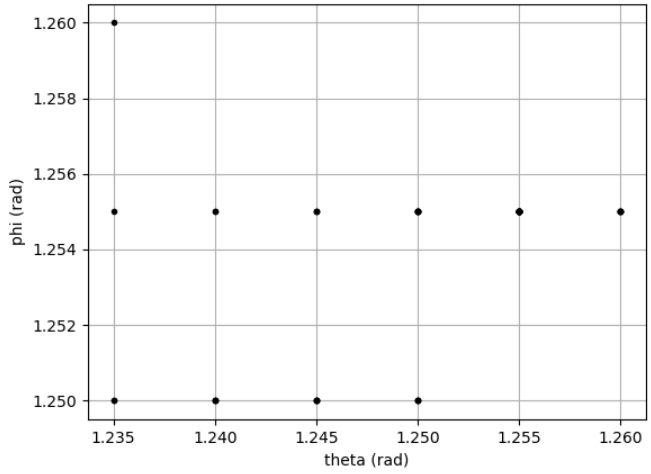


Figure 3.2: Spherical angles of the trajectories on the Moon's SOI on the 3rd stage of pruning – case 1

From figures 3.3 and 3.4, the eccentricity of the orbits relative to Earth prior to the flyby and the radius of the periapsis of the flyby are presented. From the values observed, the epoch

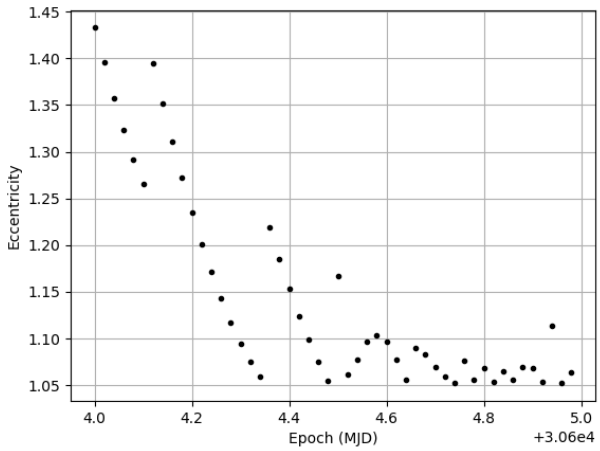


Figure 3.3: Eccentricity of the orbits relative to Earth before the flyby in the pruning 3rd stage – case 1

more likely to be more studied is between $t_i = 30604.4$ and $t_i = 30604.6$, as the most efficient trajectory tends to have lower flyby periapsis and lower eccentricity.

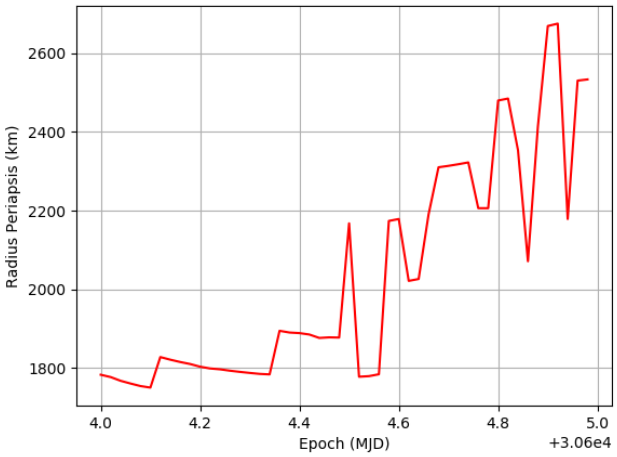


Figure 3.4: Radius of the Moon's flyby periapsis in pruning 3rd stage – Case 1

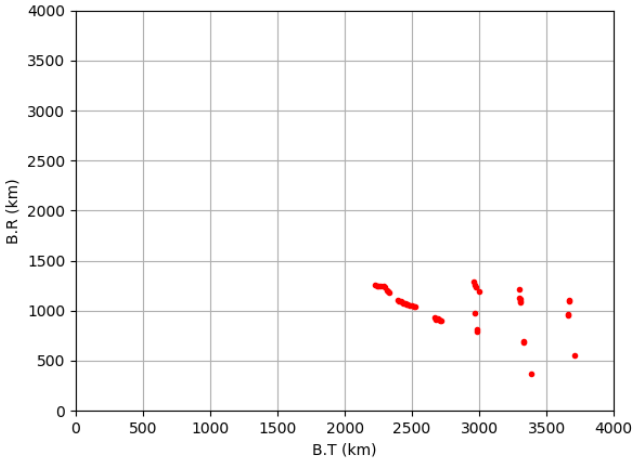


Figure 3.5: $B \cdot T$ and $B \cdot R$ parameters of the trajectories in pruning 3rd stage – Case 1

The B-plane parameters presented in figure 3.5, and eccentricity values presented in Figure 3.3 are the targeting parameters for the next step. These parameters are targeted for a first estimative value, which is optimized afterwards with the differential corrector.

3.3 Single shooting method - case 1

The results of the pruning are then studied and optimized with the Self-Adaptive Levenberg-Marquardt, as explained in the previous chapter. The position and velocity vectors from the pruning phase are propagated backwards to calculate the estimated time at the beginning of the Moon's flyby tmi and the position vector in the periapsis $\vec{r}_{periapsis}$. This position, along with tmi, the B-plane values and the eccentricity are then used to calculate the first guess.

The initial position chosen to perform the trajectory is an arbitrary position from a Keplerian low Earth orbit with the following Keplerian elements: $a = 8000km$, $e = 0.01$, $i = 10^o$, $\Omega = 0^o$, $\omega = 0^o$. The true anomaly element $\upsilon = 300^o$ is arbitrarily chosen so that the starting position is adequate to target the Moon. This orbit has the purpose of representing a general parking orbit, where the launcher deploys the spacecraft.

Before the optimization, it is important to calculate the injection epoch through the time before the flyby tmi, calculate the Moon's position at tmi and set the target for the first guess as $\vec{r}_{Moon} + \vec{r}_{periapsis}$ to shift the $\vec{r}_{periapsis}$ from Moon centred reference axes system to Earth centred. Lambert's Problem is used to obtain the velocity vector $\vec{v}_0$ that connects the initial position $\vec{r}_0$ and $\vec{r}_{periapsis}$, giving a good first guess so that the differential corrector can be applied. Instead of using tmi, an arbitrary epoch t_i is set to get the first guesses. This way it is possible to study any pruned trajectory and not being dependent of the values obtained by the pruning.

It is important to note that the B-plane parameters and the eccentricity are multiplied by a factor of 10^3, both in the function results as in the reference values. This is done because in the iterative process, the other target parameters, namely the position of Mars, has much greater absolute value, like dozens or hundreds of millions of kilometres (10^7 or $10^8\ km$), compared to the B-parameters ($10^3 km$). So, when the algorithm tries to decrease the difference between the norm of the function results and the norm of the reference vector, the fact that all the values are in the same magnitude, prevents the algorithm from reducing just the positions and increasing the difference between the function and reference B-parameters so much that the trajectory missed the Moon's SOI. When using this technique, with all parameters near the same magnitude, all of them have the same relevance, and prevents the iterations from drifting away from the B-parameters.

The pruned trajectories are then applied to the algorithm, where some results are presented in Table 3.3:

Table 3.3: 4 example results of the Lunar gravity assist manoeuvre – Case 1

$t_0(MJD)$	**Periapsis Radius** (km)	**Velocity Magnitude** (km/s)	**Characteristic Energy** $(C_3)[km^2/s^2]$
30603.3755	2419	10.5263	10.6441
30603.2704	2248	10.5152	10.4107
30603.1980	2184	10.5051	10.1994
30602.9517	1823	10.4834	9.7437

In Table 3.3, 4 different results are presented to demonstrate the parameters variations. The last result shown is the most efficient trajectory found, considering that the flyby should occur at least 50km above the Lunar surface, due to safety reasons.

Most of the results converged to the final solutions in less than 100 iterations, apart from a few exceptions that converged in less than 200. Overall, each one of the trajectories studied took less than a few minutes to obtain a solution.

The arrival epoch is the same as in the pork chop studies $t_f = 30934$.

From the table it is clear the relation between decreasing radius of the periapsis and the decreasing orbit energy and velocity magnitude.

Comparing the 4rd result from Table 3.3 with the results from Table 3.2 and the pork chop analysis, namely the characteristic energy, there is a significant reduction to obtain a trajectory to Mars. With the best result, the characteristic energy can be reduced by $0.47km^2/s^2$ using the single-shooting method.

The injection epoch t_0 influences the periapsis radius, since the sooner the spacecraft encounters the Moon, the larger the turning angle β needs to be so that the spacecraft has

the correct flight path. The orbit's energy relative to Earth also increases due to the increasing turning angle.

3.4 Graphical representation - case 1

The result obtained is now represented graphically in Python 3D plots.

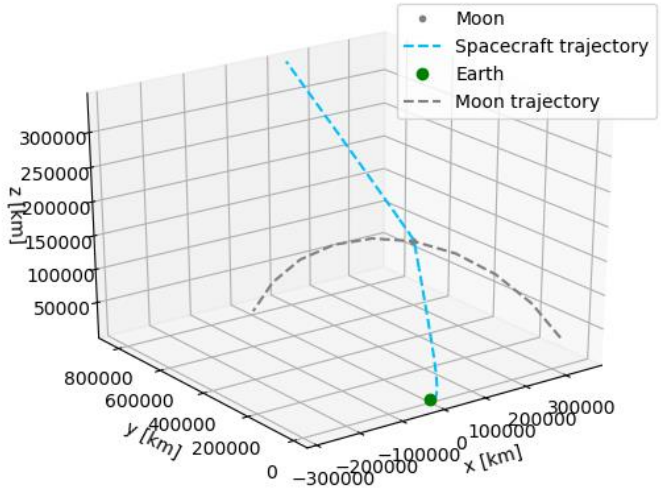


Figure 3.6: Graphical representation of the departure from Earth and the Lunar gravity assist – Case 1

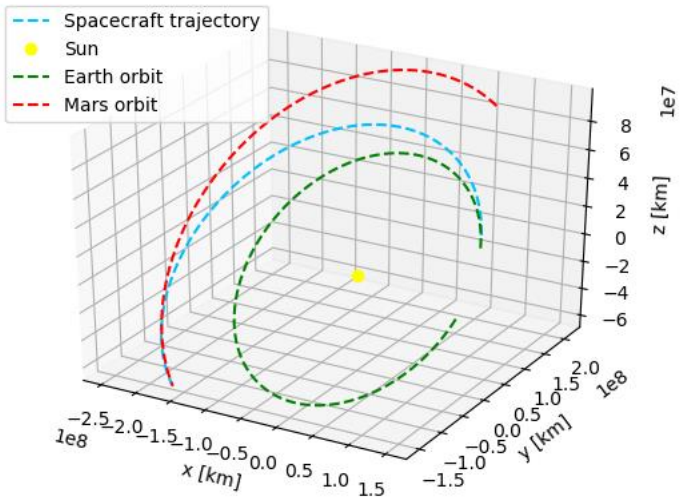


Figure 3.7: Graphical representation of the Interplanetary phase of the trajectory – Case 1

The graphs are obtained using Kepler propagation to obtain the positions of the spacecraft and using GMAT report files for the trajectory of the celestial bodies.

3.5 GMAT simulation - case 1

In order to translate the given results to a more accurate solution, and thus validating the work, the obtained results were tested in GMAT.

As the software considers all bodies in the Solar system, the results are expected to be slightly different. To obtain a valid trajectory, a new study was made with all the obtained trajectories in the previous subchapter.

The results from the GMAT analysis are represented in table 3.4:

Table 3.4- Result of the GMAT simulation – Case 1

$t_0\,(MJD)$	$t_f\,(MJD)$	r_{per} (km)	v_{mag} (km/s)	**GMAT Direct** $C_3[km^2/s^2]$	C_3 **GMAT LGA** $[km^2/s^2]$
30603.1218	30948.4127	1840	10.4882	11.0333	9.8629

From the GMAT values in table 3.4 it is possible to observe the advantage of performing the flyby. Comparing the Characteristic Energy C_3 with the direct result obtained in GMAT, there is a significant reduction, even though the launch epoch for the Lunar gravity assist trajectory is performed 14 days after the ideal transfer. The value of the reduction is $\Delta C_3 = 1.17 km^2/s^2$.

When comparing the results of GMAT with the single shooting method, it is possible to observe a slight increase in the characteristic energy. This can be explained by the influence of the Earth's gravity shortly after the departure from Earth's SOI, not considered in the patched conic trajectories method and consequently not on the single shooting method.

3.6 Pruning phase - case 2

The case 2 is presented, but in a shorter way as in the previous case the details of the process are explained. So, in this part it is presented the pruning analysis results and graphs and their discussion.

In Figure 3.8, the velocity magnitude for the pruned orbits in case 2 are presented.

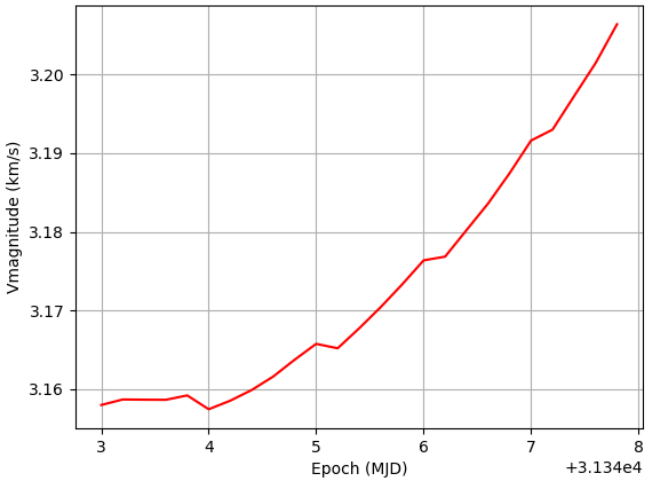


Figure 3.8: Velocity magnitude of the 1st pruning stage – Case 2

It is noticeable that the minimum velocity required for an interplanetary transfer occurs in the epoch $t_i = 31344$, as referenced in the pork chop analysis.

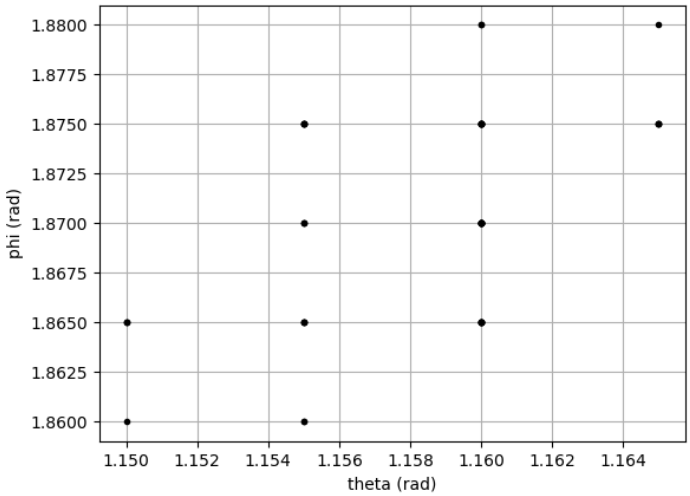


Figure 3.9: Spherical angles of the trajectories on the Moon's SOI on the 3rd stage of pruning – Case 2

In Figure 3.9, results for the spherical angles in the 3rd phase of the grid search are represented.

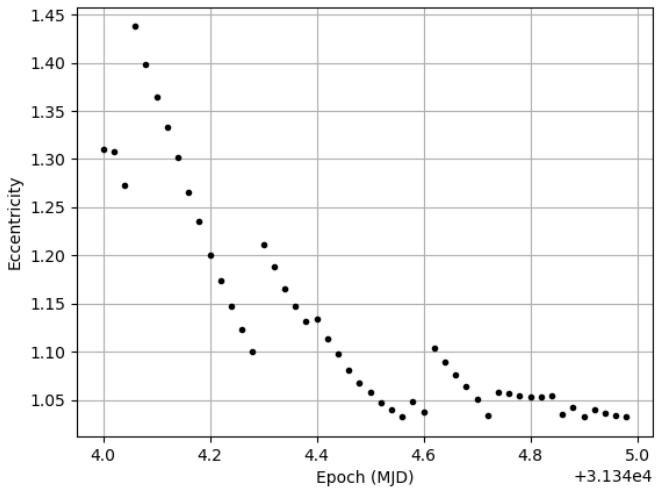


Figure 3.10: Eccentricity of the orbits relative to Earth before the flyby in the pruning 3rd stage – Case 2

The angle intervals used to obtain the results are $1.0 < \theta < 1.2$ and $1.7 < \varphi < 1.9$. The time interval is $31345 < t_i < 31346$.

From the figures 3.10 and 3.11, it is clear that the best results for a lower energy orbit can be achieved between $t_i = 31344.4$ and $t_i = 31344.6$, where both eccentricity and radius of periapsis are low, increasing the efficiency of the manoeuvre.

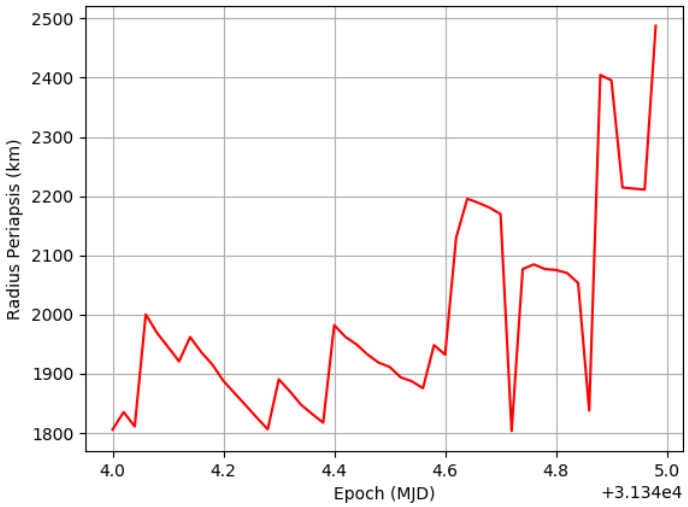


Figure 3.11: Radius of the Moon's flyby periapsis in pruning 3rd stage – Case 2

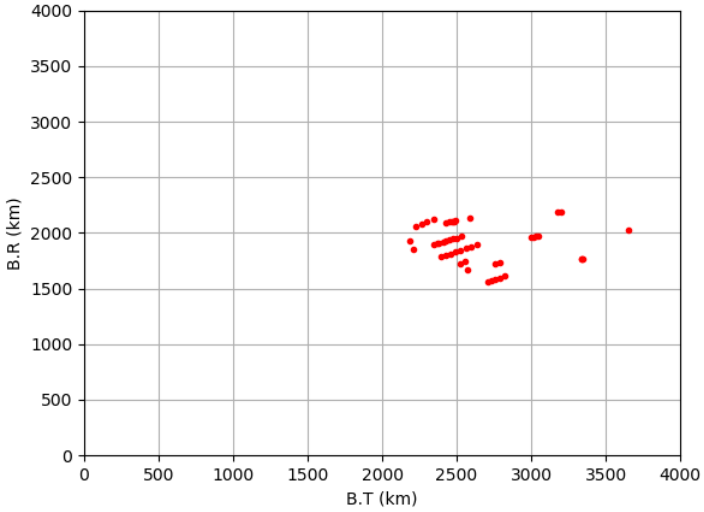


Figure 3.12 $B \cdot T$ and $B \cdot R$ parameters of the trajectories in pruning 3rd stage – Case 2

3.7 Single shooting method - case 2

In this case, the method is applied in the same way as in case 1. The initial position chosen to perform the trajectory is an arbitrary position from the same low Earth Keplerian orbit, but the true anomaly element is $\upsilon = 350^{o}$ and it is also arbitrarily chosen so that the starting position is adequate to target the Moon.

So, some of the results of the single shooting method are presented in Table 3.5.

Table 3.5: 4 example results of the Lunar gravity assist manoeuvre – Case 2

$t_0(MJD)$	**Periapsis Radius** (km)	**Velocity Magnitude** (km/s)	**Characteristic Energy** $(C_3)[km^2/s^2]$
31343.2483	2405	10.3127	5.7099
31343.0807	2164	10.3004	5.4577
31342.9223	1973	10.2899	5.2413
31342.8966	1945	10.2881	5.2033

The arrival epoch is the same as in the pork chop studies $t_f = 31629$.

The same relation between the decreasing radius of periapsis and decreasing characteristic energy is verifiable.

In this case it is evident that the characteristic energy is greatly reduced, comparing the best result with the direct transfer obtained, there is a reduction of $\Delta C_3 = 3.07 km^2/s^2$ using the single shooting.

The improved result is caused by the coincidence of the Moon being in a favourable position to perform the flyby in the same epoch as the most efficient direct transfer, even though the radius of periapsis is superior.

3.8 Graphical representation - case 2

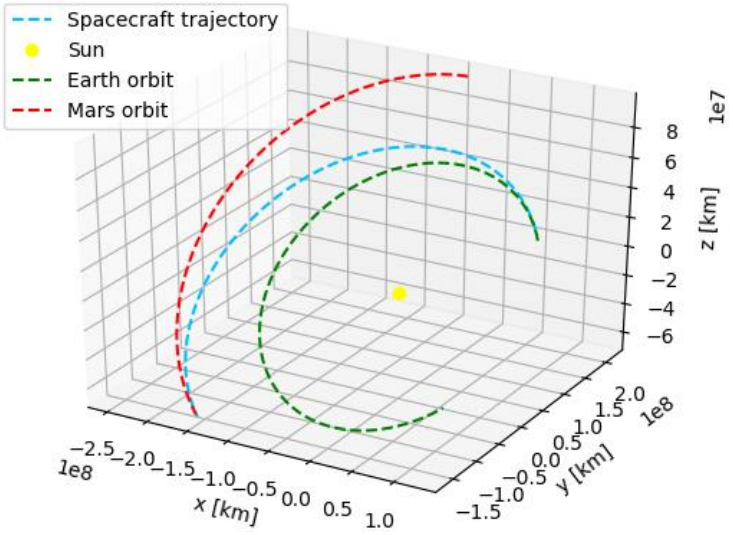


Figure 3.13: Graphical representation of the departure from Earth and the Lunar gravity assist -case 2

Figure 3.14: Graphical representation of the Interplanetary phase of the trajectory – case 2

In this subchapter the graphical representation of case 2 is presented.

3.9 GMAT simulation - case 2

The results from the GMAT simulation are presented in table 3.6:

Table 3.6- Result of the GMAT simulation – Case 2

$t_0(MJD)$	$t_f(MJD)$	r_{per} (km)	v_{mag} (km/s)	**GMAT Direct** $C_3[km^2/s^2]$	C_3 **GMAT LGA** $[km^2/s^2]$
31343.2483	31645.5052	2063	10.3451	9.2671	6.4245

In this case, there is a similar improvement as in the previous case of study. The characteristic energy has a reduction of $\Delta C_3 = 2.84 km^2/s^2$, which can be explained by the smaller radius of periapsis compared to the previous case and the fact that the Moon is in a very favourable position to perform the gravity assist manoeuvre.

It is also possible to verify that there is an increase of characteristic energy comparing to the single shooting result, due to the same implications explained in the previous case, such as the influence of the Earth on the trajectory parts that in the single-shooting method are not considered. Although all other bodies are considered in GMAT, their gravitational influence alters the final solution but it's not significant compared to Earth's gravity effect on the overall trajectory. Despite the setbacks, GMAT's algorithms managed to find a good trajectory.

3.10 Other results

In this subchapter, the results of the remaining epochs for the single shooting method (Table 3.7) and the GMAT simulation (Table 3.8) are shown:

Table 3.7: Results of the gravity assist manoeuvres with single-shooting method

$t_0(MJD)$	**Periapsis Radius** (km)	**Velocity Magnitude** (km/s)	**Direct** $(C_3)[km^2/s^2]$	**LGA** $(C_3)[km^2/s^2]$
22807.1038	1834	11.2297	8.9448	5.3130
29072.7259	1791	10.7490	15.5614	15.7081
29837.4236	1864	10.5509	12.8764	10.6237

Table 3.8: Results of the gravity assist in GMAT software

Launch window $t_0(MJD)$ **/ Mission**	$t_0(MJD)$	$t_f(MJD)$	r_{per} (km)	v_{mag} (km/s)	**Direct** $C_3[km^2/s^2]$	**LGA** C_3 $[km^2/s^2]$
MER-A	22807.3790	23011.5357	1800	11.2688	8.9448	6.2732
29085	29072.7520	29490.4874	1837	10.7891	16.3724	16.5742
29837	29837.9350	30240.3587	1794	10.5715	14.2287	11.6182

The results from the remaining launch windows where a gravity assist is performed are presented. Regarding the result for the MER-A mission, also performed in [5], there is a slight improvement of $\Delta C_3 = 0.08 km^2/s^2$. This can be explained by the lower radius of periapsis

than used in the mentioned work [5] $r_p = 1838\,km$. So, it is presumed that the result is identical. Similarly, to the mentioned work, there is a decrease of the characteristic energy of $\Delta C_3 = 2.67 km^2/s^2$. With this result, it is proven that the self-adaptive Levenberg-Marquardt algorithm is as effective as the Newton-Raphson as a differential corrector and can be effectively used for space mission design.

The result obtained in the 29085 launch window is inefficient. There is an increase in the characteristic energy of $\Delta C_3 = 0.20 km^2/s^2$ relative to the value of the direct transfer. This can be explained by the fact that during the favourable launch window, the Moon is essentially on the opposite position to the one needed to perform the gravity assist. There was an attempt to obtain an effective result both before and after the launch window. In both, epochs a trajectory was found, although the most efficient is the one presented in the Tables 3.7 and 3.8. However, the LGA trajectory after the launch window can be used as last resort, in case a mission scheduled for 29085 is unable to be launched in the exact day. A flyby can be performed approximately 14 days after the ideal direct launch window. The manoeuvre can prevent the cost increase of launching a mission outside of the direct efficient period.

In the 29837 launch window, an efficient trajectory was found. Compared to the values of the direct transfer, there is a significant decrease in the characteristic energy $\Delta C_3 = 2.61 km^2/s^2$.

The Table A.2 contains the detailed coordinates of all the injection manoeuvres performed with GMAT.

Figures 3.15 to 3.20 show the graphical representation of the results discussed in this subchapter.

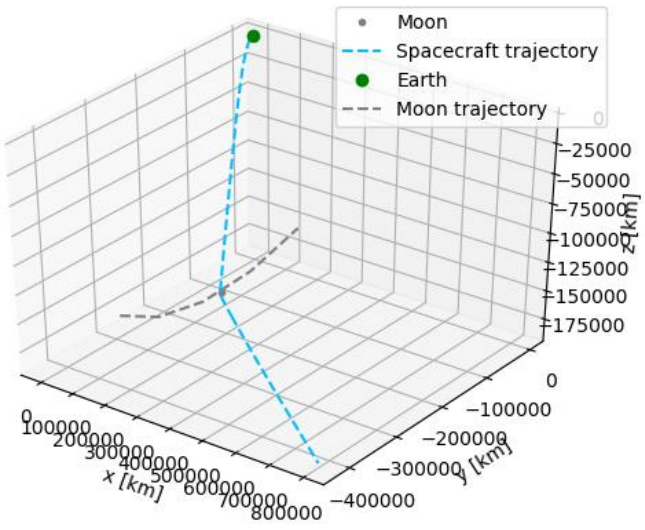


Figure 3.15: MER-A Earth departure

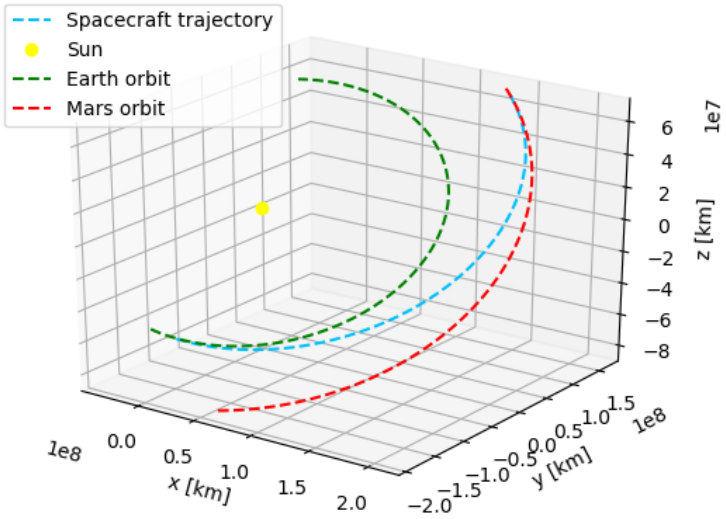


Figure 3.16: MER-A Interplanetary phase

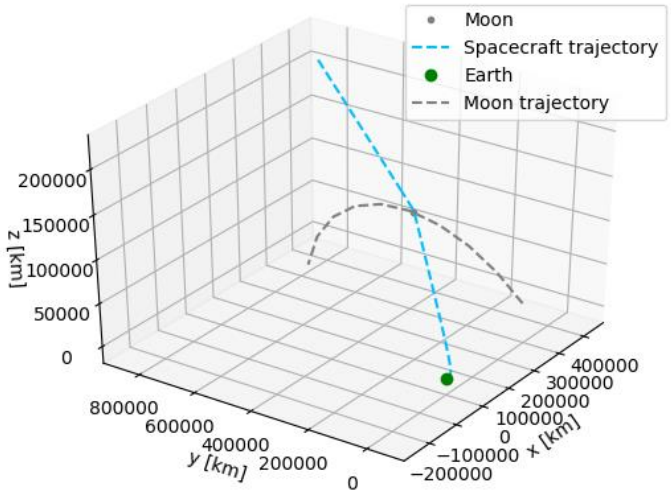


Figure 3.17: 29085 Earth departure

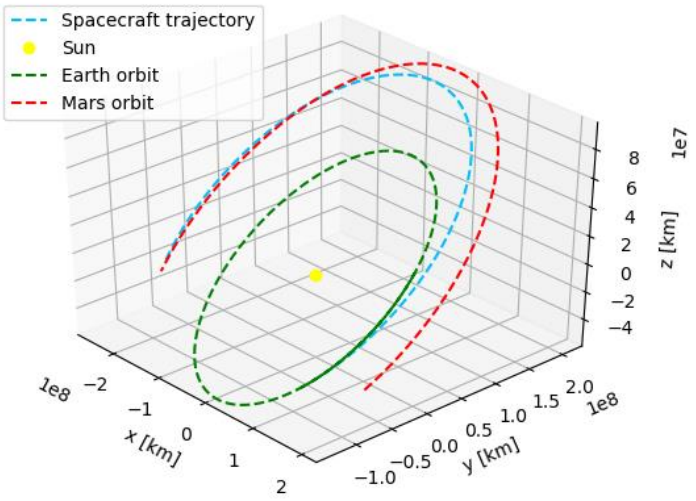


Figure 3.18: 29085 Interplanetary phase

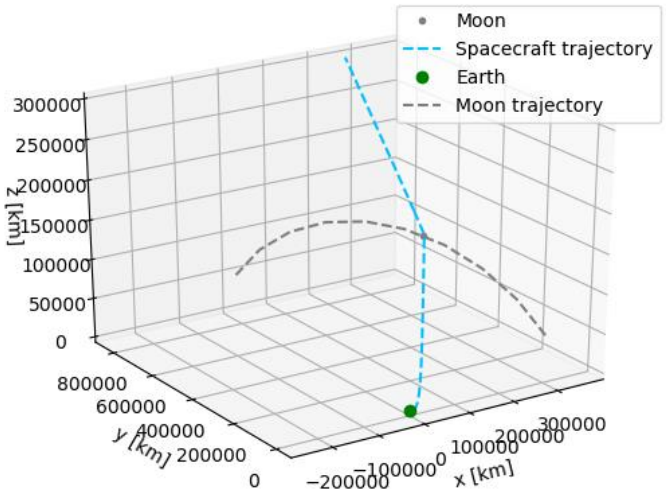


Figure 3.19: 29837 Earth departure

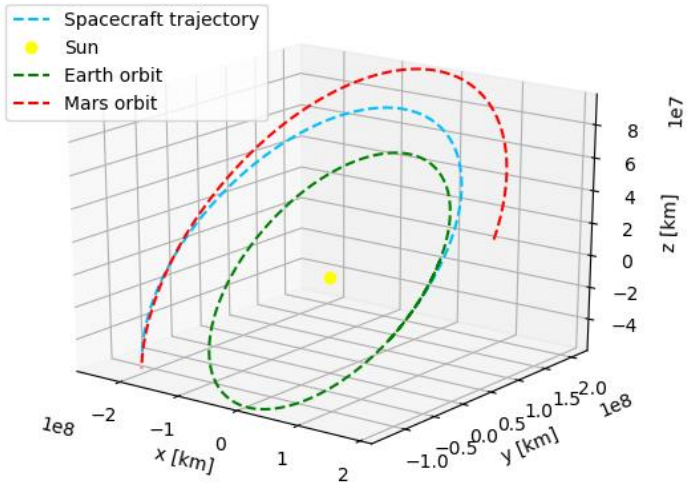


Figure 3.20: 29837 Interplanetary phase

Chapter 4

4 Conclusions

In this work, a study of trajectories to Mars with a Lunar gravity assist was made. The objective of this work was to find alternative routes to Mars that can increase the payload capacity or decrease the fuel needed of future Mars missions, as well as to analyse the potential of the self-adaptive Levenberg-Marquardt algorithm as a differential corrector.

The results were obtained using the two-body problem model, which is a simplistic but efficient method to design space mission trajectories. All results were validated using NASA's General Mission Analysis Tool (GMAT), the porch chop plots and the comparison between the direct transfers and the LGA transfers.

The proposed objectives for this work were achieved. The results include the validation of the differential corrector self-adaptive Levenberg-Marquardt algorithm with a trajectory previously studied, and the study of 4 future transfer windows. There was a solution found in all of them, and in 3 out of 4 of the Lunar gravity assist manoeuvres were effective, reducing the characteristic energy C_3 of the orbit. The results are compared with a previous study of the future Mars transfer windows and pork chop plots. It is safe to confirm that the self-Adaptive Levenberg-Marquardt algorithm is a powerful tool that can be applied to mission design, due to the fact that in all launch windows a trajectory with a Lunar gravity assist was found. In the simulations where the Lunar gravity assist performed better than the predicted direct trajectories, the decrease of orbit energy was from $1.17 km^2/s^2$ to $2.84 km^2/s^2$, which could mean a significant increase of the payload mass on most of the future missions to Mars.

The success of the results can be explained by the flexibility of the departure and arrival epochs, as well as the possible use of short or long trajectories, type I or type II, respectively. This allowed to choose the type of trajectory compatible with the lunar gravity assist manoeuvre.

The ineffectiveness of one of the studied trajectories can be explained by the fact that during the effective transfer window, the Moon is in the opposite position needed to perform the manoeuvre. So, to perform the flyby, the injection epoch needs to be approximately 14 days before or after the effective transfer window, thus increasing the characteristic energy of the trajectories.

This work proves that these types of trajectories should be considered and studied for future Mars, Venus and other missions. However, the trajectories can be considered riskier due to lower altitude flybys and trajectory correction errors can cause serious failures. This can be prevented with higher altitude flybys, that still give a significant energy increase, but have a better margin for failure.

However, there were some difficulties to translate the best results from the single-shooting method to GMAT, as the influence of bodies is not taken into account in the simplistic two-body problem model used, such as the influence of Earth's gravity in the interplanetary phase right after reaching the limit of Earth's SOI. The influence of the Earth's gravity in the flyby phase also produced changes in the GMAT trajectory, particularly because these small changes in the initial phases of the trajectory are crucial and cause massive deviation in the end of the flight path. These setbacks extended greatly the time needed to compute the trajectories in GMAT and were an inconvenience to achieve ideal results.

The main difficulties for this work can be summed up as the time of computation from GMAT, the single shooting and pruning methods. Even though the two-body problem is an acceptable simulation for real orbital mechanics, the translation of the results was difficult and time consuming.

For future work, better models can be used, to diminish the difficulty of translating from the two-body problem to a more realistic simulator. Future work suggestions also include more than one Lunar gravity assist for deep space missions, as well as Lunar gravity assist manoeuvres to other Solar System body missions or multiple Lunar flybys.

Bibliography

[1] Voyager to the Outer Planets and Into Interstellar Space. (2013). [PDF] NASA Jet Propulsion Laboratory. Available at: https://www.jpl.nasa.gov/news/fact_sheets/voyager.pdf [Accessed 11 Oct. 2019].

[2] Esa.int. (2019). The long trek. [online] Available at: http://www.esa.int/Science_Exploration/Space_Science/Rosetta/The_long_trek [Accessed 13 Oct. 2019].

[3] Yang, B., et al. (2018) Hierarchical approach for fast searching optimal launch opportunity in a wide range. Adv. Space Res., https://doi.org/10.1016/j.asr.2018.09.026

[4] Gmat.sourceforge.net. (2019). DifferentialCorrector. [online] Available at: http://gmat.sourceforge.net/docs/nightly/html/DifferentialCorrector.html [Accessed 7 Nov. 2019].

[5] Oliveira, R. (2017). Numerical Study of Earth-Mars trajectories with lunar gravity assist manoeuvres. Msc. Dissertation, Universidade da Beira Interior.

[6] Gil-Fernandez, J. (2005). ExoMars Alternative Escape Trajectories with Soyuz/Fregat. Annals of The New York Academy Of Sciences, 1065(1), 15-36. doi: 10.1196/annals.1370.011

[7] Vallado, D. and McClain, W. (2013). Fundamentals of astrodynamics and applications. Hawthorne (California): Microcosm Press.

[8] Battin, R. (2000). Introduction to the Mathematics and Methods of Astrodynamics. Reston: American Institute of Aeronautics and Astronautics.

[9] Esa.github.io. (2019). Multi revolutions Lambert Problem — pykep 2.4 documentation. [online] Available at: https://esa.github.io/pykep/examples/ex2.html [Accessed 7 Nov. 2019].

[10] Gmat.sourceforge.net. (2019). Chapter 8. Mars B-Plane Targeting. [online] Available at: http://gmat.sourceforge.net/docs/nightly/html/Mars_B_Plane_Targeting.html [Accessed 7 Nov. 2019].

[11] Cho, D., Chung, Y., & Bang, H. (2012). Trajectory correction maneuver design using an improved B-plane targeting method. Acta Astronautica, 72, 47-61. doi: 10.1016/j.actaastro.2011.11.009

[12] Fan, J., & Pan, J. (2005). Convergence Properties of a Self-adaptive Levenberg-Marquardt Algorithm Under Local Error Bound Condition. *Computational Optimization and Applications, 34*(1), 47-62. doi:10.1007/s10589-005-3074-z

[13] Yamashita, N., & Fukushima, M. (2001). On the Rate of Convergence of the Levenberg-Marquardt Method. *Topics in Numerical Analysis Computing Supplementa,* 239-249. doi:10.1007/978-3-7091-6217-0_18

[14]Fan, J., & Yuan, Y. (2005). On the Quadratic Convergence of the Levenberg-Marquardt Method without Nonsingularity Assumption. Computing, 74(1), 23-39. doi:10.1007/s00607-004-0083-1

[15] Dan, H., Yamashita, N., & Fukushima, M. (2002). Convergence Properties of the Inexact Levenberg-Marquardt Method under Local Error Bound Conditions. Optimization Methods and Software, 17(4), 605-626. doi:10.1080/1055678021000049345

[16] Hughes, S., Qureshi, R., Cooley, S., & Parker, J. (2014). Verification and Validation of the General Mission Analysis Tool (GMAT). AIAA/AAS Astrodynamics Specialist Conference. doi: 10.2514/6.2014-4151

[17]GMAT Math Spec. (2018). NASA.

Appendix A

Table A.1 Physical data of Earth, Moon, Mars and Sun

	Earth	**Moon**	**Mars**	**Sun**
Mass $[kg]$	5.9742×10^{24}	7.3483×10^{22}	6.4191×10^{23}	1.9885×10^{30}
$\mu[km^3/s^2]$	3.986×10^{5}	4902	4.305×10^{4}	1.3271×10^{11}
Period $[days]$	365.2564	27.3217	686.9801	–
$a[km]$	149.598×10^{6}	0.3844×10^{6}	227.939×10^{6}	–
$i[rad]$	0.000	0.0898	0.0323	–
e	0.0167	0.0549	0.0934	–
Equatorial radius $[km]$	6378	1738	3397	6.9570×10^{5}
SOI radius $[km]$	9.2466×10^{5}	0.6617×10^{5}	5.77278×10^{5}	–

Table A.2 Injection ephemeris in Earth MJ200Eq reference frame in GMAT

$t_0(MJD)$	$r_I\ [km]$	$r_J\ [km]$	$r_K\ [km]$	$v_I\ [km/s]$	$v_J\ [km/s]$	$v_K\ [km/s]$
22807.3786	-5950.7395	-2303.8716	-1701.4774	0.55861	-9.49263	-6.04681
29072.7519	1397.0293	-7743.3369	-1363.0928	10.42125	1.27715	2.48414
29837.9350	3989.0782	-6785.3544	-1193.1074	9.38593	3.00576	3.82442
30603.1217	3988.7141	-6785.5135	-1192.3706	9.01392	2.80708	4.56866
31343.2483	7805.9753	-1346.8180	-234.4425	5.10027	7.84662	4.40887

Appendix B

SELF-ADAPTIVE LEVENBERG-MARQUARDT IMPLEMENTATION ON PYTHON

Flávio Rosa
[1]*Aerospace Sciences Department, Universidade da Beira Interior,*
Covilhã, Portugal
[1]*flavio.rosa666@gmail.com*
(To be submitted)

Abstract. The self-adaptive Levenberg-Marquardt algorithm is an iterative method for finding the solution of a $F(x) = 0$ non-linear set of equations or their minimum. It was proposed as an improvement for the Levenberg-Marquardt (LM) method and modified LM method. This work consists of a brief presentation of the LM method, and the adaptations made for the self-adaptive LM method, and its implementation on Python 3.7. The final chapter includes the test and example application of the algorithm.
.

Keywords: *Self-adaptive Levenberg-Marquardt algorithm, non-linear equations, Python 3.7*

1. Introduction

The self-adaptive Levenberg-Marquardt (SLM) algorithm is an iterative method used to find the minimum of a system of non-linear equations or to find its solution as in [1].

$$F_n(x_1, x_2, x_{,3} \ldots, x_i) = 0 \tag{1}$$

This method has been used for various applications through numerous fields of study.
In this paper, we will do a short review of the origin of [2][3], which is itself a modification of the Newton-Gauss method, followed by a description of the SLM method. This description includes the insertion of new parameters compared with the LM algorithm, some also presented in [4] such as α and δ. These parameters help to update the damping factor μ_k (Levenberg-Marquardt parameter) in order to adjust the size of the step depending on the distance from the solution. This technique also includes the addition of the predicted reduction and an actual reduction, used to calculate their ratio r_k to update the value of α accordingly.
The SLM brings a new function q(r) with of purpose of updating the parameter α at a variable rate, allowing better numerical results.
Finally, the last part of the article it's an implementation of the SLM algorithm in the open source programming language Python 3.7.
The algorithm is tested with some example functions found in [5], and it is also applied in a set of flight equations from a flight dynamics model, to estimate the equilibrium of the aircraft on a levelled flight.
Note that this work does not intend to fully explain in detail the methods listed, and it's suggested that the reader checks the original articles [1][2][3][4].

2. Levenberg-Marquardt Algorithm

In this chapter the Levenberg-Marquardt algorithm is briefly presented and summed up to prelude the rest of the work.

2.1. Problem

From the problem

$$F_n(x_1, x_2, \ldots, x_i) = 0$$
$$F_n(x): R^n \rightarrow R^n \quad (2)$$

corresponds to *n* non-linear equations dependent on *I* variables, representing a system with

$$F_n(X) = 0 \quad (3)$$

X^T is the solution vector of said system.

2.2. Levenberg-Marquardt parameters

We consider the solution to be

$$F_n(\boldsymbol{X}) = F_n(\boldsymbol{X}_k) + \boldsymbol{d}_k \quad (4)$$

Where $\boldsymbol{d}_k$ is a correction factor calculated by:

$$\boldsymbol{d}_k = -(\boldsymbol{J}_k^T \boldsymbol{J}_k + \mu_k \boldsymbol{I})^{-1} \boldsymbol{J}_k^T F_k \quad (5)$$

$\boldsymbol{J}_k$ being the Jacobian Matrix of the system, μ_k is the damping parameter, $\boldsymbol{I}$ the identity matrix and $F_k = F(\boldsymbol{X}_k)$.

The damping parameter is a positive multiplier and was introduced in this method with the purpose of reducing the impact of the singularity of the Jacobi Matrix $\boldsymbol{J}_k$. This parameter will modify the diagonal values of $\boldsymbol{J}_k^T \boldsymbol{J}_k$, it can be increased greatly in order to reach a $\boldsymbol{d}_k$ that decreases the error $e = \left\| \boldsymbol{X} - \hat{\boldsymbol{X}} \right\|$, usually if the initial guesses are not in the neighbourhood of the solution. According to the evolution of the iterations and the μ_k update over each iteration, it will assume smaller values as e decreases and $\boldsymbol{X}$ approaches the desired values.

After obtaining the parameter $\boldsymbol{d}_k$ this one is used to obtain the new trial vector

$$\boldsymbol{X}_{k+1} = \boldsymbol{X}_k + \boldsymbol{d}_k \quad (6)$$

Although the Levenberg-Marquardt method can overcome the singularity of the Jacobi matrix in cases where the initial guess is close enough to the system solution, some problems arise when non-singularity is too strong [1].

3. Self-adaptive Levenberg-Marquardt Algorithm

3.1. Levenberg-Marquardt modifications

Over the last two decades, the LM method was improved several times, including the adding of new parameters, mainly to improve the updating of the damping parameter μ_k.

The first changes in the method were introduced in [6], choosing the LM parameter as $\mu_k = \|F_k\|^2$. Although this value could be either too large and not converge quickly or could be too small and lose its role according to [1].

Following this modification [7] proposed $\mu_k = \|F_k\|$ and concluded it has a quadratic convergence under the local error bound condition. So, the next modification to the value of this parameter came with [8], who added a positive constant $\delta \in (0,2)$: $\mu_k = \|F_k\|^\delta$. In this case the

constant causes the method to converge to the solution super linearly when $\delta \in (0,2)$ and quadratically for $\delta = 2$.
However, the last LM parameter which [1] and this paper were based, brings another variable to determine the size of the step: $\mu_k = \alpha_k \|F_k\|$. α_k is also updated along the iteration process according to the value of r_k, the ratio between the actual reduction and the predicted reduction.

$$r_k = \frac{Ared_k}{Pred_k} \tag{7}$$

$$Pred_k = \varphi_k(0) - \varphi_k(d_k) \tag{8}$$

$$Ared_k = \|F_k\|^2 - \|F(x_k + d_k)\|^2 \tag{9}$$

$$\varphi(0) = \|F_k\|^2 \tag{10}$$

$$\varphi(d_k) = \|F_k + J_k d_k\|^2 \tag{11}$$

In the previous modified LM algorithm α_k assumes several different values depending on *p0*, *p1* and *p2* parameters given in [4].

3.2 Self-adaptive Levenberg-Marquardt parameters

In the studied algorithm, the chosen value for the LM parameter is

$$\mu_k = \alpha_k \|F_k\|^\delta \tag{12}$$

$$\alpha_{k+1} = \alpha_k q_k(r_k) \tag{13}$$

a new $q(r_k)$ continuous nonnegative function appears, where $r = r_k$. Now, α_k is changed at each iteration, depending on the ratio between reductions. So r_k controls the step of the convergence by updating constantly α_k and consequently the LM parameter.
q(r) can be given by:

$$q(r) = \max\left\{\frac{1}{4}, 1 - 2(2r-1)^3\right\} \tag{14}$$

or

$$q(r) = \max\left\{\frac{1}{4}, 1 - (r - \frac{1}{2})^{\frac{1}{3}}\right\} \tag{15}$$

4. Algorithm

The algorithm presented in [1] starts with the calculation of the step $\boldsymbol{d}_k$, given the values:
$x_1 \in R^n, 0 < \delta \le 2, \alpha_1 > m > 0, 0 \le p_0 < \bar{r} < 1, k+=1$ with *m* as a positive constant to set the limit of the step, avoiding its increase near the result.

1. $(J_k^T J_k + \alpha_k \|F_k\|^\delta I) d_k = -J_k^T F_k$

The next phase is to find the reduction ratio and predict the next x_k values:

2. $r_k = \frac{Ared_k}{Pred_k}$

$$x_{k+1} = \begin{cases} x_k + d_k \; if \quad r_k > p_0, \\ x_k, \quad r_k \le p_0 \end{cases}$$

The last step is to update the parameter α_k:

3. $\alpha_{k+1} = \max\{m, \alpha_k q(r_k)\}$

Finally, go back to step 2 and repeat until $\left\| J_k^T F_k \right\| = 0$ or *k*=100.

In this algorithm the *m* constant [1] is the minimum limit for the LM parameter in order to avoid a larger step than desired near the solution.

5. Algorithm test

This chapter contains the test of the algorithm, including 6 test functions and the flight dynamics model application of a STOL (Short Take-Off and Landing) aircraft to calculate its equilibrium on a levelled flight.

5.1. Test functions

The algorithm mentioned earlier was written in the open-source programming language Python3.7 and was experimented by multiple test functions found in [5].

Constants chosen: $p_0 = 0.0001, \alpha_0 = 0.0001, m = 10^{-8}, \delta = 1$

Test function 1:

n=2, *m*=2

$f_1(x) = 10(x_2 - x_1^2)$

$f_2(x) = 1 - x_1$

$x_0 = (-1.2,1)$

$f = 0 \quad at \quad (1,1)$

Results: The algorithm converged to the solution $f = 0$ after 38 iterations.

Test function 2:

n=2, *m*=2

$f_1(x) = -13 + x_1 + ((5 - x_2)x_2 - 2)x_2$

$f_2(x) = -29 + x_1 + ((x_2 + 1)x_2 - 14)x_2$

$x_0 = (0.5,-2) \, or \quad x_0 = (1,2)$

$f = 0 \quad at \quad (5,4)$

$f = 48.9842 \ldots at \quad (11.41\ldots,-0.8968\ldots)$

Results: The algorithm converged to the solution $f = 0$ for $x_0 = (1,2)$ after 17 iterations, and to the solution $f = 48.9842$ for $x_0 = (0.5,-2)$ after 29 iterations.

Test function 5:

n=2, *m*=3

$f_1(x) = 1.5 - x_1(1 - x_2^1)$

$f_2(x) = 2.25 - x_1(1 - x_2^2)$

$f_3(x) = 2.625 - x_1(1 - x_2^3)$

$x_0 = (1,1)$

$f = 0 \quad at \quad (3,0.5)$

Results: The algorithm converged to the solution $f = 0$ for $x_0 = (1,2)$ after 27 iterations.

Test function 13:

n=4, m=4

$f_1(x) = x_1 + 10x_2)$

$f_2(x) = 5^{1/2}(x_3 - x_4)$

$f_3(x) = (x_2 - 2x_3)^2$

$f_4(x) = 10^{1/2}(x_1 - x_4)^2$

$x_0 = (3,-1,0,1)$

$f = 0 \quad at \quad (0,0,0,0)$

Results: The algorithm converged to the solution $f = 0$ for $x_0 = (0,0,0,0)$ after 28 iterations.

5.2. STOL aircraft equilibrium calculation

For the second test of this algorithm, it was used the following flight dynamics model equations, for longitudinal model:

$$\dot{u} = \frac{1}{m}(0.5\rho S \frac{u^2}{\cos^2\alpha}(C_L.\sin\alpha - C_D.\cos\alpha) + \delta_T T_{\max}\cos\varepsilon_T) - g\sin\theta - qu\tan\alpha$$

$$\dot{\alpha} = q + \frac{g\cos\alpha}{u}\cos(\theta - \alpha) - \frac{\cos\alpha}{mu}(L - \delta_T T_{\max}\sin\varepsilon_T.\cos\alpha + \delta_T T_{\max}.\cos\varepsilon_T.\sin\alpha)$$

$$\dot{q} = \frac{\rho u^2 S\bar{c}C_m}{2I_y\cos^2\alpha}$$

$$\dot{\theta} = q$$

And for the lateral-directional model:

$$\dot{v} = -\frac{QS}{m}(C_D\sin\beta - C_y\cos\beta) + g\cos\theta_0.\sin\phi - ru_0 + pw_0$$

$$\dot{p} = \frac{QSb}{I_x I_z - I_{xz}^2}(I_z C_l + I_{xz}C_n)$$

$$\dot{r} = \frac{QSb}{I_x I_z - I_{xz}^2}(I_x C_n + I_{xz}C_l)$$

$$\dot{\phi} = p + r\cos\phi.\tan\theta_0$$

Where $C_L, C_D, C_y, C_l, C_m, C_n,$ are the stability and control derivatives, m is the aircraft mass, Q is the dynamic pressure, S is the wing surface, b is the wing span, u is the horizontal speed, δ_T is the throttle fraction, $T_{\max}$ is the maximum thrust, ε_T is the angle between thrust vector and

aircraft's longitudinal axis, g is Earth's gravitational acceleration constant, q, p, r are the rate of manoeuvres, I_x, I_y, I_z, I_{xz} are the moments of inertia and ϕ is the roll angle.

The stability and control derivatives are:

$$C_L = C_{L_0} + C_{L_\alpha}\alpha + \frac{\bar{c}}{2V}(C_{L\dot{\alpha}}\dot{\alpha} + C_{L_q}q) + C_{L_{\delta_e}}\delta_e$$

$$C_D = C_{D_0} + C_{D_\alpha}\alpha + \frac{\bar{c}}{2V}(C_{D\dot{\alpha}}\dot{\alpha} + C_{D_q}q) + C_{D_{\delta_e}}\delta_e$$

$$C_y = C_{y_\beta} + C_{y_{\delta_a}}\delta_a + C_{y_{\delta_r}}\delta_r$$

$$C_l = C_{l_\beta}\beta + \frac{b}{2V}(C_{l_p}p + C_{l_r}r) + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r$$

$$C_m = C_{m_0} + C_{m_\alpha}\alpha + \frac{\bar{c}}{2V}(C_{m_{\dot{\alpha}}}\dot{\alpha} + C_{m_q}q) + C_{m_{\delta_e}}\delta_e$$

$$C_n = C_{n_\beta}\beta + \frac{b}{2V}(C_{n_p}p + C_{n_r}r) + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r$$

The variables considered for the longitudinal model are the angle of attack α, the pitch angle θ and the elevator's deflection δ_e.

The variables considered for the lateral-directional model are the yaw angle β, the ailerons and rudder deflections, δ_a, δ_r respectively.

The values for the stability and control derivatives $(C_{L_0}, C_{L_\alpha}, \ldots, C_{n_{\delta_r}})$ are:

$$C_{L_0} = 0.3, C_{L_\alpha} = 5.24, C_{L\dot{\alpha}} = 1.33, C_{L_q} = 7.83, C_{L_{\delta_e}} = 0.465$$

$$C_{D_0} = 0.036, C_{D_\alpha} = 0.67, C_{D\dot{\alpha}} = 0, C_{D_q} = 0, C_{D_{\delta_e}} = 0$$

$$C_{y_\beta} = -0.362, C_{y_{\delta_a}} = -0.233, C_{y_{\delta_r}} = 0$$

$$C_{l_\beta} = -0.125, C_{l_p} = -0.53, C_{l_r} = 0.113, C_{l_{\delta_a}} = 0.20, C_{l_{\delta_r}} = -0.024$$

$$C_{m_0} = 0, C_{m_\alpha} = -0.78, C_{m_{\dot{\alpha}}} = -6.05, C_{m_q} = -35.6, C_{m_{\delta_e}} = -2.12$$

$$C_{n_\beta} = 0.101, C_{n_p} = -0.037, C_{n_r} = -0.171, C_{n_{\delta_a}} = 0, C_{n_{\delta_r}} = 0.107$$

Note: All derivatives are in radians.

The data correspond to a levelled flight of 10000 ft (3048 m), and $M = 0.37$.

The remaining aircraft data are:

$W = 40000\ lbs$

$I_x = 273000\ slug - ft^2$

$I_y = 215000\ slug - ft^2$

$I_z = 447000\ slug - ft^2$

$I_{xz} = 0$

$S = 945\ ft^2$

$b = 96\ ft$

$\bar{c} = 10.1\ ft$

5.3. STOL aircraft equilibrium calculation results

In this subchapter, the results of the aircraft's equilibrium calculation are presented.

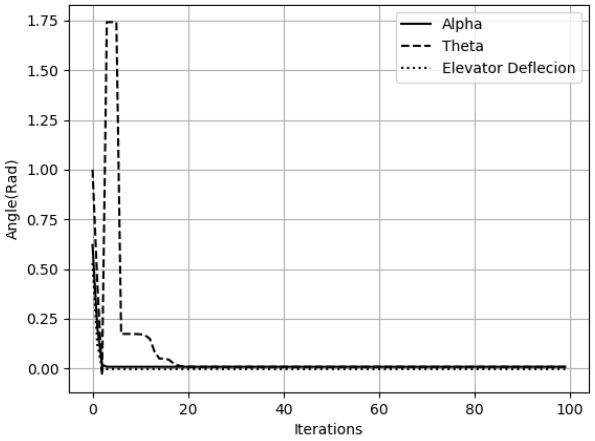


Figure 1. Longitudinal model convergence

The starting conditions for the longitudinal model were $x_0 = (1,1,1)$

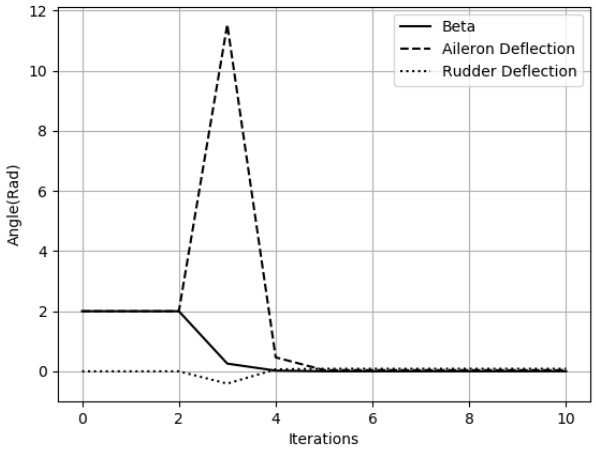


Figure 2. Lateral-directional model convergence

The starting conditions for the lateral-directional model were $x_0 = (2,2,0)$

From the longitudinal model graphic observation, we can see that it converges considerably fast, however it continues to iterate because the function reaches a minimum in the region close to zero. The final values are:

$x_f = (0.00788, 0.00786, -0.00351)$

$\alpha = 0.00788\ rad$

$\theta = 0.00786\ rad$

$\delta_e = -0.00351\ rad$

$F_1 = \dot{u} = -2.83e-04 rad/s$

$F_2 = \dot{\alpha} = -1.07e-01 rad/s$

$F_3 = \dot{q} = 2.95e-05 rad/s$

From the lateral-directional model graphic, we observe that it converges rapidly to a function zero, where the final values are:

$x_f = (0, 0.039, 0.084)$

$\beta = 0$

$\delta_a = 0.039\, rad$

$\delta_r = 0.084\, rad$

$F = \dot{v} = \dot{p} = \dot{r} = \dot{\phi} = 0$

6. Conclusions

The self-adaptive Levenberg-Marquardt algorithm was implemented successfully. Its development over the years significantly improved its converging and singular matrix problem solving abilities, resulting in a relatively simple, working and effective algorithm with various applications, one of these in the aerospace technology. This application can and should be further explored and has a lot of room to improve.

References

[1] Fan, J., & Pan, J. (2005). Convergence Properties of a Self-adaptive Levenberg-Marquardt Algorithm Under Local Error Bound Condition. *Computational Optimization and Applications, 34*(1), 47-62. doi:10.1007/s10589-005-3074-z

[2] Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics, 2*(2), 164-168. doi:10.1090/qam/10666

[3] Marquardt, D. W. (1963). An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics, 11*(2), 431-441. doi:10.1137/0111030

[4] Fan, J. (2003). A modified Levenberg-Marquardt algorithm for singular system of nonlinear equations. *Journal of Computational Mathematics, 21*(5), 625-636. Retrieved from http://www.jstor.org/stable/43693105

[5] Moré, J. J., Garbow, B. S., & Hillstrom, K. E. (1981). Testing Unconstrained Optimization Software. *ACM Transactions on Mathematical Software, 7*(1), 17-41. doi:10.1145/355934.355936

[6] Yamashita, N., & Fukushima, M. (2001). On the Rate of Convergence of the Levenberg-Marquardt Method. *Topics in Numerical Analysis Computing Supplementa*, 239-249. doi:10.1007/978-3-7091-6217-0_18

[7] Fan, J., & Yuan, Y. (2005). On the Quadratic Convergence of the Levenberg-Marquardt Method without Nonsingularity Assumption. *Computing, 74*(1), 23-39. doi:10.1007/s00607-004-0083-1

[8] Dan, H., Yamashita, N., & Fukushima, M. (2002). Convergence Properties of the Inexact Levenberg-Marquardt Method under Local Error Bound Conditions. *Optimization Methods and Software, 17*(4), 605-626. doi:10.1080/1055678021000049345

Appendix C

Earth-Mars trajectories with lunar gravity assist study using the self-adaptive Levenberg-Marquardt Algorithm

Flávio Rosa[1]

[1]Department of Aerospace Sciences, Universidade da Beira Interior
Calçada Fonte do Lameiro 6, 6200-358 Covilhã, Portugal

(To be submitted)

Abstract. In this work it is performed a numerical study of interplanetary trajectories between Earth and Mars, using the Moon to carry out a lunar gravity assist manoeuvre. The obtained results are compared with values from a direct transfer achieved with the same methods and with estimated values for the future launch windows for an interplanetary transfer between the Earth and Mars. The results are obtained with the two body problem and with the open source NASA's software GMAT. The self-adaptive Levenberg-Marquardt algorithm developed for this work in the programming language Python 3.6 is tested and used as a differential corrector to obtain the trajectories for the two body problem. The results demonstrate that the self-adaptive Levenberg-Marquardt algorithm is reliable for mission design, a lunar gravity assist can be executed in all situations studied and only in a few cases it is not viable.

Keywords: Interplanetary transfer; gravity assist; self-adaptive Levenberg-Marquardt; Mars

1. Introduction

Generally, to reach Mars a direct transfer is used, which consists of accelerating while in the parking orbit, that can be a low Earth orbit and can vary from 400 km to 1500 km of altitude. For a Mars direct transfer, this maneuver puts the spacecraft on escape velocity, in order to achieve a solar orbit that intercepts Mars orbit. The duration of an efficient journey usually is no less than 180 and not more than 450 days, depending on the position, inclination and proximity of the planets, this time can be different to perform the most efficient trajectory. In the cruise phase, between the planets, trajectory correction maneuvers can be performed, or TCM's. Finally, the orbit insertion burn is performed near the target body, to ensure that the spacecraft enters in orbit. Depending on the type of mission and its objective, the vehicle can perform more burns to achieve the desired orbit, or to achieve the trajectory to enter the atmosphere and initiate the descent to the surface landing site. On some cases, together with the retrograde burn to slow down the spacecraft, aerobraking can be used. This technique consists of using the outer layers of the atmosphere, so that atmospheric drag helps reducing the spacecraft's velocity and thus saving fuel.

As Mars exploration interest keeps growing in worldwide space agencies, the need for cheaper and alternative means of getting to the red planet are important. Same means can be used to increase the payload capacity and allow a vaster scientific payload.

Gravity assists are widely used since the first decades of deep space exploration, mainly for space missions to the outer or inner bodies of the Solar system and are an efficient method to reduce fuel mass and increase the payload capacity, although a gravity assist can extend the mission time. Several missions in space exploration history used this method to achieve their goals that otherwise wouldn't be possible with the available technology.

In order to find the most efficient launch dates and times of flight for the analysis of the future launch windows, the epochs from Yang, B. *et al.* (2018) pork chop plots are used. The self-adaptive Levenberg-Marquardt algorithm (Fan, J., & Pan, J. 2005), mentioned in this paper as SLM, is tested and used as a differential corrector to converge the trajectory describing function values to the reference values. To test this method, the mission of the MER-A Spirit Rover is analyzed. In a previous work (Oliveira, R. 2017 Msc. Dissertation), it was proven that a trajectory with a Lunar gravity assist (LGA) is possible and efficient, so the values obtained are compared to this work to check the utility of the SLM as a differential corrector. The launch windows analyzed span from 2020 to 2026. In these six years, there are four possible launch opportunities, one every two years. According to a previous study (Gil-Fernandez, J. 2005), performed for the ExoMars mission, a Lunar gravity assist can increase the mass of the vehicle injected to Mars between 9% and 10.5%.
The objectives of this work is to test and use the self-Adaptive Levenberg-Marquardt algorithm to find a way to reduce the fuel mass or increasing the payload capacity of a future mission to Mars, by taking advantage of the Moon's gravity and its favorable positions to perform a gravity assist and decrease the initial orbit energy to reach its target. The results obtained with the algorithm using the two body problem are then verified and validated in GMAT for a more realistic approach.
The software used to perform this analysis was the open source programming language Python 3.6 and NASA's General Mission Analysis Tool (GMAT) 2018 version.

2. Models and Algorithms

In this chapter, the astrodynamics models and algorithms used for this analysis are described. Note that the two-body propagation was mainly used for this analysis, being GMAT used to conclude and obtain the result, since it describes more accurately the motion of all bodies in the Solar System.

2.1 Keplerian and Cartesian coordinates

The Keplerian coordinate system defines the orbit of the spacecraft through the classical orbital elements: *a*, the semi-major axis, *e*, the eccentricity, *i*, inclination, Ω, the right ascension of the ascending node, ω, the argument of periapsis and υ, the true anomaly. The Cartesian coordinate system is inertial, the orbit is given also by six elements, the r_I, r_J, r_K and v_I, v_J, v_K components. The I axis is pointed towards the vernal equinox, J axis is 90° east in the equatorial plane and K axis extends towards the north pole of Earth. It is important to transform the coordinates, according to the calculations that best suit the models used, so the transformation equations are also represented in Vallado, D. and McClain, W. (2013):

Cartesian to Keplerian $(\vec{r}, \vec{v} \Rightarrow a, e, i, \Omega, \omega, \upsilon)$

Keplerian to Cartesian $(a, e, i, \Omega, \omega, \upsilon \Rightarrow \vec{r}, \vec{v})$

2.2 Patched conic trajectories

The method consists of an interplanetary transfer be approximated by several arcs, and in each one of these the motion of the spacecraft is described by the two-body problem, considering only the influence of the main body on the vehicle.
In this case, with the application of the patched conic trajectories method, the trajectory will be divided in 5 phases:

- The arc from Earth to the Moon's SOI, considering a hyperbolic trajectory relative to Earth;
- The arc that describe the Moon's flyby, considering a hyperbolic trajectory relative to the Moon;

- The arc from the Moon's SOI to the end of Earth's SOI, also hyperbolic;
- The arc of the interplanetary transfer, between Earth and Mars, described by an elliptic orbit relative to the Sun;
- Finally, the Mars arrival, described by a hyperbola relative to Mars.

2.3 Kepler propagation

This method predicts with a reasonable accuracy the position and velocity vectors on any moment of the orbit, given the initial position, velocity vectors and the time between the initial position and the position to be calculated.
The universal variables method of Kepler propagation is one of the methods to solve Kepler's problem, but unlike other methods mentioned in Vallado, D. and McClain, W. (2013) and Battin, R. (2000), this method allows to solve the problem in all possible conic sections so, in this case it is the best option.
The method is based on the use of f and g functions and the universal variables χ, ψ, c_2 and c_3. r_0 is the norm of the initial position vector, r is the norm of the final position, calculated in the algorithm, and Δt is the time of flight between positions.

Kepler Propagation $(\vec{r}_0, \vec{v}_0, \Delta t \Rightarrow \vec{r}, \vec{v})$**:**

$$f = 1 - \frac{\chi_n^2}{r_0} c_2 \tag{1}$$

$$\dot{f} = \frac{\sqrt{\mu}}{r \cdot r_0} \chi_n (\psi c_3 - 1) \tag{2}$$

$$g = \Delta t - \frac{\chi_n^3}{\sqrt{\mu}} c_3 \tag{3}$$

$$\dot{g} = 1 - \frac{\chi_n^2}{r} c_2 \tag{4}$$

2.4 Earth, Moon and Mars coordinates

The position and velocities of the celestial bodies were based on data from GMAT, as it is possible to retrieve report files with the planets or any other celestial bodies data. The report files are in the MJD time and present cartesian state vectors in each 1/1000 of a day approximately. If the files were used as described, the amount of time and computation needed to search for the state vector several times per iteration would be immense. So instead of this method, the report file was reduced to one state vector per day, this state vector is used to calculate the position and velocity of the body, using the Kepler Propagation, according to the Δt between the time in the report file for that day, and the time of the state vector to determine.

Kepler Propagation $(\vec{r}_{ref}, \vec{v}_{ref}, t - t_{ref} \Rightarrow \vec{r}, \vec{v})$

$\vec{r}_{ref}, \vec{v}_{ref}, t_{ref}$ are the position vector, velocity vector and time reference given by the GMAT report file.

2.5 Lambert's problem

The Lambert's problem calculates an initial velocity vector to connect any given points in space, given the time of flight but the orbit is unknown. It is known for interplanetary mission design and to construct the pork chop plots.

The method used for this work, is the Lambert's problem function from the library pykep, which is a scientific library for Python, developed by the European Space Agency for astrodynamics research (Esa.github.io. 2019).

Lambert's Problem $(\vec{r}_0, \vec{r}_f, \Delta t \Rightarrow \vec{v}_0, \vec{v}_f)$

This function is used to estimate initial velocity vectors that connects the initial starting points of the Moon's SOI, during the pruning phase, and Mars position in the arrival epoch. It is also used to calculate the first velocity vector, in the Earth's parking orbit, that connects this point and the point in the B-plane given by the pruning phase.

2.6 B-plane targeting

The B-plane is the planar coordinate system that helps targeting the Moon. It is used to aim for a specific point to perform the gravity assist. It can be seen as an attached target to the body, and it is always perpendicular to the incoming asymptote of the spacecraft approaching the body. Generally, the targeting is made in mid-course in a TCM, but in this case it is made in the injection burn. The $B \cdot T$ and $B \cdot R$ parameters represent the point where the vehicle pierces the B-plane. It is possible to obtain the $B \cdot T$ and $B \cdot R$ coordinates from the $\vec{r}$ and $\vec{v}$ coordinates (Vallado, D. and McClain, W. 2013), (Cho, D., Chung, Y., & Bang, H. 2012).

2.7 Self-adaptive Levenberg-Marquardt

The self-adaptive Levenberg-Marquardt (SLM) (Fan, J., & Pan, J. 2005) algorithm is an iterative method used to find the minimum of a system of non-linear equations or its solution. This method has been used for various applications through numerous fields of study, but in this work is used as differential corrector, in order to minimize the difference between the target values and the calculated trajectories values.
We consider the solution to be:

$$F_n(\boldsymbol{X}) = F_n(\boldsymbol{X}_k) + \boldsymbol{d}_k \tag{5}$$

Where $\boldsymbol{d}_k$ is a correction factor calculated in:

$$\boldsymbol{d}_k = -(\boldsymbol{J}_k^T \boldsymbol{J}_k + \mu_k \boldsymbol{I})^{-1} \boldsymbol{J}_k^T F_k \tag{6}$$

$\boldsymbol{J}_k$ being the Jacobian Matrix of the system, μ_k is the damping parameter, $\boldsymbol{I}$ the identity matrix and $F_k = F(\boldsymbol{X}_k)$. After the modifications mentioned above, the damping parameter is considered:

The damping parameter μ_k is a positive multiplier and was introduced in this method with the purpose of reducing the impact of the singularity of the Jacobi Matrix $\boldsymbol{J}_k$. α_k is also updated along the iteration process according to the value of r_k, the ratio between the actual reduction and the predicted reduction.

SLM: $(F_n(\boldsymbol{X}), \boldsymbol{X}_I, F_{objective} \Rightarrow \boldsymbol{X}_k, F_n(\boldsymbol{X}_k) - F_{objective})$

The iteration process starts with the calculation of the step $\boldsymbol{d}_k$, through equation (6).
The next phase is to find the reduction ratio and predict the next x_k values:

$$r_k = \frac{Ared_k}{Pred_k} \tag{7}$$

$$Pred_k = \varphi_k(0) - \varphi_k(d_k) \tag{8}$$

$$Ared_k = \|F_k\|^2 - \|F(x_k + d_k)\|^2 \tag{9}$$

$$\varphi(0) = \|F_k\|^2 \tag{10}$$

$$\varphi(d_k) = \|F_k + J_k d_k\|^2 \tag{11}$$

According to the value of r_k, a condition is established for the value of x_{k+1}:

$$x_{k+1} = \begin{cases} x_k + d_k \; if \quad r_k > p_0, \\ x_k, \quad r_k \le p_0 \end{cases} \tag{12}$$

$p_0 = 0.0001$ is a value to check if the reduction is acceptable. The last step is to update the parameter α_k:

$$\alpha_{k+1} = \max\{m, \alpha_k q(r_k)\} \tag{13}$$

$$q(r) = \max\left\{\frac{1}{4}, 1 - 2(2r-1)^3\right\} \tag{14}$$

Finally, go back to step 2 and repeat until $\|J_k^T F_k\| = 0$ or $k = 100$. In this algorithm the m constant (Fan, J., & Pan, J. 2005) is the minimum limit for the LM parameter in order to avoid a larger step than desired near the solution.

2.8 Differential corrector application

2.8.1 Pruning phase

The Self-Adaptive Levenberg-Marquardt algorithm is first used in the pruning phase, where the objective is to set several trajectories than connect the points from the grid search and Mars position. So, in this case, the trajectory starts with a Kepler Propagation from the end of the Moon's SOI until the end of Earth's SOI, followed by a Kepler Propagation from this point to Mars. So, the objective function gives the arrival point depending on the departure position $\vec{r}_i$ and initial velocity $\vec{v}_i$, as well as the departure and arrival times t_i, t_f, respectively.

$$F_{pruning}(\vec{r}_i, \vec{v}_i, t_i, t_f) \Rightarrow F_{pruning}$$

$$F_{pruning} = \begin{bmatrix} r_I \\ r_J \\ r_K \end{bmatrix} \tag{15}$$

r_I, r_J, r_K are the cartesian IJK coordinates for the spacecraft position at t_f relative to the Sun. Although the function is dependent on all the parameters, the only variable is the velocity $\vec{v}_i$. As this function defines the motion of the spacecraft according to the two-body problem, and its position in t_f, the Mars position at t_f is our objective value.

$$(F_{pruning}(\vec{v}), \vec{v}_i, r_{Mars} \Rightarrow \vec{v}_{opt}, F_{pruning} - r_{Mars})$$

$$r_{Mars} = \begin{bmatrix} r_I^{ref} \\ r_J^{ref} \\ r_K^{ref} \end{bmatrix} \tag{16}$$

$r_I^{ref}, r_J^{ref}, r_K^{ref}$ are the Mars IJK coordinates for the position of the planet in t_f relative to the Sun.
It is necessary to calculate the Jacobian matrix of the system:

$$J_k = \begin{bmatrix} \frac{\partial F(\vec{v})_I}{\partial v_I} & \frac{\partial F(\vec{v})_I}{\partial v_J} & \frac{\partial F(\vec{v})_I}{\partial v_K} \\ \frac{\partial F(\vec{v})_J}{\partial v_I} & \frac{\partial F(\vec{v})_J}{\partial v_J} & \frac{\partial F(\vec{v})_J}{\partial v_K} \\ \frac{\partial F(\vec{v})_K}{\partial v_I} & \frac{\partial F(\vec{v})_K}{\partial v_J} & \frac{\partial F(\vec{v})_K}{\partial v_K} \end{bmatrix} \tag{17}$$

The pruning analysis is performed in three stages, gradually decreasing the spherical angles window analyzed and the time interval.

2.8.2 Single-shooting method

In order to find the complete trajectory that connects the departure point and Mars with a Lunar gravity assist, the single shooting method was used. According to this method, the self-adaptive Levenberg-Marquardt algorithm was used to target simultaneously the B-plane parameters given by the pruning analysis and Mars position in t_f.

The function that describes the path is denominated:

$$F_{final}(\vec{r}_i, \vec{v}_i, t_i, t_f) \Rightarrow F_{final}$$

$$F_{final} = \begin{bmatrix} r_I \\ r_J \\ r_K \\ B \cdot T \\ B \cdot R \\ e \end{bmatrix} \tag{18}$$

r_I, r_J, r_K are the cartesian IJK coordinates for the spacecraft position at t_f relative to the Sun. $B \cdot T$ and $B \cdot R$ are the B-plane parameters calculated from the initial velocity vector and e is the orbit's eccentricity relative to Earth.
Likewise, the function variable is just the initial velocity vector $\vec{v}_i$.

$$(F_{final}(\vec{v}), \vec{v}_i, F_{goal} \Rightarrow \vec{v}_{opt}, F_{pruning}(\vec{v}_{opt}) - F_{goal})$$

$$F_{goal} = \begin{bmatrix} r_I^{ref} \\ r_J^{ref} \\ r_K^{ref} \\ B \cdot T^{ref} \\ B \cdot R^{ref} \\ e^{ref} \end{bmatrix} \tag{19}$$

The matrix F_{goal} is formed by the reference values for the algorithm to solve to, minimizing the difference between the function outcome and the objective. $r_I^{ref}, r_J^{ref}, r_K^{ref}$ are the Mars IJK coordinates for the position of the planet in t_f relative to the Sun, while $B \cdot T^{ref}$ and $B \cdot R^{ref}$ are the B-plane reference coordinates to target the gravity assist, and e^{ref} is the reference eccentricity relative to the Earth given by the pruning phase.
As this function yields six values from three variables, the Jacobian matrix for this problem is:

$$J_k = \begin{bmatrix} \frac{\partial F(\vec{v})_I}{\partial v_I} & \frac{\partial F(\vec{v})_I}{\partial v_J} & \frac{\partial F(\vec{v})_I}{\partial v_K} \\ \frac{\partial F(\vec{v})_J}{\partial v_I} & \frac{\partial F(\vec{v})_J}{\partial v_J} & \frac{\partial F(\vec{v})_J}{\partial v_K} \\ \frac{\partial F(\vec{v})_K}{\partial v_I} & \frac{\partial F(\vec{v})_K}{\partial v_J} & \frac{\partial F(\vec{v})_K}{\partial v_K} \\ \frac{\partial B \cdot T(\vec{v})}{\partial v_I} & \frac{\partial B \cdot T(\vec{v})}{\partial v_J} & \frac{\partial B \cdot T(\vec{v})}{\partial v_K} \\ \frac{\partial B \cdot R(\vec{v})}{\partial v_I} & \frac{\partial B \cdot R(\vec{v})}{\partial v_J} & \frac{\partial B \cdot R(\vec{v})}{\partial v_K} \\ \frac{\partial e(\vec{v})}{\partial v_I} & \frac{\partial e(\vec{v})}{\partial v_J} & \frac{\partial e(\vec{v})}{\partial v_K} \end{bmatrix} \quad (20)$$

3. Results

In this chapter, the results, graphs and tables obtained from the simulations are presented and discussed. The values obtained are compared with the previous work (Oliveira, R. 2017 Msc. Dissertation) in case of the MER-A mission, and the others are compared to the pork chop plots, results from Yang, B. *et al.* (2018) and to the results obtained with the single shooting method and GMAT for a direct transfer.

.

Table 1: Direct Launch Windows with results from Yang, B. *et al*. (2018) and Oliveira, R. 2017 Msc Dissertation

Departure (UTCGregorian) dd-mm-yy	Arrival (UTCGregorian) dd-mm-yy	Departure (MJD)	Arrival (MJD)	Orbit type	Direct Characteristic Energy $(C_3)[km^2/s^2]$
10-06-2003	04-01-2004	22801	23009	Short	8.94481
23-08-2020	06-10-2021	29085	29494	Long	16.5038
15-09-2022	04-10-2023	29837	30222	Long	13.7934
05-10-2024	15-09-2025	30589	30934	Long	11.1894
30-10-2026	21-08-2027	31344	31629	Long	9.1371

The purpose of the gravity assist maneuver is to reduce the characteristic energy C_3 and obtain a trajectory that accomplishes, with a small deviation, the departure and arrival dates. It is unlikely that the Moon is in the correct position to perform the gravity assist at the most efficient time to do the transfer maneuver. So, in the worst-case scenario, the injection epoch can be up to 15 days after or prior to the ideal epoch for interplanetary transfer, which can cause

a significant reduction of the orbit energy gain, or even question the viability of the gravity assist.
In order to approximate the numerical study as close to reality as possible, the General Mission Analysis Tool (GMAT) is used. GMAT is an open source software developed by a NASA's team, but also private and public contributors and it is the only open source software for mission design, analysis, optimization and navigation available. The first released version was in 2007 and since then it has been continuously improved with several accessible versions. It was validated and verified in S. P. Hughes *et al.* (2014).

3.1 Direct single shooting and GMAT values

In order to compare the results from the gravity assist, a brief study of the Mars direct transfers is performed. The self-adaptive Levenberg-Marquardt algorithm is applied the same way as in the single shooting method, only considering the departure position as the same position that is described in the work ahead to perform the gravity assist. The parts of the function that describe the motion considering the Moon are also discarded.
The results from the direct trajectory obtained from the single shooting method and the GMAT simulation are presented in table 2.

Table 2: Direct transfer results

DEPARTURE (MJD)	*Yang, B.* et al. *(2018)* $(C_3)[km^2/s^2]$	SSM DIRECT $(C_3)[km^2/s^2]$	GMAT DIRECT $(C_3)[km^2/s^2]$
29085	16.5038	15.5614	16.3724
29837	13.7934	12.8764	14.2287
30589	11.1894	10.2156	11.0333
31344	9.1371	8.2732	9.2671

3.2 Single shooting method and GMAT results

The results of the single shooting are presented in table 3 and the results of the GMAT simulations are presented in table 4.

Table 3: Results of the gravity assist maneuvers with single-shooting method

$t_0(MJD)$	Periapsis Radius (km)	Velocity Magnitude (km/s)	Direct $(C_3)[km^2/s^2]$	Characteristic Energy LGA $(C_3)[km^2/s^2]$
22807.1038 (MER-A)	1834	11.2297	8.9448	5.3130
29072.7259	1791	10.7490	15.5614	15.7081
29837.4236	1864	10.5509	12.8764	10.6237
30602.9517	1823	10.4834	10.2156	9.7437
31342.8966	1945	10.2881	8.2732	5.2033

Table 4: Results of the gravity assist in GMAT software

Launch window $t_0(MJD)$ / Mission	$t_0(MJD)$	$t_f(MJD)$	r_{per} (km)	v_{mag} (km/s)	Direct $C_3[km^2/s^2]$	C_3 LGA $[km^2/s^2]$
MER-A	22807.3790	23011.5357	1800	11.2688	8.9448	6.2732
29085	29072.7520	29490.4874	1837	10.7891	16.3724	16.5742
29837	29837.9350	30240.3587	1794	10.5715	14.2287	11.6182
30589	30603.1218	30948.4127	1840	10.4882	11.0333	9.8629
31344	31343.2483	31645.5052	2063	10.3451	9.2671	6.4245

The results from the launch windows where a gravity assist is performed are presented. Regarding to the result for the MER-A mission also performed in Oliveira, R. 2017 Msc Dissertation, there is a slight improvement of $\Delta C_3 = 0.08 km^2/s^2$. This can be explained by the lower radius of periapsis than used in the mentioned work $r_p = 1838\,km$. So, it is presumed that the result is identical. Similarly, to the mentioned work, there's a decrease of the characteristic energy of $\Delta C_3 = 2.67 km^2/s^2$. With this result, it is proven that the self-adaptive Levenberg-Marquardt algorithm is as effective as the Newton-Raphson as a differential corrector and can be used for space mission design effectively.

The result obtained in the 29085 launch window is inefficient. There is an increase in the characteristic energy of $\Delta C_3 = 0.20 km^2/s^2$ This can be explained by the fact that during the favorable launch window, the Moon is basically on the opposite position to the one needed to perform the gravity assist. There was an attempt to obtain an effective result both before and after the launch window, in both epochs a trajectory was found, although the most efficient is the one presented in the tables 3 and 4. However, the LGA trajectory after the launch window can be used as last resort, in case a mission scheduled for 29085 is unable to be launched in the exact day. A flyby can be performed approximately 14 days after the ideal direct launch window. The maneuver can prevent the cost increase of launching a mission outside of the direct efficient period.

In the 29837 launch window, an efficient trajectory was found. Comparing to the values of the direct transfer, there is a significant decrease in the characteristic energy $\Delta C_3 = 2.61 km^2/s^2$.

In the 30589 launch window, comparing the characteristic energy C_3 with the direct result obtained in GMAT, there is a significant reduction, even though the launch epoch for the Lunar gravity assist trajectory is performed 14 days after the ideal transfer. The value of the reduction is $\Delta C_3 = 1.17 km^2/s^2$.

In the last launch window, the characteristic energy has an increase of $\Delta C_3 = 2.84 km^2/s^2$, which can be explained by the fact that the Moon is in a very favorable position to perform the gravity assist maneuver.

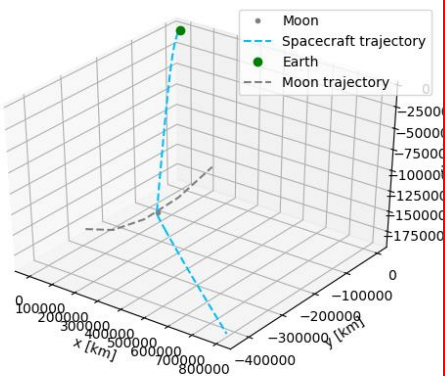
Expectedly there is a small increase of the velocity magnitude v_{mag} and characteristic energy C_3 comparing the GMAT results to SSM. This is caused by the influence of all bodies of the Solar system at all moments of the trajectory, mainly the influence of Earth shortly after the departure from Earth's SOI, as well as the influence of Earth's gravity in the phase of the flyby.

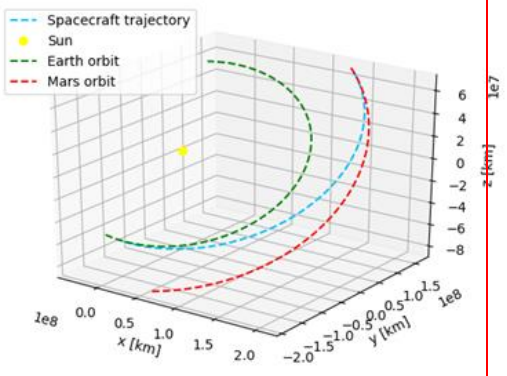
Table 5: Injection ephemeris in Earth MJ200Eq reference frame in GMAT

$t_0\,(MJD)$	r_I	r_J	r_K	v_I	v_J	v_K
22807.3786	-5950.7395	-2303.8716	-1701.4774	0.55861	-9.49263	-6.04681
29072.7519	1397.0293	-7743.3369	-1363.0928	10.42125	1.27715	2.48414
29837.9350	3989.0782	-6785.3544	-1193.1074	9.38593	3.00576	3.82442
30603.1217	3988.7141	-6785.5135	-1192.3706	9.01392	2.80708	4.56866
31343.2483	7805.9753	-1346.8180	-234.4425	5.10027	7.84662	4.40887

3.3 Graphical representations

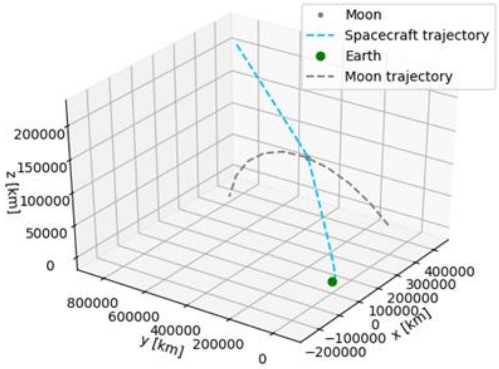
The results obtained with the SSM are now represented graphically in Python 3D plots.

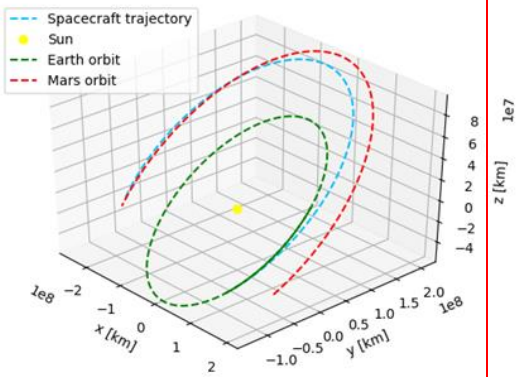




Earth departure | Interplanetary phase

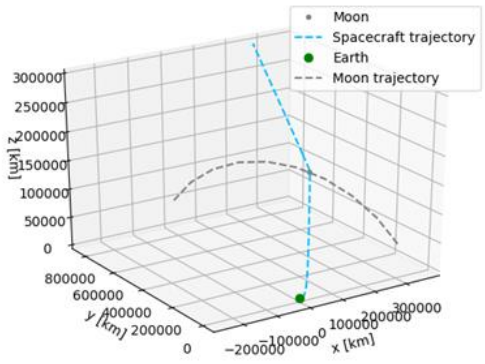
Figure 1: MER-A mission LGA



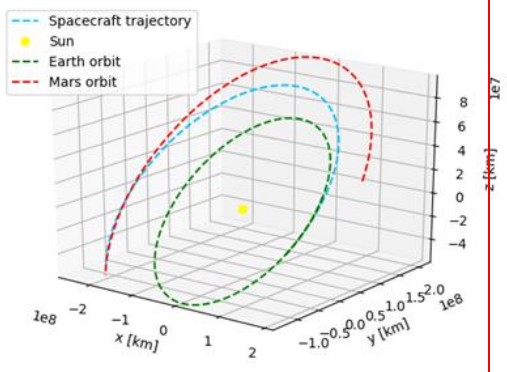


Earth departure | Interplanetary phase

Figure 2: 29085 launch window with LGA

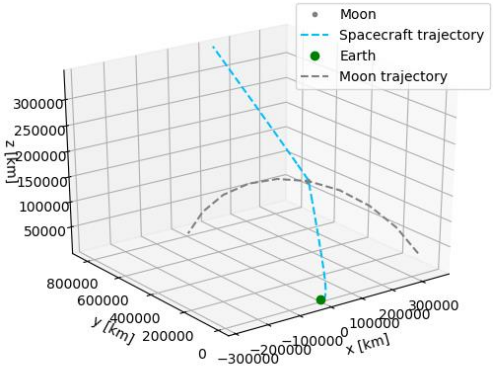


Earth departure

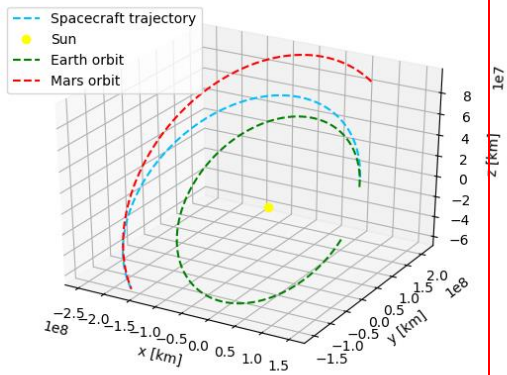


Interplanetary phase

Figure 3 29837: launch window with LGA

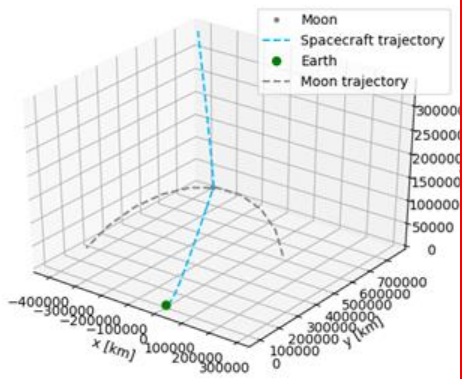


Earth departure

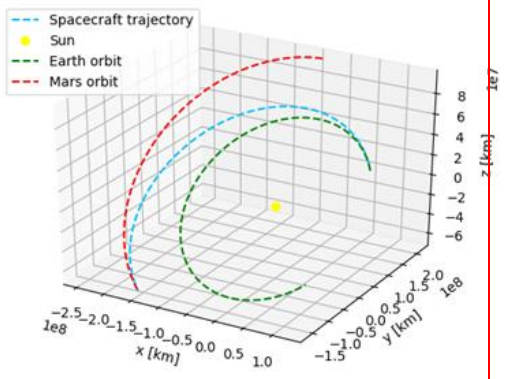


Interplanetary phase

Figure 4: 30589 launch window with LGA



Earth departure



Interplanetary phase

Figure 5: 31344 launch window with LGA

4. Conclusions

The objective of this work was to find alternative routes to Mars that can increase the payload capacity or decrease the fuel needed of future Mars missions. The results include the validation of the differential corrector SLM algorithm with a trajectory previously studied, and the study of 4 future transfer windows. There was a solution found in all of them, and in 3 out of 4 of the Lunar gravity assist maneuvers were effective, reducing the characteristic energy C_3 of the orbit. The results are compared with a previous study of the future Mars transfer windows and pork chop plots. It is safe to confirm that the self-Adaptive Levenberg-Marquardt algorithm is a powerful tool that can be applied to mission design, due to the fact that in all launch windows a trajectory with a Lunar gravity assist was found.

The success of the results can be explained by the flexibility of the departure and arrival epochs, as well as the possible use of short or long trajectories, type I or type II, respectively. This allowed to choose the type of trajectory compatible with the lunar gravity assist maneuver. In the simulations where the Lunar gravity assist performed better than the predicted direct trajectories, the decrease of orbit energy was from $1.17 km^2/s^2$ to $2.84 km^2/s^2$, which could mean a significant increase of the payload mass on most of the future missions to Mars.

This work proves that these types of trajectories should be considered and studied for future Mars, Venus and other missions. Although the trajectories can be considered riskier due to lower altitude flybys and trajectory correction errors can cause serious failures. This can be prevented with higher altitude flybys, that still give a significant energy increase, but have a better margin for failure.

References

Yang, B., *et al*. (2018) Hierarchical approach for fast searching optimal launch opportunity in a wide range. *Adv. Space Res*., **63**(1):572-588, https://doi.org/10.1016/j.asr.2018.09.026

Fan, J., & Pan, J. (2005). Convergence Properties of a Self-adaptive Levenberg-Marquardt Algorithm Under Local Error Bound Condition. *Computational Optimization and Applications*, **34**(1), 47-62. doi:10.1007/s10589-005-3074-z

Oliveira, R. (2017). Numerical Study of Earth-Mars trajectories with lunar gravity assist manoeuvres. Msc. Universidade da Beira Interior.

Gil-Fernandez, J. (2005). ExoMars alternative escape trajectories with Soyuz/Fregat. *Annals of The New York Academy of Sciences*, **1065**(1), 15-36. doi: 10.1196/annals.1370.011

Vallado, D. and McClain, W. (2013). Fundamentals of astrodynamics and applications. Microcosm Press, Hawthorne, California, USA.

Battin, R. (2000). Introduction to the Mathematics and Methods of Astrodynamics. American Institute of Aeronautics and Astronautics, Reston, USA.

Esa.github.io. (2019). Multi revolutions Lambert Problem — pykep 2.4 documentation. [online] Available at: https://esa.github.io/pykep/examples/ex2.html [Accessed 7 Nov. 2019].

Cho, D., Chung, Y., & Bang, H. (2012). Trajectory correction maneuver design using an improved B-plane targeting method. *Acta Astronautica*, **72**, 47-61. doi: 10.1016/j.actaastro.2011.11.009

S. P. Hughes, R. H. Qureshi, S. D. Cooley, and J. J. Parker, "Verification and validation of the General Mission Analysis Tool (GMAT)," *AIAA/AAS Astrodynamics Specialist Conference*, AIAA SPACE Forum, 2014. 21