



Delft University of Technology  
Faculty Electrical Engineering, Mathematics and Computer Science  
Delft Institute of Applied Mathematics

**Optimisation of Propellant Consumption for Power Limited  
Rockets.**

**What Role do Power Limited Rockets have in Future Spaceflight Missions?**

**(Dutch title: Optimaliseren van brandstofverbruik voor  
vermogen gelimiteerde raketten.**

**De rol van deze raketten in toekomstige ruimtevlucht missies. )**

A thesis submitted to the

Delft Institute of Applied Mathematics  
as part to obtain the degree of

**BACHELOR OF SCIENCE  
in  
APPLIED MATHEMATICS**

by **NATHALIE OUDHOF**

**Delft, the Netherlands  
December 2017**





**BSc thesis APPLIED MATHEMATICS**

**“ Optimisation of Propellant Consumption for Power Limited Rockets  
What Role do Power Limited Rockets have in Future Spaceflight Missions?”**

**(Dutch title: “Optimaliseren van brandstofverbruik voor vermogen  
gelimiteerde raketten  
De rol van deze raketten in toekomstige ruimtevlucht missies.)”**

**NATHALIE OUDHOF**

**Delft University of Technology**

**Supervisor**

Dr. P.M. Visser

**Other members of the committee**

Dr.ir. W.G.M. Groenevelt

Drs. E.M. van Elderen

21 December, 2017

Delft



# Abstract

In this thesis we look at the most cost-effective trajectory for power limited rockets, i.e. the trajectory which costs the least amount of propellant. First some background information as well as the differences between thrust limited and power limited rockets will be discussed. Then the optimal trajectory for thrust limited rockets, the Hohmann Transfer Orbit, will be explained. Using Optimal Control Theory, the optimal trajectory for power limited rockets can be found. Three trajectories will be discussed: Low Earth Orbit to Geostationary Earth Orbit, Earth to Mars and Earth to Saturn. After this we compare the propellant use of the thrust limited rockets for these trajectories with the power limited rockets. Here we made this comparison between a conventional thrust limited rocket with a specific impulse of 455 seconds and the VASIMR rocket for the power limited rocket. Also the initial mass of both rocket types was taken as  $5 \cdot 10^5$  kg. Lastly, we take a look at a gravity assist. Gravity assists can help reduce propellant use even further. Therefore we will once more look at the trajectory for a power limited rocket from Earth to Saturn. Only this time we use a gravity assist from Jupiter. Then we can see if the propellant use is indeed even lower when using the gravity assist. We find that the power limited rocket becomes more fuel-efficient as travel time increases. Also using a gravity assist further reduces propellant consumption.



# Contents

<b>Notation</b>	<b>1</b>
<b>1 Introduction</b>	<b>3</b>
1.1 Power Limited Rockets . . . . .	3
1.2 Goal of the Research . . . . .	3
1.3 Outline of the Thesis . . . . .	4
<b>2 Overview of Different Rocket Types</b>	<b>5</b>
2.1 A Brief History of Rockets . . . . .	5
2.2 Rocket Types . . . . .	5
2.2.1 Chemical Propulsion Systems . . . . .	6
2.2.2 Electrical Propulsion Systems . . . . .	6
2.3 Comparing Chemical and Electrical Propulsion Systems . . . . .	7
2.3.1 Hohmann Transfer Orbit . . . . .	7
2.3.2 Rocket Mass for the Hohmann Transfer Orbit . . . . .	8
2.3.3 VASIMR . . . . .	9
<b>3 The Mathematical Approach</b>	<b>11</b>
3.1 Assumptions . . . . .	11
3.2 The Cost Function . . . . .	11
3.3 Optimal Control Theory . . . . .	12
3.4 Close to Earth's Surface . . . . .	12
3.4.1 Equation for Payload . . . . .	16
<b>4 Trajectories in the Solar System</b>	<b>19</b>
4.1 Polar Coordinates . . . . .	19
4.1.1 LEO to GEO . . . . .	21
4.1.2 Earth to Mars . . . . .	23
4.1.3 Earth to Saturn . . . . .	25
4.2 Comparing to Hohmann Transfer Orbit . . . . .	27
<b>5 Trajectory to Saturn using a Gravity Assist</b>	<b>29</b>
5.1 Gravity Assist: Jupiter . . . . .	29
5.2 Rotating Reference Frame . . . . .	30
5.2.1 Simulating Jupiter's Atmosphere . . . . .	30
5.3 Effect of Jupiter on the Trajectory . . . . .	31
5.4 Removing the Singularity . . . . .	35
<b>6 Conclusion and Recommendations</b>	<b>37</b>

<b>Appendices</b>	<b>41</b>
<b>A Derivations</b>	<b>43</b>
A.1 Hamiltonian is Constant . . . . .	43
<b>B Matlab Code</b>	<b>45</b>
B.1 Close to Earth's Surface . . . . .	45
B.2 Hohmann Transfer Orbit . . . . .	45
B.3 LEO GEO . . . . .	46
B.4 Earth Mars . . . . .	48
B.5 Earth Saturn . . . . .	50
B.6 Flyby . . . . .	52

# Notation

- $t_0$  = Initial time [s].
- $t_f$  = Final time [s].
- $m(t)$  = Mass rocket at time  $t$  [kg].
- $m(t_0)$  = Initial mass of the rocket (propellant + payload) [kg].
- $m(t_f)$  = Final mass of the rocket (initial mass rocket - propellant). Also known as the payload mass [kg].
- $g$  = Earth's gravitational pull at sea level,  $9.81 \left[ \frac{m}{s^2} \right]$ .
- $\Gamma$  = Thrust acceleration  $\left[ \frac{m}{s^2} \right]$ .
- $\phi$  = Flight path angle [rad].
- $c$  = Exhaust velocity  $\left[ \frac{m}{s} \right]$ .
- $P$  = Engine power [W].
- $-\dot{m}$  = Mass flow rate  $\left[ \frac{kg}{s} \right]$ .
- $R_{\oplus}$  = Radius of the Earth,  $6.378 \cdot 10^6$  [m].
- $M_{\oplus}$  = Mass of the Earth,  $5.98 \cdot 10^{24}$  [kg].
- $M_{\odot}$  = Mass of the Sun,  $1.989 \cdot 10^{30}$  [kg].
- $M_{\text{J}}$  = Mass of Jupiter,  $1.89813 \cdot 10^{27}$  [kg].
- $G$  = Gravitational constant,  $6.67408 \cdot 10^{-11} \left[ \frac{m^3}{kg \cdot s^2} \right]$ .
- $\mu_{\oplus}$  = Standard gravitational parameter of Earth,  $GM_{\oplus} = 3.986004418 \cdot 10^{14} \left[ \frac{m^3}{s^2} \right]$ .
- $\mu_{\odot}$  = Standard gravitational parameter of the Sun,  $GM_{\odot} = 1.32712440018 \cdot 10^{20} \left[ \frac{m^3}{s^2} \right]$ .
- $\mu_{\text{J}}$  = Standard gravitational parameter of Jupiter,  $GM_{\text{J}} = 1.26686534 \cdot 10^{17} \left[ \frac{m^3}{s^2} \right]$ .



# Chapter 1

## Introduction

In recent years movies about human spaceflight are booming. Science fiction movies such as Star Wars [6] or Star Trek [1] are very popular, but also movies that depict humans staying or living on Mars, e.g. The Martian [7]. However, in reality the only astronomical object besides the Earth that man has stood foot on, is the Moon. The last manned mission to the Moon and also beyond a Low Earth Orbit (300 km above the Earth's surface), was Apollo 17 which launched in 1972 [10]. These days space agencies such as NASA, National Aeronautics and Space Administration, have plans to fly to the Red Planet (i.e. Mars) as early as 2030 [24]. Each mission naturally uses rockets. Rockets are a type of propulsion system which we call anaerobic, meaning without air. Where an airplane uses surrounding air to lift off and extract oxygen, a rocket moves in a near vacuum. Therefore a rocket needs to carry all its fuel, called propellant, on board and can only thrust forward by expelling mass. In the last century rockets mostly relied on chemical reactions to achieve thrust, but today electrical propulsion systems are seen as key for the new generation of space missions [2].

### 1.1 Power Limited Rockets

In this thesis we will focus on the electrical propulsion systems which are also known as *power limited rockets* [11]. This is because electrical propulsion systems use a certain, fixed, power unit to achieve thrust. Therefore its thrust is limited by the power available. Chemical propulsion systems, which are still popular these days, are called *thrust limited* [11]. This is because they use chemical reactions to achieve thrust. However, chemical reactions are bound by how quickly the reactions take place and the amount of energy that is released by them. Therefore they will be limited by the amount of thrust that can be generated by the chemical reactions. The most distinguishable feature between these two types of propulsion systems is that chemical propulsion systems are known for their high thrust capabilities. This is why they are often used for launch. Meanwhile, electrical propulsion systems are much more economical, meaning the overall propellant costs are lower. In this thesis we will focus on how to compare these two types of rocket propulsion systems and which one would be preferable in different types of missions.

### 1.2 Goal of the Research

The main goal of this thesis will be to find an optimal trajectory for a spacecraft that uses an electrical propulsion system, depending on its destination. In our case, we have chosen that instead of thinking of an optimal trajectory as a trajectory which takes the least time, we will minimize the propellant use. This seems more logical because we could fly at full power to the

moon and it would take little time, however the fuel costs would be enormous. Because of this, we will be looking at how the power limited rockets perform when wanting to minimize fuel use. Would flying to for example Mars take much longer than if we use the conventional, chemical rockets? Or is the difference not that significant and could power limited rockets be a part of future spaceflight missions? These are some of the questions we will be looking to answer in this dissertation.

### 1.3 Outline of the Thesis

In order to answer the questions from the previous section, we will need to take appropriate steps. First, we will go a little more in depth into the differences between electrical and chemical propulsion systems and how we will be able to compare their performances. This will be done in Chapter 2. In Chapter 3, we will take a look at the mathematics needed to minimize the fuel costs and look at the case of a homogeneous gravity field, takeoff from the Earth's surface. In chapter 4, we will then take a look at an inhomogeneous gravity field and we are interested in flying to different destinations. We will be looking at flying from a Low Earth Orbit to a Geostationary Earth Orbit. Furthermore we will also consider the trajectory when flying from the Earth to Mars and from the Earth to Saturn and compare the overall propellant costs of these trajectories to the results of chemical propulsion systems. Afterwards in chapter 5, we will take a look at a flyby. In reality, when a rocket is launched and set to fly to, for example, Mars or Jupiter, it often uses a flyby or gravity assist from Venus or the Earth in order to save fuel and reduce flight time. We will look at how the trajectory and most importantly, fuel use, changes if we use a gravity assist from Jupiter when flying to Saturn. Lastly, in chapter 6, we will conclude our results and discuss how realistic it is to use power limited rockets and for what types of missions they could be most likely used. All the used codes and some additional calculations can be found in the Appendices.

## Chapter 2

# Overview of Different Rocket Types

### 2.1 A Brief History of Rockets

The very first rocket dates back to the Greek philosopher Archytas (428 to 347 B.C.) [14]. Although this rocket was not used for outer space, it was the first device which could be called a rocket. He made a small device which was shaped like a bird. This device was then propelled by a jet of steam or compressed air [14]. After this, mostly only ‘solid propellant’ rockets were used. These were rockets that were fueled by gunpowder and originate from China [14]. It was only in the 16th and 17th century that the foundations of rocket science were laid by Galileo Galilei and Sir Isaac Newton. However rockets in the 18th and 19th century were mostly used as weapons in wars. Finally in the 20th century Konstantin Tsiolkovsky (1857-1935), Robert Goddard (1882-1945) and Hermann Oberth (1894-1989) laid the foundation of spaceflight as we know it [14]. The former two both came to the conclusion that liquid propellant rockets were better than solid propellant rockets, with Goddard successfully launching the first liquid propellant rocket in 1926 [15]. This first liquid propellant rocket was fueled by liquid oxygen and gasoline. It flew for two and a half seconds, climbed 12.5 meters, and landed 56 meters away. This may not sound like much of an achievement, but no one had ever successfully built a liquid propellant rocket before. Since then, mostly liquid propellant rockets were used for missions such as Sputnik 1, the first ever artificial Earth satellite launched by the USSR on October 4th, 1957 [17]. This rocket used liquid kerosene and oxygen as its propellant.

### 2.2 Rocket Types

Nowadays there is a wide variety of propulsion systems which can be categorized in two main categories: chemical propulsion and electrical propulsion. The main difference between these two is that chemical propulsion systems can achieve higher thrust in shorter periods of time than electrical propulsion systems [4]. This is the reason why chemical propulsion is mostly used for launch. Furthermore chemical propulsion systems are capable of making instant velocity changes whilst electrical propulsion systems need long periods of time to build up speed [4]. However, there are some electrical propulsion systems which could be very promising. Yet in order to compare performance capabilities of these rockets we first introduce the term *specific impulse*,  $I_{sp}$ . Specific impulse is denoted in seconds and represents the amount of thrust the rocket is able to generate with the weight of propellant, in Newton, expelled during 1 second [19]. It can be written as  $I_{sp} = \frac{c}{g}$ , where  $c$  is the exhaust velocity of the propellant and  $g$  is the gravitational acceleration constant at sea level,  $g = 9.81 \text{ m/s}^2$  [4]. Thus a propulsion system with a higher specific impulse uses the mass of the propellant more efficiently in creating thrust. Still these

two main categories can be broken down a bit further. We will be using the book *Understanding Space: an Introduction to Astronautics* [4] for the next subsections.

### 2.2.1 Chemical Propulsion Systems

1. **Liquid bipropellant:** This rocket type uses two liquid propellants. One is fuel, most commonly liquid hydrogen and the other is an oxidizer, usually liquid oxygen. This particular combination results in a specific impulse around 455 seconds. However one drawback is that these propellants need to be cooled at hundreds degrees Celsius below zero. This can be problematic over very long periods of time. Therefore hydrazine and nitrogen tetroxide are sometimes preferred to be used as they remain stable at room temperature. Although, the drawback is that the specific impulse of this combination is lower, around 300 seconds.
2. **Liquid monopropellant:** As its name implies, this rocket uses a single propellant. The most used monopropellant is by far hydrazine ( $N_2H_4$ ). It produces a fairly high  $I_{sp}$ , around 230 seconds, however it is still much lower than most liquid bipropellants. Monopropellant rockets are used for their simplicity and reliability, although for launch vehicles liquid bipropellant rockets are preferred because of their higher  $I_{sp}$  and thrust.
3. **Solid propellant:** Solid propellant rockets are a bit similar to liquid bipropellant rockets. They contain a mixture of fuel and oxidizer which is solidified. The most common oxidizer is ammonium perchlorate and for fuel it mostly uses powdered aluminum. Specific impulse mostly ranges from 200 to 300 seconds. These rockets are used because of their simplicity and reliability. However once started, the solid propellant rockets are difficult to stop and cannot be restarted. Instead they use all their fuel in one go.

### 2.2.2 Electrical Propulsion Systems

Electrical propulsion systems are known for their high specific impulse, yet their thrust capacities are mostly very low. Again we can divide the electrical propulsion systems in three subcategories.

1. **Electrothermal:** These types of rockets are quite similar to chemical liquid propulsion. Electrothermal rockets operate by using a propellant and heating it before accelerating the propellant and thus producing thrust. The specific impulse of this type is around 500-1000 seconds.
2. **Electromagnetic:** Electromagnetic propellant rockets use a flowing electrical current and magnetic fields for acceleration. There is a wide variety of electromagnetic propulsion rockets but most of them have an  $I_{sp}$  between 1000 and 7000 seconds.
3. **Electrostatic:** These rockets are also known as ion thrusters. They use an applied electric field to accelerate an ionized propellant. The most common used propellant is Xenon. Also the specific impulse of these rockets can be up to 10,000 seconds.

Thus in overview we know:

Type	$I_{sp}$ [s]	Thrust [N]
Liquid bipropellant	300 - 460	up to $10^7$
Liquid monopropellant	180 - 250	0.1 - 100
Solid propellant	200 - 300	$10^3$ - $10^7$
Electrothermal	500 - 1000	$10^{-2}$ - 10
Electromagnetic	1000 - 7000	$10^{-3}$ - 10
Electrostatic	2000 - 10,000	$10^{-6}$ - $10^{-3}$

**Table 2.1:** Overview of the specific impulse as well as thrust of different types of propulsion systems. [4][19]

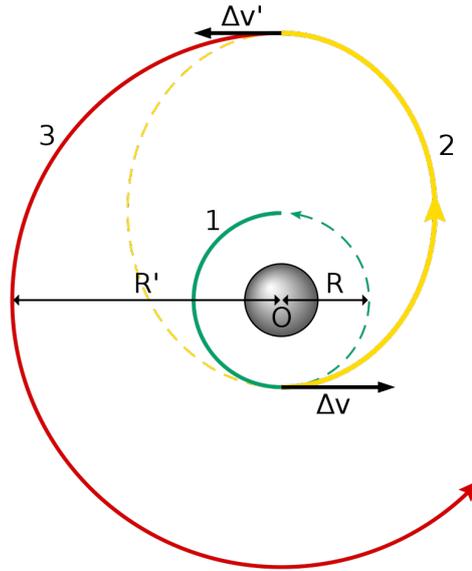
## 2.3 Comparing Chemical and Electrical Propulsion Systems

Now we need a way to compare the performance of chemical propulsion systems to electrical propulsion systems. As discussed before, in this thesis we will primarily focus on finding the optimal trajectory for an electrical propulsion system. The optimal trajectory for a chemical propulsion system has already been found by Walter Hohmann. Walter Hohmann was a German engineer born on 18th March 1880 in Hardheim, Germany. Hohmann is to this day considered one of the most important figures in laying the foundation of interplanetary travel as we know it. In 1925 he published his work ‘Die Erreichbarkeit der Himmelskörper’, The Attainability of the Heavenly Objects, where he discussed a method to move a spacecraft between two different orbits in a fuel-efficient way [3]. It is remarkable he was already thinking about this, considering it would be another 32 years before the first artificial satellite was launched [17]. The method Hohmann discussed in his publication is nowadays known as the Hohmann Transfer Orbit and is still used to this day.

### 2.3.1 Hohmann Transfer Orbit

So what does the Hohmann transfer orbit entail exactly? Well, Hohmann realized early on that minimizing fuel use would be very important if humankind were ever to travel to another planet. So the Hohmann Transfer Orbit is the most cost-effective manoeuvre for interplanetary travel. The basic concept of this manoeuvre is to move a spacecraft from one circular orbit to another circular orbit. It is assumed in this method that the orbits are in the same plane and that there are instantaneous velocity changes tangent to the initial and final orbits. This last assumption means that this manoeuvre can only be used in chemical propulsion systems as electrical propulsion systems are not capable of instantaneous velocity changes. The manoeuvre is then carried out in three steps, see figure 2.1.

At first the rocket is in circular orbit 1 with a radius  $R$  from the centre of mass,  $O$ . Then the rocket carries out an instantaneous velocity change,  $\Delta v$ , tangent to the circular orbit resulting in an elliptical orbit, orbit 2. Then when arriving at the point where orbit 2 and 3 intersect, a second instantaneous velocity change is made  $\Delta v'$ , ensuing that the rocket is now in a circular orbit at radius  $R'$ .



**Figure 2.1:** Hohmann Transfer Orbit. The rocket produces an instantaneous velocity change,  $\Delta v$ , to go from orbit 1 into the transfer orbit, orbit 2. When reaching the intersection of orbit 2 and 3, the rocket uses a second instantaneous velocity change,  $\Delta v'$ , to get into orbit 3. [8]

### 2.3.2 Rocket Mass for the Hohmann Transfer Orbit

In 1903 the Russian Konstantin Tsiolkovsky published his work where he discussed the use of the following equation.

$$\Delta v = I_{sp}g \ln \left( \frac{m_0}{m_f} \right).$$

This equation is since known as ‘Tsiolkovsky’s rocket equation’ [9]. It relates the maximum change of velocity of a rocket in terms of specific impulse,  $I_{sp}$ , standard gravity,  $g = 9.81$ , initial mass of the rocket,  $m_0$ , and final mass of the rocket,  $m_f$ . When knowing the  $\Delta v$  necessary for the trajectory, the propellant costs can be calculated with Tsiolkovsky’s rocket equation. However, we first need the means to find the  $\Delta v$  that is necessary for a trajectory. For this we need the mechanical energy, which is defined as the potential energy added to the kinetic energy. So,  $E = \frac{1}{2}mv^2 - \frac{\mu m}{R}$ . This equation shows the relationship between the spacecraft’s mass,  $m$ , its velocity,  $v$ , its position,  $R$ , and the standard gravitational parameter of the centre of mass,  $\mu$ . Here  $\mu$  is defined as  $\mu = GM$ , with  $M$  the mass of the centre of mass around which we are rotating. However, to generalize this equation we do not want to worry about the mass of the rocket. Therefore we introduce the term *specific mechanical energy*, which is merely  $\varepsilon = \frac{E}{m}$  [4]. Thus this means:

$$\varepsilon = \frac{v^2}{2} - \frac{\mu}{R}.$$

Simply rearranging this equation gives us an expression for the velocity of the rocket at position  $R$ , namely:

$$v = \sqrt{2 \left( \frac{\mu}{R} + \varepsilon \right)}. \quad (2.1)$$

From *Understanding Space: an Introduction to Astronautics* [4] chapters 4 and 6, we know that the specific mechanical energy is also equal to

$$\varepsilon = \frac{-\mu}{2a}. \quad (2.2)$$

Here  $a$  is known as the *semi-major axis*, which is simply half of the longest diameter of an ellipse. Note that if we have a circle, the semi-major axis is the same as the radius of the circle. Now we can calculate the  $\Delta v$  necessary for the trajectory. This will be carried out in a few steps. First we will calculate  $\Delta v_1$  which takes the spacecraft from orbit 1 into the transfer orbit, orbit 2, see figure 2.1. This  $\Delta v_1$  is equal to the velocity needed for the spacecraft to enter the transfer orbit at radius  $R$ , minus the velocity of the spacecraft in orbit 1. Furthermore  $\Delta v_2$  is equal to the velocity of the spacecraft in orbit 2, minus the velocity of the spacecraft in the transfer orbit at radius  $R'$ . Thus:

$$\begin{aligned}\Delta v_1 &= |v_{\text{transfer at orbit 1}} - v_{\text{orbit 1}}|, \\ \Delta v_2 &= |v_{\text{orbit 2}} - v_{\text{transfer at orbit 2}}|.\end{aligned}$$

From equation (2.1) we know how to calculate these velocities, hence:

$$\begin{aligned}v_{\text{orbit 1}} &= \sqrt{2 \left( \frac{\mu}{R} + \varepsilon_{\text{orbit 1}} \right)}, \\ v_{\text{transfer at orbit 1}} &= \sqrt{2 \left( \frac{\mu}{R} + \varepsilon_{\text{transfer}} \right)}, \\ v_{\text{transfer at orbit 2}} &= \sqrt{2 \left( \frac{\mu}{R'} + \varepsilon_{\text{transfer}} \right)}, \\ v_{\text{orbit 2}} &= \sqrt{2 \left( \frac{\mu}{R'} + \varepsilon_{\text{orbit 2}} \right)}.\end{aligned}$$

Finally, the total  $\Delta v$  can be calculated as  $\Delta v = \Delta v_1 + \Delta v_2$ . Then only Tsiolkovsky's equation is needed in order to calculate the propellant necessary for the trajectory. Thus we know how to calculate the propellant costs of a thrust limited rocket which uses a Hohmann Transfer Orbit. Nevertheless, we are also interested to see how the power limited rockets compare to thrust limited rockets on flight time. We want to know the time of flight (TOF) of the Hohmann transfer orbit [4]. This is equal to:

$$TOF^1 = \pi \sqrt{\frac{a_{\text{transfer}}^3}{\mu}}.$$

Still, because of the extensive differences in rocket types as discussed in the last section, we will need to make a distinct choice for our chemical rocket which uses a Hohmann transfer orbit and our power limited rocket for which we will determine the transfer orbit in the next chapters. Only then can we make a fair comparison.

### 2.3.3 VASIMR

Now that we know how to calculate the propellant use for chemical propulsion systems, we have to make some decisions. We choose one specific form of chemical rocket and compare its performance to one specific form of electrical rocket. We have chosen to take a liquid bipropellant rocket as the chemical rocket as it is the most commonly used in this category. More specifically we will be using the specifications of a bipropellant rocket using liquid oxygen and liquid hydrogen, meaning we will assume an  $I_{sp}$  of 455 seconds [4]. Moreover, the electrical propulsion rocket of choice will be the VASIMR rocket. VASIMR stands for: Variable Specific Impulse Magnetoplasma Rocket and it is a type of electromagnetic thruster. It has an operational power

---

<sup>1</sup>In this equation we always use SI-units.

of around 59 MW and a variable  $I_{sp}$  of between 10,000 and 30,000 seconds [23]. Also its thrust capacities are quite high when comparing it to other types of electrical propulsion. The VASIMR rockets are seen as the bridge between high  $I_{sp}$  and low thrust and low  $I_{sp}$  and high thrust. It takes the best of both worlds. Throughout the entire thesis, these two rockets will be used to compare chemical propulsion to electrical propulsion.

## Chapter 3

# The Mathematical Approach

In this chapter we deal with the mathematics needed to be able to calculate an optimal trajectory for the power limited rockets. In our case, this means finding the most cost-effective trajectory where we minimize fuel use. In order to minimize the use of fuel for a power limited rocket, we have to minimize the cost function:  $\int_{t_0}^{t_f} \frac{1}{2} \Gamma^2 dt$  [11]. Later on in this chapter, the derivation of this function will be discussed. In this equation,  $\Gamma$  is the thrust magnitude of the rocket which can be varied. But first, we will agree on some assumptions.

### 3.1 Assumptions

Naturally it would be desirable to create an as realistic trajectory as possible. Though for the sake of simplicity, some assumptions need to be made. In this section we merely specify the assumptions that will hold throughout the entire thesis unless otherwise declared. The assumptions are:

1. The only forces acting on the spacecraft are gravity and thrust. (No atmospheric drag, solar radiation etc.)
2. All astronomical objects are perfect spheres.
3. Each planet's orbit around the Sun is perfectly circular.
4. We will be working in a two-dimensional plane.
5. Orbits and the trajectory of the rocket are in the same plane.

### 3.2 The Cost Function

In the introduction of this chapter the cost function has already been expressed. The derivation of this cost function will be made using the book *Optimal Control with Aerospace Applications* [11], chapter 10. The thrust acceleration,  $\Gamma$ , can be derived from the conservation of momentum:

$$m\Gamma = -\dot{m}c \quad (3.1)$$

Furthermore, power is the energy consumption per unit time. Thus power is:

$$P = -\frac{1}{2} \dot{m}c^2 \quad (3.2)$$

Combining these two equations we can derive an expression for the propellant costs.

$$\frac{\Gamma^2}{2P} = \frac{\dot{m}^2 c^2}{-2 \frac{\dot{m} c^2}{2}} = \frac{-\dot{m}}{m^2} = \frac{d}{dt} \left( \frac{1}{m} \right).$$

Thus the propellant costs, which we will call  $J$ , will be proportional to

$$\frac{1}{m(t_0)} - \frac{1}{m(t_f)} = \int_{t_0}^{t_f} \frac{\Gamma^2}{2P} dt.$$

In order to maximize the final mass, for a given value of initial mass, the engine needs to run at maximum power  $P = P_{max}$  [11]. For this reason, the final cost functional which is used throughout this thesis is  $J = \int_{t_0}^{t_f} \frac{1}{2} \Gamma^2 dt$ .

### 3.3 Optimal Control Theory

Now that we have agreed on several assumptions, we need to take a closer look at how to mathematically minimize the cost function,  $J = \int_{t_0}^{t_f} \frac{1}{2} \Gamma^2 dt$ . This will be done using the Hamiltonian, in particular the Hamiltonian of Optimal Control Theory [5] which was developed by Lev Pontryagin (1908 - 1988). The Hamiltonian of optimal control theory can be used to minimize a function which is subject to several conditions. So for problems that can be written as:

$$\begin{aligned} \min V &= \int_{t_0}^{t_f} F(t, \mathbf{x}, \mathbf{u}) dt, \\ \text{subject to } \mathbf{x}' &= f(t, \mathbf{x}, \mathbf{u}), \\ \mathbf{x}(t_0) &= \text{given}, \\ \mathbf{x}(t_f) &= \text{given}. \end{aligned}$$

Here  $F(t, \mathbf{x}, \mathbf{u})$  is the function we want to minimize and  $\mathbf{u}$  is the steering variable(s). In our case the steering variables are thrust magnitude,  $\Gamma$ , and thrust direction, given by the angle  $\phi$  [5]. Furthermore  $\mathbf{x}'$  is known as the equations of motion and  $\mathbf{x}$  is the vector of the position and velocity variables. So,  $\mathbf{x}'$  will describe the equations of motion of our rocket. Now the Hamiltonian can be formulated as:

$$H(t, \mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}) = F(t, \mathbf{x}, \mathbf{u}) + \boldsymbol{\lambda} f(t, \mathbf{x}, \mathbf{u}).$$

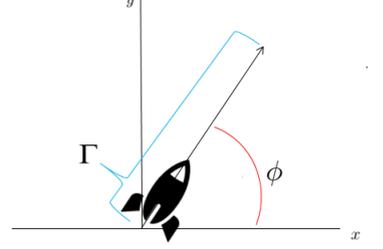
Knowing we want to optimize the steering variable(s),  $\mathbf{u}$ , the Hamiltonian is subject to  $\frac{\partial H}{\partial \mathbf{u}} = 0$  [5]. Furthermore, when the Hamiltonian is formed, we can write  $\mathbf{x}' = \frac{\partial H}{\partial \boldsymbol{\lambda}}$  and  $\boldsymbol{\lambda}' = -\frac{\partial H}{\partial \mathbf{x}}$ . If  $x_i(t_f)$  is not given, ergo it is 'free', then we use the transversality that its costate variable  $\lambda_i(t_f) = 0$  [5]. When subsequently solving these ordinary differential equations, we finally obtain the desired trajectory. It is also good to note that if the final time,  $t_f$ , is not specified ('free'), then it must hold that  $H(t_f) = 0$ . Lastly whenever the Hamiltonian does not directly depend on time  $t$ , meaning there is no  $t$  term in the Hamiltonian, then the Hamiltonian is constant [5]. Proof of this can be found in Appendix A.1.

### 3.4 Close to Earth's Surface

In order to further illustrate the use of the Hamiltonian as described in the previous section, we will look at the case of a homogeneous gravity field. This means we want to find the trajectory

for a rocket at launch, where we will minimize the cost function mentioned before,  $J = \int_0^{t_f} \frac{1}{2} \Gamma^2 dt$ . First we need to think about how to describe the motion of the rocket. We are working with Cartesian coordinates, meaning the position of the rocket is described in  $x, y$  and the velocity in  $v_x, v_y$ .

For the equations of motion we will have to look at the derivatives of  $x, y, v_x, v_y$ . Where we note that the change in velocity for  $v_x$  depends on the thrust magnitude,  $\Gamma$ , as well as flight path angle,  $\phi$ . For  $v_y$  this is also the case, except we also have the gravitational acceleration in the negative  $y$ -direction,  $g$ . Furthermore,  $g$  will be taken as a constant as we are only interested in takeoff from the Earth's surface, meaning the change in  $g$  is negligible. This means the motion of the rocket can be described with:



**Figure 3.1:** Rocket launch with thrust magnitude  $\Gamma$  and flight path angle  $\phi$ . [12]

$$\begin{aligned} x' &= v_x \\ y' &= v_y \\ v_x' &= \Gamma \cos(\phi) \\ v_y' &= \Gamma \sin(\phi) - g \end{aligned}$$

Knowing we want to minimize the cost function mentioned before, this results in the following Hamiltonian:

$$H(t) = \frac{1}{2} \Gamma^2 + \lambda_1 v_x + \lambda_2 v_y + \lambda_3 \Gamma \cos(\phi) + \lambda_4 (\Gamma \sin(\phi) - g)$$

From the Hamiltonian theorem [5] we know that:

$$\begin{aligned} x' &= \frac{\partial H}{\partial \lambda_1}, & y' &= \frac{\partial H}{\partial \lambda_2}, & v_x' &= \frac{\partial H}{\partial \lambda_3}, & v_y' &= \frac{\partial H}{\partial \lambda_4}, \\ \lambda_1' &= -\frac{\partial H}{\partial x}, & \lambda_2' &= -\frac{\partial H}{\partial y}, & \lambda_3' &= -\frac{\partial H}{\partial v_x}, & \lambda_4' &= -\frac{\partial H}{\partial v_y}. \end{aligned}$$

In our case the steering-variables which we want to optimize are, as mentioned before,  $\Gamma$  and  $\phi$ . This means we solve  $\frac{\partial H}{\partial \Gamma} = 0$  and  $\frac{\partial H}{\partial \phi} = 0$ . This gives us:

$$\cos(\phi) = \frac{-\lambda_3}{\sqrt{\lambda_3^2 + \lambda_4^2}}, \quad \sin(\phi) = \frac{-\lambda_4}{\sqrt{\lambda_3^2 + \lambda_4^2}}.$$

We also obtain  $\Gamma = -\lambda_3 \cos(\phi) - \lambda_4 \sin(\phi) = \sqrt{\lambda_3^2 + \lambda_4^2}$ . After elimination of the steering variables, our Hamiltonian reduces to:

$$H(t) = -\frac{1}{2} (\lambda_3^2 + \lambda_4^2) + \lambda_1 v_x + \lambda_2 v_y - g \lambda_4.$$

And we are left with eight Ordinary Differential Equations, ODEs:

$$\begin{aligned} x' &= v_x, & y' &= v_y, & v_x' &= -\lambda_3, & v_y' &= -\lambda_4 - g, \\ \lambda_1' &= 0, & \lambda_2' &= 0, & \lambda_3' &= -\lambda_1, & \lambda_4' &= -\lambda_2. \end{aligned}$$

These can easily be solved by hand:

$$\begin{aligned}
\lambda_1(t) &= \lambda_1, \\
\lambda_2(t) &= \lambda_2, \\
\lambda_3(t) &= -\lambda_1 t + c_1, \\
\lambda_4(t) &= -\lambda_2 t + d_1, \\
v_x(t) &= \frac{1}{2}\lambda_1 t^2 - c_1 t + c_2, \\
v_y(t) &= \frac{1}{2}\lambda_2 t^2 - d_1 t - gt + d_2, \\
x(t) &= \frac{1}{6}\lambda_1 t^3 - \frac{1}{2}c_1 t^2 + c_2 t + c_3, \\
y(t) &= \frac{1}{6}\lambda_2 t^3 - \frac{1}{2}d_1 t^2 - \frac{1}{2}gt^2 + d_2 t + d_3.
\end{aligned}$$

Where  $c_1, c_2, c_3, d_1, d_2, d_3$  are the integration constants. In order to solve these integration constants, we need begin and end conditions. As discussed before, we are only interested in takeoff of the rocket, from the Earth's surface. This means that our initial conditions are  $x(0) = 0$ ,  $y(0) = 0$ ,  $v_x(0) = 0$ ,  $v_y(0) = 0$ . We are interested in flying to an altitude of  $h$  kilometres and to be in orbit around the Earth at this altitude. This means for our end conditions that the position of  $x$  at end time,  $t_f$ , is free. This means instead of  $x(t_f)$  having a given value, we know that the costate variable belonging to  $x$ ,  $\lambda_1$  is zero at end time  $t_f$ . Furthermore, in order to be in orbit at a given altitude  $h$ , we solve that the gravitational force near the Earth's surface:  $F_g = M_1 \cdot g$ , must be equal to the centripetal force:  $F_c = \frac{M_1 v_x^2}{(R + h)}$ . Here  $M_1$  is the mass of the rocket and  $R$  denotes the radius Earth. Solving this for  $v_x$  when the spacecraft is close to the Earth, meaning  $R + h \approx R$ , gives:  $v_x = \sqrt{g \cdot R}$ . Naturally the end condition for  $y$  is to be at an altitude of  $h$  kilometres and the end condition for  $v_y(t_f) = 0$ , because at  $t_f$  we are in orbit. In overview:

$$\begin{aligned}
x(0) &= 0, & \lambda_1(t_f) &= 0, \\
y(0) &= 0, & y(t_f) &= h, \\
v_x(0) &= 0, & v_x(t_f) &= \sqrt{g \cdot R}, \\
v_y(0) &= 0, & v_y(t_f) &= 0.
\end{aligned}$$

Because  $t_f$  is free, we have the additional condition that  $H(t_f) = 0$ . Solving the equations using the conditions we obtain:

$$\lambda_1 = 0, \tag{3.3}$$

$$\lambda_2 = -\frac{12h}{t_f^3}, \tag{3.4}$$

$$\lambda_3 = -\frac{\sqrt{g \cdot R}}{t_f}, \tag{3.5}$$

$$\lambda_4(t) = -\frac{6h}{t_f^2} + \frac{12ht}{t_f^3} - g, \tag{3.6}$$

$$x(t) = \frac{t^2 \sqrt{g \cdot R}}{2t_f}, \quad (3.7)$$

$$y(t) = \frac{3ht^2}{t_f^2} - \frac{2ht^3}{t_f^3}, \quad (3.8)$$

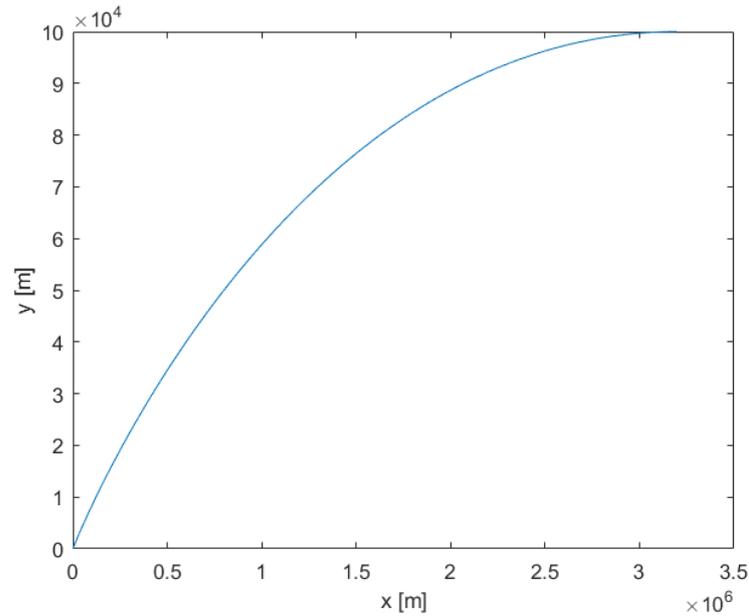
$$v_x(t) = \frac{t \sqrt{g \cdot R}}{t_f}, \quad (3.9)$$

$$v_y(t) = \frac{6ht}{t_f^2} - \frac{6ht^2}{t_f^3}. \quad (3.10)$$

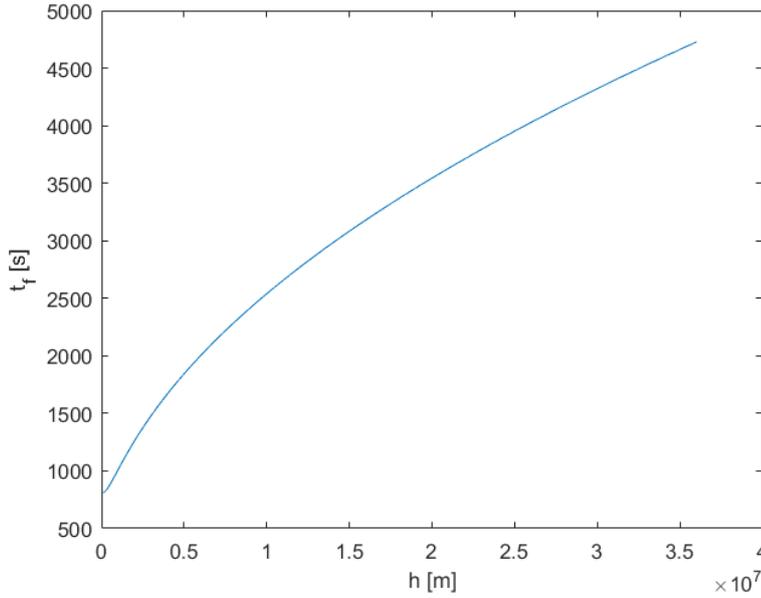
Having derived this, we still want to know what the optimal end time,  $t_f$ , is depending on the altitude we are flying to. We can find this using the condition  $H(t_f) = 0$  and knowing  $H(t)$  is constant (Appendix A.1), so  $H(0) = H(t) = H(t_f) = 0$ . Using this we obtain:

$$t_f(h) = \sqrt{\frac{R}{2g} \left( 1 + \sqrt{1 + \frac{144h^2}{R^2}} \right)}. \quad (3.11)$$

Using these functions, equations (3.3 - 3.10), we can now plot the trajectory of our spacecraft. The trajectory of the spacecraft is shown in figure 3.2, where the rocket flies to an altitude of 100 kilometres. This means the end time  $t_f$  for this trajectory was roughly  $t_f \approx 806$  s.



**Figure 3.2:** Trajectory of the spacecraft flying to an altitude of 100 kilometres from the Earth's surface.



**Figure 3.3:** End time  $t_f$  depending on altitude  $h$ .

We also make a figure of the optimum end time,  $t_f$ , depending on the altitude which we want to reach. Figure 3.3 shows that the optimal end time almost linearly depends on the altitude, as  $h$  becomes large.

### 3.4.1 Equation for Payload

Now in order to be able to make a comparison between the propellant mass used when using a Hohmann Transfer Orbit and the transfer orbit for an electrical propulsion system with the method as described in the previous sections, we need an expression for the mass of the rocket. The expression for the mass of the rocket using a Hohmann transfer orbit, has been explained in section 2.3.2. Nevertheless, we still need an expression for the mass of the rocket using an electrical propulsion system. For this, remember equations (3.1) and (3.2) from section 3.2. From equation (3.1), we conclude that the time-dependant exhaust velocity is  $c = \sqrt{-\frac{2P}{\dot{m}}}$ . Furthermore, note that from the previous section we know that  $\Gamma = \sqrt{\lambda_3^2 + \lambda_4^2}$ . Combining equation (3.1) and (3.2) we can eliminate  $c$  and we obtain an expression for the mass of the rocket at time  $t_f$  which is known as the mass of the payload.

$$\Gamma = -\frac{\dot{m}}{m} \sqrt{\frac{-2P}{\dot{m}}} = \frac{1}{m} \sqrt{-2P\dot{m}} = \sqrt{\lambda_3^2 + \lambda_4^2}, \quad (3.12)$$

$$-2P \frac{\dot{m}}{m^2} = \lambda_3^2 + \lambda_4^2, \quad (3.13)$$

$$\frac{d}{dt} \left( \frac{1}{m} \right) = -\frac{\dot{m}}{m^2} = \frac{\lambda_3^2 + \lambda_4^2}{2P}, \quad (3.14)$$

$$\frac{1}{m(t_0)} = \frac{1}{2P} \int_{t_0}^{t_0} (\lambda_3^2 + \lambda_4^2) dt + C = C, \quad (3.15)$$

$$m(t_f) = \frac{1}{\frac{1}{m(t_0)} + \frac{1}{2P} \int_{t_0}^{t_f} (\lambda_3^2 + \lambda_4^2) dt}. \quad (3.16)$$

Using this equation, we can calculate the mass of the rocket at end time  $t_f$ . Thus when knowing the initial mass  $m(t_0)$ , the propellant costs can be calculated. The power of the rocket will be taken to be equal to 59 MW as we take the specifications of a VASIMR rocket. Additionally we have to agree on the initial mass of the rocket as it is necessary when calculating the final mass. The initial mass of the rocket will be taken to be equal to  $m(t_0) = 5 \cdot 10^5 \text{ kg}$ , as this is a realistic initial mass for a rocket [18]. Using equation (3.16), we can calculate the final mass of the rocket for the trajectory from figure 3.2.

$$m(806) = \frac{1}{\frac{1}{5 \cdot 10^5} + \frac{1}{2 \cdot 59 \cdot 10^6} \int_0^{806} \left( - \left( \frac{\sqrt{g \cdot R}}{806} \right)^2 + \left( -\frac{6h}{806^2} + \frac{12ht}{806^3} - g \right)^2 dt \right)},$$

$$\approx 1.511 \cdot 10^3 \text{ kg}.$$

We can see that a lot of fuel has been used. This is argumentative as the gravitational pull of the Earth is largest at launch, therefore a lot of propellant has to be used in order to lift off.



## Chapter 4

# Trajectories in the Solar System

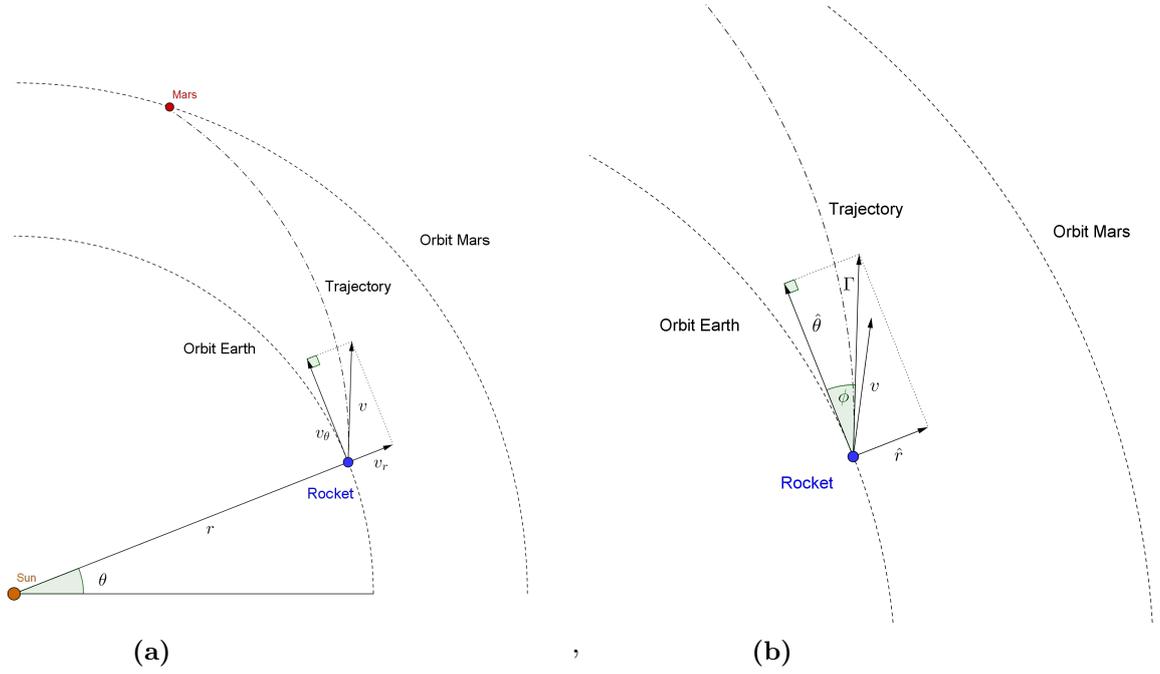
Now that we know the mathematics for finding the optimal trajectory, we apply the method to finding optimal trajectories in the Solar System. We will focus on three different trajectories. The first will be flying from a Low Earth Orbit (LEO), to a Geostationary Earth Orbit (GEO). The second one will be flying from the Earth to Mars and the third one from the Earth to Saturn. The gravitational pull from the Earth or Sun, whichever we use as centre of mass, will no longer be constant. This means we are working in an inhomogeneous gravity field, where the further we are away from the centre of mass, the lower its gravitational pull will be. In the first part of the chapter we will be working in polar coordinates as they reduce the complexity of our equations. This means we need to write our equations of motion in polar coordinates and derive the Hamiltonian in the same way as discussed in the previous chapter. After we have derived the Hamiltonian and acquired our ODEs, we can take a look at the three different trajectories. To conclude this chapter, we will take look at comparing the propellant use of the three trajectories to the propellant use of the Hohmann Transfer Orbits.

### 4.1 Polar Coordinates

The first step into obtaining the trajectories is writing the equations of motion in polar coordinates. We have chosen to work in polar coordinates as they reduce complexity of the equations. This now means we no longer have the equations of motion in the form of  $x, y, v_x, v_y$ . Instead we will now look at the radius,  $r$ , angle,  $\theta$ , radial velocity,  $v_r$ , and tangential velocity,  $v_\theta$ . Figure 4.1a, may give a good rendition of what we mean by these. The figure shows how the rocket would fly from Earth to Mars with showing what  $r, \theta, v_r$  and  $v_\theta$  are. The same principle holds if we fly from LEO to GEO or from the Earth to Saturn.

Now in order to acquire the equations of motion, we use Kepler's law of planetary motion [4]. Here  $\mathbf{r} = r \cdot \hat{\mathbf{r}}$  where  $r$  is the distance (radius) from centre of mass to the rocket and  $\hat{\mathbf{r}}$  is a unit vector pointing towards the rocket. We can find the equations of motion knowing  $\mathbf{r}' = \mathbf{v}$ . This gives us:

$$\begin{aligned}\mathbf{r}' &= \hat{\mathbf{r}}'r + \hat{\mathbf{r}}r', & \Leftrightarrow \\ \mathbf{v} &= \hat{\boldsymbol{\theta}}\theta'r + \hat{\mathbf{r}}r' = v_r\hat{\mathbf{r}} + v_\theta\hat{\boldsymbol{\theta}}.\end{aligned}$$



**Figure 4.1:** On the left the polar coordinates can be seen when flying from the Earth to Mars. The figure on the right gives a rendition of the thrust vector  $\Gamma$ . [12]

Here  $\hat{\theta}$  is a unit vector, tangent to  $\hat{r}$ . Furthermore we also need the acceleration vector:

$$\begin{aligned} \mathbf{r}'' &= \frac{d}{dt} (v_r \hat{r} + v_\theta \hat{\theta}), \\ &= v_r' \hat{r} + v_r \hat{\theta} \theta' + v_\theta' \hat{\theta} + v_\theta (-\hat{r} \theta'), \\ &= \left( v_r' - \frac{v_\theta^2}{r} \right) \hat{r} + \left( \frac{v_r v_\theta}{r} + v_\theta' \right) \hat{\theta}. \end{aligned}$$

Using Newton's second law, we then know that this acceleration vector must be equal to:

$$\mathbf{r}'' = \left( v_r' - \frac{v_\theta^2}{r} \right) \hat{r} + \left( \frac{v_r v_\theta}{r} + v_\theta' \right) \hat{\theta} = \frac{-GM}{r^2} \hat{r} + \Gamma.$$

Where  $\Gamma = (\hat{r} \sin(\phi) + \hat{\theta} \cos(\phi))\Gamma$ , see figure 4.1b. This means we have the equations of motion as follows:

$$\begin{aligned} r' &= v_r, \\ \theta' &= \frac{v_\theta}{r}, \\ v_r' &= \frac{v_\theta^2}{r} - \frac{GM}{r^2} + \Gamma \sin(\phi), \\ v_\theta' &= -\frac{v_r v_\theta}{r} + \Gamma \cos(\phi). \end{aligned}$$

Thus giving us the following Hamiltonian:

$$H(t) = \frac{1}{2}\Gamma^2 + \lambda_1 v_r + \lambda_2 \frac{v_\theta}{r} + \lambda_3 \left( \frac{v_\theta^2}{r} - \frac{GM}{r^2} + \Gamma \sin(\phi) \right) + \lambda_4 \left( -\frac{v_r v_\theta}{r} + \Gamma \cos(\phi) \right).$$

Same as before, we want to optimize (minimize)  $H$ , with respect to  $\Gamma$  and  $\phi$ . Therefore we solve  $\frac{dH}{d\phi} = 0$ , giving us:

$$\cos(\phi) = \frac{-\lambda_4}{\sqrt{\lambda_3^2 + \lambda_4^2}}, \quad \sin(\phi) = \frac{-\lambda_3}{\sqrt{\lambda_3^2 + \lambda_4^2}}.$$

And  $\frac{dH}{d\Gamma} = 0$  gives us:

$$\Gamma = \frac{\lambda_3^2}{\sqrt{\lambda_3^2 + \lambda_4^2}} + \frac{\lambda_4^2}{\sqrt{\lambda_3^2 + \lambda_4^2}} = \sqrt{\lambda_3^2 + \lambda_4^2}.$$

Substituting these into the Hamiltonian reduces it to:

$$H(t) = -\frac{1}{2}(\lambda_3^2 + \lambda_4^2) + \lambda_1 v_r + \lambda_2 \frac{v_\theta}{r} + \lambda_3 \left( \frac{v_\theta^2}{r} - \frac{GM}{r^2} \right) - \lambda_4 \frac{v_r v_\theta}{r}.$$

Using the same method as we did in the previous chapter, we obtain eight ODEs:

$$\begin{aligned} r' &= v_r, & \theta' &= \frac{v_\theta}{r}, & v_r' &= -\lambda_3 + \frac{v_\theta^2}{r} - \frac{GM}{r^2}, & v_\theta' &= -\lambda_4 - \frac{v_r v_\theta}{r}, \\ \lambda_1' &= \lambda_2 \frac{v_\theta}{r^2} + \lambda_3 \left( \frac{v_\theta^2}{r^2} - \frac{2GM}{r^3} \right) - \lambda_4 \frac{v_r v_\theta}{r^2}, & \lambda_2' &= 0, & \lambda_3' &= -\lambda_1 + \lambda_4 \frac{v_\theta}{r}, & \lambda_4' &= \frac{\lambda_2}{r} - \frac{2\lambda_3 v_\theta}{r} + \frac{\lambda_4 v_r}{r}. \end{aligned}$$

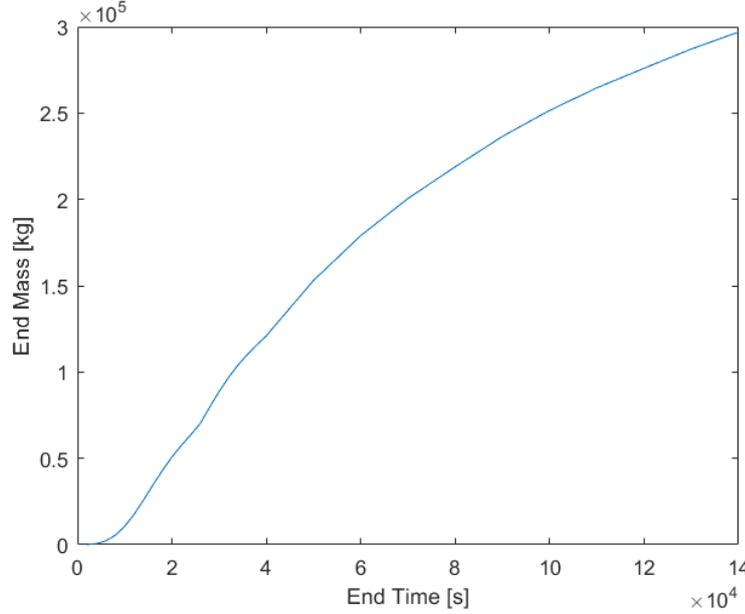
We can no longer solve these differential equations analytically. Therefore we need to numerically solve these equations. The program that has been chosen for this task is Matlab [22]. Furthermore we will be using the built-in command BVP4C [13], which solves boundary value problems for ordinary differential equations. In order to use this however, we again need certain begin and end conditions which will depend on the destination. We will be looking at three different trajectories. As decided in 3.4.1, for every trajectory we will take the initial mass of the rocket to be equal to  $m(t_0) = 5 \cdot 10^5 \text{ kg}$ .

#### 4.1.1 LEO to GEO

As stated before, for the first trajectory we are interested in flying from a Low Earth Orbit (LEO), to a Geostationary Earth Orbit (GEO) [4]. A Low Earth Orbit in this case is chosen to be 300 kilometres above the Earth's surface. The Earth's core is chosen as the centre of mass which means that we want to fly from a radius of 6,678  $km$  to around 42,164  $km$ . Furthermore  $\theta(0) = 0$ , because we want to launch from a point on the  $x$ -axis. Lastly, we choose  $v_r(0) = 0$  and  $v_\theta(0) = \sqrt{\frac{\mu_\oplus}{r(0)}}$  because we start in a circular orbit at radius  $r(0)$ . This was calculated by once more solving  $F_g = F_c$ , only we used  $F_g = \mu_\oplus \frac{M_1}{r(0)}$  instead of  $F_g = M_1 \cdot g$  as the latter equation only holds if the rocket is very close to the Earth. Here  $M_1$  is again the mass of the rocket. We also need end conditions as we have eight differential equations to solve. The end conditions for  $r$  is clear, the angle  $\theta$  is free, therefore  $\lambda_2(t_f) = 0$ . From the last section we know that  $\lambda_2' = 0$ , therefore  $\lambda_2 = 0$  for every  $t$ . Lastly we choose that we are again in a circular orbit at  $r(t_f)$  and thus  $v_r(t_f) = 0$  and  $v_\theta(t_f) = \sqrt{\frac{\mu_\oplus}{r(t_f)}}$ . It was chosen that instead of using the unit metres for distance, we will be using kilometres. For time, we will keep working in the SI-unit, seconds. All variables such as  $\mu_\oplus$ , which in SI-units is measured in  $[\frac{m^3}{s^2}]$ , have naturally been converted accordingly. So in overview we have the following boundary conditions of the problem:

$$\begin{aligned} t_0 &= 0 \text{ [s]}, & r(0) &= 6678 \text{ [km]}, & \theta(0) &= 0 \text{ [rad]}, & v_r(0) &= 0 \text{ [} \frac{km}{s} \text{]}, & v_\theta(0) &= \sqrt{\frac{\mu_\oplus}{r(0)}} \text{ [} \frac{km}{s} \text{]}, \\ t_f &= T \text{ [s]}, & r(T) &= 42164 \text{ [km]}, & \lambda_2(T) &= 0 \text{ [rad]}, & v_r(T) &= 0 \text{ [} \frac{km}{s} \text{]}, & v_\theta(T) &= \sqrt{\frac{\mu_\oplus}{r(T)}} \text{ [} \frac{km}{s} \text{]}. \end{aligned}$$

In the previous chapter we used the Hamiltonian function and the condition  $H(T) = 0$ , to find an explicit value for the optimal end time,  $T$ , using the equations of our lambdas. However, now it is not possible to analytically find an expression for our lambdas. Therefore we chose to take this problem on differently. Instead we varied  $T$  and observed how the Hamiltonian and also end mass,  $m(t_f) = m(T)$ , changed depending on the end time used. For our trajectory from LEO to GEO we obtained the following graph for  $m(T)$ :

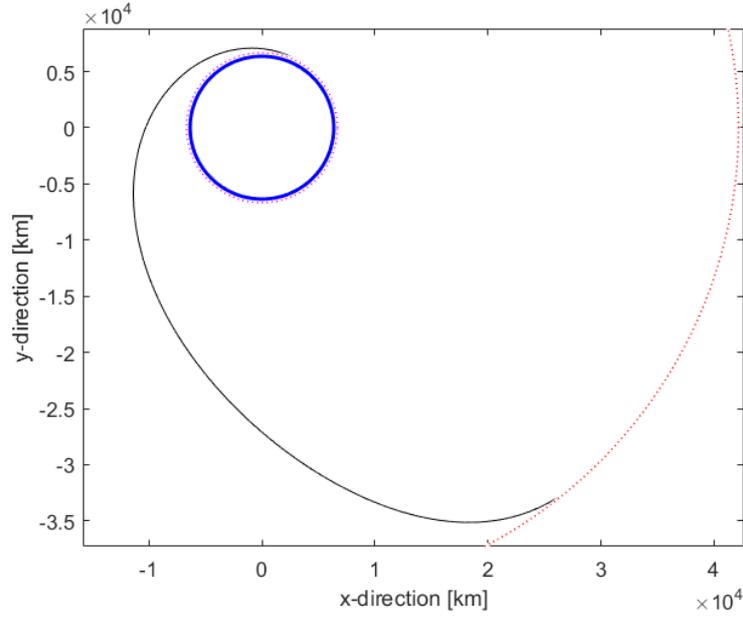


**Figure 4.2:** End mass of the rocket when flying from LEO to GEO depending on end time chosen.

What can be seen from this figure is that the larger the end time was chosen, the higher the end mass was. Therefore we can conclude that it is most economical to fly for as long as possible. However, in reality there comes a time where the advantages of mass-saving are outweighed by the time it takes. Therefore for the sake of making a graph of the trajectory, the end time that was chosen was to be equal to the end time of the Hohmann Transfer Orbit of this trajectory. This is equal to:

$$TOF = \pi \sqrt{\frac{\left(\frac{6678 \cdot 10^3 + 42164 \cdot 10^3}{2}\right)^3}{\mu_{\oplus}}} \approx 1.899 \cdot 10^4 \text{ s.}$$

Using this end time, we acquire the trajectory in figure 4.3. In this figure, Earth is denoted by the blue circle and the black line indicates the trajectory of the spacecraft. Also the LEO orbit (magenta) and GEO orbit (red) can be seen in the graph. For this trajectory, the payload mass (end mass) of the spacecraft is  $m(T) \approx 4.70 \cdot 10^4 \text{ kg}$ .



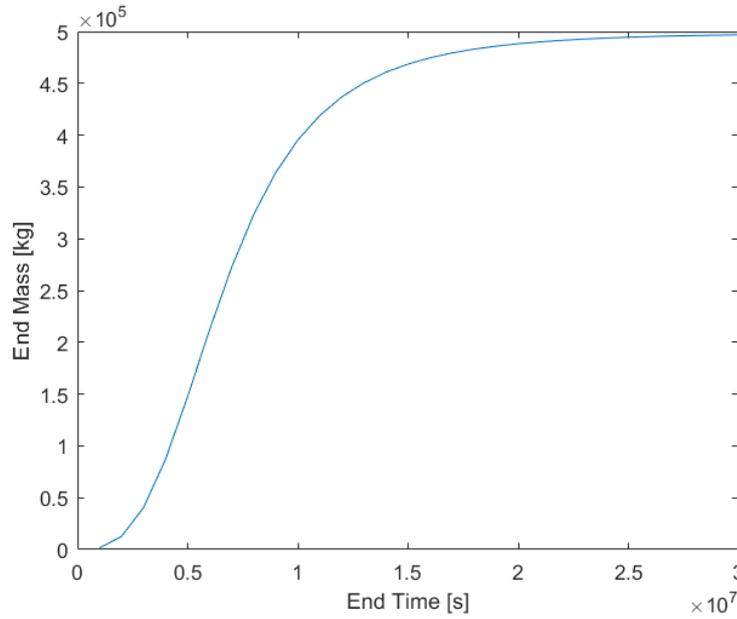
**Figure 4.3:** Trajectory when flying from a Low Earth Orbit (magenta dotted line) to a Geostationary Earth Orbit (red dotted line). The circumference of the Earth is denoted by the blue circle.

#### 4.1.2 Earth to Mars

Now that we know how to fly from a Low Earth Orbit to a Geostationary Earth Orbit, the equations of motion remain the same. When wanting to fly to a different destination, the only thing that needs to be altered are the boundary conditions. In this case, we want to start at Earth orbit and fly to a Mars orbit. Furthermore, the Sun is put at the centre and is also the centre of mass, whilst the Earth is on the  $x$ -axis at  $r(0)$  distance from the Sun. For now we still use kilometres as the unit for distance and seconds as the unit for time. This means we obtain the following boundary conditions:

$$\begin{aligned}
 t_0 = 0 [s], \quad r(0) = 1.496 \cdot 10^8 [km], \quad \theta(0) = 0 [rad], \quad v_r(0) = 0 \left[ \frac{km}{s} \right], \quad v_\theta(0) = \sqrt{\frac{\mu_\odot}{r(0)}} \left[ \frac{km}{s} \right], \\
 t_f = T [s], \quad r(T) = 2.279 \cdot 10^8 [km], \quad \lambda_2(T) = 0 [rad], \quad v_r(T) = 0 \left[ \frac{km}{s} \right], \quad v_\theta(T) = \sqrt{\frac{\mu_\odot}{r(T)}} \left[ \frac{km}{s} \right].
 \end{aligned}$$

It is important to note that from now on we will be taking  $\mu_\odot$ , instead of  $\mu_\oplus$  as the Sun is now the centre of mass. Other than that, we use the same exact method as before. We take several values of the end time,  $T$ , and look for which  $T$ , the end mass is largest. This results in the following graph:

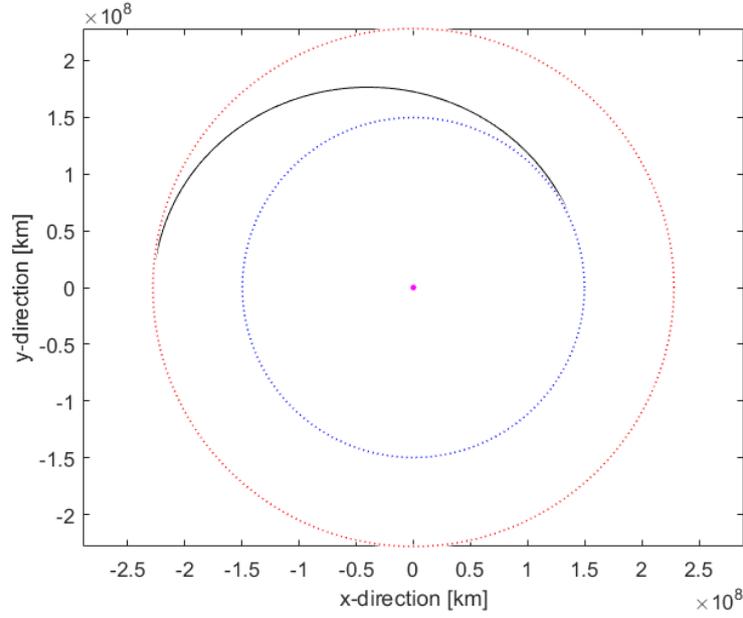


**Figure 4.4:** End mass for the trajectory from Earth to Mars, depending on end time.

What is notable when comparing this graph to figure 4.2 from the previous section, is that in this graph we can clearly see that at around  $1 \cdot 10^7$  seconds the gradient starts to decrease and the end mass seems to converge to  $5 \cdot 10^5 \text{ kg}$ , which was the initial mass. However in figure 4.2 this was not the case. This could be because we have not calculated the end mass for a large enough end time, however the point from which the gradient starts to decrease in figure 4.4, is much lower than the TOF of the Hohmann Transfer Orbit. This TOF is namely equal to:

$$TOF = \pi \sqrt{\frac{\left(\frac{1.49598023 \cdot 10^{11} + 2.279392 \cdot 10^{11}}{2}\right)^3}{\mu_{\odot}}} \approx 2.2366 \cdot 10^7 \text{ s.}$$

But when looking back at figure 4.2, we see that the end mass still is significantly higher if we would take a higher end time than the TOF for the LEO to GEO trajectory. This difference is most likely because the gravitational pull from the Earth is much higher when flying from LEO to GEO than the gravitational pull from the Sun when flying from the Earth to Mars. Now when taking the TOF of the Hohmann Transfer Orbit as the end time, the trajectory when flying from the Earth to Mars can be found in figure 4.5. In this graph the Sun is denoted by the magenta dot, the Earth's orbit can be seen as the blue dotted line and Mars is denoted by the red dotted line. The black line is once again the trajectory of the spacecraft. This end time then corresponds to a payload mass of  $m(T) \approx 4.915 \cdot 10^5 \text{ kg}$ .



**Figure 4.5:** Trajectory when flying from an Earth orbit (blue dotted line) to a Mars orbit (red dotted line) with the Sun (magenta) as the centre of mass.

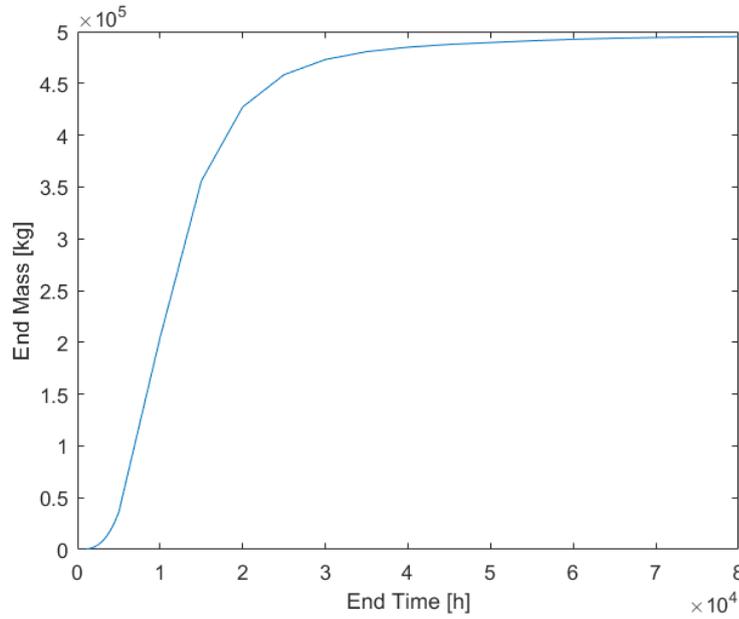
### 4.1.3 Earth to Saturn

Lastly we want to take a look at the trajectory when flying from the Earth to Saturn. As before, the only part that has to be altered are the begin and end conditions, but this time also the units will be changed. The Sun is the centre of mass and again placed in the origin. Furthermore the Earth is put on the  $x$ -axis at a distance of  $r(0)$  from the Sun. Now instead of working in kilometres and seconds as we did in the previous two sections, we decided to work with gigametres (i.e.  $10^9$  metres) and hours for respectively the units of distance and time. This means that instead of taking  $\mu_{\odot} = 1.32712440018 \cdot 10^{20} [\frac{m^3}{s^2}]$ , we have  $\mu_{\odot} = \frac{1.32712440018 \cdot 10^{20} \cdot (10^{-9})^3}{(1/3600)^2} \approx 1.716065223 \frac{Gm^3}{h^2}$ . Naturally all other variables were also converted accordingly. This results in the boundary following conditions:

$$t_0 = 0 [h], \quad r(0) = 1.496 \cdot 10^2 [Gm], \quad \theta(0) = 0 [rad], \quad v_r(0) = 0 \left[ \frac{Gm}{h} \right], \quad v_{\theta}(0) = \sqrt{\frac{\mu_{\odot}}{r(0)}} \left[ \frac{Gm}{h} \right],$$

$$t_f = T [h], \quad r(T) = 1.43353 \cdot 10^3 [Gm], \quad \lambda_2(T) = 0 [rad], \quad v_r(T) = 0 \left[ \frac{Gm}{h} \right], \quad v_{\theta}(T) = \sqrt{\frac{\mu_{\odot}}{r(T)}} \left[ \frac{Gm}{h} \right].$$

Now we can, again, see how the end mass of the rocket changes depending on the end time. This results in figure 4.6:

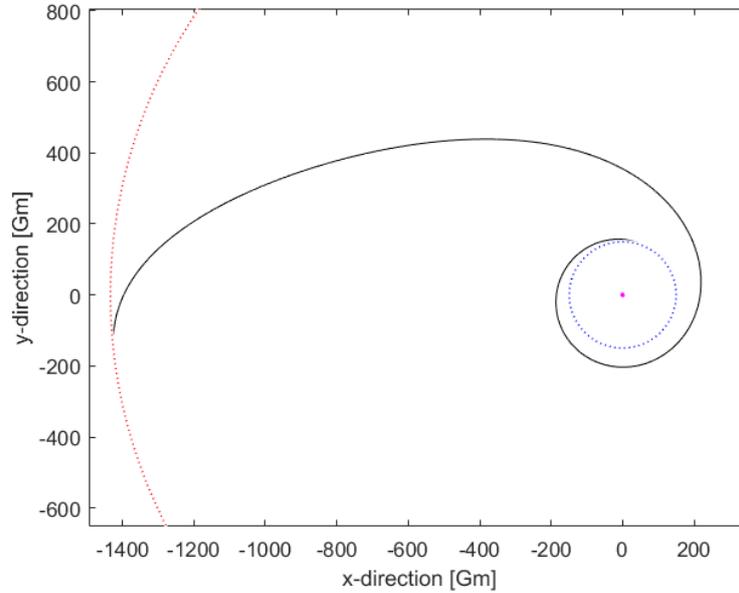


**Figure 4.6:** End mass for the trajectory from Earth to Saturn, depending on end time.

From this graph we can surmise that our conclusions from the previous two sections still hold. Furthermore we note that this graph is very similar to figure 4.4. In this case around  $2 \cdot 10^4 h$  the gradient starts to decrease and it seems to converge to an end mass of  $5 \cdot 10^5 kg$ . Also, we again make a graph of the trajectory where we take the end time as the TOF of the Hohmann transfer orbit, thus:

$$TOF = \pi \sqrt{\frac{\left(\frac{1.49598023 \cdot 10^{11} + 1.43353 \cdot 10^{12}}{2}\right)^3}{\mu_{\odot}}} \approx 1.9205 \cdot 10^8 s \approx 53348 h.$$

Taking this end time, we obtain the trajectory in figure 4.7. Here the Earth's orbit is seen as the blue dotted line, Saturn's orbit can be seen as the red dotted line. The Sun is again denoted by the magenta dot and the trajectory is denoted by the black line. With this trajectory, the payload mass of the spacecraft will be equal to  $m(T) \approx 4.904 \cdot 10^5 kg$ .



**Figure 4.7:** Trajectory when flying from an Earth orbit (blue dotted line) to a Saturn orbit (red dotted line) with the Sun (magenta) as the centre of mass.

## 4.2 Comparing to Hohmann Transfer Orbit

Now that we know the propellant costs of the trajectories for power limited rockets, we need to calculate the propellant costs for our liquid bipropellant rocket that uses a Hohmann Transfer Orbit. This has been done using section 2.3.2 and with the help of Matlab, a little program was made that easily calculates the  $\Delta v$  necessary and uses it to calculate the end mass,  $m(t_f)$ . The initial mass was chosen, same as before, to be equal to  $m(t_0) = 5 \cdot 10^5 \text{ kg}$ . The results from this, as well as the results from the last section, can be seen in overview in the following table:

**Table 4.1:** Hohmann Transfer Orbit final mass versus VASIMR trajectory final mass using the same end time.

Trajectory	Final mass Hohmann [kg]	TOF [s]	Final mass VASIMR [kg]
LEO GEO	$2.0904 \cdot 10^5$	$1.8990 \cdot 10^4$	$4.7004 \cdot 10^4$
Earth Mars	$1.4280 \cdot 10^5$	$2.2366 \cdot 10^7$	$4.9153 \cdot 10^5$
Earth Saturn	$1.4718 \cdot 10^4$	$1.9205 \cdot 10^8$	$4.9039 \cdot 10^5$

What we can see from this table is that the VASIMR rocket is better for longer trajectories. We can clearly see that when flying a short distance such as LEO to GEO, the liquid bipropellant rocket using liquid oxygen and liquid hydrogen, has around 40% of the initial mass left. Whilst the VASIMR rocket only has around 9.4% of the initial mass left. When flying a moderate distance such as from the Earth to Mars, the results of the chemical rocket and the VASIMR rocket are quite close, with the VASIMR rocket being slightly better. Yet when flying to very large distances such as from the Earth to Saturn or even further, it is quite clear that the VASIMR rocket is much more fuel-efficient. This is in accordance with what we expected. From this we can presume that power limited rockets are suitable for interplanetary or even interstellar missions.

Even so, something else we can take a look at is the total energy that is necessary for each trajectory. For the power limited rockets, this can easily be calculated as  $E = P \cdot t$ . Knowing  $P = 59 \text{ MW}$  and  $t$  is equal to the end time,  $T$ , the total energy can be calculated. For the Hohmann Transfer Orbit we use the following formula:  $E = \frac{m(t_0) - m(t_f)}{2} \cdot c^2$  as this is the total combustion energy. Remember that  $c$  is the exhaust velocity of the rocket and thus equal to  $c = I_{sp} \cdot g$ , where  $g = 9.81 \frac{m}{s^2}$  and in section 2.3.3 we had chosen  $I_{sp}$  to be equal to 455 seconds. The energy consumption for each trajectory as well as rocket type can be seen in the following table:

**Table 4.2:** Total energy consumption for each trajectory.

Trajectory	Hohmann Transfer Orbit [J]	VASIMR Trajectory [J]
LEO GEO	$2.8984 \cdot 10^{12}$	$1.1204 \cdot 10^{12}$
Earth Mars	$3.5583 \cdot 10^{12}$	$1.3196 \cdot 10^{15}$
Earth Saturn	$4.8342 \cdot 10^{12}$	$1.3312 \cdot 10^{16}$

It can be seen from this table the energy consumption for the LEO GEO trajectory with the Hohmann Transfer Orbit is higher than the energy consumption for the VASIMR rocket. For the other two trajectories, the energy consumption is much lower for the Hohmann Transfer Orbit than for the VASIMR trajectory. However, something which is interesting to note is that the energy consumption for the Hohmann Transfer Orbit increases more, the longer the trajectory is. For the VASIMR rocket on the other hand, the energy consumption also increases, but the increase is smaller the longer the trajectory is. This means that when the trajectory is long enough, the VASIMR trajectory will probably at one point start costing less energy than a Hohmann Transfer Orbit. This again supports the hypothesis that the power limited rockets are better for deep space missions.

## Chapter 5

# Trajectory to Saturn using a Gravity Assist

In this chapter we will be taking a look at a flyby, also known as a gravity assist. Gravity assists are used for many missions that travel to Jupiter or further planets. They are especially used because a gravity assist reduces propellant use, as well as flight time [20]. For a gravity assist, we will need to take the gravitational pull of another planet and not only the gravitational pull of the Sun, into account. This concept is often referred to as a three-body problem [21]. In reality, missions often use multiple flybys via several planets. Examples are the Cassini mission or Voyager 1 and Voyager 2 [20]. For example the Cassini mission used a gravity assist twice from Venus, once from the Earth and once from Jupiter before flying on to Saturn [16].

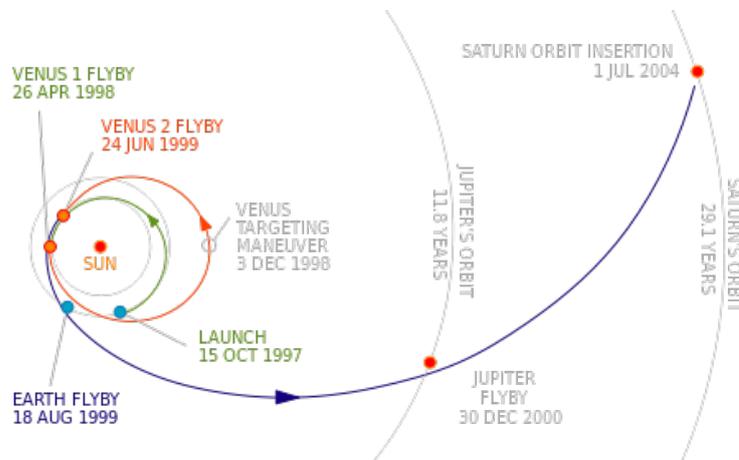


Figure 5.1: Gravity assists used by the Cassini mission. [16]

### 5.1 Gravity Assist: Jupiter

However in our case it would be a bit ambitious to try and recreate this trajectory as we would need to take into account the gravitational pull of not only Venus, but also from the Earth, Jupiter and the Sun. In our case we have decided to take a look at a gravity assist from Jupiter when flying from the Earth to Saturn. Jupiter is the largest planet in our Solar system and therefore the largest effect can be seen when using its gravitational pull. In the previous chapter we already have taken a look at the propellant use for the trajectory from the Earth to Saturn and we are interested in comparing these results to the propellant use when using the flyby.

The concept of a gravity assist is that the spacecraft flies by a planet and uses the gravitational pull of the planet to achieve a higher velocity. The key to understanding why a gravity assist works, is to consider two different points of view, i.e. reference frames. One of the reference frames is known as the inertial frame. In this frame the Sun is put at the centre with the planets rotating around it. But because we are interested in flying by Jupiter, which rotates around the Sun, we will be working in the rotating reference frame. In this frame, we look at the planet's point of view, where Jupiter's position is fixed. Now when flying by Jupiter in the rotating reference frame, we only see a perturbation of the trajectory. The planet speeds up on approach, however it also slows down as it departs. When consequently converting back to the inertial frame, we can see that the spacecraft indeed gets a velocity boost of the encounter with Jupiter [20]. This is due to the fact that Jupiter rotates around the Sun.

## 5.2 Rotating Reference Frame

When working in the rotating reference frame, a few things change for the equations of motion. We are still working with polar coordinates, however the additional gravitational pull of Jupiter has to be taken into account, as well as the rotation of Jupiter. This results in the following equations:

$$\begin{aligned} r' &= v_r, \\ \theta' &= \frac{v_\theta}{r}, \\ v_r' &= \frac{v_\theta^2}{r} + 2v_\theta\Omega + \Omega^2 r - \frac{\mu_\odot}{r^2} - \mu_J \frac{r - R \cos(\theta)}{(r^2 + R^2 - 2rR \cos(\theta))^{3/2}} - \Gamma \sin(\phi), \\ v_\theta' &= -\frac{v_\theta v_r}{r} - 2\Omega v_r - \mu_J \frac{R \sin(\theta)}{(r^2 + R^2 - 2rR \cos(\theta))^{3/2}} - \Gamma \cos(\phi). \end{aligned}$$

In these equations  $\Omega$  is the angular velocity of Jupiter in unit hours, so  $\Omega = \frac{2\pi}{4332.53 \cdot 24}$ . Also,  $R$  is the distance of Jupiter from the Sun and Jupiter is fixed on the  $x$ -axis. Because Jupiter will be seen as a point mass, we will be faced with an additional problem. The spacecraft will want to fly as close to Jupiter as possible, because then the velocity gain will be highest. However, if we were to fly infinitely close to Jupiter, then the term  $(r^2 + R^2 - 2rR \cos(\theta))$  will approximate zero. This would cause problems in our program as Matlab will start to divide zero by zero. Also, in reality we would not fly infinitely close to Jupiter as this would mean we are flying 'through' the planet, which would cause us to crash. Therefore, we will need a way to make sure the spacecraft does not fly too close to Jupiter. The method we have chosen as the most realistic way to do this, is by adding a friction force. This friction force can be interpreted as a simulation of Jupiter's atmosphere. The atmosphere will slow down the spacecraft as it comes too close to Jupiter. Through this method, the spacecraft will not come too close to Jupiter as it wants to use Jupiter's gravitational pull, without being slowed down. This should mean that the program will not encounter the problems as before.

### 5.2.1 Simulating Jupiter's Atmosphere

Now we only need a way to simulate Jupiter's atmosphere. What we want is to slow down the spacecraft whenever it comes too close to Jupiter. Thus we will need an additional term in our equations for  $v_r$  and  $v_\theta$ . This term was chosen to be:  $-\mathbf{v} C e^{-(r-r_J)^2/D^2}$ . Here  $C$  is the drag coefficient and  $D$  is a sphere of influence. This term makes sure that the closer we fly by Jupiter, the more the spacecraft is slowed down. The drag coefficient  $C$  was chosen to be around 0.05

and initially we wanted to take the radius of Jupiter as its sphere of influence. However, the time step which we would need to take in Matlab when Jupiter is this size, would result in a time step of  $dt = \sqrt{\frac{D^3}{\mu_{\uparrow}}} = \sqrt{\frac{0.069^3}{0.0016418575}} \approx 1.6 \cdot 10^3 \text{ s} < 1 \text{ h}$ . This is not practical in the program as it would take too much time for each run. This led to the conclusion to take the radius of Jupiter the same as the Sun's while still taking the same mass as before. This results in a time step of  $\sqrt{\frac{0.7^3}{0.0016418575}} \approx 14 \text{ h}$ . Therefore it was chosen to take  $D = 0.7 \text{ Gm}$  as this reduces computation time.

Thus finally we have our equations of motion, where for the sake of brevity we define  $b = |\mathbf{r} - \mathbf{r}_{\uparrow}| = \sqrt{r^2 + R^2 - 2rR \cos(\theta)}$ .

$$\begin{aligned} r' &= v_r, \\ \theta' &= \frac{v_{\theta}}{r}, \\ v_r' &= \frac{v_{\theta}^2}{r} + 2v_{\theta}\Omega + \Omega^2 r - \frac{\mu_{\odot}}{r^2} - \mu_{\uparrow} \frac{r - R \cos(\theta)}{b^3} - \Gamma \sin(\phi) - v_r C e^{-b^2/D^2}, \\ v_{\theta}' &= -\frac{v_{\theta} v_r}{r} - 2\Omega v_r - \mu_{\uparrow} \frac{R \sin(\theta)}{b^3} - \Gamma \cos(\phi) - v_{\theta} C e^{-b^2/D^2}. \end{aligned}$$

### 5.3 Effect of Jupiter on the Trajectory

With the equations of motion acquired in the previous section, we can use the Hamiltonian and obtain our eight ODEs as we did in the previous chapters.

$$\begin{aligned} H(t) &= \frac{1}{2}\Gamma^2 + \lambda_1 v_r + \lambda_2 \frac{v_{\theta}}{r} + \lambda_3 \left( \frac{v_{\theta}^2}{r} + 2v_{\theta}\Omega + \Omega^2 r - \frac{\mu_{\odot}}{r^2} - \mu_{\uparrow} \frac{r - R \cos(\theta)}{b^3} - \Gamma \sin(\phi) \right. \\ &\quad \left. - v_r C e^{-b^2/D^2} \right) + \lambda_4 \left( -\frac{v_{\theta} v_r}{r} - 2\Omega v_r - \mu_{\uparrow} \frac{R \sin(\theta)}{b^3} - \Gamma \cos(\phi) - v_{\theta} C e^{-b^2/D^2} \right). \end{aligned}$$

Wanting to optimize  $\Gamma$  and  $\phi$  we obtain:

$$\cos(\phi) = \frac{\lambda_4}{\sqrt{\lambda_3^2 + \lambda_4^2}}, \quad \sin(\phi) = \frac{\lambda_3}{\sqrt{\lambda_3^2 + \lambda_4^2}}.$$

And  $\frac{dH}{dt} = 0$  gives us:

$$\Gamma = \frac{\lambda_3^2}{\sqrt{\lambda_3^2 + \lambda_4^2}} + \frac{\lambda_4^2}{\sqrt{\lambda_3^2 + \lambda_4^2}} = \sqrt{\lambda_3^2 + \lambda_4^2}.$$

After elimination, the Hamiltonian reduces to:

$$\begin{aligned} H(t) &= -\frac{1}{2}(\lambda_3^2 + \lambda_4^2) + \lambda_1 v_r + \lambda_2 \frac{v_{\theta}}{r} + \lambda_3 \left( \frac{v_{\theta}^2}{r} + 2v_{\theta}\Omega + \Omega^2 r - \frac{\mu_{\odot}}{r^2} - \mu_{\uparrow} \frac{r - R \cos(\theta)}{b^3} - v_r C e^{-b^2/D^2} \right) \\ &\quad + \lambda_4 \left( -\frac{v_{\theta} v_r}{r} - 2\Omega v_r - \mu_{\uparrow} \frac{R \sin(\theta)}{b^3} - v_{\theta} C e^{-b^2/D^2} \right). \end{aligned}$$

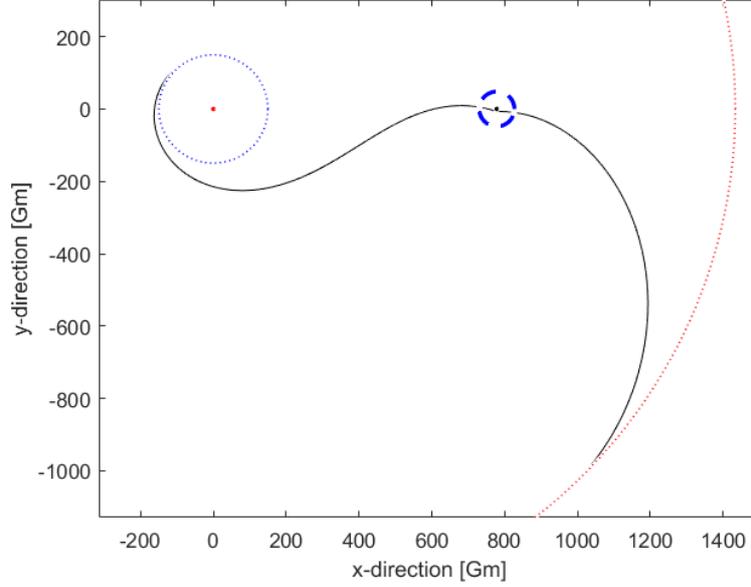
After some calculations, we now have our eight ordinary differential equations:

$$\begin{aligned}
r' &= v_r, \\
\theta' &= \frac{v_\theta}{r}, \\
v_r' &= \frac{v_\theta^2}{r} + 2v_\theta\Omega + \Omega^2 r - \frac{\mu_\odot}{r^2} - \mu_\gamma \frac{(r - R \cos(\theta))}{b^3} - \lambda_3 - v_r C e^{-b^2/D^2}, \\
v_\theta' &= -\frac{v_\theta v_r}{r} - 2\Omega v_r - \mu_\gamma \frac{(R \sin(\theta))}{b^3} - \lambda_4 - v_\theta C e^{-b^2/D^2}, \\
\lambda_1' &= \frac{\lambda_2 v_\theta}{r^2} - \lambda_3 \left( -\frac{v_\theta^2}{r^2} + \Omega^2 + \frac{2\mu_\odot}{r^3} - \frac{\mu_\gamma}{b^3} + \frac{3\mu_\gamma (r - R \cos(\theta))(2r - 2R \cos(\theta))}{2b^5} + \frac{v_r C (2r - 2R \cos(\theta)) e^{-b^2/D^2}}{D^2} \right) \\
&\quad - \lambda_4 \left( \frac{v_\theta v_r}{r^2} + \frac{3\mu_\gamma (R \sin(\theta))(2r - 2R \cos(\theta))}{2b^5} + \frac{v_\theta C (2r - 2R \cos(\theta)) e^{-b^2/D^2}}{D^2} \right), \\
\lambda_2' &= -\lambda_3 \left( -\frac{\mu_\gamma R \sin(\theta)}{b^3} + \frac{3\mu_\gamma (r - R \cos(\theta)) R r \sin(\theta)}{b^5} + \frac{2v_r C R r \sin(\theta) e^{-b^2/D^2}}{D^2} \right) \\
&\quad - \lambda_4 \left( -\frac{\mu_\gamma R \cos(\theta)}{b^3} + \frac{3\mu_\gamma R^2 r (\sin(\theta))^2}{b^5} + \frac{2v_\theta C R r \sin(\theta) e^{-b^2/D^2}}{D^2} \right), \\
\lambda_3' &= -\lambda_1 - \lambda_4 \left( -\frac{v_\theta}{r} - 2\Omega \right) + \lambda_3 C e^{-b^2/D^2}, \\
\lambda_4' &= -\frac{\lambda_2}{r} - \lambda_3 \left( \frac{2v_\theta}{r} + 2\Omega \right) + \frac{\lambda_4 v_r}{r} + \lambda_4 C e^{-b^2/D^2}.
\end{aligned}$$

As we stated before, we are interested in flying from the Earth to Saturn with a gravity assist from Jupiter. As we are solving this with BVP4C, which is a boundary value problem solver, we cannot specify where we would like to be ‘in between’. Therefore our begin conditions have to be chosen in such a way that we are indeed flying by Jupiter. This means that instead of taking the initial angle of our rocket to be equal to zero as we did before, we will now be varying it in order to see what the effect of Jupiter is on the trajectory. The other begin and end conditions stay the same as in the last chapter. This means we have:

$$\begin{aligned}
t_0 &= 0 [h], & r(0) &= 1.496 \cdot 10^2 [Gm], & \theta(0) &= \text{varied} [rad], & v_r(0) &= 0 \left[ \frac{Gm}{h} \right], & v_\theta(0) &= \sqrt{\mu_\odot r(0)} \left[ \frac{Gm}{h} \right], \\
t_f &= T [h], & r(T) &= 1.4333 \cdot 10^3 [Gm], & \lambda_2(T) &= 0 [rad], & v_r(T) &= 0 \left[ \frac{Gm}{h} \right], & v_\theta(T) &= \sqrt{\mu_\odot r(T)} \left[ \frac{Gm}{h} \right].
\end{aligned}$$

Yet, even with the additional friction term whenever we got too close when flying ‘over’ Jupiter, the program started making huge errors. However, something we did succeed in, was flying ‘underneath’ Jupiter as can be seen in the following graph.



**Figure 5.2:** Graph of the trajectory in the rotating reference frame. The Earth's orbit (blue dotted line), Saturn's orbit (red dotted line) and the trajectory (black line) can be seen. The Sun is denoted by the red dot in the origin and Jupiter can be seen at a distance  $R$  from the Sun on the  $x$ -axis. Around Jupiter its circle of influence can be seen as the blue dashed line. The spacecraft clearly experiences the gravitational pull of Jupiter inside this circle of influence.

This graph shows the trajectory of the rocket in the rotating reference frame, where we used the initial angle  $\theta(0) = 0.34\pi$ . Jupiter was fixed on the  $x$ -axis at a distance  $R$  from the Sun. We can clearly see from this graph that as the spacecraft enters Jupiter's circle of influence, it experiences the gravitational pull of Jupiter. This circle of influence was calculated in the following way [4]:

$$R_{SOI} \approx R \cdot \left( \frac{M_{J_+}}{M_{\odot}} \right)^{2/5} = 7.785 \cdot 10^{11} \cdot \left( \frac{1.898 \cdot 10^{27}}{1.989 \cdot 10^{30}} \right)^{2/5} \approx 4.81 \cdot 10^{10} \text{ m} = 48.1 \text{ Gm}.$$

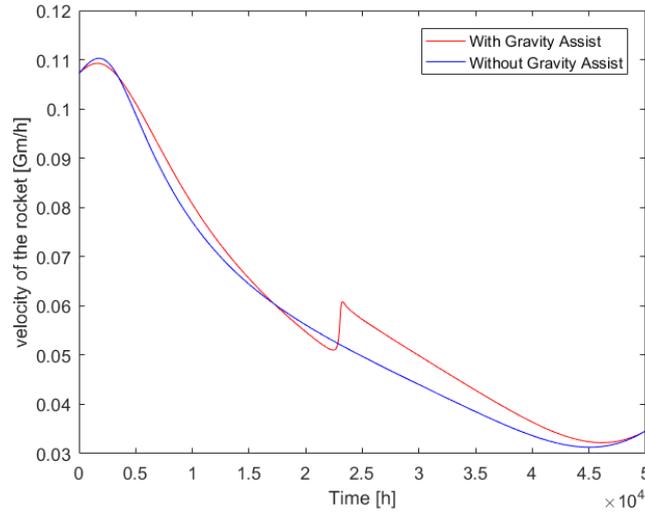
Normally this is called the sphere of influence but as we are working in a two-dimensional plane, we call it the circle of influence. What is important to note, is that we used an end time of 50,000 hours instead of 53,000 hours as we had in the previous chapter. This was because we had too many problems with Matlab when taking a larger end time. Using this end time however, we obtain the following end mass for the trajectory from Earth to Saturn with and without the gravity assist:

Trajectory	End time [h]	Payload Mass [kg]
With Gravity Assist	50,000	$4.896594 \cdot 10^5$
Without Gravity Assist	50,000	$4.893771 \cdot 10^5$

**Table 5.1:** Payload mass of the trajectory Earth to Saturn with and without a gravity assist.

From this table, we can see that using a gravity assist indeed saves propellant. The difference is not that significant, only around 280 kilograms of propellant were saved. It can also be seen from the following graph that the spacecraft indeed got a velocity boost from Jupiter. In this graph we show the velocity of the rocket using the gravity assist in red and without the gravity

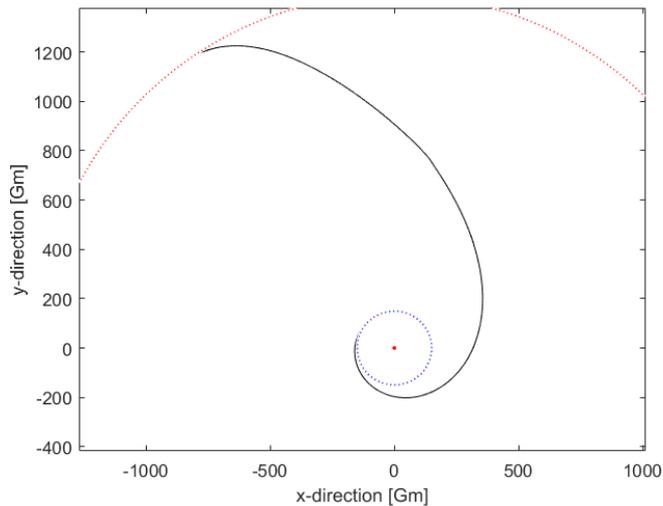
assist in blue. It can be clearly seen that the spacecraft's velocity was higher after the encounter with Jupiter.



**Figure 5.3:** The velocity of the spacecraft can be seen with and without using a gravity assist. Around  $2.3 \cdot 10^4$  hours, the spacecraft that uses the gravity assist obtains a velocity boost as it flies by Jupiter.

Also converting figure 5.2 back to the inertial frame, we can see a very small perturbation in the trajectory of the spacecraft. This can be seen in figure 5.4.

From this we can conclude that using a gravity assist is indeed beneficial when wanting to save propellant costs. Although we did not succeed in obtaining the desired graph, the velocity boost when flying underneath Jupiter is still enough to save propellant. When flying over Jupiter, the velocity boost as well as propellant savings, would likely be a lot higher.



**Figure 5.4:** The trajectory of the spacecraft as seen in the inertial frame. A small perturbation in the trajectory can be seen. Again the blue dotted line denotes the Earth's orbit, the red dotted line denotes Saturn's orbit and the trajectory is denoted by the black line. Also the Sun can be seen as the red dot in the origin.

## 5.4 Removing the Singularity

We also thought of an additional way to try to ensure the spacecraft would not have any trouble flying through Jupiter. The idea was to remove the singularity that we have in  $(R, 0)$ , the point where Jupiter is. Therefore, it was decided that an additional term is necessary to omit the singularity. This means our final equations of motion are:

$$\begin{aligned} r' &= v_r, \\ \theta' &= \frac{v_\theta}{r}, \\ v_r' &= \frac{v_\theta^2}{r} + 2v_\theta\Omega + \Omega^2 r - \frac{\mu_\odot}{r^2} - \mu_\gamma \frac{(r - R \cos(\theta))(1 - e^{-b^2/D^2})}{b^3} - \Gamma \sin(\phi) - v_r C e^{-b^2/D^2}, \\ v_\theta' &= -\frac{v_\theta v_r}{r} - 2\Omega v_r - \mu_\gamma \frac{(R \sin(\theta))(1 - e^{-b^2/D^2})}{b^3} - \Gamma \cos(\phi) - v_\theta C e^{-b^2/D^2}. \end{aligned}$$

This should mean it is no longer a problem to fly ‘through’ Jupiter as it makes sure that when we are flying through Jupiter, the acceleration becomes finite. However, as it is still more realistic if the rocket would not fly that close to Jupiter, we keep the equations for the atmospheric drag in our equations of motion.

Using these new equations of motion, we derived the Hamiltonian once more as well as the differential equations. For the Hamiltonian we obtained:

$$\begin{aligned} H(t) &= \frac{1}{2}\Gamma^2 + \lambda_1 v_r + \lambda_2 \frac{v_\theta}{r} + \lambda_3 \left( \frac{v_\theta^2}{r} + 2v_\theta\Omega + \Omega^2 r - \frac{\mu_\odot}{r^2} - \mu_\gamma \frac{(r - R \cos(\theta))(1 - e^{-b^2/D^2})}{b^3} - \Gamma \sin(\phi) \right. \\ &\quad \left. - v_r C e^{-b^2/D^2} \right) + \lambda_4 \left( -\frac{v_\theta v_r}{r} - 2\Omega v_r - \mu_\gamma \frac{(R \sin(\theta))(1 - e^{-b^2/D^2})}{b^3} - \Gamma \cos(\phi) - v_\theta C e^{-b^2/D^2} \right). \end{aligned}$$

In the same way as in the previous section we know that:

$$\cos(\phi) = \frac{\lambda_4}{\sqrt{\lambda_3^2 + \lambda_4^2}}, \quad \sin(\phi) = \frac{\lambda_3}{\sqrt{\lambda_3^2 + \lambda_4^2}}.$$

And also:

$$\Gamma = \frac{\lambda_3^2}{\sqrt{\lambda_3^2 + \lambda_4^2}} + \frac{\lambda_4^2}{\sqrt{\lambda_3^2 + \lambda_4^2}} = \sqrt{\lambda_3^2 + \lambda_4^2}.$$

After elimination, the Hamiltonian reduces to:

$$\begin{aligned} H(t) &= -\frac{1}{2}(\lambda_3^2 + \lambda_4^2) + \lambda_1 v_r + \lambda_2 \frac{v_\theta}{r} \\ &\quad + \lambda_3 \left( \frac{v_\theta^2}{r} + 2v_\theta\Omega + \Omega^2 r - \frac{\mu_\odot}{r^2} - \mu_\gamma \frac{(r - R \cos(\theta))(1 - e^{-b^2/D^2})}{b^3} - v_r C e^{-b^2/D^2} \right) \\ &\quad + \lambda_4 \left( -\frac{v_\theta v_r}{r} - 2\Omega v_r - \mu_\gamma \frac{(R \sin(\theta))(1 - e^{-b^2/D^2})}{b^3} - v_\theta C e^{-b^2/D^2} \right). \end{aligned}$$

Now the eight ODEs are as follows:

$$\begin{aligned}
r' &= v_r, \\
\theta' &= \frac{v_\theta}{r}, \\
v_r' &= \frac{v_\theta^2}{r} + 2v_\theta\Omega + \Omega^2 r - \frac{\mu_\odot}{r^2} - \mu_{\Upsilon} \frac{(r - R \cos(\theta))(1 - e^{-b^2/D^2})}{b^3} - \lambda_3 - v_r C e^{-b^2/D^2}, \\
v_\theta' &= -\frac{v_\theta v_r}{r} - 2\Omega v_r - \mu_{\Upsilon} \frac{(R \sin(\theta))(1 - e^{-b^2/D^2})}{b^3} - \lambda_4 - v_\theta C e^{-b^2/D^2}, \\
\lambda_1' &= \frac{\lambda_2 v_\theta}{r^2} - \lambda_3 \left( -\frac{v_\theta^2}{r^2} + \Omega^2 + \frac{2\mu_\odot}{r^3} - \frac{\mu_{\Upsilon}(1 - e^{-b^2/D^2})}{b^3} + \frac{3\mu_{\Upsilon}(r - R \cos(\theta))(2r - 2R \cos(\theta))(1 - e^{-b^2/D^2})}{2b^5} \right. \\
&\quad \left. - \frac{\mu_{\Upsilon}(r - R \cos(\theta))(2r - 2R \cos(\theta))e^{-b^2/D^2}}{b^3 \cdot D^2} + \frac{v_r C(2r - 2R \cos(\theta))e^{-b^2/D^2}}{D^2} \right) \\
&\quad - \lambda_4 \left( \frac{v_\theta v_r}{r^2} + \frac{3\mu_{\Upsilon}(R \sin(\theta))(1 - e^{-b^2/D^2})(2r - 2R \cos(\theta))}{2b^5} - \frac{\mu_{\Upsilon} R \sin(\theta)(2r - 2R \cos(\theta))e^{-b^2/D^2}}{b^3 \cdot D^2} \right. \\
&\quad \left. + \frac{v_\theta C(2r - 2R \cos(\theta))e^{-b^2/D^2}}{D^2} \right), \\
\lambda_2' &= -\lambda_3 \left( -\frac{\mu_{\Upsilon} R \sin(\theta)(1 - e^{-b^2/D^2})}{b^3} + \frac{3\mu_{\Upsilon}(r - R \cos(\theta))Rr \sin(\theta)(1 - e^{-b^2/D^2})}{b^5} \right. \\
&\quad \left. - \frac{2\mu_{\Upsilon}(r - R \cos(\theta))Rr \sin(\theta)e^{-b^2/D^2}}{b^3 \cdot D^2} + \frac{2v_r C Rr \sin(\theta)e^{-b^2/D^2}}{D^2} \right) \\
&\quad - \lambda_4 \left( -\frac{\mu_{\Upsilon} R \cos(\theta)(1 - e^{-b^2/D^2})}{b^3} + \frac{3\mu_{\Upsilon} R^2 r (\sin(\theta))^2 (1 - e^{-b^2/D^2})}{b^5} - \frac{2\mu_{\Upsilon} R^2 (\sin(\theta))^2 r e^{-b^2/D^2}}{b^3 \cdot D^2} \right. \\
&\quad \left. + \frac{2v_\theta C Rr \sin(\theta)e^{-b^2/D^2}}{D^2} \right), \\
\lambda_3' &= -\lambda_1 - \lambda_4 \left( -\frac{v_\theta}{r} - 2\Omega \right) + \lambda_3 C e^{-b^2/D^2}, \\
\lambda_4' &= -\frac{\lambda_2}{r} - \lambda_3 \left( \frac{2v_\theta}{r} + 2\Omega \right) + \frac{\lambda_4 v_r}{r} + \lambda_4 C e^{-b^2/D^2}.
\end{aligned}$$

With these equations and the same boundary conditions as in the previous section, we tried to make a graph where we are flying ‘over’ Jupiter. Sadly enough, we were not successful. The most likely explanation is that the problem is very sensitive to the initial conditions. Even a very small alteration in the begin angle  $\theta(0)$  has a very large impact. Therefore the optimal trajectory cannot be found. Still we did manage to get some results in the previous section where we could see that using a gravity assist is beneficial. This is in accordance with what one would expect.

## Chapter 6

# Conclusion and Recommendations

In this thesis the goal was to find the most cost-effective trajectory for a power limited rocket. The power limited rocket of choice was the VASIMR rocket. In order to make an estimation of how cost-effective the trajectory was, we compared it to the most cost-effective trajectory of the thrust limited rocket. Therefore we first explained how the optimal trajectory of the thrust limited rocket is found. This uses two instantaneous velocity boosts to go from one circular orbit to another and is called the Hohmann Transfer Orbit. Furthermore, using the Tsiolkovsky rocket equation, we calculated the propellant costs for the Hohmann Transfer Orbit. Where we used the characteristics of the liquid bipropellant rocket using liquid oxygen and liquid hydrogen. Thus we used a specific impulse of 455 seconds. Then using Optimal Control Theory we calculated the optimal trajectory for the power limited rockets. After deriving the function for the mass of the rocket, we could also calculate the propellant costs for this trajectory. We looked at three different trajectories. The trajectory from a Low Earth Orbit to a Geostationary Earth Orbit, the trajectory from Earth to Mars and the trajectory from Earth to Saturn. After calculating the propellant costs for both types of rockets we could clearly see that the power limited rockets are suited for missions that cover great distances, such as interplanetary or even interstellar missions. When calculating the total energy for each trajectory, we also came to this conclusion. Additionally we calculated the end mass of the rocket for various different end times. From this we concluded that the power limited rocket becomes more fuel-efficient as travel time increases. But in order to make a comparison between the power and thrust limited rockets, we used the same end time for both. As the time of flight of the Hohmann Transfer Orbit is fixed, this end time was chosen. However, one thing that should be noted is that we did not take some things into account. For example when flying from the Earth to Saturn or from the Earth to Mars, we only took the gravitational pull of the Sun into account. For a realistic conclusion, the gravitational pull of the Earth as well as Saturn should be taken into account. This may show that the power limited rockets simply have too little thrust capabilities in order to get away from the Earth in the first place. Realistically, spaceflight missions will always need thrust limited rockets for the first stage of liftoff from the Earth as they can achieve enough thrust to get away from the Earth. This is also because of the atmospheric drag of the Earth, something we also did not take into account.

The last subject which was discussed in this thesis was the use of a gravity assist. We were interested to see if the gravity assist had a positive outcome on the cost-effectiveness of the trajectory. In reality they are commonly used for space missions with destinations beyond Mars. Therefore we took a second look at the Earth to Saturn trajectory where we used a gravity assist from Jupiter. This means that we not only took the gravitational pull of the Sun into account, but also of Jupiter. When flying close to Jupiter, we expected to see a peak in

the velocity of our rocket as it got a velocity boost. We were not able find the exact optimal trajectory. We came to the conclusion that the problems we encountered were due to the fact that the spacecraft would fly too close by Jupiter as Jupiter was seen as a point-mass in the program. This caused problems as Matlab would try to divide zero by zero. Therefore we decided that we needed a friction force in order to make sure the spacecraft did not come too close to Jupiter. We did this by simulating Jupiter's atmosphere. The closer the spacecraft came to Jupiter's centre, the more it would be slowed down. As the program was written in such way that the most cost-effective trajectory would be found, this helped to make sure the spacecraft did not come too close. Doing this, we obtained a graph where we flew 'underneath' Jupiter and its gravitational pull could clearly be seen as it perturbed the trajectory. Also from this we could see that the end mass of the rocket was slightly higher than when the trajectory was done without a gravity assist. Additionally we indeed saw that the rocket got a velocity boost after flying past Jupiter. However, we were not able to fly 'over' Jupiter. This would most likely result in a much higher velocity boost as well as save much more propellant. Therefore we tried to remove the singularity in Jupiter all-together. Unfortunately, this seemed to help little as we still got big errors whenever we tried to fly close to Jupiter. Even after removing the singularities in the differential equations, introducing smooth friction force as well as making the time steps sufficiently small, the system of differential equations remains too unstable. Small changes in the initial conditions have very large outcomes on the trajectory. This led to the conclusion that the optimal trajectory cannot be found.

To surmise, we can conclude that power limited rockets will most likely be recommended for interplanetary or even interstellar missions. Because of their low thrust capabilities they need a long time to built up speed, but they are very economical for long distances. Using a gravity assist will also most likely be beneficial when wanting to minimize propellant costs.

# Bibliography

- [1] Abrams, J.J. (Producer), & Lin, J (Director). (2016). Star Trek Beyond [Motion Picture]. United States: Paramount Pictures.
- [2] ESA. (2015, June 16). What is Electrical Propulsion. Retrieved from [http://www.esa.int/Our\\_Activities/Space\\_Engineering\\_Technology/What\\_is\\_Electric\\_propulsion](http://www.esa.int/Our_Activities/Space_Engineering_Technology/What_is_Electric_propulsion)
- [3] Hohmann, W. (1960). The Attainability of Heavenly Bodies. Retrieved from [https://ia800502.us.archive.org/12/items/nasa\\_techdoc\\_19980230631/19980230631.pdf](https://ia800502.us.archive.org/12/items/nasa_techdoc_19980230631/19980230631.pdf)
- [4] J Sellers, J. (2005). Understanding Space: an Introduction to Astronautics (3rd ed.). New York, USA: The McGraw-Hill Companies.
- [5] Keijzer, M. (2016, April). Optimale besturingstheorie [Hand-Out]. Delft University of Technology.
- [6] Kennedy, K. (Producer), & Abrams, J.J. (Director). (2015). Star Wars: The Force Awakens [Motion Picture]. United States: Walt Disney Studios Motion Pictures.
- [7] Kinberg, S. (Producer), & Scott, R (Director). (2015). The Martian [Motion Picture]. United States: 20th Century Fox.
- [8] Leafnode. (2007, April 3). Illustration of Hohmann transfer orbit [Illustration]. Retrieved from [https://commons.wikimedia.org/wiki/File:Hohmann\\_transfer\\_orbit.svg](https://commons.wikimedia.org/wiki/File:Hohmann_transfer_orbit.svg)
- [9] Levavasseur, D. (2004, January 1). The Tsiolkovsky formula. Retrieved from [http://www.astrosurf.com/levavasseur/tsiolkovsky/tsiolkovsky\\_fichiers/Tsiolkovsky.pdf](http://www.astrosurf.com/levavasseur/tsiolkovsky/tsiolkovsky_fichiers/Tsiolkovsky.pdf)
- [10] List of 6 Manned Moon Landings. (2012). Retrieved from <http://historylists.org/events/list-of-6-manned-moon-landings.html>
- [11] Longuski, J. M., Guzmán, J. J., & Prussing, J. E. (2014). General Theory of Optimal Rocket Trajectories. *Optimal Control with Aerospace Applications*. El Segundo, USA: Microcosm Press and Springer.4
- [12] Markus Hohenwarter et al. (2011). GeoGebra [Software]. Retrieved from <https://www.geogebra.org/?lang=nl>
- [13] Mathworks. (n.d.). bvp4c. Retrieved from <https://nl.mathworks.com/help/matlab/ref/bvp4c.html>
- [14] NASA. (n.d.). A Pictorial History of Rockets. Retrieved from [https://www.nasa.gov/pdf/153410main\\_Rockets\\_History.pdf](https://www.nasa.gov/pdf/153410main_Rockets_History.pdf)
- [15] NASA. (n.d.). Brief History of Rockets. Retrieved from [https://www.grc.nasa.gov/WWW/k-12/TRC/Rockets/history\\_of\\_rockets.html](https://www.grc.nasa.gov/WWW/k-12/TRC/Rockets/history_of_rockets.html)

- [16] NASA. (n.d.). Cassini Trajectory. Retrieved from <https://saturn.jpl.nasa.gov/resources/1776/>
- [17] NASA. (n.d.). First Satellites. Retrieved November 29, 2017, from <https://www.jpl.nasa.gov/jplhistory/early/firstsatellites.php>
- [18] NASA. (n.d.). Mars Science Laboratory. Retrieved from <https://mars.nasa.gov/msl/mission/spacecraft/>
- [19] Olympio, J. (2009). Optimisation and optimal control methods for planet sequence design of low-thrust Interplanetary transfer problems with gravity-assists. Retrieved from <https://pastel.archives-ouvertes.fr/tel-00365115>
- [20] Shortt, D. (2013, September 27). Gravity Assist. Retrieved from <http://www.planetary.org/blogs/guest-blogs/2013/20130926-gravity-assist.html>
- [21] The Editors of Encyclopædia Britannica. (1998, July 20). Three-body Problem. Retrieved December 13, 2017, from <https://www.britannica.com/topic/three-body-problem>
- [22] The MathWorks, Inc. (2011). MatLab Student Version with Simulink, Version r2016b [Software]. Retrieved from <http://www.mathworks.com/>
- [23] VASIMR. (n.d.). Retrieved November 29, 2017, from [http://www.projectrho.com/public\\_html/rocket/engine1ist.php#vasimr](http://www.projectrho.com/public_html/rocket/engine1ist.php#vasimr)
- [24] Wilson, J., & Dunbar, B. (2017, August 4). Journey to Mars Overview. Retrieved December 8, 2017, from <https://www.nasa.gov/content/journey-to-mars-overview>

# Appendices



# Appendix A

## Derivations

### A.1 Hamiltonian is Constant

We want to show that the Hamiltonian is constant. In order to show this we merely have to show that  $\frac{dH}{dt} = 0$ . For this we only assume that  $H$  does not directly depend on  $t$ , thus  $\frac{\partial H}{\partial t} = 0$ . Now if we want to show  $\frac{dH}{dt} = 0$ , we show:

$$\begin{aligned}\frac{dH}{dt} &= \frac{\partial H}{\partial t} + \frac{\partial H}{\partial \mathbf{x}} \dot{\mathbf{x}} + \frac{\partial H}{\partial \boldsymbol{\lambda}} \dot{\boldsymbol{\lambda}} + \frac{\partial H}{\partial \Gamma} \dot{\Gamma} + \frac{\partial H}{\partial \phi} \dot{\phi}, \\ &= 0 + \frac{\partial H}{\partial \mathbf{x}} \frac{\partial H}{\partial \boldsymbol{\lambda}} + \frac{\partial H}{\partial \boldsymbol{\lambda}} \cdot -\frac{\partial H}{\partial \mathbf{x}} + \frac{\partial H}{\partial \Gamma} \cdot 0 + \frac{\partial H}{\partial \phi} \cdot 0, \\ &= 0.\end{aligned}$$

And thus from this we can conclude that the Hamiltonian is constant.



# Appendix B

## Matlab Code

### B.1 Close to Earth's Surface

```
clear all
format long
h = 100*10^3; % Altitude to which we are flying [m]
R = 6378*10^3; % Radius of the Earth [m]
g = 9.81; % The Earth's gravitational pull at sea level [m/s^2]
T = sqrt( R/(2*g) * (1+sqrt(1+(144*h.^2)/R^2))); % Calculate optimal end time T for given altitude [s]
t = 0:1:T;
for i = 1:length(t)
    x(i) = (sqrt(g*R)*t(i)^2)/(2*T);
    y(i) = (3*h*t(i)^2)/T^2 - (2*h*t(i)^3)/T^3;
    vx(i) = (sqrt(g*(R+h))*t(i))/T;
    vy(i) = (6*h*t(i))/T^2 - (6*h*t(i)^2)/T^3;
end
figure(1)
plot(x,y) ; xlabel('x [m]') ; ylabel('y [m]')
```

```
clear all
format long
R = 6378*10^3; % Radius of the Earth [m]
g = 9.81; % The Earth's gravitational pull at sea level [m/s^2]
h = 100*10^3: 500: 36*10^6; % Altitude as a range to see how optimal end time changes [m]
for i = 1:length(h)
    T(i) = sqrt( R/(2*g) * (1+sqrt(1+(144*h(i).^2)/R^2)));
end
plot(h,T) ; xlabel('h [m]') ; ylabel('t_f [s]')
```

### B.2 Hohmann Transfer Orbit

```
function [ v ] = deltav( mu,epsilon , r )
%calculating the velocity depending on the orbit and energy
v = sqrt(2*(mu/r + epsilon));
end

function [ trans ] = epsilon( mu,a )
% Calculates the specific mechanical energy, using a given standard
% gravitational parameter and semi major axis.

trans = mu/(2*a);
end

function [ m ] = mass( deltav , isp , m0 )
%Calculating the mass at time t_f. Formula was derived from Tsiolkovsky's rocket equation.
g = 9.81;
m = m0/(exp(deltav/(isp*g)));
end

format long

%% LEO GEO
%Calculates the total delta v necessary for the trajectory LEO to GEO. All
%units are SI units.
muA = 3.986e14; LEO = 6678e3; GEO = 6378e3+35786e3;
e1 = epsilon(muA,LEO);
etrans = epsilon(muA, (GEO+LEO)/2);
e2 = epsilon(muA,GEO);
vleo = deltav(muA, e1,LEO);
vtrans = deltav(muA,etrans,LEO);
vtrans1 = deltav(muA,etrans,GEO);
vgeo = deltav(muA,e2,GEO);
```

```

vtot1 = abs(vtrans vleo)+abs(vgeo vtrans1);

%% EARTH MARS
%Calculates the total delta v necessary for the trajectory Earth to Mars. All
%units are SI units.
muS = 1.32712440018e20; rearth = 1.49598023e11; rmars = 2.279392e11;
e2 = epsilon(muS, rearth);
etrans2 = epsilon(muS, (rearth+rmars)/2);
e3 = epsilon(muS, rmars);
vearth = deltav(muS, e2, rearth);
vtrans2 = deltav(muS, etrans2, rearth);
vtrans3 = deltav(muS, etrans2, rmars);
vmars = deltav(muS, e3, rmars);
vtot2 = abs(vtrans2 vearth)+abs(vmars vtrans3);

%% EARTH SATURN
%Calculates the total delta v necessary for the trajectory Earth to Saturn. All
%units are SI units.
muS = 1.32712440018e20; rearth = 1.49598023e11; rsat = 1.43353e12;
e4 = epsilon(muS, rearth);
etrans3 = epsilon(muS, (rearth+rsat)/2);
e5 = epsilon(muS, rsat);
vearth = deltav(muS, e4, rearth);
vtrans4 = deltav(muS, etrans3, rearth);
vtrans5 = deltav(muS, etrans3, rsat);
vsat = deltav(muS, e5, rsat);
vtot3 = abs(vtrans4 vearth)+abs(vsat vtrans5);

%% Calculating the end mass of the rocket
% Here an initial mass of 2e6 kg is chosen as well as a specific impulse of
% 455 s, this value corresponds to taking a liquid bipropellant rocket
% using liquid oxygen and liquid hydrogen.
m0 = 5e5; isp = 455;
mass1 = mass(vtot1, isp, m0)
mass2 = mass(vtot2, isp, m0)
mass3 = mass(vtot3, isp, m0)

%% TIME OF FLIGHTS
% The total time of flight for each orbit is calculated.
tof1 = pi*sqrt((((LEO+GEO)/2)^3)/muA)
tof2 = pi*sqrt((((rearth+rmars)/2)^3)/muS)
tof3 = pi*sqrt((((rearth+rsat)/2)^3)/muS)

```

## B.3 LEO GEO

```

function [ res ] = bcs( ya, yb )
% Boundary conditions
h = 300; % Initial altitude [km]
rearth = 6378; % Radius of the Earth [km]
mu = 3.986004418e5; % Standard gravitational parameter of the Earth [km^3/s^2]
r0 = h+rearth; % LEO radius [km]
reind = rearth+35786; % GEO radius [km]

res = [ya(1) (r0)
ya(2)
ya(3)
ya(4) sqrt(mu/(r0))
yb(1) (reind)
yb(3)
yb(4) sqrt(mu/(reind))
yb(6)
];

end

function [ xprime ] = D(tau, x)
% These are the differential equations where x can be read as x =
% (r, theta, vr, vtheta, lambda2, lambda3, lambda4)
mu = 3.986004418e5; % Standard gravitational parameter of the Earth [km^3/s^2]
xprime = [x(3);
x(4)/x(1);
x(7)+(x(4)^2/x(1) - mu/(x(1)^2));
x(8) x(3)*x(4)/x(1);
x(7)*((x(4)^2)/(x(1)^2) - 2*mu/(x(1)^3)) - x(8)*x(3)*x(4)/(x(1)^2);
0;
x(5)+x(8)*x(4)/x(1);
2*x(7)*x(4)/x(1)+x(8)*x(3)/x(1)
];

end

clear all; close all; clc;
% Main program for the trajectory LEO to GEO
h = 300; % Initial altitude [km]
rearth = 6378; % Radius of the Earth [km]
mu = 3.986004418e5; % Standard gravitational parameter of the Earth [km^3/s^2]
r0 = h+rearth; % LEO radius [km]
reind = rearth+35786; % GEO radius [km]

option = bvpset('Nmax',10000,'RelTol',1e-11);
Teind = 19000; % End time [s]
Nt = 1000; % Number of steps in each interval

```

```

Tinit = 500; % Initial time [s]
tau = linspace(0,Tinit,Nt);
yinit = [r0 0 0 sqrt(mu/r0) 0 0 0 0]; % Initial guess
solinit = bvpinit(tau,yinit);
sol = bvp4c(@D,@bcs,solinit,option);
Z = deval(sol,tau);
Tlst = [2*Tinit:Tinit:Teind];
for T=Tlst
    T
    tau = linspace(0,T,Nt);
    solinit.y = Z;
    solinit.x = tau;
    sol = bvp4c(@D,@bcs,solinit,option);
    Z = deval(sol,tau);
end
tau = linspace(0,Teind,Nt);
solinit.y = Z;
solinit.x = tau;
sol = bvp4c(@D,@bcs,solinit,option);
Z = deval(sol,tau);

tau2 = linspace(0,T,5*Nt);
for q = 1:8
    Z2(q,:) = spline(tau,Z(q,:),tau2);
end
tau = tau2;
Z = Z2;
time = tau;
figure(1)
plot(Z(1,:).*cos(Z(2,:)), Z(1,:).*sin(Z(2,:)), 'k')
xlabel('x direction [km]')
ylabel('y direction [km]')
axis equal
hold on
viscircles([0,0], rearth, 'Color', 'b')
hold on
viscircles([0,0], reind, 'Color', 'r', 'LineStyle', ':', 'Linewidth', 0.9)
hold on
viscircles([0,0], r0, 'Color', 'm', 'LineStyle', ':', 'Linewidth', 0.9)
figure(2)
subplot(2,3,1)
plot(time, Z(1,:)); xlabel('time [s]'); ylabel('r [km]');
subplot(2,3,2)
plot(time, Z(2,:)); xlabel('time [s]'); ylabel('theta');
subplot(2,3,3)
plot(time, Z(3,:)); xlabel('time [s]'); ylabel('vr [km/s]');
subplot(2,3,4)
plot(time, Z(4,:)); xlabel('time [s]'); ylabel('vtheta [km/s]');
subplot(2,3,5)
plot(time, Z(7,:)); xlabel('time [s]'); ylabel('l3 [km/s^2]');
subplot(2,3,6)
plot(time, Z(8,:)); xlabel('time'); ylabel('l4 [km/s^2]');
Y = Z(7,:).^2 + Z(8,:).^2;
Eindmassa = 1/(1/m0 + trapz(tau,Y)/(2*P));
%%
figure(3)
plot(Z(1,:).*cos(Z(2,:)), Z(1,:).*sin(Z(2,:)))
axis equal
hold on
viscircles([0,0], rearth, 'Color', 'b')
hold on
for k=1:length(time)
    plot(Z(1,k).*cos(Z(2,k)), Z(1,k).*sin(Z(2,k)), 'or', 'MarkerSize', 5, 'MarkerFaceColor', 'r');
    pause(.001)
end

function [Eindmassa] = minverbruik(Teind,Tinit,Nt)
% Calculating the end mass of the rocket depending on end time, Teind. If
% statements were used as the program sometimes had difficulties with
% accuracy.
mu = 3.986004418e5; % Standard gravitational parameter of the Sun [km^3/s^2]
h = 300; % Initial altitude [km]
rearth = 6378; % Radius of the Earth [km]
r0 = h+rearth; % LEO radius [km]
reind = rearth+35786; % GEO radius [km]
m0 = 5e5; % Initial mass of the rocket [kg]
P = 59; % Power of the rocket [kg km^2/s^3]

option = bvpset('Nmax',10000,'RelTol',1e-11);
tau = linspace(0,Tinit,Nt);
yinit = [r0 0 0 sqrt(mu/r0) 0 0 0 0];
solinit = bvpinit(tau,yinit);
sol = bvp4c(@D,@bcs,solinit,option);
Z = deval(sol,tau);

if Teind<=25000
    Tlst = 2*Tinit:Tinit:Teind;
elseif Teind<=39000
    Tlst = [2*Tinit:Tinit:25000, 25050:50:26000, 27000:Tinit:Teind];
elseif Teind<=52000
    Tlst = [2*Tinit:Tinit:25000, 25050:50:26000, 27000:Tinit:39000, 39960,41000:Tinit:Teind];
elseif Teind<=66000
    Tlst = [2*Tinit:Tinit:25000, 25050:50:26000, 27000:Tinit:39000, 39960,41000:Tinit:52000,52960,53570,54600,55000:Tinit:Teind];
elseif Teind<=80000
    Tlst = [2*Tinit:Tinit:25000, 25050:50:26000, 27000:Tinit:39000, 39960,41000:Tinit:

```

```

:52000,52960,53570,54600,55000:Tinit:66000,68000:500:Teind];
elseif Teind<=92000
    Tlst = [2*Tinit:Tinit:25000, 25050:50:26000, 27000:Tinit:39000, 39960,41000:Tinit
:52000,52960,53570,54600,55000:Tinit:66000,68000:500:80000, 80200, 82000:Tinit:Teind];
elseif Teind<=100000
    Tlst = [2*Tinit:Tinit:25000, 25050:50:26000, 27000:Tinit:39000, 39960,41000:Tinit
:52000,52960,53570,54600,55000:Tinit:66000,68000:500:80000, 80200, 82000:Tinit:92000,
93500,94500, 96000:Tinit:Teind];
for T=Tlst
    tau = linspace(0,T,Nt);
    solinit.y = Z;
    solinit.x = tau;
    sol = bvp4c(@D,@bcs,solinit,option);
    Z = deval(sol,tau);
end

tau = linspace(0,Teind,Nt);
solinit.y = Z;
solinit.x = tau;
sol = bvp4c(@D,@bcs,solinit,option);
Z = deval(sol,tau);

tau2 = linspace(0,Teind,5*Nt);
for q = 1:8
    Z2(q,:) = spline(tau,Z(q,:),tau2);
end
tau = tau2;
Z = Z2;
Y = Z(7,:).^2 + Z(8,:).^2;

Eindmassa = 1/(1/m0 + trapz(tau,Y)/(2*P));
end

clear all; close all; clc;
% Use the function minverbruik, to calculate the end mass for multiple end
% times.
Nt = 1000;
lst = 10000:10000:140000;
Tinit = 1000;
stap = 1;
Use = zeros(length(lst),1);
for i = lst
    Use(stap) = minverbruik(i,Tinit,Nt);
    stap = stap + 1
end

figure(1)
plot(lst,Use); xlabel('End Time [s]'); ylabel('End Mass [kg]');

```

## B.4 Earth Mars

```

function [ res ] = bcs2( ya,yb )
% Boundary conditions
r0 = 1.496e8; % Distance Sun to Earth [km]
rend = 2.279e8; % Distance Sun to Mars [km]
mu = 1.32712440018e11; % Standard gravitational parameter of the Sun [km^3/s^2]
res = [ya(1) (r0)
ya(2)
ya(3)
ya(4) sqrt(mu/(r0))
yb(1) rend
yb(3)
yb(4) sqrt(mu/rend)
yb(6)
];
end

function [ xprime ] = D2(tau,x)
% These are the differential equations where x can be read as x =
% (r, theta, vr, vtheta, lambda1, lambda2, lambda3, lambda4)
mu = 1.32712440018e11; %Standard gravitational parameter of the Sun in km^3/s^2
xprime = [x(3);
x(4)/x(1);
x(7)+(x(4)^2/x(1) - mu/(x(1)^2));
x(8) - x(3)*x(4)/x(1);
x(7)*((x(4)^2)/(x(1)^2) - 2*mu/(x(1)^3)) - x(8)*x(3)*x(4)/(x(1)^2);
0;
x(5)+x(8)*x(4)/x(1);
2*x(7)*x(4)/x(1)+x(8)*x(3)/x(1)
];
end

clear all; close all; clc;
% Main program for the trajectory Earth to Mars
rearth = 6378; % Radius of the Earth [km]
rsun = 6.957e5; % Radius of the Sun [km]
r0 = 1.496e8; % Distance Sun to Earth [km]
rend = 2.279e8; % Distance Sun to Mars [km]
mu = 1.32712440018e11; % Standard gravitational parameter of the Sun [km^3/s^2]

option = bvpset('Nmax',10000,'RelTol',10e-13);
Tend = 2.23e7; % End time [s]

```

```

Nt = 1000; % Number of steps in each interval
Tinit = 1e6; % Initial time [s]
tau = linspace(0,Tinit,Nt);
yinit = [r0 0 0 sqrt(mu/r0) 0 0 0 0]; % Initial guess
solinit = bvpinit(tau,yinit);
sol = bvp4c(@D2,@bcs2,solinit,option);
Z = deval(sol,tau);

for T=2*Tinit:Tinit:floor(Tend/Tinit)*Tinit
    T
    tau = linspace(0,T,Nt);
    solinit.y = Z;
    solinit.x = tau;
    sol = bvp4c(@D2,@bcs2,solinit,option);
    Z = deval(sol,tau);
end

tau = linspace(0,Tend,Nt);
solinit.y = Z;
solinit.x = tau;
sol = bvp4c(@D2,@bcs2,solinit,option);
Z = deval(sol,tau);

tau2 = linspace(0,T,5*Nt);
for q = 1:8
    Z2(q,:) = spline(tau,Z(q,:),tau2);
end
tau = tau2;
Z = Z2;
phi = acosd( Z(8,:) ./ (sqrt(Z(7,:).^2 + Z(8,:).^2) ));
time = tau;
figure(1)
plot(Z(1,:).*cos(Z(2,:)), Z(1,:).*sin(Z(2,:)),'k')
xlabel('x direction [km]')
ylabel('y direction [km]')
axis equal
hold on
viscircles([0,r0], rearth,'Color','b')
hold on
viscircles([0,0],rsun, 'Color','m')
hold on
viscircles([0,0],rend, 'Color','r','LineStyle',':','Linewidth',0.9)
hold on
viscircles([0,0], r0, 'Color','b','LineStyle',':','Linewidth',0.9)
figure(2)
subplot(2,3,1)
plot(time, Z(1,:)); xlabel('time [s]'); ylabel('r [km]');
subplot(2,3,2)
plot(time, Z(2,:)); xlabel('time [s]'); ylabel('theta');
subplot(2,3,3)
plot(time, Z(3,:)); xlabel('time [s]'); ylabel('vr [km/s]');
subplot(2,3,4)
plot(time, Z(4,:)); xlabel('time [s]'); ylabel('vtheta [km/s]');
subplot(2,3,5)
plot(time, Z(7,:)); xlabel('time [s]'); ylabel('13 [km/s^2]');
subplot(2,3,6)
plot(time, Z(8,:)); xlabel('time'); ylabel('14 [km/s^2]');
%%
figure(3)
plot(Z(1,:).*cos(Z(2,:)), Z(1,:).*sin(Z(2,:)))
axis equal
hold on
viscircles([0,r0], rearth,'Color','b')
hold on
viscircles([0,0], rsun, 'Color','r')
for k=1:length(time)
    plot(Z(1,k).*cos(Z(2,k)), Z(1,k).*sin(Z(2,k)),'or','MarkerSize',5,'MarkerFaceColor','r');
    pause(.001)
end

function [Eindmassa] = minverbruikmars(Teind,Tinit,Nt)
% Calculating the end mass of the rocket depending on end time, Teind.
mu = 1.32712440018e11; % Standard gravitational parameter of the Sun [km^3/s^2]
r0 = 1.496e8; % Distance Sun to Earth [km]
m0 = 5e5; % Initial mass of the rocket [kg]
P = 59; % Power of the rocket [kg km^2/s^3]
option = bvpset('Nmax',100000,'RelTol',1e-13);
tau = linspace(0,Tinit,Nt);
yinit = [r0 0 0 sqrt(mu/r0) 0 0 0 0];
solinit = bvpinit(tau,yinit);
sol = bvp4c(@D2,@bcs2,solinit,option);
Z = deval(sol,tau);

for T=2*Tinit:Tinit:floor(Tend/Tinit)*Tinit
    tau = linspace(0,T,Nt);
    solinit.y = Z;
    solinit.x = tau;
    sol = bvp4c(@D2,@bcs2,solinit,option);
    Z = deval(sol,tau);
end

tau = linspace(0,Teind, Nt);
solinit.y = Z;
solinit.x = tau;
sol = bvp4c(@D2,@bcs2,solinit,option);
Z = deval(sol,tau);

```

```

tau2 = linspace(0,Teind,5*Nt);
for q = 1:8
    Z2(q,:) = spline(tau,Z(q,:),tau2);
end
tau = tau2;
Z = Z2;
Y = Z(7,:).^2 + Z(8,:).^2;
Eindmassa = 1/(1/m0 + trapz(tau,Y)/(2*P));
Hamilton = .5.*(Z(7,:).^2 + Z(8,:).^2) + Z(5,:).*Z(3,:) + Z(7,:).*((Z(4,:).^2)./Z(1,:)) - mu./(Z(1,:).^2) - Z(8,:).*(Z(3,:).*Z(4,:)./Z(1,:));
end

clear all; close all; clc;
% Use the function minverbruik, to calculate the end mass for multiple end
% times.
stap = 1;
format long
Nt = 100;
lst = 0.1e7:1e6:3e7;
Use = zeros(length(lst),1);
for i = 1:lst
    [Use(stap)] = minverbruikmars(i,Tinit,Nt);
    stap = stap + 1
end

figure(1)
plot(lst,Use); xlabel('End Time [s]'); ylabel('End Mass [kg]');

```

## B.5 Earth Saturn

```

function [ res ] = bcs4( ya,yb )
% Boundary conditions
r0 = 1.496e2; % Distance Sun to Earth [Gm]
reind = 1.43353e3; % Distance Sun to Saturn [Gm]
mu = 1.719953;
%mu = 1.32712440018e20*(1e9)^3/((1/3600)^2); % Standard gravitational parameter of the Sun [Gm^3/h^2]
res = [ya(1) r0
      ya(2)
      ya(3)
      ya(4) (sqrt(mu/(r0)))
      yb(1) reind
      yb(3)
      yb(4) (sqrt(mu/(reind)))
      yb(6)
      ];
end

function [ xprime ] = D4(tau,x)
% These are the differential equations where x can be read as x =
% (r, theta, vr, vtheta, lambda1, lambda2, lambda3, lambda4)
mu = 1.719953;
%mu = 1.32712440018e20*(1e9)^3/((1/3600)^2); % Standard gravitational parameter of the Sun [Gm^3/h^2]
xprime = [x(3);
          x(4)/x(1);
          x(7)+(x(4)^2/x(1) - mu/(x(1)^2));
          x(8) - x(3)*x(4)/x(1);
          x(7)*((x(4)^2)/(x(1)^2) - 2*mu/(x(1)^3)) - x(8)*x(3)*x(4)/(x(1)^2);
          0;
          x(5)+x(8)*x(4)/x(1);
          2*x(7)*x(4)/x(1)+x(8)*x(3)/x(1)
          ];
end

clear all; close all; clc;
% Main program for the trajectory Earth to Mars
rsun = 6.957e1; % Radius of the Sun [Gm]
r0 = 1.496e2; % Distance Sun to Earth [Gm]
reind = 1.43353e3; % Distance Sun to Saturn [Gm]
mu = 1.719953; % 1.32712440018e20*(1e9)^3/((1/3600)^2); % Standard gravitational parameter of the Sun [Gm^3/h^2]

option = bvpset('Nmax',100000,'RelTol',10e-13);
Teind = 53000; % End time [h]
Nt = 1000; % Number of steps in each interval
Tinit = 1000; % Initial time [h]
tau = linspace(0,Tinit,Nt);
yinit = [r0 0 0 sqrt(mu/r0) 0 0 0 0]; % Initial guess
solinit = bvpinit(tau,yinit);
sol = bvp4c(@D4,@bcs4,solinit,option);
Z = deval(sol,tau);
if Teind <= 50000
    Tlst = 2*Tinit:Tinit:Teind;
elseif Teind <= 80000
    Tlst = [2*Tinit:Tinit:52000, 52005:5:52100, 53000:500:Teind];
elseif Teind > 80000
    Tlst = [2*Tinit:Tinit:52000, 52005:5:52100, 53000:1000:80000, 80100:50:Teind];
end
for T=Tlst
    T
    tau = linspace(0,T,Nt);
    solinit.y = Z;
end

```

```

        solinit.x = tau;
        sol = bvp4c(@D4,@bcs4,solinit,option);
        Z = deval(sol,tau);
end

tau = linspace(0,Teind,Nt);
solinit.y = Z;
solinit.x = tau;
sol = bvp4c(@D4,@bcs4,solinit,option);
Z = deval(sol,tau);

tau2 = linspace(0,T,5*Nt);
for q = 1:8
    Z2(q,:) = spline(tau,Z(q,:),tau2);
end
tau = tau2;
Z = Z2;
time = tau;
figure(1)
plot(Z(1,:) .* cos(Z(2,:)), Z(1,:) .* sin(Z(2:)), 'k')
xlabel('x direction [Gm]')
ylabel('y direction [Gm]')
axis equal
hold on
viscircles([0,r0], 0.006378, 'Color', 'b')
hold on
viscircles([0,0],rsun, 'Color', 'm')
hold on
viscircles([0,0],reind, 'Color', 'r', 'LineStyle', ':', 'Linewidth', 0.9)
hold on
viscircles([0,0], r0, 'Color', 'b', 'LineStyle', ':', 'Linewidth', 0.9)
figure(2)
subplot(2,3,1)
plot(time, Z(1,:)); xlabel('time [h]'); ylabel('r [Gm]');
subplot(2,3,2)
plot(time, Z(2,:)); xlabel('time [h]'); ylabel('theta');
subplot(2,3,3)
plot(time, Z(3,:)); xlabel('time [h]'); ylabel('vr [Gm/h]');
subplot(2,3,4)
plot(time, Z(4,:)); xlabel('time [h]'); ylabel('vtheta [Gm/h]');
subplot(2,3,5)
plot(time, Z(7,:)); xlabel('time [h]'); ylabel('l3 [Gm/h^2]');
subplot(2,3,6)
plot(time, Z(8,:)); xlabel('time [h]'); ylabel('l4 [Gm/h^2]');
%%
figure(3)
plot(Z(1,:) .* cos(Z(2,:)), Z(1,:) .* sin(Z(2:)))
axis equal
hold on
viscircles([0,r0], 6378e6, 'Color', 'b')
hold on
viscircles([0,0], rsun, 'Color', 'r')
for k=1:length(time)
    plot(Z(1,k) .* cos(Z(2,k)), Z(1,k) .* sin(Z(2,k)), 'or', 'MarkerSize', 5, 'MarkerFaceColor', 'r');
    pause(.001)
end

function [Eindmassa] = minverbruiksat(Teind,Tinit,Nt)
% Calculating the end mass of the rocket depending on end time, Teind. If
% statements were used as the program sometimes had difficulties with
% accuracy.
rsun = 6.957e11; % Radius of the Sun [Gm]
r0 = 1.496e2; % Distance Sun to Earth [Gm]
reind = 1.43353e3; % Distance Sun to Saturn [Gm]
mu = 1.719953e3; % 1.32712440018e20*(1e9)^3/((1/3600)^2); % Standard gravitational parameter of the Sun [Gm^3/h^2]
m0 = 5e5; % Initial mass of the rocket [kg]
P = 59*46656e6; % Power of the rocket [kg Gm^2/h^3]

option = bvpset('Nmax',100000, 'RelTol', 1e-13);
tau = linspace(0,Tinit,Nt);
yinit = [r0 0 0 sqrt(mu/r0) 0 0 0 0];
solinit = bvpinit(tau,yinit);
sol = bvp4c(@D4,@bcs4,solinit,option);
Z = deval(sol,tau);

if Teind <= 50000
    Tlst = 2*Tinit:Tinit:Teind;
elseif Teind <= 80000
    Tlst = [2*Tinit:Tinit:52000, 52005:5:52100, 53000:500:Teind];
elseif Teind > 80000
    Tlst = [2*Tinit:Tinit:52000, 52005:5:52100, 53000:1000:80000, 80300:1000:Teind];
end

for T=Tlst
    tau = linspace(0,T,Nt);
    solinit.y = Z;
    solinit.x = tau;
    sol = bvp4c(@D4,@bcs4,solinit,option);
    Z = deval(sol,tau);
end

tau = linspace(0,Teind, Nt);
solinit.y = Z;
solinit.x = tau;
sol = bvp4c(@D4,@bcs4,solinit,option);
Z = deval(sol,tau);

```

```

tau2 = linspace(0,Teind,5*Nt);
for q = 1:8
    Z2(q,:) = spline(tau,Z(q,:),tau2);
end
tau = tau2;
Z = Z2;
Y = Z(7,:).^2 + Z(8,:).^2;

Eindmassa = 1/(1/m0 + trapz(tau,Y)/(2*P));
end

clear all; close all; clc;
% Use the function minverbruik, to calculate the end mass for multiple end
% times.
stap = 1;
format long
Tinit = 1000;
Nt = 1000;
lst = 5000:5000:80000;
Use = zeros(length(lst),1);
for i = lst
    [Use(stap)] = minverbruiksat(i,Tinit,Nt);
    stap = stap + 1
end
%%
figure(1)
plot(lst,Use); xlabel('End Time [h]'); ylabel('End Mass [kg]');

```

## B.6 Flyby

```

function [ res ] = bcs3( ya,yb )
% Boundary conditions
r0 = 1.496e2; % Distance from the Earth to the Sun [Gm]
reind = 1.43353e3; % Distance from Saturn to the Sun [Gm]
mus = 1.719953 ; % Standard gravitational parameter of the Sun [Gm^3/h^2]
omega = (2*pi)/(4332.71*24); % Orbital speed of Jupiter [rad/h]
res = [ya(1) (r0)
       ya(2) (0.34)*pi
       ya(3)
       ya(4) (sqrt(mus/r0) omega*r0)
       yb(1) reind
       yb(3)
       yb(4) (sqrt(mus/reind) omega*reind)
       yb(6)
       ];
end

function [ xprime ] = D3(tau,x)
% Differential equations in the rotating reference frame
r0 = 1.496e2; % Distance from the Earth to the Sun [Gm]
R = 7.785e2; % Distance from Jupiter to the Sun [Gm]
reind = 1.43353e3; % Distance from Saturn to the Sun [Gm]
mus = 1.719953 ; % Standard gravitational parameter of the Sun [Gm^3/h^2]
muv = 0.0016418575; % 1.26686534e17*(1e 9)^3/((1/3600)^2) % Standard gravitational parameter Jupiter [
Gm^3/h^2]
omega = (2*pi)/(4332.71*24); % Orbital speed of Jupiter [rad/h]
rsun = 6.957e 1; % Radius of the Sun [Gm]
c1 = 0.05; % Drag coefficient
D = 0.7; % Radius of influence
% Differential equations only with Jupiter's atmosphere
a = x(3);
b = x(4)/x(1);
c = x(7)+x(4)^2/x(1) mus/(x(1)^2)+2*x(4)*omega+omega^2*x(1) (muv*(x(1) R*cos(x(2))))/((x(1)^2+R^2 2*
R*x(1)*cos(x(2)))^(3/2)) x(3)*c1*exp((x(1)^2+R^2 2*R*x(1)*cos(x(2)))/D^2);
d = x(8) x(3)*x(4)/x(1) 2*omega*x(3) (muv*(R*sin(x(2))))/((x(1)^2+R^2 2*R*x(1)*cos(x(2)))^(3/2)) x(4)*
c1*exp((x(1)^2+R^2 2*R*x(1)*cos(x(2)))/D^2);
e = (x(6)*x(4)/(x(1))^2 + x(7)*((x(4)^2)/(x(1)^2) (x(3)*c1*(2*x(1) 2*R*cos(x(2))))*exp((x(1)^2+R^2 2*R
*x(1)*cos(x(2)))/D^2))/D^2 omega^2 2*mus/(x(1)^3) + muv/((x(1)^2+R^2 2*R*x(1)*cos(x(2)))^(3/2)
) (3/2)*(muv*(x(1) R*cos(x(2)))*(2*x(1) 2*R*cos(x(2))))/((x(1)^2+R^2 2*R*x(1)*cos(x(2)))^(5/2)
) x(8)*(x(3)*x(4)/x(1)^2 + (x(4)*c1*(2*x(1) 2*R*cos(x(2))))*exp((x(1)^2+R^2 2*R*x(1)*cos(x(2))
)/D^2))/D^2 + (3/2)*(muv*R*sin(x(2)))*(2*x(1) 2*R*cos(x(2)))/((x(1)^2+R^2 2*R*x(1)*cos(x(2)))^(5/2)
));
f = x(7)*((muv*R*sin(x(2)))/((x(1)^2+R^2 2*R*x(1)*cos(x(2)))^(3/2)) + (3*muv*(x(1) R*cos(x(2)))*R*x
(1)*sin(x(2)))/((x(1)^2+R^2 2*R*x(1)*cos(x(2)))^(5/2)) + (2*x(3)*c1*R*x(1)*sin(x(2))*exp((x(1)^2+
R^2 2*R*x(1)*cos(x(2)))/D^2))/D^2 x(8)*((muv*R*cos(x(2)))/((x(1)^2+R^2 2*R*x(1)*cos(x(2)))
^(3/2)) + (3*muv*(x(1)*R^2*(sin(x(2)))^2))/((x(1)^2+R^2 2*R*x(1)*cos(x(2)))^(5/2)) + (2*x(4)*c1*R*x
(1)*sin(x(2))*exp((x(1)^2+R^2 2*R*x(1)*cos(x(2)))/D^2))/D^2);
g = x(5) x(8)*(x(4)/x(1) 2*omega)+x(7)*c1*exp((x(1)^2+R^2 2*R*x(1)*cos(x(2)))/D^2);
h = x(6)/x(1) x(7)*(2*x(4)/x(1)+ 2*omega) + x(8)*x(3)/x(1)+x(8)*c1*exp((x(1)^2+R^2 2*R*x(1)*cos(x(2)
))/D^2);
xprime = [a;b;c;d;e;f;g;h];

% Differential equations with Jupiter's atmosphere as well as singularity
% removal

%r = x(1); theta = x(2); vr = x(3); vtheta = x(4);
% b = (r^2 + R^2 2*R*r*cos(theta))^(1/2);
% B = (r R*cos(theta));
% Em = exp((b^2)/(D^2));
% a = vr;

```

```

% k=   vtheta/r;
% c=   x(7)+vtheta^2/r   mus/(r^2)+2*vtheta*omega+omega^2*r (muv*B*(1 Em))/(b^3) vrc1*Em;
% d=   x(8) vrvtheta/r 2*omega*vrc1((muv*R*sin(theta))*(1 Em))/(b^3) vtheta*c1*Em;
% e=   (x(6)*vtheta)/r^2 + x(7)*((vtheta^2)/(r^2) (vrc1*2*B*Em)/D^2   omega^2   2*mus/(r^3) + (muv*(1
Em))/(b^3)   (3/2)*(muv*B*2*B*(1 Em))/(b^5) + (muv*2*B*B*Em)/(b^3*D^2))   x(8)*(vrvtheta/r^2 + (
vtheta*c1*2*B*Em)/(D^2) + (3/2)*(muv*R*sin(theta)*2*B*(1 Em))/(b^5)   (muv*R*sin(theta)*2*B*Em)/(D
^2*b^3));
% f=   x(7)*((muv*R*sin(theta)*(1 Em))/b^3 + (3*muv*B*R*r*sin(theta)*(1 Em))/b^5   (2*muv*B*R*r*sin(
theta)*Em)/(b^3*D^2) + (2*vrc1*R*r*sin(theta)*Em)/D^2)   x(8)*((muv*R*cos(theta)*(1 Em))/(b^3) +
(3*muv*(r*R^2*(sin(theta))^2)*(1 Em))/(b^5)   (2*muv*R^2*(sin(theta))^2*r*Em)/(b^3*D^2)+ (2*vtheta*
c1*R*r*sin(theta)*Em)/D^2);
% g=   x(5)   x(8)*(vtheta/r   2*omega)+x(7)*c1*Em;
% h=   x(6)/r   x(7)*(2*vtheta/r+ 2*omega) + x(8)*vrc1+x(8)*c1*Em;
% xprime = [a;k;c;d;e;f;g;h];
end

clear all;close all; clc;
format longEng
% Main program for the trajectory Earth to Saturn using a gravity assist
% from Jupiter
r0 = 1.496e2; % Distance from the Earth to the Sun [Gm]
R = 7.785e2; % Distance from Jupiter to the Sun [Gm]
reind = 1.43353e3; % Distance from Saturn to the Sun [Gm]
mus= 1.719953 ; % Standard gravitational parameter of the Sun [Gm^3/h^2]
muv= 0.0016418575; % 1.26686534e17*(1e 9)^(3)/((1/3600)^2); % Standard gravitational parameter Jupiter
[Gm^3/h^2]
omega = (2*pi)/(4332.71*24); % Orbital speed of Jupiter [rad/h]
m0 = 5e5 ; % Initial mass of the rocket [kg]
P = 59*46656e 6; % Power of the rocket [kg Gm^2/h^3]
rsun = 6.957e 1; % Radius of the Sun [Gm]
option = bvpset('Nmax',20000,'RelTol',10e 13);
Teind = 50000; % End time [h]
Nt = 1000; % Number of steps
Tinit = 50; % Initial time [h]
tau = linspace(0,Tinit,Nt);
yinit = [r0 (0.34)*pi 0 sqrt(mus/r0) omega*r0 1e 9 1e 9 1e 7 1e 6]; % Initial guess
solinit = bvpinit(tau,yinit);
sol = bvp4c(@D3T,@bcs3T,solinit,option);
Z = deval(sol,tau);
Tlst = [2*Tinit:Tinit:49800, 49805:5:Teind];
%Tlst = [2*Tinit:Tinit:16250, 16300:2:16500, 16550: 50: Teind];% , 49510: 1: Teind];
%Tlst = [2*Tinit:Tinit:23000, 23100:50:48500, 48510: 10 :49360, 49400:10:Teind];
for T=Tlst %floor(Teind/Tinit)*Tinit
    T
    tau = linspace(0,T,Nt);
    solinit.y = Z;
    solinit.x = tau;
    sol = bvp4c(@D3T,@bcs3T,solinit,option);
    Z = deval(sol,tau);
end

tau2 = linspace(0,Teind,5*Nt);
for q = 1:8
    Z2(q,:) = spline(tau,Z(q,:),tau2);
end
tau = tau2;
Z = Z2;
time = tau;
Y = Z(7,:).^2 + Z(8,:).^2;
Eindmassa = 1/(1/m0 + trapz(tau,Y)/(2*P))
figure(1)
plot(Z(1,:).*cos(Z(2,:)+omega*tau), Z(1,:).*sin(Z(2,:)+omega*tau),'k')
xlabel('x direction [km]')
ylabel('y direction [km]')
axis equal
hold on
viscircles([0,r0], rsun,'Color','b')
hold on
viscircles([0,0],rsun, 'Color','r')
% hold on
% viscircles([0,0],reind, 'Color', 'b','LineStyle',' ')
hold on
viscircles([R,0],rsun, 'Color', 'k')
figure(2)
subplot(2,3,1)
plot(time, Z(1,:)); xlabel('time') ; ylabel('r');
subplot(2,3,2)
plot(time, Z(2,:)+omega*tau); xlabel('time') ; ylabel('theta');
subplot(2,3,3)
plot(time, Z(3,:)); xlabel('time') ; ylabel('vr');
subplot(2,3,4)
plot(time, Z(4,:)+omega*Z(1,:)); xlabel('time') ; ylabel('vtheta');
subplot(2,3,5)
plot(time, Z(7,:)); xlabel('time') ; ylabel('l3');
subplot(2,3,6)
plot(time, Z(8,:)); xlabel('time') ; ylabel('l4');
figure(3)
plot(time, Z(3,:).^2+(Z(4,:)+omega*Z(1,:)).^2,'r')
hold on
plot(time, (2*mus./Z(1,:)), 'k')
%%
figure(4)
plot(Z(1,:).*cos(Z(2,:)), Z(1,:).*sin(Z(2,:)));
axis equal

```

```
hold on
viscircles([r0,0], rsun, 'Color', 'b')
hold on
viscircles([0,0], rsun, 'Color', 'r')
hold on
viscircles([R,0], rsun, 'Color', 'k')
hold on
viscircles([R,0], 48.1, 'Color', 'b', 'LineStyle', ' ')
for k=1:Nt/100:length(time)
    plot(Z(1,k).*cos(Z(2,k)), Z(1,k).*sin(Z(2,k)), 'or', 'MarkerSize', 5, 'MarkerFaceColor', 'r');
    pause(.001)
end
```