Aerogravity assists

Hypersonic maneuvering to improve planetary gravity assists

J.R. Hess

Delft University of Technology

TU Delft Delft University of Technology

Challenge the future

Aerogravity assists

Hypersonic maneuvering to improve planetary gravity assists

by

J.R. Hess

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Wednesday February 24, 2016 at 10:00 AM.

Student number:	4087321	
Thesis committee:	Dr. ir. E. Mooij,	TU Delft, supervisor
	Ir. K.J. Sudmeijer,	TU Delft, supervisor
	Prof. dr. ir. P. N.A.M. Visser,	TU Delft
	Ir. K. Cowan,	TU Delft

"I guess you could call it a "failure", but I prefer the term "learning experience"

MARK WATNEY

Acknowledgments

This Master of Science (MSc) thesis is the final milestone of my education at Delft University of Technology. When I started in 2010 with my Bachelor Aerospace Engineering, I was mainly interested in aircraft. I thought that the space-part was a nice bonus, but it was not the reason I had choosen Aerospace Engineering. However, as they years went by, I became more and more interested in the space aspect of the curriculum. I selected the Master track Space Exploration at the department of Astrodynamics and Space Missions, where I also carried out my MSc thesis on a topic that covered my interest in both spaceflight and atmospheric flight: aerogravity assists.

Although a MSc thesis is an individual project, a number of people were involved that made the performed research a success. First of all, I want to express my sincere gratitude towards my thesis supervisors Erwin Mooij and Kees Sudmeijer for the useful comments, remarks and insights they provided during our weekly meetings. Next, I want to thank Dominic Dirkx for answering my (possibly annoying) questions about C++ and Tudat. Also, to my fellow students at the 9th floor, thank you for the time we spent together working our own, each others or random problems.

In addition, I want to thank Steven Plimpton and Michael Gallis, the developers of SPARTA at Sandia National Laboratories, for taking the time to answer my questions about their software package. Furthermore, I would also like to thank Dario Izzo, member of ESA's Advanced Concepts Team and developer of PaGMO, the optimization toolbox used in this thesis, for his assistance in tracking down a bug in one of the optimization algorithms. Finally, I would like to thank my Dad for his support and encouragement during my graduation.

J.R. Hess
Delft, February 2016

Abstract

Interplanetary missions use gravitational slingshots around planetary bodies to adjust their heliocentric velocity or inclination for quite some time. The momentum exchange that can be achieved during a so-called gravity assist is limited by the mass of the planetary body. To overcome this limitation, an aerogravity assist was proposed, a maneuver where, in addition to the gravitational forces, use is made of aerodynamic forces to increase the bending angle of the velocity, hence increasing the momentum exchange. To investigate the effects of the aerogravity assist, the following research question was proposed:

> **How efficient can the interplanetary orbital inclination and velocity be changed with an aerogravity assist?**

A simulator was developed that is capable of simulating both the gravitational and aerodynamic forces on a vehicle during an aerogravity assist. It was determined that waverider is a type of vehicle suitable for aerogravity assists due to their large lift-to-drag ratio, which reduces the energy dissipation in the atmosphere. The aerodynamic characteristics of a number of waverider shapes were evaluated, after which the one with the largest lift-to-drag ratio was selected. Furthermore, a numerical optimization algorithm was used to develop a reference trajectory planner. Finally, a guidance algorithm based on the tracking of drag accelerations was developed and tested to investigate if the found trajectories would still be feasible under the influence of uncertainties and perturbations.

The angle over which the trajectory is bent is a measure for the effectiveness of the aerogravity assist. Using the reference trajectory planner, the maximum possible atmospheric bending angle was investigated for an aerogravity assist at Mars and Jupiter for different initial velocities. From this analysis, it was concluded that extremely high velocities were involved in the aerogravity assist at Jupiter, which resulted in large mechanical and thermal loads. These loads would limit the achievable bending angle when the velocities become too large. For the entry velocities investigated, the velocity bending angle could be increased by 10% for high entry (80.0 km/s) velocities and up to 143% for a relatively low entry velocity (68.0 km/s). For an entry velocity of 80.0 km/s, the initial heat-flux peak exceeded the imposed constraints, which prevented the optimization algorithm of finding any solutions. The maximum velocity bending angle that could be achieved at Jupiter was 125.1° at an entry velocity of 68.0 km/s. At Mars, although the heat loads were still larger than for an Earth entry, it is believed that thermal protection systems can be designed that could handle the heat loads. The velocity bending angle could be increased by 490% to 818% depending on the arrival velocity, with a maximum velocity bending angle of 178.5° at an entry velocity of 9.0 km/s.

To investigate the effect of an aerogravity assist on an actual mission, two existing missions has been selected: Rosetta for Mars and Ulysses for Jupiter. Although both spacecraft did not have an aerodynamic shape, which means an aerogravity assist could not have been performed during the actual mission, it has been assumed that these vehicles would have had the geometry of a waverider. During the investigation of Rosetta swing-by at Mars, a reference trajectory was generated to investigate the amount of velocity decrease that could have been achieved using an aerogravity assist. It was determined that the reduction in velocity could be increased by 167% with respect to a gravity assist: from 2.3 km/s for a gravity assist to 6.2 km/s for an aerogravity assist. For Jupiter, it was investigated if the orbital inclination could be changed using the aerodynamic force only. As the entry velocity exceeded 80.0 km/s, the heat flux constrained was removed from the trajectory optimization to allow the optimization algorithm to find solutions. It was possible to change the orbital inclination by 54.2°, but at an extremely large heat load of $40,620\,\mathrm{W/cm^2}$. This reconfirms that even though orbital inclination changes are possible using aerodynamic forces, Jupiter is unsuitable for aerogravity assists due to the high velocities and large heat loads associated with an atmospheric maneuver at this planet.

Finally, using the aerogravity assist trajectory found for Rosetta, which was generated generated with the reference trajectory planner, the guidance algorithm was tested. The guidance algorithm was capable of tracking a drag reference under the influence of uncertain initial flight-path angles. The maximum offset in velocity bending angle occurred for a steep entry and was 1.06°, while the maximum offset in hyperbolic

excess velocity occurred during a shallow entry and was 1.88 m/s. Furthermore, the tracking was also successful when a more accurate atmosphere model and perturbations were taken into account. For this analysis, the maximum offset in velocity bending angle and hyperbolic excess velocity were 1.24° and 2.14 m/s respectively.

List of Symbols

Roman Symbols

a	Semi-major axis	[m]
$\boldsymbol{C}^{A/B}$	Transformation matrix from frame B to frame A	[−]
C_C	Side force coefficient	[−]
C_D	Drag coefficient	[−]
C_L	Lift coefficient	[−]
C_p	Normal pressure coefficient	[−]
C_t	Shear stress coefficient	[−]
$C_{p,b}$	Base pressure coefficient	[−]
$\mathbf{c}$	Constraint	[−]
D	Drag acceleration	[N/kg]
d	Slope	[−]
E	Specific energy	[J/kg]
e	Eccentricity	[−]
F	Bridging function	[−]
F	Force	[N]
G	Gravitational constant ($= 6.67259 \times 10^{-11}$)	[$\mathrm{m}^3/\mathrm{kg}/\mathrm{s}^2$]
g	Gravitational acceleration	[m/s^2]
h	Step size	[−]
h_s	Scale height	[m]
i	Inclination	[rad]
k	Boltzmann constant ($= 1.38064852 \times 10^{-23}$)	[$\mathrm{m}^2\,\mathrm{kg}/\mathrm{s}^2/\mathrm{K}$]
Kn	Knudsen number	[−]
L	Characteristic length	[m]
L	Lift acceleration	[N/kg]
l	Length	[m]
M	Mach number	[−]
M	Mass of major body	[kg]
M	Molar mass	[kg/mol]
m	Mass	[kg]

n_G	G-load	$[-]$
$\mathscr{N}$	Guidance node	$[-]$
$\boldsymbol{n}$	Normal vector	$[-]$
N_A	Avogadro's number ($= 6.0221409 \times 10^{23}$)	[1/mol]
N_p	Number of guidance nodes	$[-]$
q	Heat flux	[W/m^2]
R	Distance w.r.t. center of an object	[m]
R_N	Nose radius	[m]
$\mathscr{R}$	Universal gas constant ($= 8.3145$)	[J/K/mol]
Pr	Prandtl number	$[-]$
S	Area	[m^2]
s	Molecular speed ratio	$[-]$
T	Temperature	[K]
t	Time	[s]
$\boldsymbol{t}$	Tangential vector	$[-]$
V_∞^+	Departure hyperbolic excess velocity	[m/s]
V_∞^-	Arriving hyperbolic excess velocity	[m/s]
w	Width	[m]
$\dot{x}, \dot{y}, \dot{z}$	Cartesian velocity components	[m/s]
x, y, z	Cartesian position components	[m]
$\boldsymbol{z}^*$	Reference point	$[-]$

Greek Symbols

α	Angle of attack	[rad]
β	Angle of sideslip	[rad]
γ	Flight-path angle	[rad]
γ	Specific heat ratio	$[-]$
$\boldsymbol{\Gamma}$	Guidance matrix	$[-]$
δ	Divided difference	$[-]$
δ	Latitude	[rad]
δ	Oblique shock angle	[rad]
δ	Velocity bending angle	[rad]
ϵ	Maximum energy of attraction between colliding molecules	[m^2 kg/s^2]
θ	Atmospheric bending angle	[rad]
θ	True anomaly	[rad]

θ	Wedge angle	[rad]
$\boldsymbol{\lambda}$	Weight vector	[−]
λ	Mean free path length	[m]
μ	Gravitational parameter	[$\mathrm{m}^3/\mathrm{s}^2$]
ρ	Atmospheric density	[kg/m^3]
σ	Bank angle	[rad]
σ	Collision diameter	[m]
σ_N	Normal angular momentum coefficient	[−]
σ_T	Tangential angular momentum coefficient	[−]
τ	Longitude	[rad]
χ	Heading angle	[rad]
ω	Argument of periapse	[rad]
ω	Rotational velocity	[rad/s]
$\boldsymbol{\Omega}$	Decision variable space	[−]
Ω	Right ascension of the ascending node	[rad]

Subscripts

∞	Free-stream condition
aero	Aerodynamic
cb	Central body
conv	Convective
grav	Gravitational
l	Lower
P	Planetary
p	Planform
pert	Perturbing
rad	Radiative
ref	Reference
SoI	Sphere of influence
u	Upper
w	Wall

Acronyms

AGA	Aerogravity Assist
DE	Differential Evolution
DSMC	Direct Simulation Monte Carlo
ESA	European Space Agency
GA	Gravity Assist
GNC	Guidance, Navigation and Control
GRAM	Global Reference Atmospheric Model
lb	Lower bound
MOEA/D	Multiobjective Evolutionary Algorithm based on Decomposition
NASA	National Aeronautics and Space Administration
PaGMO	Parallel Global Multiobjective Optimizer
RAAN	Right Ascension of Ascending Node
RK87DP	Runge-Kutta 8(7) Dormand-Prince
RKF45	Runge-Kutta-Fehlberg 4(5)
RKF56	Runge-Kutta-Fehlberg 5(6)
RKF78	Runge-Kutta-Fehlberg 7(8)
SPARTA	Stochastic Parallel Rarefied-gas Time-accurate Analyzer
Tudat	TU Delft Astrodynamics Toolbox
ub	Upper bound

Contents

1
Introduction

Currently, gravity assists are commonly used in interplanetary missions to adjust the heliocentric velocity or inclination of a spacecraft to obtain trajectories to targets that would not be available with propulsive maneuvers only. Disadvantage, however, is that gravity assists can result in a longer flight duration, and the velocity gain that can be achieved from such a maneuver is limited. An aerogravity assist does not only depend on the use of gravitational forces, but also the aerodynamic forces to increase the heliocentric velocity. This makes it possible to increase the momentum exchange.

1.1 Background

People have gazed upon night sky for thousands of years. Early astronomers observed objects moving among the stars, also known as planets. Later, once the telescope was invented, these planets could be observed in more detail. In more recent time, in the scientists' and researchers' pursuit for more information about the origin of the Solar System, the development of the Solar System and search for life, space agencies have sent spacecraft and probes to other planets. This made it possible to observe the planets from a closer distance, which allowed them to make observations, which could not be performed from Earth itself.

The first interplanetary probe, although not successful, was the Venera-1, developed as a part of the Soviet Union's Venera program to explore Venus. Using a propulsive maneuver, it was put into a heliocentric trajectory heading for Venus, but contact was lost before it actually arrived at Venus. The first successful probe that visited Mars was the Mariner 4, developed by NASA. Similarly to the Venera-1, it left its Earth parking orbit and entered a Mars Transfer Orbit using a propulsive maneuver.

Both Venus and Mars, being the two planets in the Solar System closest to Earth, can be reached directly using propulsive maneuvers only. Jupiter can also be reached from Earth directly using a propulsive maneuver, but since the velocity change for such a trajectory requires a lot of propellant, it is not used very often. The velocity change that can be obtained from the available launch vehicles and thrusters is sufficient to reach these few planets. Objects that are located at much larger distance from Earth cannot be reached by using propulsive maneuvers only. This would require such a large amount of propellant, that it is not feasible to launch it in the first place. To overcome this limitation, use has been made of planetary bodies that can be reached to increase the heliocentric velocity of a spacecraft. Such a maneuver is also called a gravity assist, gravitational sling-shot or swing-by, and consists of a hyperbolic trajectory around a planetary body, resulting in a rotation of the velocity vector in the heliocentric reference frame, which allows to increase or decrease the spacecraft's heliocentric velocity or change its inclination.

Two excellent examples that demonstrates the effectiveness of gravity assists are ESA's Rosetta mission and NASA's New Horizons mission. As described by Villefranche et al. (1997), the Rosetta spacecraft performed two gravity assists around Earth and one around Mars to obtain a trajectory that would be heading to the comet 67P/Churyumov-Gerasimenko. The New Horizons probe used the gravitational acceleration of Jupiter to boost its trajectory to Pluto and is currently on its way to the Kuiper Belt. These two missions would not have been possible if these gravity assist were not included in their mission planning.

As the concept of a gravity assist can be difficult to grasp at first, an analogy is presented in the cartoon in Fig. 1.1. The child throws a baseball with a velocity of 30 miles per hour away from him. This is also the velocity the baseball has with respect to the Sun, which is considered to be sitting on a stationary platform. With respect to the arriving train, the baseball is approaching with a velocity of 80 miles per hour since the

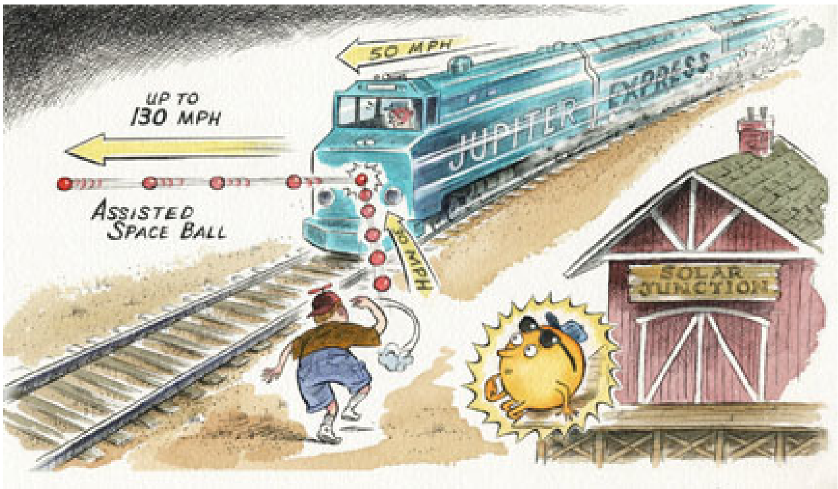


Figure 1.1: Analogy to the gravity assist (Courtesy NASA/JPL-Caltech)

velocity of the train is 50 miles per hour with respect to the stationary platform. Assuming no energy is lost when the baseball interacts with the surface of the train, the ball bounces from the train with a velocity of 80 miles per hour as well. However, since the train's velocity is 50 miles per hour, the velocity of the baseball with respect to the stationary platform is 130 miles per hour, a velocity increase of 100 miles per hour.

Although the benefit of a gravity assist is clear, it is limited by the momentum exchange that can be obtained from a planetary body. According to Armellin et al. (2007), this momentum exchange is determined by the eccentricity of the hyperbolic orbit around the planetary body, which is defined by the arrival velocity (V_∞^-), the pericenter radius and the planetary mass. The arrival velocity is usually very large and the pericenter radius is limited by the planetary radius, so large momentum exchange can be obtained from planetary bodies with a large mass.

To increase the efficiency of the gravity assist, McRonald and Randolph (1990) proposed the aerogravity assist maneuver. In addition to the gravitational acceleration, aerodynamic lift is used to increase centripetal acceleration. This will allow to increase the bending angle and, as a result, increase the momentum exchange. In addition, performing out-of-plane maneuvers, which can be achieved by changing the direction of the lift vector by means of the bank angle, can result in an adjustment of the inclination of the orbit.

An aerogravity assist is not the only interplanetary maneuver that makes use of atmospheres. Two other mission concepts that use an atmosphere to their advantage are aerobraking and aerocapture. Aerobraking is a mission concept that uses atmospheric drag to achieve a decrease in semi-major axis (Uesugi et al., 1988). The main advantage is that no propellant has to be used to lower the orbital altitude, although a propulsive maneuver is required to enter an elliptical orbit around the planet. To achieve a decrease in semi-major axis, a large number of passes through the atmosphere is required, which increase the time to complete this maneuver. For example, the Magellan aerobraking operations at Venus had a duration of 70 days, during which the apoapsis altitude decreased from 8,500 km to 550 km (Lyons et al., 1995). As a spacecraft during aerobraking operations do not enter the very dense layers of the atmosphere, the heat loads will not become too excessive. For example, the Mars Reconnaissance Orbiter (Lyons, 2002) expected a maximum heat flux of $0.7\,\mathrm{W/cm^2}$. Also, as only drag is required to reduce the apoapsis altitude, it is not necessary to have a capsule or a lifting body as a vehicle shape.

For an aerocapture, thermal protection and vehicle shape are two very important aspects. According to Spilker (2005), during an aerocapture, the hyperbolic orbit of the approaching spacecraft is changed into an elliptical orbit by means of aerodynamic drag. In contrast to aerobraking, an aerocapture has never been performed during an actual mission. As the energy reduction required to change a hyperbolic orbit into an elliptical orbit can be quite large, a spacecraft has to enter the atmosphere of a planet much deeper compared to an aerobraking maneuver. The larger density at these lower altitudes, as well as the high velocities that are related to a hyperbolic orbit, result in large heat loads on the vehicle. The analysis performed by Lockwood (2004), where an aerocapture at Neptune was studied, shows that the convective stagnation point heat flux can be as high as $8,000\,\mathrm{W/cm^2}$. By comparison, the Space Shuttle experienced a maximum heat flux of $100\,\mathrm{W/cm^2}$ (Bansal and Lamon, 2014).

Where the goal of both aerobraking and aerocapture is to decelerate by means of atmospheric drag with respect to the planetary body, an aerogravity assist is used to change a vehicle's state with respect to the Sun. This is achieved by cruising through the atmosphere, which allows to rotate the departure velocity vector over a larger angle with respect to the arrival velocity vector compared to a regular gravity assist. As the energy dissipation due to drag reduces the possible length of the cruise phase, it is preferred to use a vehicle with a large lift-to-drag ratio (McRonald and Randolph, 1990).

The geometry of an aerogravity assist is depicted in Fig. 1.2. A vehicle arrives at a planet with the hyperbolic excess velocity $\boldsymbol{V}_{\infty}^{-}$. If no atmosphere would be present, the velocity vector would be bent over an angle $\delta_{GA,0}$, indicated by the dashed trajectory labeled *Osculating outgoing*. The atmosphere, however, becomes noticeable at a distance $\boldsymbol{R}_0$. At this distance, it has been defined that the atmospheric phase is started, which ends once the vehicle passes the edge of the atmosphere again at $\boldsymbol{R}_f$. An indication of the length of this atmospheric phase is given by the *atmospheric bending angle* θ_{AGA}, which is defined as the angle between $\boldsymbol{R}_0$ and $\boldsymbol{R}_f$. The vehicle leaves the planet with the hyperbolic excess velocity $\boldsymbol{V}_{\infty}^{+}$. The arrival trajectory a vehicle would have had to leave the planet with $\boldsymbol{V}_{\infty}^{+}$ in case there was no atmosphere is indicated by the dashed trajectory labeled *Osculating incoming*. As a result of the atmospheric maneuver, the velocity vector is rotated over the *velocity bending angle* δ_{AGA}.

Even though a vehicle with a high lift-to-drag ratio is desired, drag will result in energy dissipation, which in turn leads to a reduction of the magnitude of the departure velocity V_{∞}^{+}. For a hyperbolic orbit with no interaction with an atmosphere holds $V_{\infty}^{-} = V_{\infty}^{+}$, but in case of an aerogravity assist this relation becomes $V_{\infty}^{-} > V_{\infty}^{+}$. Although it might seem like this energy loss can make an aerogravity assist unattractive, this is not the case. First of all, the velocity bending angle can still be increased, giving more flexibility in mission

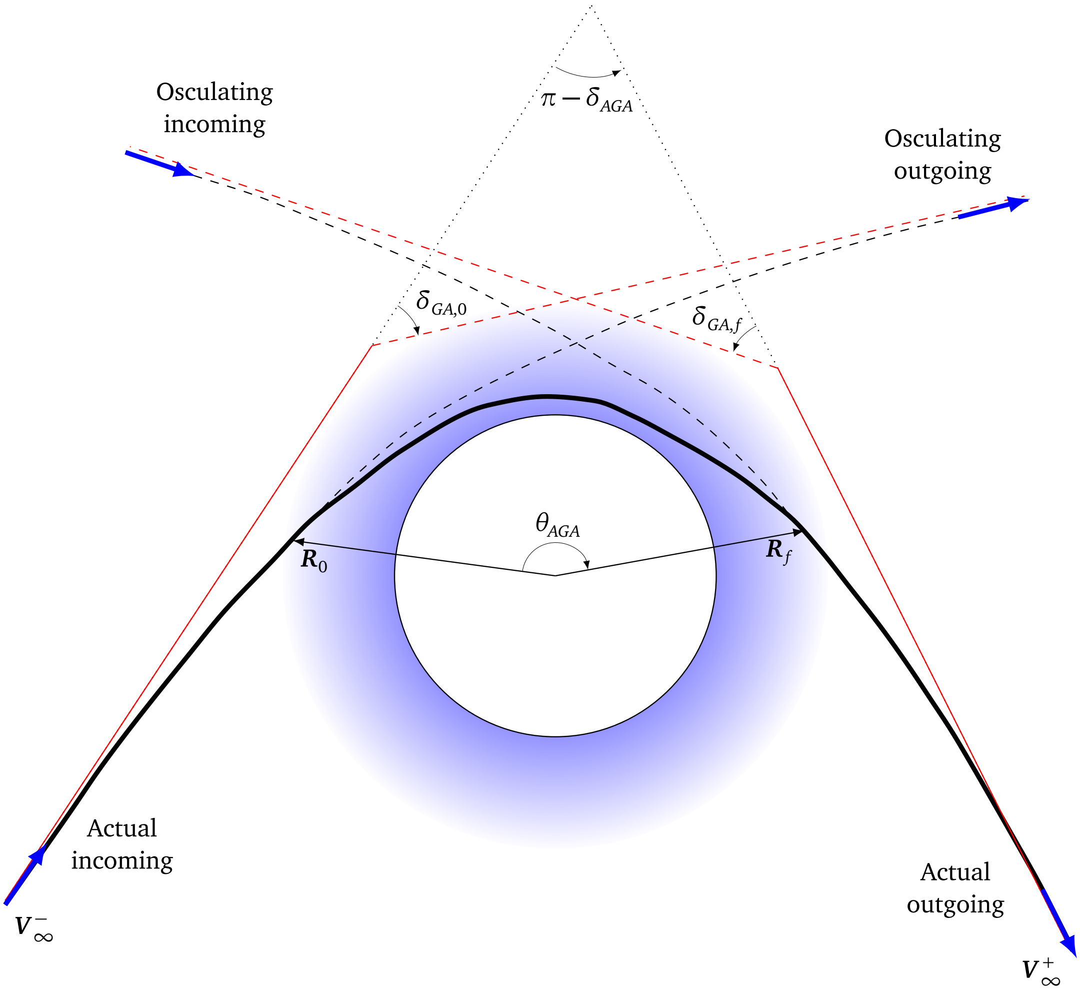


Figure 1.2: Geometry of an aerogravity assist, adapted from Casoliva et al. (2008)

design. In addition, the velocity loss due to drag is only with respect to the assisting planetary body. The heliocentric velocity or inclination change is also a function of the atmospheric bending angle (McRonald and Randolph, 1990), which is increased with respect to a regular gravity assist.

To date, although the benefits are clear, no actual aerogravity assist has been performed. The high velocities, heat loads and uncertainties associated with an aerogravity assist make it an extremely challenging maneuver. A robust guidance system and thermal protection system will be a few of the necessities to be able to incorporate an aerogravity assist in future interplanetary mission.

1.2 Research goal

The study presented in this thesis focuses on the analysis of the benefits of an aerogravity assist maneuver. Therefore, the following research question was formulated:

> **How efficient can the interplanetary orbital inclination and velocity be changed with an aerogravity assist?**

To be able to answer this question, a number of tasks was defined. First of all, a simulator should be developed that can simulate an aerogravity assist trajectory. Second, a suitable vehicle should be selected and for this vehicle, the aerodynamic characteristics should be determined. Next, a trajectory planner should be developed that can generate trajectories which achieve specified objectives while remaining within certain constraints. Finally, a guidance algorithm should be designed and tested, which uses the trajectories obtained with the trajectory planner as a reference. By completing these tasks, an extensive tool will be obtained, which can be used to perform detailed analysis on aerogravity assist trajectories. This will provide an answer to the research question.

1.3 Outline of thesis

The development of a simulator requires a number of reference frames, in which the equations of motion can be defined. Both the equations of motion and the reference frames including the corresponding transformations will be presented in Chapter 2. The equations of motion require a number of properties of the environment, such as the atmospheric density and the gravitational acceleration. Chapter 3 will therefore provide an overview of the atmospheric and gravitational models that will be used, as well as the method used to obtain the heliocentric position and velocity of Solar System bodies.

As aerodynamic forces are essential to an aerogravity assist, a vehicle and its aerodynamic characteristics should be chosen. Chapter 4 will first discuss the vehicles that are considered for an aerogravity assist, also known as waveriders. This is followed by an approach to determine the aerodynamic properties of a waveridere in different flow regimes. Next, a trade-off between different waverider geometries is made. This chapter will end with a method of estimating the radiative and convective heat flux on a vehicle.

A number of numerical methods will be used, such as interpolation, differentiation, integration and optimization. Chapter 5 provides a discussion on the used techniques, the justification of these techniques and their implementation in the simulator. During the development of the simulator, use has been made of many different tools, subroutines and software packages. Chapter 6 presents an overview of the developed simulator, as well as the performed verification steps.

Using numerical optimization method in combination with the developed simulator, a reference trajectory planner is described in Chapter 7. Given a number of operational and control constraints, the reference trajectory planner is able to provide a trajectory that has been optimized with respect to a specified objective. Using the trajectory planner, a number of reference trajectories is generated to explore the possibilities of an aerogravity assist. The guidance algorithm, which should guide a vehicle based on a reference trajectory, is designed and tested in Chapter 9.

The research performed in this thesis is wrapped up in Chapter 10, where the conclusions and recommendations are presented.

2 Flight Dynamics

The flight dynamics used to develop the trajectory simulator will be discussed in this chapter. First of all, the reference frames in which the equations of motion are defined and the transformation between these reference frames are shown in Section 2.1. Next, in Section 2.2, the actual equations of motion are presented. Finally, a number of variables, described by the general term flight parameters, are defined in Section 2.3, and help interpreting the output of the equations of motion

2.1 Reference frames

Equations of motion are defined in a specified reference frame. For the work performed in this thesis, a number of reference frames is defined, which are presented in this section. In addition, the transformations between the different frames are shown in the second half in this section. The reference frames, which are listed below, are adapted from Mooij (1997).

- The **ecliptic reference frame**, indicated by a subscript E, is positioned at the center of mass of the Sun. The $\boldsymbol{X_E}$ axis points towards the mean equinox at J2000, the $\boldsymbol{Y_E}$ axis lays in the same plane as the Earth's orbit, perpendicular to $\boldsymbol{X_E}$. The $\boldsymbol{Z_E}$ axis completes the right-handed orthogonal reference frame. This frame is required to determine the state of the vehicle with respect to the Sun.

- The **inertial planetocentric reference frame**, indicated by a subscript I, is positioned at the center of mass of the central body. The $\boldsymbol{Z_I}$ axis points towards the north of the central body and is aligned with the rotation axis of the body. The $\boldsymbol{X_I}$ axis points towards a zero longitude at J2000 and the $\boldsymbol{Y_I}$ axis completes the right-handed orthogonal reference frame. This is the reference frame in which the equations of motion will be defined.

- The **rotating planetocentric reference frame**, indicated by a subscript R, is positioned at the center of mass of the central body. The $\boldsymbol{Z_R}$ axis points towards the north of the central body and is aligned with rotation axis of the body. The $\boldsymbol{X_R}$ axis, located in the equatorial plane, points towards zero longitude. The $\boldsymbol{Y_R}$ axis completes the right-handed orthogonal reference frame. Using this rotating planetocentric reference frame, it is possible to define the state of the vehicle with respect to the surface of the planetary body.

- The **vertical reference frame**, indicated by a subscript V, is positioned at the center of mass of the vehicle. The $\boldsymbol{Z_V}$ axis points towards the center of mass of the central body. The $\boldsymbol{X_V}$ axis points towards the north. The $\boldsymbol{Y_V}$ axis completes the right-handed orthogonal reference frame. By using this frame, it is possible to use more intuitive variables to describe the state of the vehicle, such as the flight-path angle and heading angle.

- The **body-fixed reference frame**, indicated by a subscript B, is positioned at the center of mass of the vehicle. The $\boldsymbol{X_B}$ axis lies in the plane of symmetry of the vehicle and points towards the front of the vehicle. The $\boldsymbol{Z_B}$ axis lies also in the plane of symmetry of the vehicle and points downwards. The $\boldsymbol{Y_B}$ axis completes the right-handed orthogonal reference frame.

- The **aerodynamic reference frame**, indicated by a subscript *A*, is positioned at the center of mass of the vehicle. The $\boldsymbol{X_A}$ axis points in the direction of the airspeed. The $\boldsymbol{Z_A}$ axis is pointed in the opposite direction of the aerodynamic lift vector. The $\boldsymbol{Y_A}$ axis completes the right-handed orthogonal reference frame. By using this frame, in combination with the body-fixed reference frame discussed before, the orientation of the vehicle with respect to the free-stream velocity can be defined, necessary to describe the aerodynamic forces on the vehicle.
- The **trajectory reference frame**, indicated by a subscript T, is positioned at the center of mass of the vehicle. The $\boldsymbol{X_T}$ axis points in the direction of the airspeed. The $\boldsymbol{Z_T}$ axis lies in the vertical plane and points towards the surface. The $\boldsymbol{Y_T}$ axis completes the right-handed orthogonal reference frame. This frame is required to describe bank reversals.

In addition to the definition of the reference frame, it is necessary to obtain the relation between the different reference frames, such that the state in one frame can be transformed to another frame. These relations are obtained by performing one or more elementary transformations. Each of these transformations represent a rotation over an angle θ about one of the three principle axes, X, Y and Z, as shown in Eqs. (2.1) to (2.3), respectively (Wie, 2008).

$$\boldsymbol{C}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \tag{2.1}$$

$$\boldsymbol{C}_y(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \tag{2.2}$$

$$\boldsymbol{C}_z(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{2.3}$$

These elementary transformations are orthonormal matrices. A property of such matrices is that the inverse is equal to the transpose. Therefore, if a transformation from frame A to frame B is defined, the transformation from frame B to frame A is given by the transpose of the transformation matrix of the transformation from frame A to frame B. This property can be summarized by:

$$\boldsymbol{C}^{A/B} = \left[\boldsymbol{C}^{B/A}\right]^T \tag{2.4}$$

Using Eqs. (2.1) to (2.3) in a specific order, the transformation matrices that will be discussed in the remainder of this section can be derived. In front of each of the transformation matrices, the sequence of elementary transformations with their corresponding rotation angle is given.

First of all, the transformation between the inertial planetocentric reference frame, the rotating planetocentric reference frame en the vertical reference frame will be discussed. These three reference frames are depicted in Fig. 2.1.

The rotating planetocentric reference frame is obtained when the inertial planetocentric reference frame is rotated over an angle $\omega_{cb}t$ around the $\boldsymbol{Z_I}$ axis, where ω_{cb} is the rotational rate of the central body around its rotation axis and t is the time since a specified reference time. Basically, $\omega_{cb}t$ represents the angle the planet has rotated since a specified reference time, which is chosen to be epoch J2000. The transformation from the inertial planetocentric reference frame to the rotating planetocentric reference frame can be described by the following transformation matrix:

$$\boldsymbol{C}^{R/I} = \boldsymbol{C}_z(\omega_{cb}t) = \begin{bmatrix} \cos\omega_{cb}t & \sin\omega_{cb}t & 0 \\ -\sin\omega_{cb}t & \cos\omega_{cb}t & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.5}$$

The vertical reference frame is related to the rotating planetocentric reference frame by two angles: the longitude τ and the latitude δ. The longitude defined as a rotation around the $\boldsymbol{Z_R}$ axis, positive in eastward direction. The latitude is defined positive upwards from the equatorial plane. The transformation matrix from the rotating planetocentric reference frame to the vertical frame is given by:

$$\boldsymbol{C}^{V/R} = \boldsymbol{C}_y(\varphi + \frac{\pi}{2})\boldsymbol{C}_z(\theta) = \begin{bmatrix} -\sin\delta\cos\tau & -\sin\delta\sin\tau & \cos\delta \\ -\sin\theta & \cos\theta & 0 \\ -\cos\delta\cos\tau & -\cos\delta\sin\tau & -\sin\delta \end{bmatrix} \tag{2.6}$$

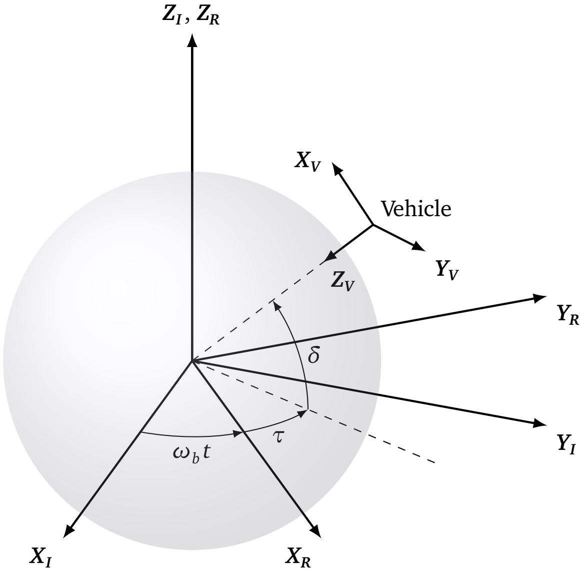


Figure 2.1: Inertial planetocentric reference frame (index I), rotating planetocentric reference frame (index R) and vertical reference frame (index V), adapted from Mooij (1997)

Next, the relation between the trajectory reference frame and the vertical reference frame will be discussed. Both frames are shown in Fig. 2.2. Two angles are important: the heading angle χ and the flight-path angle γ. The heading angle is the angle between the north and the projection of the velocity vector in the local vertical plane. The flight-path angle is the angle between the velocity vector and the local vertical plane. The transformation from vertical reference frame to the trajectory reference frame is given by:

$$\boldsymbol{C}^{T/V} = \boldsymbol{C}_y(\gamma)\boldsymbol{C}_z(\chi) = \begin{bmatrix} \cos\gamma\cos\chi & \cos\gamma\sin\chi & -\sin\gamma \\ -\sin\chi & \cos\chi & 0 \\ \sin\gamma\cos\chi & \sin\gamma\sin\chi & \cos\gamma \end{bmatrix} \tag{2.7}$$

The relation between the trajectory reference frame and the aerodynamic reference frame is described by a rotation of the bank angle σ around the $\boldsymbol{X_T}$ axis, which is pointing in the direction of the velocity vector. Both frames are shown in Fig. 2.3. The transformation between the trajectory reference frame and the aerodynamic reference frame is given by:

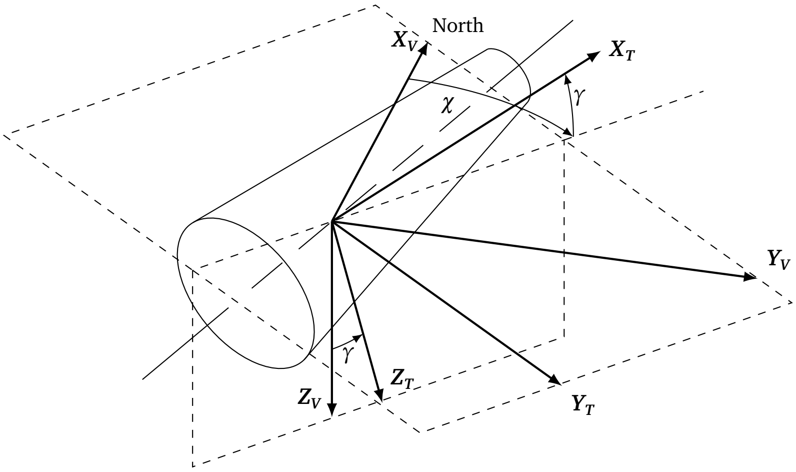


Figure 2.2: Trajectory reference frame (index T) and vertical reference frame (index V), adapted from Mooij (1997)

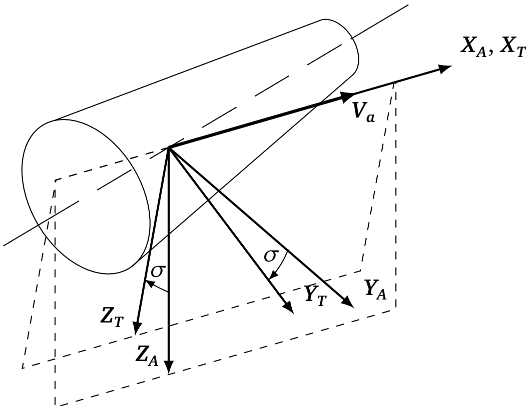


Figure 2.3: Aerodynamic reference frame (index *A*) and trajectory reference frame (index *T*), adapted from Mooij (1997)

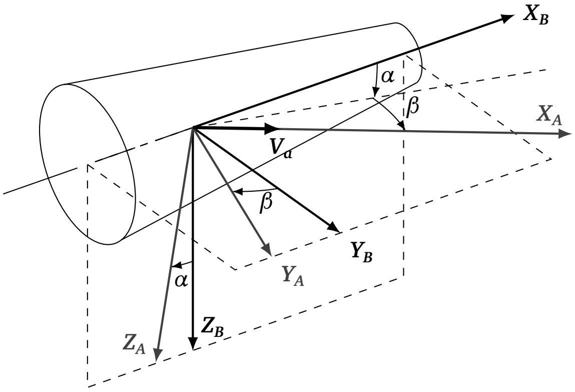


Figure 2.4: Body-fixed reference frame (index *B*) and aerodynamic reference frame (index *A*), adapted from Mooij (1997)

$$\boldsymbol{C}^{A/T} = \boldsymbol{C}_x(\sigma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\sigma & \sin\sigma \\ 0 & -\sin\sigma & \cos\sigma \end{bmatrix} \tag{2.8}$$

The transformation from the body-fixed reference frame to the aerodynamic reference frame is related to the angle of attack and the angle of sideslip. The angle of attack is defined positive for a nose-up attitude, the angle of sideslip is defined positive for a nose-left attitude, as is shown in Fig. 2.4. The transformation between the two reference frames given by:

$$\boldsymbol{C}^{A/B} = \boldsymbol{C}_z(\beta)\boldsymbol{C}_y(-\alpha) = \begin{bmatrix} \cos\alpha\cos\beta & \sin\beta & \sin\alpha\cos\beta \\ -\cos\alpha\sin\beta & \cos\beta & -\sin\alpha\sin\beta \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \tag{2.9}$$

Finally, the ecliptic reference frame is shown in Fig. 2.5, and is necessary to determine the state vector of a vehicle with respect of the Sun. The transformation from the inertial planetocentric reference frame, which is defined as the orientation of the rotating planetocentric reference frame at epoch J2000, to the ecliptic reference frame is obtained from the SPICE library incorporated in the TU Delft Astrodynamics Toolbox (Tudat), see also Section 6.1. This transformation is related to the axial tilt of a planet's rotation axis with respect to its orbital axis.

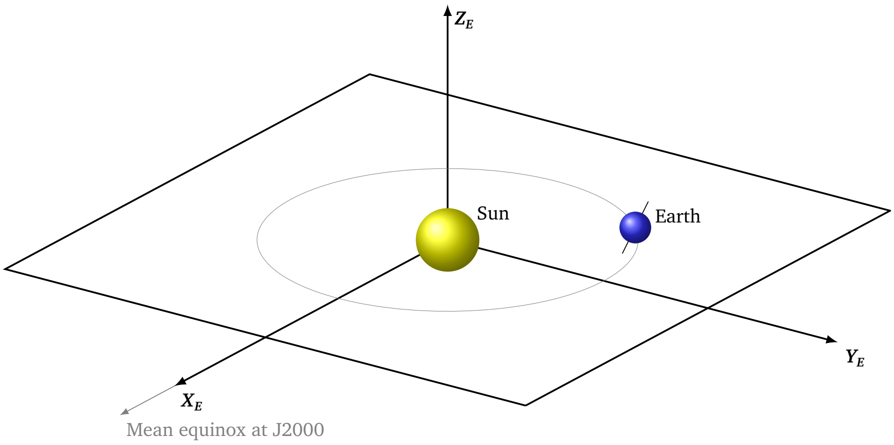


Figure 2.5: Ecliptic J2000 reference frame, adapted from Kemble (2006)

2.2 Equations of Motion

The movement of a particle, object or vehicle can be described by a set of mathematical equations based on Newton's second law. This law could be defined in basically any reference frame imaginable, as long as it is an inertial frame. The equations could also be adapted such that they would be valid in rotating reference frames, but this would the introduction of apparent forces. For the simulation, the inertial planetocentric reference frame has been selected to define the equations of motion. By defining the equations of motion in this reference frame, only a reference frame transformation on the aerodynamic forces needs to be performed.

Some assumptions were made when establishing the equations of motion. These assumptions simplify the equations of motion, reducing the required computational and implementation effort, and are listed below:

- The mass of the vehicle does not change over time. So it is assumed that no propellant tanks are depleted or that the mass of the thermal protection systems is reduced due to ablation. This assumption removes all the mass-varying terms from the equations of motion.
- The vehicle is assumed to be a rigid body. This implies that any deformations due to aerodynamic or thermal loading, as well as that the movement of anything within the vehicle, is neglected.
- Only the translational motion is considered in the equations of motion. The rotational accelerations on the vehicle are not taken into account.

The resulting force on a point mass can determined by combining the gravitational, aerodynamic and perturbing forces. By defining the position vector in an inertial frame $\mathbf{R}^I = \begin{bmatrix} x_I & y_I & z_I \end{bmatrix}^T$ and combining the external forces with Newton's second law, the following equation can be obtained:

$$\boldsymbol{F}^I_{ext} = m\frac{d^2\boldsymbol{R}^I}{dt^2} = \boldsymbol{F}^I_{grav} + \boldsymbol{F}^I_{aero} + \boldsymbol{F}^I_{pert} \tag{2.10}$$

All the external forces in Eq. (2.10) are defined in the inertial planetocentric reference frame, indicated by the superscript I. The external forces consist of the gravitational forces of the central body, the aerodynamic forces and the perturbing forces.

The gravitational force $\boldsymbol{F}^I_{grav}$ on the vehicle can be derived from Newton's Law of Universal Gravitation. When using this law, it is assumed that the central body is a homogeneous sphere, and all perturbing gravitational forces are neglected. A model for the central gravity force will be discussed in Section 3.1.1.

Once the atmosphere becomes noticeable, the aerodynamic lift and drag will become of major importance as well. The aerodynamic coefficient C_L and C_D are defined with respect to the aerodynamic reference frame. As a result, Eqs. (2.11) and (2.12) express the lift L and drag D acceleration in the aerodynamic reference frame as well. In these equations, ρ is the atmospheric density, V_A is the airspeed, S_{ref} is the reference area and m is the vehicle mass.

$$L = \frac{1}{2}\rho V_A^2 \frac{S_{ref} C_L}{m} \tag{2.11}$$

$$D = \frac{1}{2}\rho V_A^2 \frac{S_{ref} C_D}{m} \tag{2.12}$$

The aerodynamic force on a vehicle $\boldsymbol{F}^I_{aero}$ can be expressed in vector form. Using Eqs. (2.11) and (2.12), as well as using a transformation as defined in Section 2.1 from the aerodynamic reference frame to the inertial planetocentric reference frame, the aerodynamic force vector can be constructed:

$$\boldsymbol{F}^I_{aero} = m\, \boldsymbol{C}^{I/R}\boldsymbol{C}^{R/V}\boldsymbol{C}^{V/T}\boldsymbol{C}^{T/A}\begin{bmatrix} -D & 0 & -L \end{bmatrix}^T \tag{2.13}$$

The perturbing forces $\boldsymbol{F}^I_{pert}$ are considered to be forces other than the aerodynamic and central gravity forces. Chapter 3 will contain a brief analysis on the different forces and their magnitude will be compared to determine if they need to be included. As will be shown, the J_2 effect is the largest perturbing acceleration. However, as the set-up of the simulator is quite general, the accelerations due to a third body will also be included in the simulator.

The equations of translational motion is related to the derivative of the position vector, and is given by:

$$\dot{\boldsymbol{R}}^I = \frac{d\boldsymbol{R}^I}{dt} = \boldsymbol{V}^I \tag{2.14}$$

Combining $\boldsymbol{F}^I_{grav}$, $\boldsymbol{F}^I_{aero}$ and $\boldsymbol{F}^I_{pert}$ using Eq. (2.10), the total external force on the vehicle can be obtained. However, for the numerical integrator, it is more convenient to express the equations of motion in terms of accelerations. From Newton's second law, as was shown in Eq. (2.10), it can be found that the force and the acceleration are related by the vehicle mass m. By dividing Eq. (2.10) by the vehicle mass, the acceleration due to the external forces can be obtained. Combining these accelerations with equations of translational motion form the complete set of equations of motion in the inertial planetocentric reference frame. Then, a state vector $\boldsymbol{X}^I$ is defined as:

$$\boldsymbol{X}^I = \begin{pmatrix} \boldsymbol{R}^I & \boldsymbol{V}^I \end{pmatrix}^T = \begin{pmatrix} x_I & y_I & z_I & \dot{x}_I & \dot{y}_I & \dot{z}_I \end{pmatrix}^T \tag{2.15}$$

Using the state vector from Eq. (2.15), a differential equation in the form $\dot{\boldsymbol{X}}^{\boldsymbol{I}} = f(t, \boldsymbol{X}^I)$ can be formed using where the function $f(t, \boldsymbol{X}^I)$ is given by Eqs. (2.10) and (2.14). These differential equations can be propagated using a numerical integrator, such that, given an initial state, the resulting trajectory can be obtained.

2.3 Flight parameters

For numerical integration, it is convenient to integrate in Cartesian coordinates. However, these coordinates are not easy to interpret and therefore no conclusions could be drawn from the output of the numerical integration in Cartesian coordinates. In addition, a number of variables is necessary to perform the frame transformations presented in Section 2.1. It is, however, possible to obtain these variables from the Cartesian state, as will be discussed in this section. The derivation of the equations shown in this section can be found in Mooij (1997).

First of all, two vectors will be defined that are derived from the Cartesian state, namely a position vector $\boldsymbol{R}^I$ and a velocity vector $\boldsymbol{V}^I$. The definition of these two vectors, given in the inertial planetocentric reference frame, is given by:

$$\boldsymbol{R}^I = \begin{bmatrix} x_I & y_I & z_I \end{bmatrix}^T \tag{2.16}$$

$$\boldsymbol{V}^I = \begin{bmatrix} \dot{x}_I & \dot{y}_I & \dot{z}_I \end{bmatrix}^T \tag{2.17}$$

For a number of flight variables, the position and velocity in the rotating planetocentric reference frame is required. The transformation from Eq. (2.5) is used to obtain the vectors $\boldsymbol{R}^R$ and $\boldsymbol{V}^R$ in the rotating planetocentric reference frame:

$$\boldsymbol{R}^R = \boldsymbol{C}^{R/I}\, \boldsymbol{R}^I \tag{2.18}$$

$$\boldsymbol{V}^R = \boldsymbol{C}^{R/I}\, \left(\boldsymbol{V}^I - \boldsymbol{\omega}_{cb} \times \boldsymbol{R}^I \right) \tag{2.19}$$

in which $\boldsymbol{\omega}_{cb}$ is a vector containing the rotational velocity of the central body. Note that $\boldsymbol{V}^R$, in case no winds are taken into account, is the velocity of the vehicle with respect to the atmosphere.

From the position and velocity vectors in the rotating planetocentric reference frame, the latitude δ and the longitude τ can be obtained using:

$$\delta = \arcsin\left(\frac{z_R}{||\boldsymbol{R}^R||} \right) \tag{2.20}$$

$$\tau = \arctan\left(\frac{y_R}{x_R} \right) \tag{2.21}$$

To determine the heading and flight-path angle, another transformation is necessary. This time, the velocity vector in the planetocentric reference $\boldsymbol{V}^R$ frame needs to be transformed to obtain the velocity the vertical reference frame $\boldsymbol{V}^V$:

$$\boldsymbol{V}^V = \boldsymbol{C}^{V/R}\, \boldsymbol{V}^R \tag{2.22}$$

Now consider the three components of $\boldsymbol{V}^V$ to be v_N, v_E and v_D respectively. From these three velocity components, the heading angle χ and the flight-path angle γ can be obtained using:

$$\chi = \arctan\left(\frac{v_N}{v_E}\right) \tag{2.23}$$

$$\gamma = -\arcsin\left(\frac{v_D}{||\boldsymbol{V}^V||}\right) \tag{2.24}$$

Using the position vector $\boldsymbol{R}^R$ and the radius of the planetary body, the altitude can be determined. The altitude is defined as the closest distance with respect to the surface of the planetary body. Assuming a spherical planet with radius R_p, the altitude h of the vehicle with respect to the surface is given by:

$$h = \sqrt{x_R^2 + y_R^2 + z_R^2} - R_p \tag{2.25}$$

Finally, although not necessary for the integration itself, the mechanical load on the vehicle will be discussed. To prevent any structural damage, the mechanical load will need to be limited to a maximum value. For this reason, the mechanical load n_G will also be calculated for each time step, and is a function of the aerodynamic accelerations. The expression used to calculate the mechanical load using:

$$n_G = \frac{\sqrt{L^2 + D^2}}{g_0} \tag{2.26}$$

in which $g_0 = 9.81\,\mathrm{m/s^2}$, and L and D are the lift- and drag accelerations given by Eqs. (2.11) and (2.12).

3 Environment

By assuming an unpowered vehicle, an aerogravity assist maneuver is dominated by gravitational and aerodynamic accelerations. The magnitude of the different accelerations depend on the environment in which a vehicle has to operate. To obtain these accelerations, environmental models are used. Very basic models exist with limited accuracy, while the more detailed ones require more computational effort. Taking this in mind, the different environmental models used to simulate an aerogravity assist are described in this chapter. First of all, the models for the gravity field are discussed in Section 3.1. The atmosphere models will be presented in Section 3.2. Finally, the method of obtaining the ephemerides of planetary bodies is shown in Section 3.3.

3.1 Gravity Field

During an aerogravity assist, the vehicle is under the influence of gravitational acceleration of a central body. These gravitational accelerations are determined by the mass of the central body, as well as the mass distribution of the planet. This requires a model of the the gravity field of the planets. In addition, perturbing forces will be exerted on the vehicle. The accelerations due to the central gravity field and perturbing accelerations due to the uneven mass distribution and third-bodies will be discussed in this section.

Figure 3.1 shows the magnitude of the different accelerations during a typical aerogravity assist at Mars. In the capture phase, where the atmospheric density is very low, the central gravitational acceleration and the perturbing J_2 acceleration are the most dominant accelerations. However, as the density increases, the aerodynamic acceleration becomes larger. This is the largest acceleration until the vehicle leaves the atmosphere again. The perturbing acceleration due to the Sun is two orders of magnitude lower than the smallest aerodynamic acceleration, and can therefore be considered to be negligible.

The first gravitational acceleration that will be discussed is the central gravitational acceleration (Section 3.1.1). As the acceleration due to J_2 is larger than the aerodynamic acceleration during the capture phase, this acceleration will be discussed next, as well as the possibility of adding higher-order terms of the spherical harmonics model (Section 3.1.2). As the setup of the simulator is quite general, the acceleration of perturbing bodies will also be included (Section 3.1.3), although it will hardly have any impact on the results. However, including this acceleration will make it possible to include perturbation of third bodies in future research that makes use of the developed simulator.

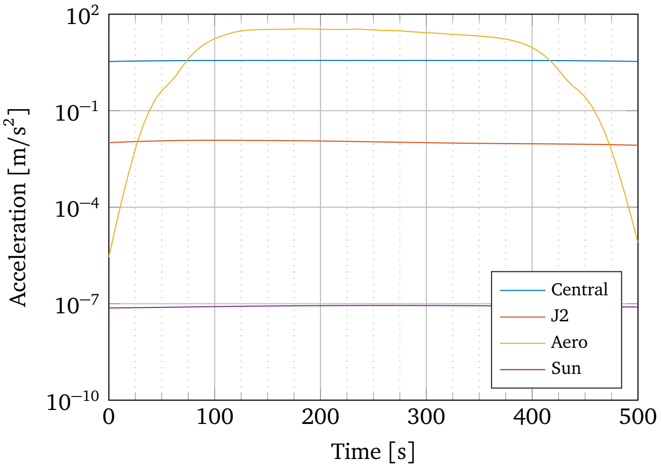


Figure 3.1: Magnitude of different accelerations during a typical aerogravity assist at Mars

3.1.1 Central gravity

The gravitational attraction force between two bodies which are considered to be point masses m_1 and m_2, is described by Newton's law of universal gravitation, given by:

$$\mathbf{F} = -G\frac{m_1 m_2}{R^3}\boldsymbol{R} \tag{3.1}$$

where G is the gravitational constant ($G = 6.67259 \times 10^{-11}\,\mathrm{m^3/kg/s^2}$) and $\boldsymbol{R}$ is the position vector of m_1 relative to m_2 (Wakker, 2010). Equation (3.1) does not only holds for point masses, but also for bodies with constant mass distribution. In that case, $\mathbf{R}$ gives the position of the center of mass of m_1 relative to m_2. Combining Eq. (3.1) with Newton's second law, and using $M = m_1 + m_2$ results in:

$$\frac{d^2\mathbf{R}}{dt^2} = -\frac{GM}{R^3}\boldsymbol{R} = -\frac{\mu}{R^3}\boldsymbol{R} \tag{3.2}$$

in which the gravitational parameter μ is introduced. If it is assumed that the mass of a vehicle is much smaller than the body it is orbiting (i.e., $m_2 \ll m_1$), then M is the mass of the central body. The masses and radii of the main bodies in the Solar System can be obtained from SPICE kernels, see also Section 6.1. As an overview, the masses and radii of major Solar System bodies are listed in Table 3.1.

Table 3.1: Masses and radii of major Solar System body (Lissauer and De Pater, 2013)

Body	Mean Radius (km)	Mass ($\times 10^{24}$ kg)
Mercury	2440	0.3302
Venus	6051.8	4.8685
Earth	6371.0	5.9736
Mars	3389.9	0.641 85
Jupiter	71 492	1898.6
Saturn	60 268	568.46
Uranus	25 559	86.832
Neptune	24 766	102.43
Sun	696 000	1.989×10^6

3.1.2 J$_2$ acceleration

A planetary body never has a constant mass distribution. This would make Eq. (3.2) invalid. To take the unequal mass distribution in a body into account, the gravitational potential U can be used (Wakker, 2010). The gravitational potential is defined by:

$$\begin{aligned} U = -\frac{GM}{R}\Bigg[1 - \sum_{n=2}^{\infty} J_n \left(\frac{R_p}{R}\right)^n P_n(\sin\delta) \quad + \\ \sum_{n=2}^{\infty}\sum_{m=1}^{n} J_{n,m}\left(\frac{R_p}{R}\right)^n P_{n,m}(\sin\delta)\left\{\cos m(\Lambda - \Lambda_{n,m})\right\}\Bigg] \end{aligned} \tag{3.3}$$

in which R_p is the mean equatorial radius of the body, and δ is the latitude. To obtain gravitational acceleration in a desired direction, the derivative of U should be taken with respect to that direction. Furthermore, J_n, $J_{n,m}$ and $\Lambda_{n,m}$ are model parameters. P_n and $P_{n,m}$ are Legendre polynomials and associated Legendre functions of the first kind.

As Eq. (3.3) consists of a summation of a central gravity term (see also Section 3.1.1) and perturbing terms corresponding to a zonal ($m = 0$), sectoral ($n = m$) or tesseral ($n \neq m$) harmonics coefficients, the perturbing acceleration due to the irregularity of the gravity field can expressed for each coefficient. The first four zonal harmonics coefficients are listed in Table 3.2.

For all planetary bodies listed in Table 3.2, the J$_2$ coefficient is the largest, while all other terms are, except for Venus, several orders of magnitude smaller. This indicates that the acceleration due to J$_2$ is the

Table 3.2: Zonal harmonics coefficients for axisymmetric bodies (Lissauer and De Pater, 2013)

Body	J_2	J_3	J_4	J_6
	$(\times 10^{-6})$	$(\times 10^{-6})$	$(\times 10^{-6})$	$(\times 10^{-6})$
Mercury	22.5		6.5	
Venus	4.46	−1.93	−2.38	
Earth	1082.627	−2.532	−1.620	−0.21
Mars	1960.5	31.5	−15.5	
Jupiter	14 696.4		−587	35
Saturn	16 290.7		−936	86
Uranus	3343.5		−28.9	
Neptune	3410		−35	

most dominant perturbing acceleration due to the irregularity of the gravity field, which is the reason only this coefficient is included in the simulator. The tesseral and sectoral harmonics coefficient are smaller than the J_2 coefficient as well, and will be negligible with respect to the aerodynamic and central gravity accelerations as was shown in Fig. 3.1.

By only considering the J_2 term from Eq. (3.3), the gravitational force can be expressed by:

$$\boldsymbol{F}^I_{J2} = -m\frac{3}{2}\mu J_2 \frac{R_p^2}{R^5}\begin{bmatrix} x_I(1-5z_I^2/R^2) \\ y_I(1-5z_I^2/R^2) \\ z_I(3-5z_I^2/R^2) \end{bmatrix} \tag{3.4}$$

in which m is the mass of the vehicle, μ is the gravitational parameter, J_2 is the second zonal harmonics coefficient, R_p is the radius of the planetary body, R is the radial distance of the vehicle with respect to the center of the planetary body, and x_I, y_I and z_I are the position coordinates of the vehicle in the inertial planetocentric reference frame (Wakker, 2010).

3.1.3 Third body perturbations

Accelerations caused by bodies a spacecraft is not orbiting are called third-body perturbations. An example of this type of perturbation is an acceleration caused by the Moon on an Earth-orbiting satellite, see Fig. 3.2.

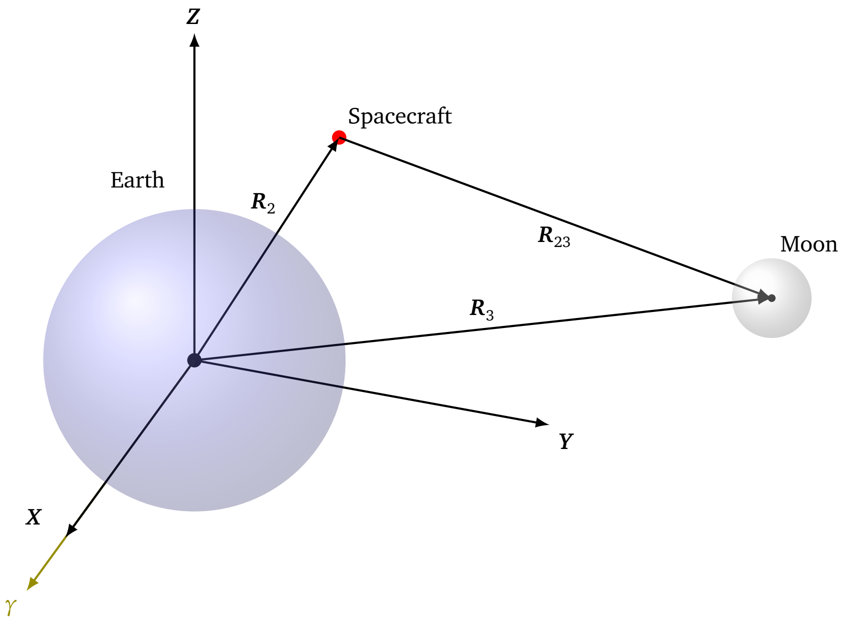


Figure 3.2: Third body perturbation (Curtis, 2014)

The general equation for the acceleration due to n perturbing bodies is described by:

$$\frac{d^2\boldsymbol{R}}{dt^2} = \sum_{\substack{j=i \\ j\neq i}}^{n-1} \mu_j \left(\frac{\boldsymbol{R}_j - \boldsymbol{R}_i}{|\boldsymbol{R}_j - \boldsymbol{R}_i|^3} - \frac{\boldsymbol{R}_j}{|\boldsymbol{R}_j|^3} \right) \tag{3.5}$$

in which μ_j is the gravitational parameter of body j and $\boldsymbol{R}_j$ and $\boldsymbol{R}_i$ are the position vector of the perturbing body j and the vehicle with respect to the central body respectively. Simplifying Eq. (3.6) to the case where there is only one perturbing body, such as the case in Fig. 3.2, is given by:

$$\frac{d^2\boldsymbol{R}}{dt^2} = \mu_3 \left(\frac{\mathbf{R}_{23}}{R_{23}^3} - \frac{\mathbf{R}_3}{R_3^3} \right) \tag{3.6}$$

in which μ_3 is the gravitational parameter of the third body, $\mathbf{R}_{23}$ is the relative distance with respect to the spacecraft and the third body, and $\mathbf{R}_3$ is the distance from the main body to the perturbing body.

As can be observed from Eqs. (3.5) and (3.6), not only the mass of the third body is important, but also the distance with respect to the spacecraft and the main body. This distance can be determined using the ephemerides of the Solar-System bodies. A method of obtaining these ephemerides will be discussed in Section 3.3.

To determine whether an acceleration due to third bodies should be considered as a perturbation, the concept of the sphere of influence should be introduced. The sphere of influence is defined as the a sphere around a planetary body where the gravitational force of that body is dominant. All gravitational forces due to other bodies can be considered as perturbations. The radius of the sphere of influence can be approximated by:

$$R_{SoI} = a \left(\frac{m}{M} \right)^{2/5} \tag{3.7}$$

in which R_{SoI} is the radius of the sphere of influence, a is the semi-major axis of the the smaller body with respect to the larger body, and m and M are the masses of the smaller and larger body respectively (Wakker, 2010). It was found that the largest third-body acceleration was caused by the Sun, and the magnitude of this acceleration was much lower than, for example, the perturbation due to J_2 effect, as shown in Fig. 3.1. Therefore, it would be acceptable to neglect third-body accelerations. However, for the sake of completeness, the perturbing acceleration of the Sun has been included into the simulator. If, in future analysis, the effect of perturbing accelerations due to, for example, moons, needs to be investigated, the used acceleration model can be extended quite easily.

3.2 Atmosphere

An atmosphere is the enabling factor for an aerogravity assist. The density and temperature of an atmosphere is therefore required. In general, the density and temperature in the atmosphere is dependent on altitude, but temporal and spatial variations exist as well. In this section, an analytic model is presented to describe the atmospheric properties and aavailable atmospheric models for different planets are discussed. The content of this section is based on Tewari (2007) and Lissauer and De Pater (2013).

First, the general description of an exponential atmosphere model will be discussed in Section 3.2.1. However, more detailed atmosphere models exist. For Earth, Mars and Venus, more detailed atmosphere models are discussed in Sections 3.2.2 to 3.2.4. Even though only atmosphere models for Mars and Jupiter will be used in this thesis, more atmosphere models are included in the software such that they can be used in future research, and will therefore also be discussed in this section. Finally, in Section 3.2.5, the composition of the different atmospheres is discussed.

3.2.1 Exponential atmosphere

An exponential atmospheric model is certainly not the most accurate atmospheric model available, but it provides a rough estimate of the atmospheric properties without being too computationally intensive. For the gas giants, no detailed atmosphere model was publicly available. For these planets, an exponential atmosphere model is necessary. This section will describe the principles of an exponential atmosphere.

Table 3.3: Scale heights for different atmospheres (Williams, 2014)

Planets	Scale height [km]	Surface density [kg/m^3]	Surface pressure [bar]	Avg. temperature [K]
Earth	8.5	1.217	1.014	288
Mars	11.1	0.020	6.36	210
Venus	15.9	65	92	737
Jupiter	27	0.16†	$\gg$1000	165†
Saturn	59.5	0.19†	$\gg$1000	134†
Uranus	27.7	0.42†	$\gg$1000	76†
Neptune	20.3	0.45†	$\gg$1000	72†

†at 1 bar

The atmosphere of a planet is formed by gasses that are kept close to the planetary surface by gravitational forces. The thermodynamic properties of the gases in the atmosphere define the aerothermodynamic loads on vehicles passing through an atmosphere. Variables that represent a thermodynamic state are temperature T, pressure p and density ρ. For a perfect gas, the relation between these three variables is given by:

$$p = \rho R T \tag{3.8}$$

Here, R is the specific gas constant, which depends on the gas. The specific gas constant for a gas with molecular mass m is given by:

$$R = \frac{\mathscr{R}}{m} \tag{3.9}$$

where $\mathscr{R}$ is the universal gas constant ($\mathscr{R} = 8.3145 \ \frac{\mathrm{J}}{\mathrm{K \cdot mol}}$). If it is assumed that the gas in the atmosphere is at rest in a vertical direction, the pressure and gravity forces balance. This can be expressed in the hydrostatic equation, given by:

$$dp = -\rho g dh \tag{3.10}$$

which holds for a thin slice of the atmosphere. Using the previously mentioned equations, it is possible to derive a relation for the density as a function of altitude. This results in the barometric law, given by:

$$\frac{p}{p_0} = e^{-\int_0^h dh/h_s(h)} \tag{3.11}$$

in which h_s is the scale height of the atmosphere, given by

$$h_s = \frac{R(h)T(h)}{g(h)} \tag{3.12}$$

Combining both Eqs. (3.10) and (3.11), an analytic expression for the density as a function of altitude can be obtained, given by:

$$\frac{\rho}{\rho_0} = e^{-\frac{h-h_0}{h_s}} \tag{3.13}$$

The temperature, pressure and density of a planet can therefore be described by Eqs. (3.8) and (3.13). A value for the scale height h_s is required to use these equations, as well as a value for the surface density and pressure. In Table 3.3, an overview of scale heights, surface densities and surface pressures for different atmospheres are given.

3.2.2 Earth atmospheric model

A commonly used model for the Earth's atmosphere is called the 1976 US Standard Atmosphere. A standard atmosphere is a proposed vertical distribution of atmospheric properties such as temperature, density and pressure which, by international agreement, represents the yearly average, mid-latitude, atmospheric conditions (Mooij, 2013). This 1976 US Standard Atmosphere model is given by a number of tabulated vales. Using these tabulated values and the equations from Section 3.2.1, the temperature, density and pressure can be determined at each altitude in the Earth's atmosphere.

A different empiric model of the Earth's atmosphere has been developed by the US Naval Research Laboratory, and is called the NRLMSISE-00. Developed in 2000, this model does include temporal and spatial variations and is also depending on solar activity. The model is based on measurements from a mass spectrometer and an incoherent scatter radar. The model, written in FORTRAN, is available for free (Picone et al., 2003).

Another atmosphere model is the so-called Earth Global Reference Atmospheric Model (Earth-GRAM), which is an engineering-level model of the Earth's atmosphere and is described by Justus et al. (2004). Reference atmospheres contain, in contrast to standard atmospheres, seasonal and spatial influences on the atmospheric properties, as well as aspects like geomagnetic and solar effects. Not only is Earth-GRAM capable of providing the density and temperature as a function of altitude, latitude and longitude, it provides also other information such as atmospheric constituents. From sea-level up to an altitude of 27 km, Earth-GRAM is based on the National Oceanic and Atmospheric Administration Global Upper Air Climate Atlas (GUACA). Up to an altitude of 120 km, the data is based on the Middle Atmosphere Program (MAP). At altitudes larger than 120 km, the atmospheric model is based on the NASA Marshall Engineering Thermosphere (MET) model. Next to the spatial variation of atmospheric parameters, the Earth-GRAM also includes a variation of the atmospheric parameters as a function of time.

For the developed simulator, only a single detailed atmospheric model will be included per planetary body. Therefore, a trade-off needs to be made between these three different atmospheric models. The US76 is a model which is relatively easily to implement since it consists only of a table with numerical data and a few simple equations. However, this type of model does not include spatial or temporal variations, which are included in Earth-GRAM and NRLMSISE-00. Another benefit of the Earth-GRAM model is that it is well documented, properly validated and there are software examples available for interfaces with different programming languages. So, the preferred atmospheric model is the Earth-GRAM model. However, according Barbara Fawcett from NASA Marshall Space Flight Center, it was determined that Earth-GRAM is only available to citizens from the United States (personal communication, March 2015). Therefore, the NRLMSISE-00 model was used instead.

3.2.3 Mars atmospheric model

The atmospheric model that is commonly used for Mars is called the Mars Global Reference Atmospheric Model (Mars-GRAM) (Justh et al., 2006). This atmospheric model has been used to predic the aerobraking for the the Mars Global Surveyor mission. Although in early versions of Mars-GRAM data from missions to Mars has been used, newer versions use information based on the NASA Ames Mars General Circulation Model (MGCM) and the University of Arizona Mars Thermospheric General Circulation Model (MTGCM).

Mars-GRAM, as shown in Justus et al. (2002), provides the variation of temperature, density and pressure and wind components as a function of altitude, latitude, time of day and the celestial longitude of the Sun as viewed from Mars. The MGCM provides provides data from the surface to an altitude of 80 km. From 80 km to 170 km, the MTGCM is used. Above 170 km, a modified Stewart-type thermospheric model is used which is dependent on latitude and longitude. This modified model includes also dependence on the solar activity.

A different Martian atmospheric model, called the Mars Climate Database, is described by Millour et al. (2014), and is the required model for all ESA missions. The Mars Climate Database is derived from the General Circulation Model numerical simulations and has been validated by observational data. The database also includes post-processing schemes such as high-resolution spatial interpolation of the environmental data. The most important output of the database are parameters like the temperature, density and pressure, but it is also capable of providing other variables, like the concentrations of different species, dust mass mixing ratios and winds.

For the simulation of an aerogravity assist around Mars, only a single detailed atmospheric model will be implemented. Therefore, a trade-off was made between Mars-GRAM and the Mars Climate Database. A comparison between the two models has been performed by Carbonne et al. (2011), and it was concluded that both models are coherent in cold and average cases, while a comparison performed by Mooij et al. (2006) shows a preference for the Mars Climate Database. As the Mars-GRAM model was readily available, it was decided to use this atmosphere model instead of the Mars Climate Database. In addition, the Mars-GRAM and Venus-GRAM have been constructed almost identically, which would require a more or less identical interface between the FORTRAN code and C++ code for both models. Therefore, using Mars-GRAM will reduce the required programming effort.

3.2.4 Venus atmospheric model

For Venus, the Venus Global Reference Atmospheric Model (Venus-GRAM) is a commonly used atmospheric model developed by NASA. For altitudes lower than 250 km, the Venus-GRAM is based on the Venus International Reference Atmosphere (VIRA). The Venus-GRAM has extended the VIRA model to an altitude of 1000 km (Justh et al., 2006).

After contact with Hilary Justh from NASA Marshall Space Flight Center, it was determined that Venus-GRAM is publicly available (personal communication, March 2015). Since this appears to be the most complete, well-documented and verified model available, this model will be included in the developed simulator.

3.2.5 Atmospheric composition

The composition of the atmosphere determines the magnitude of the convective heat, as will be shown in Section 4.5.1. Lissauer and De Pater (2013) provides an overview of the atmospheric composition for the different planetary atmospheres, which is shown in Tables 3.4 and 3.5.

From the detailed atmospheric models, it is possible to obtain the atmospheric composition as a function of altitude. For Earth, Mars and Venus, the atmospheric composition as a function of altitude is shown in Fig. 3.3. From this figure, it can be observed that only approximately at altitudes lower than 100 km, the composition of the atmosphere matches the values shown in Table 3.4. At higher altitudes, the composition will start to differ from the values in Table 3.4. As the aerodynamic heating will not only depend on the atmospheric composition but also on the atmospheric density, it is can be concluded that the variation of the atmospheric composition will hardly have any effect on the aerodynamic heating as the composition remains constant for the altitudes where an aerogravity assist will be executed.

3.3 Ephemerides

To calculate the acceleration due to perturbing bodies, their relative position is necessary and the heliocentric velocity of a vehicle can be determined by adding the planetocentric velocity of the vehicle to the velocity of the planetary body. These two examples illustrate the need of a method of estimating the positions of the different planetary bodies. Although analytical models exist, see for example Standish (2011), it is more convenient to use an off-the-shelf solution. The SPICE toolkit (Arora and Russell, 2010), which has an interface with the TU Delft Astrodynamic Toolbox (Tudat), is therefore the preferred solution to obtain ephemerides data.

The SPICE toolkit has been developed by the Navigation and Ancillary Information Facility, which is a part of NASA. It consists of data sets which contains navigation and other ancillary information, such as spacecraft ephemerides, planet ephemerides and physical, dynamical and cartographic constants of different target bodies.

SPICE makes use of Jet Propulsion Laboratory Development Ephemerides, or short DE, followed by a number which indicate the version of the Development Ephemeris. By default, Tudat contains DE421, which was released by JPL in 2008. With respect to earlier versions, it contains updated ephemeris data for several spacecraft such as the Mars spacecraft and Venus Express, as well as some updated measurements for the Moon and Pluto. DE421 covers the period from 1900 to 2050. The most recent version of the Development Ephemeris, DE430, was released in 2013. It has a much longer time span than DE421: from 1550 to 2650. Although it covers a longer time span and it contains the most current version of the Development Ephemeris, the improved accuracy will not be noticeable for the application it is mainly intended: estimating the third-body perturbations and transforming the planetocentric state of a vehicle to a heliocentric state. Therefore, the default DE421 that is included in Tudat will be used.

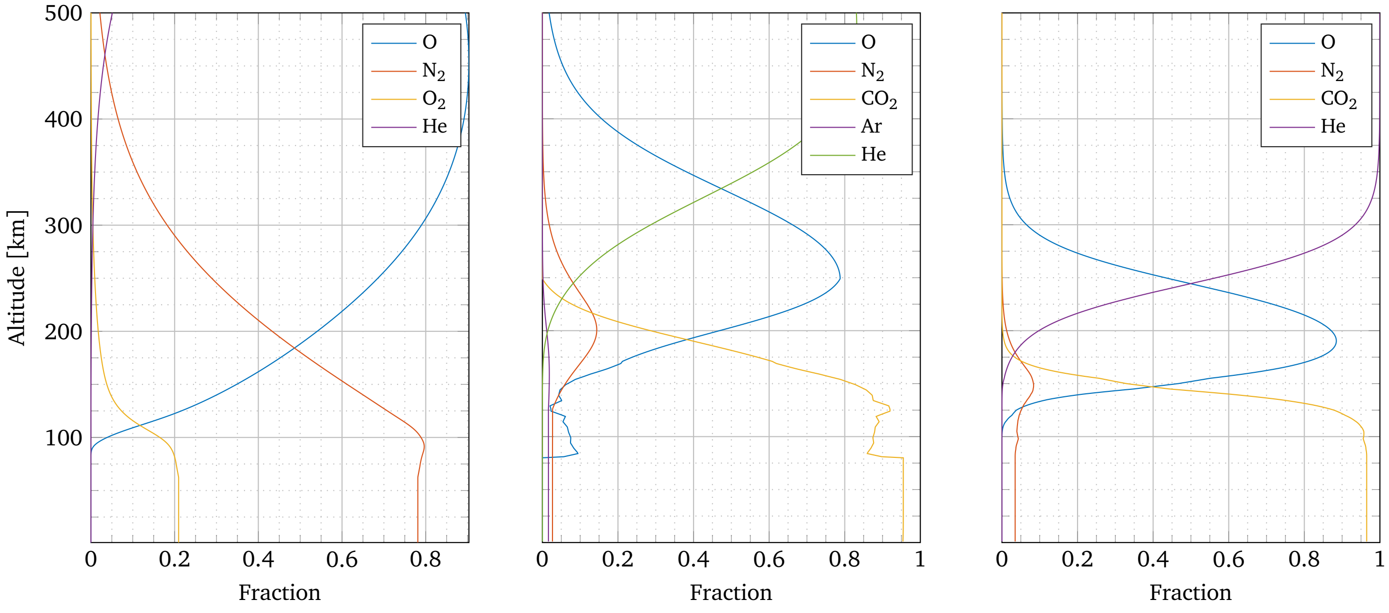
Table 3.4: Atmospheric composition of the terrestrial planets (Lissauer and De Pater, 2013)

Constituents	Earth	Venus	Mars
N_2	0.7808	0.035	0.027
O_2	0.2095	0-20 ppm	0.0013
CO_2	345 ppm	0.965	0.953
CH_4	2 ppm		10-250 ppb
H_2O	<0.03	30 ppm	<100 ppm
Ar	0.009	70 ppm	0.016
CO	0.2 ppm	20 ppm	700 ppm
O_3	10 ppm		0.01 ppm
SO_2	< 2 ppb	100 ppm	

ppm = Parts Per Million, ppb = Parts Per Billion

Table 3.5: Atmospheric composition of the giant planets (Lissauer and De Pater, 2013)

Constituents	Jupiter	Saturn	Uranus	Neptune
H_2	0.864	0.88	0.83	0.82
He	0.136	0.119	0.15	0.15
H_2O	$> 4.2 \times 10^{-4}$			
CH_4	2.0×10^{-3}	4.5×10^{-3}	0.023	0.03
NH_3	7×10^{-4}	5×10^{-4}		
H_2S	7.7×10^{-5}			



(a) Earth **(b)** Mars **(c)** Venus

Figure 3.3: Atmospheric composition of the terrestrial planets

4
Aerothermodynamics

Aerodynamic forces on a vehicle are necessary to be able to perform an aerogravity assist. Most conventional (re-)entry vehicles, such as capsules or winged vehicles such as the Space Shuttle, have a low lift-to-drag ratio, which results in a relatively large energy dissipation due to atmospheric drag. As will be discussed in Section 4.2, most authors suggest the use of a waverider for aerogravity assists. However, a limited amount of aerodynamic data is available for waveriders. This requires the use of a method, which capable of determining the aerodynamic characteristics of a waverider.

In Section 4.3, the methods used to determine the aerodynamic forces on an arbitrary waverider shape is discussed. Since these methods depend on the type of flow, different flow regimes are first discussed in Section 4.1. Using these methods, a trade-off between different waverider shapes can be made based on the found aerodynamic characteristics. This will be shown in Section 4.4. Finally, due to the high velocities that will be encountered during an aerogravity assist, large heat fluxes are expected. Therefore, a method for estimating the aerodynamic heating will be presented in Section 4.5.

4.1 Flow regimes

The aerodynamic characteristic depend not only on the shape of the vehicle, but also on the type of flow. Three flow regimes can be identified: the free molecular flow regime, the transition flow regime and the continuum flow regime (Regan and Anandakrishnan, 1993). The flow regimes can be identified by means of the Knudsen number. The Knudsen number, according to Anderson (2006), is a non-dimensional number and depends on the mean free path length:

$$Kn = \frac{\lambda}{L} = \frac{m_p / \left(\sqrt{2}\pi\sigma^2\rho\right)}{L} \tag{4.1}$$

in which λ is the mean free path length, L is a characteristic length, m_p is the mass of the particles, σ is the collision diameter of the particles and ρ is the free stream density. The three flow regimes identified in Regan and Anandakrishnan (1993) are:

Free molecular flow	$Kn > 10$
Transition flow	$0.001 < Kn < 10$
Continuum flow	$Kn < 0.001$

In the free molecular flow regime, the mean free path length of the gas is so large that the flow cannot be considered as a continuum flow. The flow should therefore be considered as individual particles moving in a rectilinear path. The continuum flow regime is the flow regime where the collisions between particles predominate, which is not the case in the free molecular flow regime. In the transition between these two flow regimes, the assumptions of free molecular flow or continuum flow are no longer completely valid.

4.2 Vehicle

As was explained in Chapter 1, a vehicle that generates lift is necessary to perform an aerogravity assist. A vehicle with a large lift-to-drag ratio is desirable to prevent the vehicle from losing too much energy due to atmospheric drag. This eliminates vehicle designs such as capsules and winged entry vehicles. Lohar et al. (1994), Lavagna et al. (2005) and Armellin et al. (2007) all propose a waverider as a vehicle for an aerogravity assist due to its high lift-to-drag ratio. An example of the geometry of a waverider is shown in Fig. 4.1a.

Starkey and Lewis (1999) states that a waverider can be described by five parameters, namely l, w, θ, δ, and n. The parameters w and l correspond to the width and length of the vehicle, as is shown in Fig. 4.1b. The parameters θ and δ are the wedge angle and the oblique shock angle, respectively, shown in Fig. 4.1c. The parameter n is related to the curvature of the planform. Using these five parameters, the upper and lower surface, can be described:

$$y_p = Ax^n \tag{4.2}$$

$$z_u = (y_u/B)^{1/n} \tag{4.3}$$

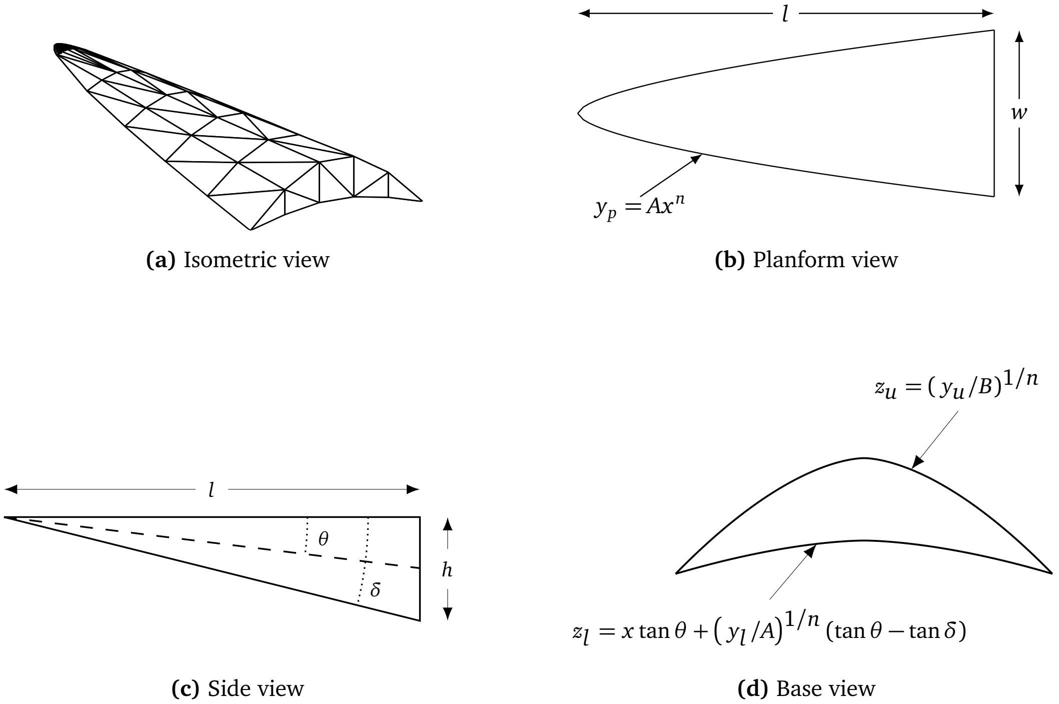
$$z_l = x\tan\theta + (y_l/A)^{1/n}(\tan\theta - \tan\delta) \tag{4.4}$$

In these equations, $A = \frac{1}{2}w/l^n$ and $B = A/\tan^n\delta$. The geometry of the planform is described by Eq. (4.2). The curvature of the upper surface is given by z_u, while the lower surface is defined by z_l. Both are shown in Fig. 4.1d.

To be able to compare the aerodynamics of one vehicle with another, a reference area is defined. For a waverider, the planform area will be used. The planform area can be calculated, according to Starkey and Lewis (1999), using:

$$S_{ref} = \frac{wl}{n+1} \tag{4.5}$$

Varying the five parameters of the waverider has an influence on the shape of the vehicle. Fig. 4.2 shows how the waverider shape varies for different values of n and A, which is a function of the width and length.



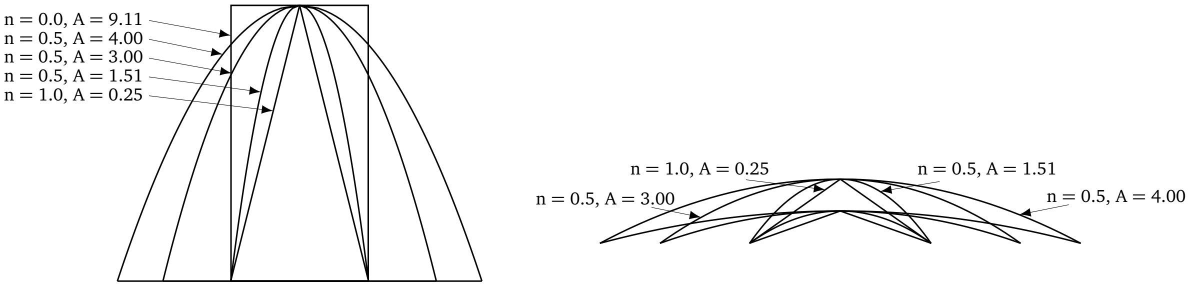
(a) Isometric view

(b) Planform view

(c) Side view

(d) Base view

Figure 4.1: Geometry of a typical waverider, adapted from Starkey and Lewis (1999)



(a) Planform

(b) Base

Figure 4.2: Influence of n and A on waverider planform and base shape (Starkey and Lewis, 1999)

For $n = 0$, the waverider planform is a rectangle, for $n = 1$, the planform is a triangle. For values of $0 < n < 1$, a curved planform is created. Furthermore, from Fig. 4.2, it can be seen that for increasing values of A the width of the vehicle increases, which is in agreement with the definition of A.

4.3 Aerodynamics

The aerodynamic force on a vehicle can be determined using Eqs. (2.11) and (2.12), and can be used for any vehicle shape as long as the aerodynamic coefficients C_L and C_D are known, as well as the reference area S_{ref}. The reference area for a waverider can be obtained from Eq. (4.5). In this section, the method used to obtain the lift and drag coefficients will be discussed.

Assume that an arbitrary vehicle shape can be discretized by a total of N panels. Each panel has a normal unit vector $\boldsymbol{n}_i$ and a tangential unit vector $\boldsymbol{t}_i$. The normal unit vector is defined positive pointing outwards, while the unit tangential vector is defined by:

$$\boldsymbol{t}_i = \frac{(\boldsymbol{n}_i \times \boldsymbol{V}) \times \boldsymbol{n}_i}{\|(\boldsymbol{n}_i \times \boldsymbol{V}) \times \boldsymbol{n}_i\|} \tag{4.6}$$

where $\boldsymbol{V}$ is the velocity vector of the incoming flow. Assuming that for each panel the pressure coefficient $C_{p,i}$ and the tangential coefficient $C_{t,i}$ are available, then the aerodynamic force can be obtained by integrating the forces as a result of normal pressure and shear stress per panel over the entire vehicle:

$$\boldsymbol{F}^B_{aero} = -\sum_{i=1}^{N} \left(C_{p,i} A_i \boldsymbol{n}_i + C_{t,i} A_i \boldsymbol{t}_i \right) \tag{4.7}$$

in which A_i represents the area of the ith panel. The minus sign was introduced to reverse the direction of the normal vector and tangential vector, since this is the direction in which the normal pressure and shear stress act. The aerodynamic force coefficients in the aerodynamic reference frame can then be obtained by:

$$\boldsymbol{C}^A_{F,aero} = \begin{pmatrix} C_L \\ C_S \\ C_D \end{pmatrix} = \frac{1}{S_{ref}} \boldsymbol{C}^{A/B} \boldsymbol{F}^B_{aero} \tag{4.8}$$

Note that the transformation matrix from the body-fixed reference frame to the aerodynamic reference frame $\boldsymbol{C}^{A/B}$ is a function of the angle of attack α and the sideslip angle β as shown in Eq. (2.9). If, for a given vehicle shape, the angle of attack, the angle of sideslip and the Mach number are varied, the aerodynamic coefficients calculated using Eqs. (4.7) and (4.8) can be stored in an aerodynamic database. To obtain the aerodynamic coefficients for an arbitrary combination of angle of attack, angle of sideslip and Mach number, a numerical interpolator can be used.

The normal- and tangential vectors, as well as the areas of each panels can be determined from geometry. The normal pressure coefficient and shear stress coefficients, however, are still missing. These coefficients are a function of the composition and density of the flow, as well as velocity and the orientation of the vehicle with respect to the flow. It is therefore not trivial to obtain these coefficients. For the study presented in this thesis, two methods of determining the normal pressure coefficients and shear stress coefficients are compared.

4.3.1 Local inclination method

The first method that is considered is a so-called local inclination method. As discussed in Anderson (2006), the normal pressure and shear stress in these methods only depend on the inclination θ of a panel with respect to the free-stream flow, as shown in Fig. 4.3. The normal pressure and shear stress for panels that are not directly in view of the free-stream flow are set to zero. These panels are considered to be in the "shadow" of the vehicle. Different relations exist for the three flow regimes, as will be shown below.

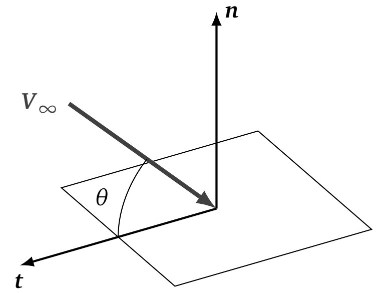


Figure 4.3: Definition of local inclination θ

- The pressure coefficient for a **continuum flow** can be estimated by the Modified Newtonian method (Anderson, 2006). The shear stress in this method is zero, whereas the normal pressure coefficient is given by:

$$C_p = C_{p,max} \sin^2 \theta \tag{4.9}$$

in which

$$C_{p_{max}} = \frac{2}{\gamma M_\infty^2} \left[\left[\frac{(\gamma+1)^2 M_\infty^2}{4\gamma M_\infty^2 - 2(\gamma-1)} \right]^{\gamma/(\gamma-1)} \left[\frac{1-\gamma+2\gamma M_\infty^2}{\gamma+1} \right] - 1 \right]$$

where γ is the specific heat ratio of the gas and M_∞ is the Mach number of the free-stream. Eq. (4.9) is therefore a function of the composition of the gas and the velocity of the flow. For $\gamma = 1$ and $M \to \infty$, the magnitude of $C_{p,max}$ becomes 2, which results in $C_P = 2\sin^2\theta$. This corresponds to the definition of the straight, non-modified, Newtonian Method, which is also given by Anderson (2006).

- For a **free molecular flow**, the normal pressure coefficient and the shear stress coefficient are given by Hart et al. (2014), and are shown in Eqs. (4.10) and (4.11) respectively.

$$\begin{aligned} C_p = \frac{1}{s^2} \Bigg[\left(\frac{2-\sigma_N}{\sqrt{\pi}} s \sin\theta + \frac{\sigma_N}{2} \sqrt{\frac{T_w}{T_\infty}} \right) e^{-(s\sin\theta)^2} & + \\ \left\{ (2-\sigma_N) \left((s\sin\theta)^2 + \frac{1}{2} \right) + \frac{\sigma_N}{2} \sqrt{\frac{\pi T_w}{T_\infty}} s \sin\theta \right\} (1 + \mathrm{erf}(s\sin\theta)) \Bigg] \end{aligned} \tag{4.10}$$

$$C_t = \frac{\sigma_T \cos\theta}{s\sqrt{\pi}} \left[e^{-s^2 \sin^2\theta} + \sqrt{\pi} s \sin\theta [1 + \mathrm{erf}(s\sin\theta)] \right] \tag{4.11}$$

In Eqs. (4.10) and (4.11) T_w is the wall temperature, T_∞ is the free-stream temperature, M_∞ is the free-stream Mach number and σ_N and σ_T are the normal and tangential angular momentum coefficients respectively. Furthermore, C_p and C_t for a free molecular flow are also a function of the molecular speed ratio s, which is given by:

$$s = M_\infty \sqrt{\frac{\gamma}{2}} \tag{4.12}$$

- For the **transition flow regime**, Ronse (2013) has shown a function that can combine the results of the continuum flow and the free molecular flow depending on the Knudsen number of the flow. This bridging function is given by:

$$F = \sin^2 \left(\pi \frac{2 + \log_{10} Kn}{8} \right) \tag{4.13}$$

If F is equal to 0, the flow corresponds to a continuum flow whereas if F is equal to 1, the flow corresponds to a free molecular flow. The function F is related to the normalized force coefficient

$\bar{C}_X$ as shown in Eq. (4.14). The subscript *free* corresponds to the free molecular flow and *cont* to the continuum flow regime.

$$\bar{C}_X = \frac{C_X - C_{X_{cont}}}{C_{X_{free}} - C_{X_{cont}}} = F \tag{4.14}$$

Ronse (2013) developed software that implements these local inclination methods, of which only the method for continuum flow was implemented in the TU Delft Astrodynamics Toolbox (Tudat). The source code of the work performed by Ronse (2013) was available, and was used to develop aerodynamic databases for a waverider shapes. This tool required a quadrilateral mesh to be able to compute the aerodynamic coefficients. An example of a quadrilateral mesh for a waverider is shown in Fig. 4.4a.

4.3.2 Direct Simulation Monte Carlo method

The second method that was considered was the Direct Simulation Monte Carlo (DSMC) method (Bird, 2013). At Knudsen numbers larger than 0.05, the continuum description of a gas is not valid since the mean free path can no longer be neglected. This means that the Navier-Stokes equations that are usually solved in Computational Fluid Dynamics programs cannot be used anymore.

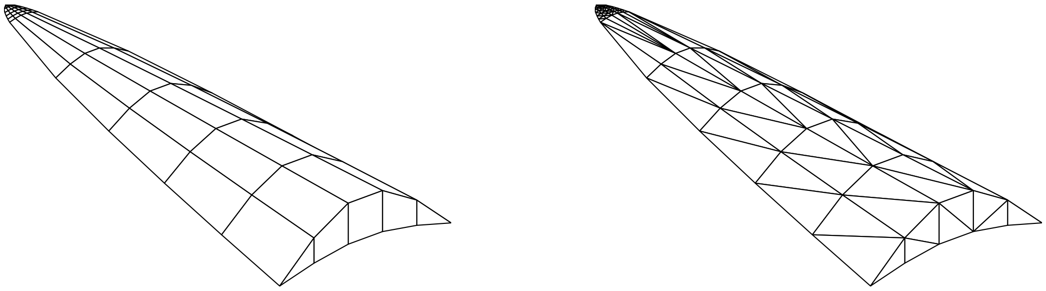
The Direct Simulation Monte Carlo method, however, can be used in the rarefied flow regime. It can be considered as a Monte Carlo method for solving the Boltzmann equation, an equation that describes the behavior of a gas. In theory, the Boltzmann equation is also valid for a continuum flow, at which it would reproduce the results of the Navier-Stokes equation. However, according to Alexander and Garcia (1997), the simulation of a gas at standard conditions in a volume of one cubic millimeter would require a computational time of 10^2 years using a teraflop machine. From this, it can be concluded that it is only feasible to use the DSMC method at high Knudsen numbers where the flow can be considered to be rarefied.

Consider a group of identical particles forming a gas without an internal degree of freedom, where the molecules in the gas are considered to be point masses and therefore cannot rotate of vibrate. Assume that each particle or molecule in the gas has a position $\boldsymbol{x} = (x_1, x_2, x_3)$ and a velocity $\boldsymbol{c} = (c_1, c_2, c_3)$, which is a function of time. The behavior of a gas can be described by a distribution function $f(\boldsymbol{x}, \boldsymbol{c}, t)$. Then, the standard notation of the Boltzmann equation, as described by Kremer (2010), is shown below, where it is implied that a summation is performed over the indices $i = 1, 2, 3$.

$$\frac{\partial f}{\partial t} + c_i \frac{\partial f}{\partial x_i} + F_i \frac{\partial f}{\partial c_i} = \left[\frac{\partial f}{\partial t}\right]_{coll} \tag{4.15}$$

In Eq. (4.15), the term $[\partial f / \partial t]_{coll}$ describes the change in the distribution function f due to collisions. The derivation of this partial derivative is beyond the scope of the thesis work. Furthermore, the particles are subjected to an external force field $\boldsymbol{F} = (F_1, F_2, F_3)$. Implementation of the Boltzmann equation to develop a DSMC method was not feasible within the time frame of a MSc thesis project. For this reason, an open source DSMC simulator was used.

Several DSMC software tools are available. The one used in this thesis work is called the Stochastic Parallel Rarefied-Gas Time-accurate Analyzer, or SPARTA for short (Gallis et al., 2014). This tool, developed



(a) Quadrilateral mesh

(b) Triangular mesh

Figure 4.4: Different types of meshes

by Sandia National Laboratories, a United States Department of Energy laboratory, is distributed as an open source code under the General Public License (GPL).

SPARTA makes use of an input file which contains all the parameters to start a simulation. The dimensions of the simulation box, the velocity of the flow, the number density composition of the background gas and the vehicle shape are only a few of the parameters that can be defined in the input file. SPARTA is only able to read a triangulated surface mesh consisting of a list of vertices and a list of triangles. An example of a triangulated surface mesh of a waverider is shown in Fig. 4.4b. One of the requirements of the surface mesh is that it should be watertight, such that no particles can enter the vehicle. A more elaborate discussion on the use of SPARTA is given in Appendix A.

There are many options for presenting the results of a simulation in SPARTA. For example, the flow of particles can be printed to an image file and the state of each particle can be written to a text file. Not only the state of the particles, but also the state of each cell of the computational grid as well as the state at each panel of the surface can be printed to an image file or written to a text file. Especially that last option is useful for determining the aerodynamic forces on a vehicle shape, since SPARTA calculates the normal pressure and shear stress for each of the panels of a vehicle. Integrating the normal pressure and shear stress over the entire vehicle shape will, as shown earlier in this chapter, result in the total aerodynamic force on the vehicle.

4.3.3 Comparison of methods

The local inclination method for free molecular flow was compared with the results of a DSMC simulation, as shown in Fig. 4.5. The DSMC method is not suitable to simulate continuum flow due to long computation times, which is the reason it is not compared with the Modified Newtonian method. As is visible from Fig. 4.5, a discontinuity occurs between approximately −5° and 0° for the local inclination method, something that one would not expect. This discontinuity does not appear in the results of the DSMC simulation.

An explanation for the discontinuity can be found using Eq. (4.11) and the fact that for panels in the shadow zone, the coefficients C_p and C_t are zero. Consider a single panel at a particular angle of attack such that the local inclination θ of that panel with respect to the flow is close to zero. Since it scales with $\cos\theta$, the C_t is maximum for small local inclination angles. If the angle of attack is adjusted only a little such that the panel is in the shadow zone, the complete contribution of the shear stress disappears for that particular panel, whereas it had a maximum value earlier. It is believed that this behavior is the result of the discontinuities in Fig. 4.5.

The location of the discrete jumps also correspond with the shape of the waverider used. For the result from Fig. 4.5, a waverider with a wedge angle of 5° was used. This basically means that at an angle of attack of −5°, the behavior of C_t discussed in the previous paragraph occurs at the bottom surface of the waverider. This same behavior occurs at the upper surface at an angle of attack of 0°. Especially in Fig. 4.5b, the discrete jumps at 0° and −5° are clearly visible, which corresponds with their expected location.

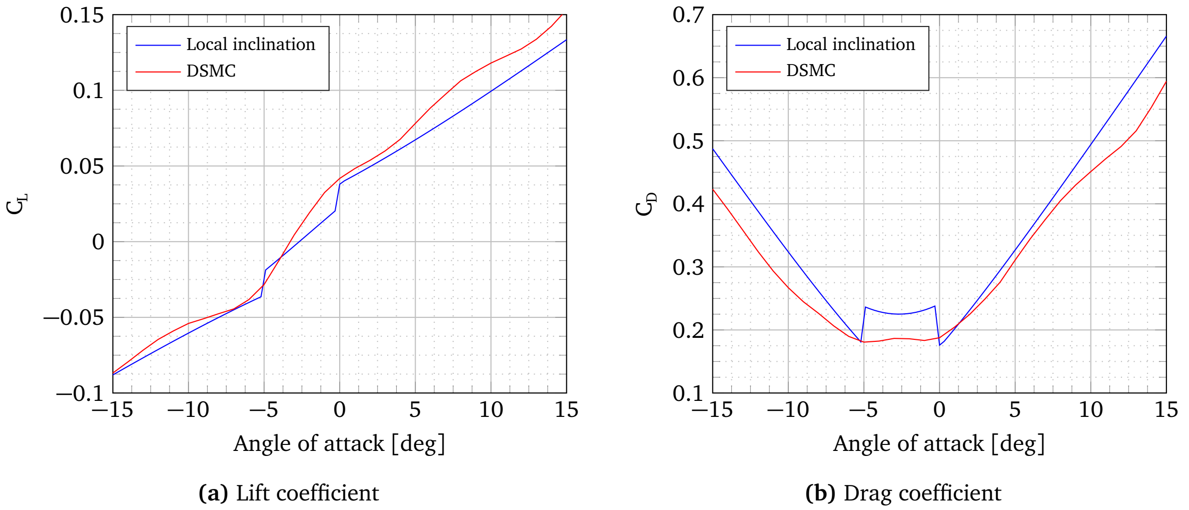
The reason the local inclination method for a free molecular flow worked well in Ronse (2013), is most likely the result of the vehicle shape used. The model of the Delta-K upper stage model used by him has more curves and contains no major sharp edges, something, which is the case for a waverider. A local inclination method to estimate the aerodynamic coefficients in the free molecular flow regime is therefore not suitable for waveriders. Instead, a DSMC proves to be a better method for modeling the aerodynamic coefficients for a rarefied flow.

4.3.4 Base pressure

The software developed by Ronse (2013) assumed that for panels that were in the shadow zone of the flow, the pressure coefficient would be zero. This would also be the case for the base pressure, i.e., the pressure at the rear of the waverider. Gabeaud (1950) proposed a correction term to take into account this base pressure. This correction term, that should be applied to the panels that form the base of the waverider, is given by:

$$C_{p,b} = \frac{2}{\gamma M_\infty^2}\left\{\left(\frac{2}{\gamma+1}\right)^{1.4}\left(\frac{1}{M_\infty}\right)^{2.8}\left[\frac{2\gamma M_\infty^2-(\gamma-1)}{\gamma+1}\right]-1\right\} \tag{4.16}$$

It is clear that if M_∞ approaches infinity, $C_{p,b}$ becomes zero. This yields that for large Mach numbers, the contribution of the correction term becomes negligible and the Modified Newtonian method will become



(a) Lift coefficient

(b) Drag coefficient

Figure 4.5: Comparison of local inclination method and DSMC for a waverider with $w = 2.0\,\text{m}$, $l = 5.0\,\text{m}$, $n = 0.6$, $\theta = 5.0°$ and $\delta = 7.0°$

more accurate. The result of adding the base pressure is visible in Fig. 4.6. If the standard Modified Newtonian method will be used without base pressure correction, the drag coefficient increases as the Mach number increases from $M_\infty = 1$ onward. However, it is expected that around $M_\infty = 1$ a peak in the drag coefficient would occur, which is clearly not the case. Note that Fig. 4.6 is hatched between $M_\infty = 1$ and $M_\infty = 5$, as the Modified Newtonian method can only be applied for $M_\infty > 5$. When applying the correction term, however, this peak in the drag coefficient does occur which confirms the successful implementation the Gabeaud base-pressure correction.

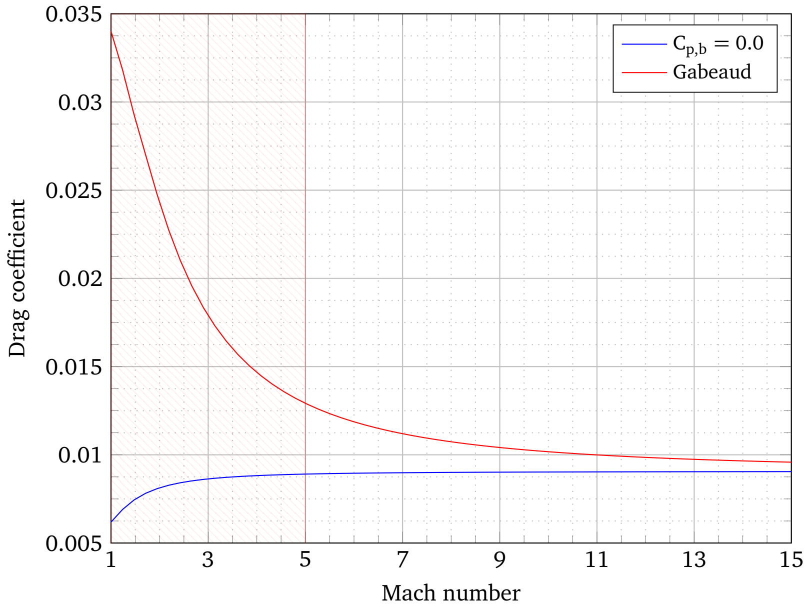


Figure 4.6: Effect of applying the Gabeaud base pressure correction term for a waverider with $w = 2.0\,\text{m}$, $l = 5.0\,\text{m}$, $n = 0.6$, $\theta = 5.0°$ and $\delta = 7.0°$ at $\alpha = 0°$

4.4 Vehicle selection

The shape of the waverider that will be used in an aerogravity assist maneuver has a large impact on the performance of such a maneuver. If, for example, a waverider is selected, which has a rather large drag coefficient, it is likely that the atmospheric bending angle that can be achieved is less due to the larger energy loss. Furthermore, even if it is assumed that the same bending angle would be achieved, the departure velocity will be less in case of a high-drag waverider shape.

To asses the performance of different waverider shapes, as well as the influence of the design parameters, a number of waverider shapes was generated and evaluated using SPARTA. The design parameters θ, δ and n were varied in the following ranges, while keeping the width and length at 2 and 5 m respectively.

$$\begin{aligned}\theta &\in \{5°, 7°, 9°, 12°, 15°, 18°\}\\ \delta &\in \{5°, 7°, 9°, 12°, 15°, 18°\}\\ n &\in \{0.5, 0.6, 0.7, 0.8\}\end{aligned}$$

Using these sets of θ, δ and n resulted in a total of 144 different waverider shapes. A number of geometries for varying θ, δ and n is shown in Figs. 4.8a to 4.8h. Each of these waverider shapes was evaluated using SPARTA, using a flow velocity of 8000 m/s, a number density of 1.2×10^{19} m^{-3} and using an atmospheric composition of 0.79 N_2 and 0.21 O_2. It was observed that varying the atmospheric composition had only minor effect on the results and will not be further investigated. Furthermore, the ratio of physical particles to simulation particles was 0.5×10^{16}, using a diffuse collision model with an accommodation coefficient of 0.9. The maximum lift-to-drag ratio for each of the 144 waverider shapes is plotted in Fig. 4.7 as a function of the volume of the waverider.

The reason that the maximum lift-to-drag ratio is plotted as a function of volume is that the volume is a measure for the amount of payload that can be fitted within the waverider. So ideally, the volume is as large as possible. However, larger volume might also result in an increase in the amount of drag, which decreases the lift-to-drag ratio. Therefore, lift-to-drag ratio and volume are two contradictory requirements. From the 144 waverider shapes, three specific configurations will be highlighted, each in a different color in Fig. 4.7. These three cases are:

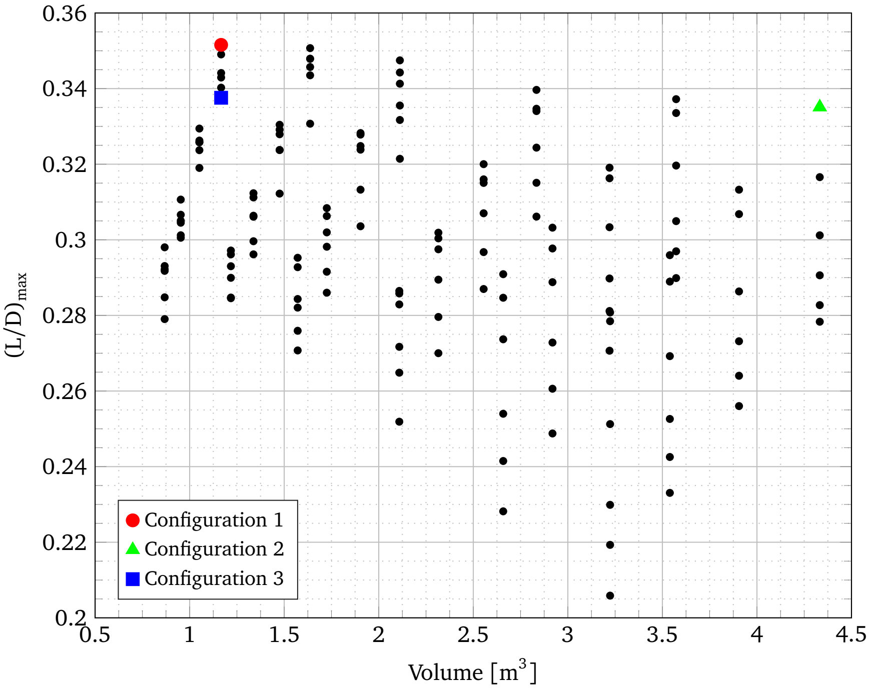
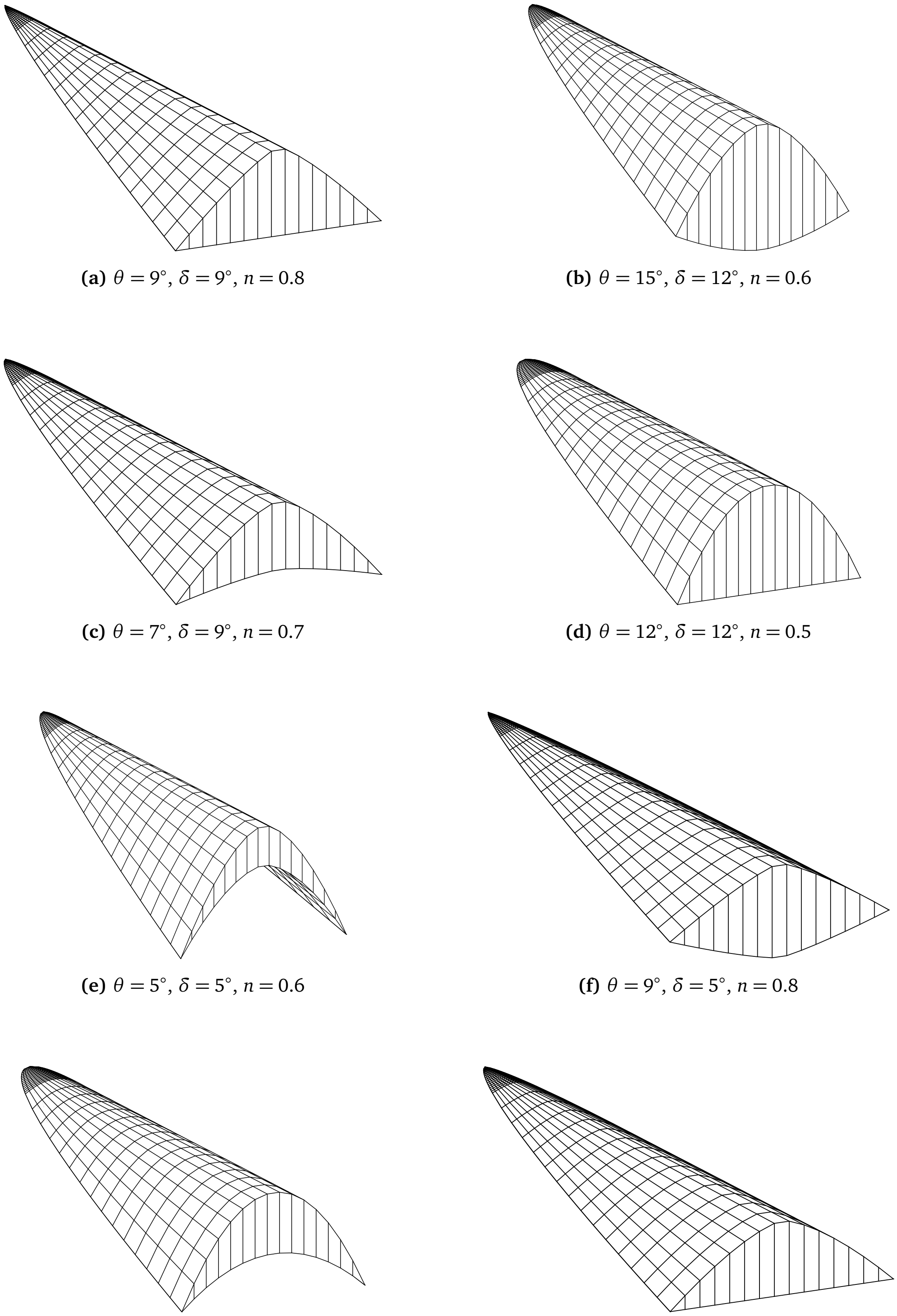


Figure 4.7: Maximum lift-to-drag ratio for 144 waverider shapes



(a) $\theta = 9°$, $\delta = 9°$, $n = 0.8$

(b) $\theta = 15°$, $\delta = 12°$, $n = 0.6$

(c) $\theta = 7°$, $\delta = 9°$, $n = 0.7$

(d) $\theta = 12°$, $\delta = 12°$, $n = 0.5$

(e) $\theta = 5°$, $\delta = 5°$, $n = 0.6$

(f) $\theta = 9°$, $\delta = 5°$, $n = 0.8$

(g) $\theta = 7°$, $\delta = 12°$, $n = 0.5$

(h) $\theta = 7°$, $\delta = 7°$, $n = 0.7$

Figure 4.8: A selection of different waverider geometries

- **Configuration 1** Maximum lift-to-drag ratio
- **Configuration 2** Maximum vehicle volume with with maximum lift-to-drag ratio
- **Configuration 3** Minimum drag coefficient at maximum lift-to-drag ratio

The lift-to-drag ratio as a function of the angle of attack of these three waveriders obtained using SPARTA is plotted in Fig. 4.9a, their geometries are shown in Figs. 4.10a to 4.10c and their properties are listed in Table 4.1.

Table 4.1: Three selected configurations and their properties

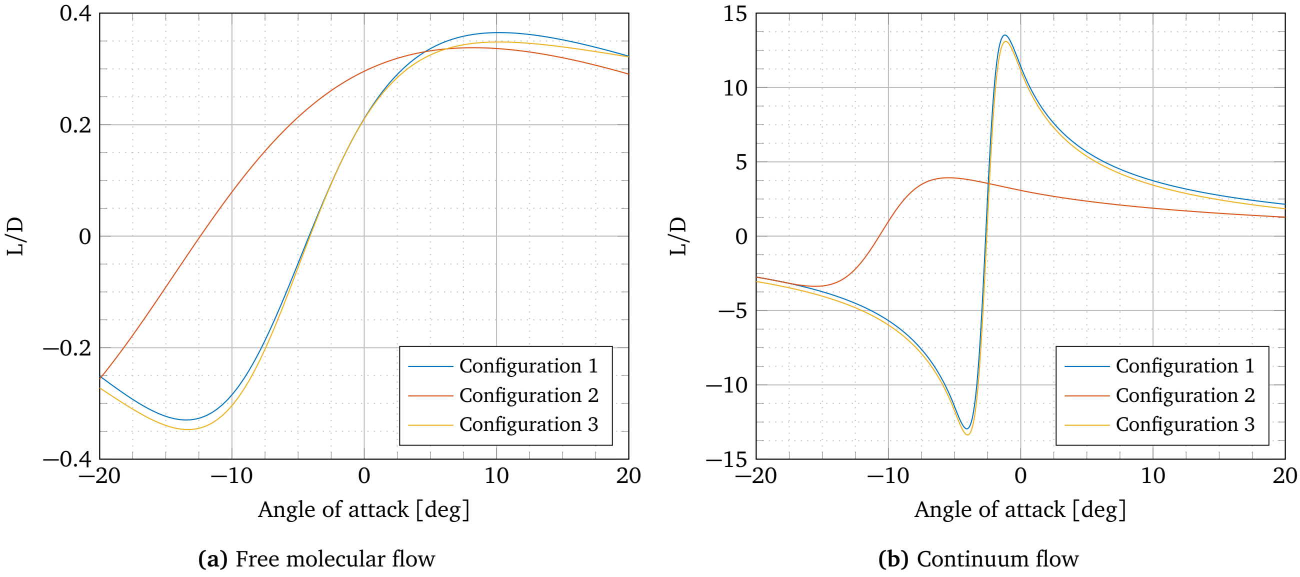
Config.	w	l	n	θ	δ	L/D	C_L	C_D	V
1	2.0	5.0	0.5	5.0	9.0	0.3516	0.2215	0.6300	1.17
2	2.0	5.0	0.5	18.0	18.0	0.3351	0.3784	1.1292	4.33
3	2.0	5.0	0.5	5.0	5.0	0.3376	0.1734	0.5138	1.17

$[w] = [l] = m$, $[\theta] = [\delta] = °$, $[V] = m^3$

Configuration 1, which has the largest lift-to-drag ratio of all the evaluated vehicles, has a relatively small volume compared to Configuration 2, which is the vehicle with the largest volume. Configuration 2 has a 240% increase in volume compared to Configuration 1, although the maximum lift-to-drag ratio decreases by 4.7%. The small decrease in the maximum lift-to-drag ratio seems to be a good trade of the increase that is gained in the volume, but Configuration 2 has also 78% larger drag coefficient at maximum lift-to-drag ratio. The origin of this increase in drag coefficient can be explained by the large increase in frontal area, which is clearly visible in Fig. 4.10b. One might consider to increase the width and the length of a vehicle rather than increasing the values for θ and δ for increasing the available volume to avoid the increase in drag coefficient.

Configuration 3, which is the configuration with the smallest drag coefficient at maximum lift-to-drag ratio, has the same volume as Configuration 1. The maximum lift-to-drag ratio for Configuration 3 is, however, is approximately 4.0% less than the maximum lift-to-drag ratio of Configuration 1. The drag coefficient for Configuration 3 is 18.4% smaller than the drag coefficient at maximum lift-to-drag ratio for Configuration 1. This lower drag coefficient could reduce the energy loss in the atmosphere.

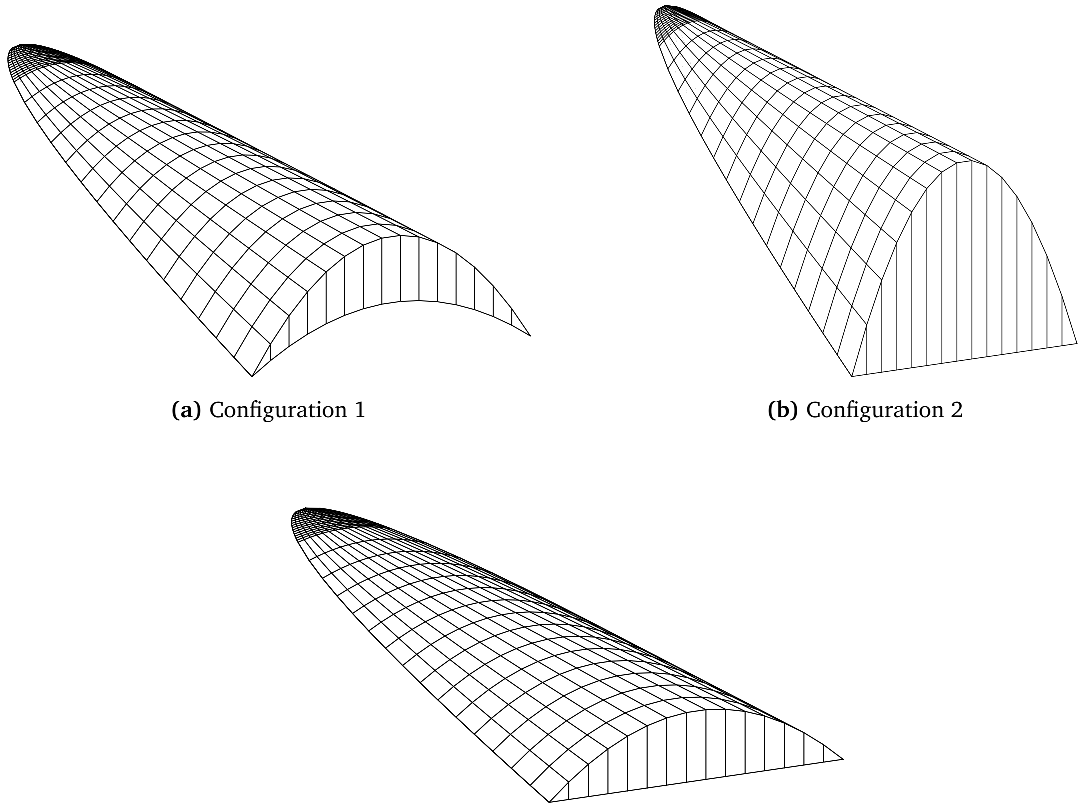
In addition to the free molecular flow regime, in which the vehicle configuration analysis has been performed so far, it is also necessary to obtain the aerodynamic coefficients for the same three vehicles in the



(a) Free molecular flow

(b) Continuum flow

Figure 4.9: Lift-to-drag ratio as a function of the angle of attack for the three selected vehicle configurations in free molecular flow ($V = 8,000\,\mathrm{m/s}$) and continuum flow ($M = 40$)



(a) Configuration 1

(b) Configuration 2

(c) Configuration 3

Figure 4.10: The three selected configurations

continuum flow regime. These coefficients were obtained using the Modified Newtonian method. The lift-to-drag ratio for the three configurations in continuum flow for $M = 40$ is shown in Fig. 4.9b.

When comparing the results from Fig. 4.9a with the results found for the coefficients in the free molecular flow, similar conclusions can be drawn. The lift-to-drag versus angle of attack curve for Configuration 1 and 3 are almost identical and Configuration 2 has a significantly lower lift-to-drag ratio, the result of the large drag due to the large frontal area. The magnitude of the lift-to-drag ratio as well as the drag coefficient agree with the results that were found in Sandlin and Pessin (1993).

Although it would be interesting to investigate relation between the three selected waverider shapes and the impact they have on the possible aerogravity assists, it was too time consuming to do this. Instead, since literature suggests that high lift-to-drag ratios are desirable, Configuration 1 will be used in the remainder of the performed research. For Configuration 1, two aerodynamic databases are made: one for the continuum flow generated with the aerodynamic code from Ronse (2013) and one for the free-molecular flow generated with SPARTA. The lift- and drag coefficient are determined as a function of the velocity of the flow and the angle of attack. The aerodynamic coefficients for the transition flow regime are determined at runtime, where the results from the free-molecular flow regime and the continuum flow regime are both averaged using the Eq. (4.13).

For the equations of motion, it is also necessary to determine the reference area and the vehicle mass. Based on values found in literature, a vehicle mass of 800 kg has been used. The reference area used to determine the aerodynamic forces can be determined using Eq. (4.5), and is $6.67\,\mathrm{m}^2$ for Configuration 1. For an aerogravity assist at Mars, this reference area did not result in sufficiently large lift force, which would result in a skip out of the atmosphere even with a full lift-down bank-angle profile. To account for this limitation of the selected waverider dimensions, while still being able to fly at maximum lift-to-drag ratio, the reference area for the analysis on Mars was increased by a factor 10. This would, taking into account the same aspect ratio of 2:5, result in a waverider with $l = 15.8\,\mathrm{m}$ and $w = 6.32\,\mathrm{m}$. It is expected that these dimensions are beyond the current launch capabilities, and it seems unlikely that such a vehicle will be able to maneuver in the recent future.

Of course, increasing the angle of attack will also result in a larger lift force, but will automatically increase the drag which will in turn increase the energy dissipation, limiting the possible atmospheric bending angles. By investigating the effect of the reference area in more detail, possibly in combination with a larger angle of attack, the required reference area that is necessary could be reduced to more acceptable values, but this is left for future research. If from this research, it is shown that a reference area would be required indeed exceeds the current launch capabilities, alternative solutions to this problem should be found. A suggestion to increase the reference area can be derived from the concept of inflatable aeroshells, as for example discussed by Player et al. (2005). This is a concept used in the design of aero-assisted entries, and allows to increase the size of the aeroshell at much larger sizes than possible using rigid aeroshells. So-far, the geometry of inflatable aeroshells is often similar to those of blunt entry capsules, which is not the geometry required for an aerogravity assist. The concept of inflatable aeroshells could prove a valuable means of increasing the reference area of a waverider while remaining within launch vehicle constraints.

4.5 Aerodynamic heating

Viscous effects during interaction of the flow with the boundary layer result in a decrease in kinetic energy of the flow. Part of this energy increases the internal energy of the gas, whereas another part will be transferred from the gas into the body.

There are three different mechanisms in aerodynamic heating. The first mechanism is thermal conduction, which occurs if there is a temperature gradient in a gas. The second mechanism is thermal radiation, which occurs when the temperature of a gas is large enough such that it starts to radiate. The last mechanism is catalytic recombination of atomic gases, which occurs when molecules are formed on the surface and they give up their latent heat of dissociation to the surface (J.J. Bertin, 1992). For the latter mechanism, specialized software is required to estimate the heat loads, which is not available. Also, when looking at literature, for example Armellin et al. (2007) and Casoliva et al. (2008), only the convective and radiative heat loads are taken into consideration for aerogravity assists. Therefore, for only the first two mechanisms, engineering models will be presented to estimate the aerodynamic heating.

4.5.1 Convective heating

To precisely predict the convective heating, the three-dimensional flow around a body should be determined using an Computational Fluid Dynamics (CFD) technique. Using boundary conditions on the outer edge, it would be possible to determine the boundary layer profile and, as a result, obtain the temperature gradient in the boundary layer, from which the convective heat flux could be obtained. However, calculating the heat distribution in three dimensions using this method is extremely time consuming and computationally intensive.

A general stagnation-point convective-heating equation was proposed by Sutton and Graves (1971). This method takes into account the composition of the atmosphere. The expression for the convective heating is given by

$$q_c = K\sqrt{\frac{p_s}{R_N}}\left(h_s - h_w\right) \tag{4.17}$$

in which q_c is the convective heat flux in MW/m^2, p_s is the stagnation pressure in bars, R_N is the nose radius, h_s is the total enthalpy and h_w is the wall enthalpy.

The constant K is the heat-transfer coefficient. This coefficient is a function of the atmospheric composition, as can be seen in Eq. (4.18). In this equation, Pr is the Prandtl number, $c_{o,i}$ is the mass fraction of species i, $\gamma_{o,i}$ is the transport parameter and $M_{o,i}$ is the molecular mass. The unit of K is kg/s $\cdot$ m$^{-3/2}$ $\cdot$ atm$^{-1/2}$.

$$K = \frac{0.0885}{Pr^{0.6}}\left(\sum \frac{c_{o,i}}{M_{o,i}\gamma_{o,i}}\right)^{-1/2} \tag{4.18}$$

The transport parameter γ in Eq. (4.18) is a function of the collision diameter σ and the ratio of the maximum energy of attraction between colliding molecules over the Boltzmann constant, ϵ/k. The definition of the transport parameter is given by Sutton and Graves (1971) and is shown below:

Table 4.2: Molecular properties (Svehla, 1962)

Molecule	M	σ	ϵ/k
	g/mol	Å	K
H	1.008	2.708	37.0
N	14.008	3.298	71.4
O	16.000	3.050	106.7
H_2	2.016	2.827	59.7
N_2	28.02	3.798	71.4
O_2	32.000	3.467	106.7
CO	28.01	3.690	91.7
CO_2	44.01	3.941	195.2
Ar	39.944	3.542	93.3
He	4.003	2.551	10.22
H_2O	18.02	2.641	809.1

$1\text{Å} = 10^{-10}$ m

$$\gamma_i = \frac{1}{\sigma_i^2 \left(\frac{\epsilon_i}{k}\right)^{0.15}} \tag{4.19}$$

In Table 4.2, the molecular mass M, the collision diameter σ and the ratio of the maximum energy of attraction between colliding molecules over the Boltzmann constant ϵ/k are shown for the molecules that are the most significantly present in the atmospheres of the planets considered, as was discussed in Section 3.2.5.

Note that Eq. (4.17) uses the stagnation pressure and enthalpy to calculate the convective heat flux. It is more common to express the heat flux as a function of density and velocity, which is also done in for example Casoliva et al. (2008). Furthermore, Eq. (4.17) is using atmospheres as unit for pressure. After some derivation, and using the more conservative cold-wall model, it can be shown that Eq. (4.17) can also be expressed by

$$q_c = k_c \sqrt{\frac{\rho}{R_N}} V^3 \tag{4.20}$$

in which

$$k_c = \left[\frac{1}{4}\sqrt{2}(101325)^{-1/2}\right] K$$

In contrast to Eq. (4.17), Eq. (4.20) uses only SI units. The unit of q_c is therefore now W/m^2. From Eq. (4.20), it is clear that the convective heat flux is a function of atmospheric density ρ, velocity V and the atmospheric composition.

4.5.2 Radiative heating

At high velocities, which occur during an aerogravity assist, radiative heating is a significant thermal load as well. It can even become larger than the convective heating. For accurate calculation of the radiative heat flux, the composition of the gas should be calculated, including the ionized, dissociated and recombined molecules. Since this is a computational intensive process, it is more convenient to use analytical expressions to calculate the radiative heat flux.

An expression for the radiative heat flux in W/m^2 is given by Tauber and Sutton (1991) and is shown in Eq. (4.21), in which R_N is the nose radius in meters, ρ is the atmospheric density in kg/m^3, $f(V)$ are tabulated values and C, a and b are constants or functions of density or velocity and depend on the planet of interest.

$$q_{rad} = C R_N^a \rho^b f(V) \tag{4.21}$$

The tabulated values for Earth are listed in the first two columns of Table 4.3. The constants for Earth are given below:

$$C = 4.736 \times 10^8$$
$$a = 1.072 \times 10^6 V^{-1.88} \rho^{-0.328}$$
$$\text{if } 1 \leq R_N \leq 2 \quad a \leq 0.6$$
$$\text{if } 2 \leq R_N \leq 3 \quad a \leq 0.5$$
$$b = 1.22$$

The lower limit for the nose radius in these equations is 1 m. The nose radii of waveriders are often much smaller than 1 meter. This problem was also found by Lohar et al. (1994). The solution was to just assume a nose radius of 1 m, since only the range of the heat flux was of concern. A more detailed discussion on the nose radius will be given in Section 4.5.3. The tabluated values for Mars are listed in the last two columns of Table 4.3. The constants for Mars are given below:

$$C = 2.35 \times 10^8$$
$$a = 0.526$$
$$b = 1.19$$

Different relations exist for Venus entries, as given in Tauber et al. (2012). For velocities smaller than 10, 028 m/s, the radiative heat flux is given by:

$$q_{rad} = 9.497 \times 10^{-63} V^{18} \rho^{1.2} R_N^{0.49} \tag{4.22}$$

For velocities larger than 10, 028 m/s and smaller than 12, 000 m/s, the radiative heat flux is given by:

$$q_{rad} = 2.195 \times 10^{22} V^{7.9} \rho^{1.2} R_N^{0.49} \tag{4.23}$$

Since there is little knowledge of the gas giants, specialized equations or tables do not exist for the calculation of the radiative heat transfer. Existing work on entry in the Jovian atmosphere, given by Tauber et al. (1999), use numerical codes for the calculation of radiative heating. Although it can be expected that

Table 4.3: Radiative heating velocity functions for Earth and Mars (Tauber and Sutton, 1991)

V, m/s	$f_E(V)$	V, m/s	$f_M(V)$
9000	1.5	6000	0.2
9250	4.3	6150	1.0
9500	9.7	6300	1.95
9750	19.5	6500	3.42
10 000	35	6700	5.1
10 250	55	6900	7.1
10 500	81	7000	8.1
10 750	115	7200	10.2
11 000	151	7400	12.5
11 500	238	7600	14.8
12 000	359	7800	17.1
12 500	495	8000	19.2
13 000	660	8200	21.4
13 500	850	8400	24.1
14 000	1065	8600	26.0
14 500	1313	8800	28.9
15 000	1550	9000	32.8
15 500	1780		
16 000	2040		

an engineering level equation for the ratiative heating on the gas giants would have the form of Eq. (4.21), the corresponding constants and tabulated values remain unavailable.

However, for Jupiter, a study for an entry probe was performed by Ritter et al. (2006). A empirical relation for the radiative heating was derived from several, unspecified data sets. The radiative heat q_{rad} in W/m^2 for a Jovian entry is given by:

$$q_{rad} = 9.7632379^{-40}(2R_N)^{-0.17905}(\rho)^{1.763827469}(V)^{10.993852} \tag{4.24}$$

For Eq. (4.24), it was assumed that the Jovian atmosphere consists of 89% H_2 and 11% He. The equation was validated by Ritter et al. (2006) with in-flight measurements from the Gallileo probe.

It was not possible to obtain similar relations for radiative heating on Saturn, Uranus or Neptune. However, the atmospheric composition of these three planets is roughly similar to the atmospheric composition of Jupiter. Therefore, due to the lack of alternatives, Eq. (4.24) could also be used to estimate the radiative heat flux on Saturn, Uranus and Neptune.

4.5.3 Nose radius and thermal protection

The convective and radiative heating are both a function of the nose radius of the vehicle. For the used geometries of the waveriders, as shown in Section 4.2, the nose radius would be close to zero, since all these vehicles have a sharp leading edge. By looking at Eqs. (4.17) and (4.21), this would result in an infinitely large heat flux. To still be able to give an estimate for the heat flux, a nose radius of 1 m has been used. This is in agreement with the value of the nose radius that has been used by Lohar et al. (1994) and Casoliva et al. (2008).

Even with this theoretical nose radius of 1 m, the experienced heat loads will become very large. In Chapter 7, the heat flux constraint will be set to 500 W/cm^2 for a maneuver around Mars, and to 6000 W/cm^2 for a maneuver around Jupiter. By comparison, the largest heat flux on the Space Shuttle's nose cone during its re-entry from orbit was approximately 100 W/cm^2 (Bansal and Lamon, 2014).

Thermal protection systems are necessary to handle these high heat loads. Dijkstra (2012) mentions the same problems for an Earth entry and a vehicle with a sharp nose. There it was concluded that an active cooling system was necessary. Such a system would require liquid water and might prove to be challenging, since the temperature of a vehicle in interplanetary space is well below the freezing point of water.

A more straightforward choice for the thermal protection system would be the use of ablative material. Being a commonly used solution for thermal protection systems, ablative materials are, according to Laub (2003), capable of withstanding high heating rates and loads through phase changes and mass loss. All the planetary entry probes from NASA used this type of thermal protection. Ablative materials are often reinforced composites. When heated, the resin of the composite is transformed into gaseous products which enters the boundary layer of the flow, which reduces the convective heating. Mazzaracchio and Marchetti (2010) distinguish two types of ablative materials: charring and non-charring. Charring materials do undergo a chemical reaction where the material pyrolyses whereas non-charring materials do not.

During a hypersonic flight, the temperature increases near the vehicles surface due to the generated bow shock, as well as in the the boundary layer. Due to conduction, heat is transferred from the outside of the thermal protection system to the internal layers. After sufficient heating, the ablative material in the inner layers of the thermal protection system changes its state. The charring material decomposes and as a results in a gaseous product which leaves a porous residue called "char". The pressure that is generated pushes the gases through the pores of the char into the boundary layer, greatly reducing convective heating of the body. Non-charring ablators, in contrast to charring ablators, only erode or change phase at the surface. This type of ablators is often much more resistant to shear stresses than charring ablators, since non-charring ablators are not porous.

Table 4.4 shows an overview of existing ablative materials that are or could be used in the design of a thermal protection system. It can be concluded that all of the materials listed in Table 4.4 can withstand the heat flux of 500 W/cm^2 that is expected at Mars. For Jupiter, only the Hertiage Carbon phenolic should be able to withstand the expected heat flux of 6000 W/cm^2.

If multiple gravity assist will be performed, an ablative thermal protection system might not be the best solution. As an ablative material will burn up, extra ablative material should be added to the thermal protection system to be able to execute any subsequent aerogravity assist. A better solution might be to make use of a reusable system, such as the thermal protection system of the Space Shuttle.

Even though the thermal protection system might be one of the most crucial elements to successfully perform an aerogravity assist, its design is beyond the scope of the performed research in this thesis. It is expected, given the properties of the materials listed in Table 4.4, that a thermal protection system can be designed that makes an aerogravity assist possible.

Table 4.4: Ablative TPS materials (Venkatapathy et al., 2010)

Material	**Flight qual/TRL**	**Heat flux** W/cm^2	**Pressure** atm
PICA	Stardust	$\sim$1200	$<$1
Avcoat	Apollo	$\sim$1000	$<$1
ACC	Genesis	$>$2000	$>$1
BPA	TRL 3-4	$\sim$1000	$\sim$ 1
PhenCarb Family	TRL 5-6	1000-4000	$>$1
3DQP	TRL 4	$\sim$5000	$>$1
Heritage Carbon phenolic	Venus, Jupiter	10,000-30,000	$\gg$1

5
Numerical Methods

This chapter will discuss a number of methods that was used in the aerodynamic database generation, simulator, trajectory optimization and guidance algorithm. First of all, Section 5.1 will discuss the numerical integration method used to propagate the equations of motion, and a trade-off has been made between different types of integration schemes. Next, in Section 5.2, the numerical interpolation scheme used for the trajectory optimization will be presented. Section 5.3 discusses the numerical differentiation scheme used to determine derivatives, which is required by the guidance algorithm. Finally, Section 5.4 gives a short summary of the numerical optimization algorithm used, and an analysis on the population size for this algorithm is performed.

5.1 Numerical integration

Numerical integration is necessary to propagate a given initial state to a later time step using a set of differential equations. In the simulation of an aerogravity assist, the numerical integrator propagates the initial state using the equations of motion presented in Section 2.2. In this section, the basics of the numerical integration methods considered are discussed. In addition, a trade-off between different integration methods is presented.

Based on Bayen and Siauw (2015), the general form of an ordinary differential equation can be given by:

$$\frac{d^n x}{dt^n} = f\left(t, \frac{dx}{dt}, \frac{d^2x}{dt^2}, \frac{d^3x}{dt^3}, \dots, \frac{d^{(n-1)}x}{dt^{(n-1)}}\right) \tag{5.1}$$

As could be seen in Section 2.2, the equations of motion based on Newton's second law have the form $\ddot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x})$, which follows the form of an ordinary differential equation as given by Eq. (5.1). A basic method of solving a set of ordinary differential equations is a fixed step-size method such as an Euler method, which is a method that uses a single function evaluation to estimate the state at the next time-step. Higher order methods that use more function evaluations, such as Runge-Kutta 4, are more accurate, but the larger number of function evaluations increases the required computational effort.

The error made during each time-step, depends on the magnitude of the used step-size and the order of the integration method. Small step sizes require a large number of steps, in turn increasing the required functions evaluations and hence the computation time. A higher order method performs more function evaluations per step. Furthermore, with a uniform grid, which is the result of a fixed step-size, the error made during the integration varies per step as the error depends on the function and its derivatives.

To reduce the computational effort, an adaptive step-size method is used. The idea behind such a method is that two different integration methods are used, for example *M1* and *M2*. Assuming that *M2* is a method with a higher order than *M1*, it is considered that the solution found using *M2* is closer to the exact solution than the solution found with *M1*. So when evaluating the same time-step using the two methods, the difference between the two methods could be considered as a measure for the error made with the lower-order method. If the error is larger than the allowed tolerance, the step size should be decreased and it should be increased if the error is smaller than the allowed tolerance. This is the working principle behind variable step-size integration methods.

In the TU Delft Astrodynamics Toolbox (Tudat), several variable step-size methods are available, namely Runge-Kutta-Fehlberg 4(5), Runge-Kutta-Fehlberg 5(6), Runge-Kutta-Fehlberg 7(8) and Runge-Kutta 8(7) Dormand-Prince. For convenience, these four methods will be abbreviated by RKF45, RKF56, RKF78 and RK87DP. Since these methods all have a different order and therefore a varying number of function evaluations per step, it is necessary to perform a trade-off between these four methods as the outcome might be problem specific.

The different configurations of the numerical integrators will be compared with a baseline solution. Ideally, this baseline solution would correspond to an analytical solution, such that only the performance of the integrator will be estimated. Unfortunately, no analytical solution is available. Instead, as a baseline, a trajectory generated using the RKF78 with a tolerance of $O(10^{-15})$ will be selected. By using a small tolerance value, it is assumed that this solution will be closer to an actual solution than a solution obtained with a larger tolerance value.

The guidance algorithm (see Chapter 9) will require a constant step-size. This requires the numerical integrator to have a step end at each guidance interval. As an example, consider that a guidance update occurs every 1.0 s. This means that if the variable step-size integrator proposes a step-size that exceeds the interval of 1.0 s, the step-size will be decreased such that the guidance update can still be provided at the guidance interval. By decreasing the step-size in this way, a higher accuracy is obtained.

The results of the trade-off between different numerical integrators is shown in Table 5.1. In this table, the maximum error in altitude with respect to the baseline trajectory is shown as a function of the number of function evaluations required for a specific orbit. The effect of enforcing a fixed guidance interval is visible in this table. For both RKF56 and RKF78, even though the tolerance is increased from 10^{-10} to 10^{-11}, the number of function evaluations is identical. At these tolerances, the enforced guidance interval is controlling the step-size. At higher tolerances (10^{-12} to 10^{-14}), the guidance interval is not always sufficient to achieve the specified tolerances, so step-sizes smaller than the guidance interval are also used. This increases the number of steps required and hence the number of function evaluations.

For the RKF56 integrator, even though the number of function evaluations is smaller than for RKF78 and RK87DP, the maximum error in altitude is approximately 50 m. At lower tolerances, the number of function evaluations for RKF78 and RK87DP are approximately equal, but the errors are smaller than for RKF56. At a tolerance of 10^{-14}, the number of function evaluations increases tremendously to a total of 138541 evaluations. The error for this tolerance exceeds 350 m, and is expected that this the result of the round-off errors involved in the large number of evaluations. Even though this tolerance is not taken into account, the errors that occur using a RK87DP are still larger than RKF78 while it is expected that these two integrators have similar performance. This problem was also addressed by Ronse (2013), but no conclusive explanation could be given. RK87DP will therefore not be used.

Table 5.1: Comparison of the performance of RKF56, RKF78 and RK87DP

Method	Tolerance	Evaluations	Error [m]
RKF56	10^{-10}	19 107	50.697
	10^{-11}	19 107	50.697
	10^{-12}	19 134	50.509
	10^{-13}	19 125	50.550
	10^{-14}	20 052	50.482
RKF78	10^{-10}	27 599	0.305 75
	10^{-11}	27 599	0.305 75
	10^{-12}	27 599	0.305 75
	10^{-13}	27 599	0.305 75
	10^{-14}	27 755	0.305 67
RK87DP	10^{-10}	27 599	17.239
	10^{-11}	27 664	16.398
	10^{-12}	28 132	9.6924
	10^{-13}	35 750	9.4376
	10^{-14}	138 541	352.92

The performance of RKF45 was also assessed, but is not shown in Table 5.1. The errors for this integrator were in the order of 10^4 times as large as the errors obtained with the three other integrators. It can therefore be concluded that Runge-Kutta-Fehlberg 4(5) is not a suitable integration method.

When only considering RKF56 and RKF78, higher tolerances does not lead to an increase in accuracy as the accuracy is mainly dominated by the size of the guidance interval. As an error of 50 m for the RKF56 integrator is considered to be quite large, especially compared to the 0.3 m error of the RKF78 integrator, the RKF78 integrator with a tolerance of 10^{-10} is used in the analysis.

5.2 Interpolation

Assuming one has a finite set of data points, a method of obtaining intermediate values is by fitting a curve through the data points. Many different implementations of this method, also called interpolation, are available, each with their own strengths and weaknesses. Common interpolation methods such linear interpolation and cubic spline interpolation are already incorporated in the TU Delft Astrodynamics Toolbox (Tudat) and will therefore not be discussed in this section.

For the trajectory optimization, as will be shown in Section 7.1, a finite number of guidance nodes for the bank angle, angle of attack or both are defined. This means that an interpolation technique should be used to obtain intermediate values. Cubic splines, a method already included in Tudat, would provide a function that is continuous in the first and second derivatives, something that is not the case for linear interpolation. However, cubic splines can result in overshoots (see also Fig. 5.1), which is undesirable.

An interpolator that does not have this behavior, but is smooth in the first derivative is a Hermite spline, which is based on piecewise cubic polynominals. Consider the ith interval of length h_i, such that $h_i = x_{i+1} - x_i$. Then, the first divided difference δ_i is given by:

$$\delta_i = \frac{y_{i+1} - y_i}{h_i}$$

Furthermore, the slope at x_i is equal to $d_i = P'(x_i)$, where $P(x)$ is the interpolant on the interval $x_i \leqslant x \leq x_{i+1}$. This interpolant $P(x)$ for a Hermite spline can be expressed using local variables $s = x - x_i$ and $h = h_i$ using:

$$P(x) = \frac{3h^2 - 2s^3}{h^3} y_{i+1} + \frac{h^3 - 3hs^2 + 2s^3}{h^3} y_i + \frac{s^2(s-h)}{h^2} d_{i+1} + \frac{s(s-h)^2}{h^2} d_i \tag{5.2}$$

Eq. (5.2) is a cubic polynomial in x that satisfies for interpolation conditions: two for each boundary of an interval and two on the derivative values:

$$\begin{aligned} P(x_i) &= y_i\,, & P(x_{i+1}) &= y_{i+1} \\ P'(x_i) &= d_i\,, & P'(x_{i+1}) &= d_{i+1} \end{aligned}$$

The boundary values of an interval are obviously obtained from the data set that is interpolated. However, the derivatives on the boundaries of an interval are not necessarily known. These derivatives are selected such that the function does not locally overshoot the data values. If the divided differences δ_i and δ_{i-1} have an opposing sign or one of them is equal to zero, then the point x_i is a local minimum. For this reason, the derivative at such a point is set to zero, or $d_i = 0$.

If the divided differences δ_i and δ_{i-1} have the same sign, then the derivative d_i is determined by a weighted harmonic mean, as given by:

$$\frac{w_1 + w_2}{d_i} = \frac{w_1}{\delta_{i-1}} + \frac{w_2}{\delta_i} \tag{5.3}$$

in which

$$w_1 = 2h_i + h_{i-1}, \quad w_2 = h_i + 2h_{i-1}$$

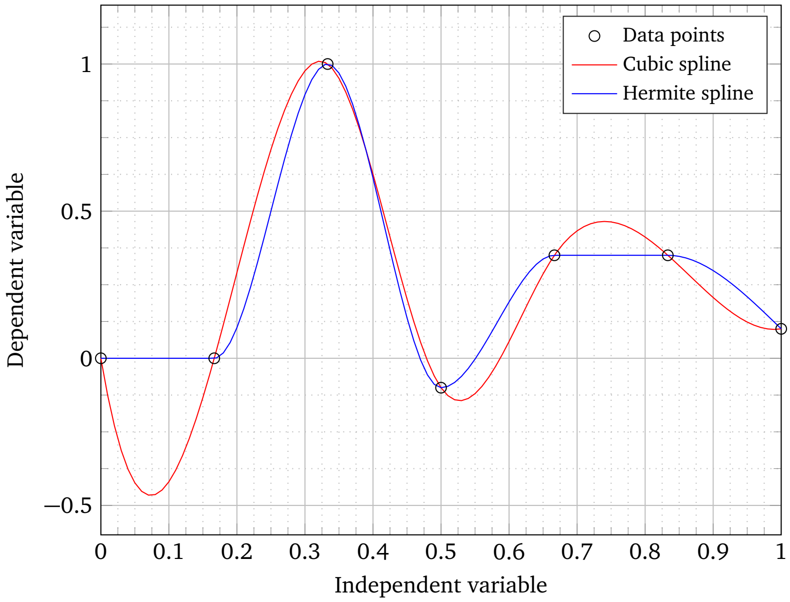


Figure 5.1: Comparison of a Hermite spline and a cubic spline

So far, the slope of the polynomial is determined for interior data points. For the edge of the data interval, one-sided, shape-preserving, three-point formula is required. Consider the first two intervals h_1 and h_2 as well as the divided differences δ_1 and δ_2. Then, the following initial estimate is used for the derivative $d_{1,est}$:

$$d_{1,est} = \frac{(2h_1 + h_2)\delta_1 - h_1\delta_2}{h_1 + h_2} \tag{5.4}$$

If the sign of $d_{1,est}$ is different from or equal to the sign of δ_1, then the actual derivative $d_1 = 0$. In case the absolute value of $d_{1,est}$ is three times larger than the value of δ_1, *and* δ_1 and δ_2 have opposite signs, then $d_1 = 3\delta_1$. If neither are true, then $d_1 = d_{1,est}$. The same approach holds for the the derivative at the other edge of the data interval.

In Fig. 5.1, a comparison between a Hermite spline and a cubic spline is shown. From Fig. 5.1, the benefit of a Hermite spline becomes clear. For two data points at an equal y-value, the Hermite spline will produce an straight line whereas a cubic spline can exceed these values at intermediate values of the independent variable, as can be seen for the first two data points in Fig. 5.1. This behavior of a Hermite spline is more desirable for the interpolation of the guidance nodes, which is the reason this interpolation technique was selected.

5.3 Numerical differentiation

The guidance algorithm, which will be presented in Chapter 9, requires first- and second-order derivatives of a number of variables. Since no analytical expressions are available for these variables, one has to resort to numerical differentiation techniques. These techniques are often based on finite differences

In general, three different finite difference schemes are available: central differencing, forward differencing and backward differencing. Assume one wants to obtain a derivative at $x = x_i$. A central differencing scheme uses data points at $x < x_i$ as well as at $x > x_i$ to estimate the derivative, while a forward differencing scheme only uses data points at $x \geq x_i$ and a backwards differencing scheme only uses data points at $x \leq x_i$. A visual example of these three schemes is shown in Fig. 5.2.

Since central differencing uses both data points before and after the point of interest, this method can provide a more accurate approximation of the numerical derivative and is therefore the preferred approach. To obtain the derivative of the drag reference trajectory, it will be possible to use central differencing since all the data points are known in advance. However, for the guidance algorithm, only the historic values of variables such as the drag coefficient and the drag acceleration are known, which makes central differencing

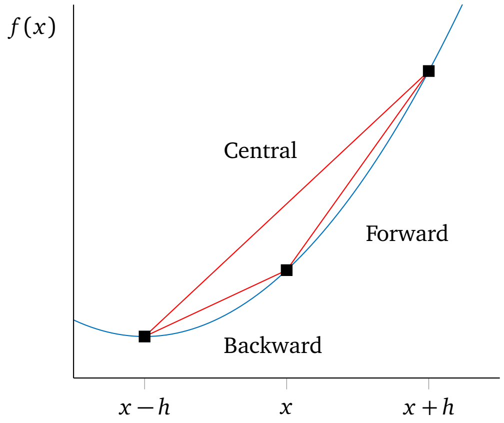


Figure 5.2: Forward-, backward- and central-difference approximations Wilmott et al. (1995)

unsuitable. Instead, backward differencing will be used to obtain the derivatives of variables in the guidance algorithm.

Chapra and Canale (2009) provides the equations to approximate a numerical derivative using central- and backward differencing. Assuming a function $f(x)$ and a step size h, an approximation for the derivative of $f(x)$ using central differencing is given by:

$$f'(x) \approx \frac{f(x+h)-f(x-h)}{2h} + O(h^2) \tag{5.5}$$

Similarly, an approximation for the derivative of $f(x)$ using backward differencing is given by:

$$f'(x) \approx \frac{f(x)-f(x-h)}{h} + O(h) \tag{5.6}$$

Note that Eq. (5.6) has an error proportional to h, which is one order less accurate than the central differencing scheme. It is possible to obtain the same order of error by adding one additional function evaluation (Chapra and Canale, 2009):

$$f'(x) \approx \frac{3f(x)-4f(x-h)+f(x-2h)}{2h} + O(h^2) \tag{5.7}$$

In addition, for some parameters, the second derivative is required. Instead of using Eq. (5.6) twice, a different backwards difference scheme can be formulated to estimate the second derivative:

$$f''(x) \approx \frac{f(x)-2f(x-h)+f(x-2h)}{h^2} + O(h) \tag{5.8}$$

Note that Eq. (5.8) has also an accuracy which is one order less than Eq. (5.6). However, by adding one additional function evaluation, a higher accuracy can be obtained:

$$f''(x) \approx \frac{f(x)-5f(x-h)+4f(x-2h)-f(x-3h)}{h^2} + O(h^2) \tag{5.9}$$

The TU Delft Astrodynamics Toolbox (Tudat) contains a software routine for central differencing. It is therefore necessary to develop numerical derivative approximation using backward differencing. Backward differencing is only possible if there is already a number of data points is available. Until sufficient data points are available to evaluate Eqs. (5.7) and (5.9), Eqs. (5.6) and (5.8) will be used to provide an initial estimate of the first and second derivative, respectively.

5.4 Numerical optimization

As will be explained Chapter 7, the reference trajectory is obtained by defining a number of guidance nodes for the bank angle, of the angle of attack or for both. These guidance nodes are interpolated using the interpolation method discussed in Section 5.2, such that a continuous function is obtained for the bank angle and/or angle of attack history. However, a random set of guidance nodes will certainly not result in a bank angle and/or angle of attack history that is optimal. Therefore, numerical optimization is used find a set of guidance nodes that will provide a bank-angle and/or angle-of-attack history that achieves the optimum.

The numerical optimization tools that are commonly used at the Astrodynamics and Space Missions department is the Parallel Global Multiobjective Optimizer (PaGMO), which is a toolbox developed by ESA (Biscani et al., 2010). It is equipped with many single- and multi-objective optimization algorithms, and it is capable of solving the following general optimization problem

$$\begin{aligned} \text{find:} \quad & \boldsymbol{x} \in \boldsymbol{\Omega} \\ \text{to minimize:} \quad & \boldsymbol{f}(\boldsymbol{x}) \\ \text{subject to:} \quad & \mathbf{lb} \leq \boldsymbol{x} \leq \mathbf{ub} \\ & \boldsymbol{c}(\boldsymbol{x}) = \mathbf{0} \\ & \boldsymbol{c}_{in}(\boldsymbol{x}) \leq \mathbf{0} \end{aligned}$$

in which $\boldsymbol{x}$ is the decision vector, $\boldsymbol{\Omega}$ represents the decision variable space, $\boldsymbol{f}(\boldsymbol{x})$ is the objective function, **lb** and **ub** are the lower and upper bound respectively and $\boldsymbol{c}(\boldsymbol{x})$ represents the constraints. Since the objective of the optimization of the guidance nodes is to achieve a trajectory that maximizes the bending angle as well as to achieve a predetermined inclination, a multi-objective algorithm is needed.

5.4.1 Algorithm description

A recently developed multi-objective optimization algorithm that is also implemented in PaGMO is the MOEA/D algorithm, an abbreviation for Multi-Objective Evolutionary Algorithm based on Decomposition (Zhang and Li, 2007). It divides a multi-objective problem into many single objective optimization subproblems in such a way that the solution for each of the subproblems is a part of the optimal pareto front of the multi-objective problem.

The first step of the MOEA/D algorithm is to divide the multi-objective problem into many single-objective problems. This process is called decomposition. Three different approaches of decomposing the problem were proposed by Zhang and Li (2007), namely the Weighted Sum approach, the Tchebycheff approach and the Boundary Intersection approach. Since the MOEA/D implementation in PaGMO makes use of the Tchebycheff approach, only this approach will be discussed in this report.

Consider a weight vector $\boldsymbol{\lambda} = [\lambda_1, \dots \lambda_m]^T$, which is used to weigh each of the m objectives of a multi-objective problem. Furthermore, for $\boldsymbol{\lambda}$ holds that each $\lambda_i \geq 0$ and $\sum_{i=1}^{m} \lambda_i = 1$. Then, using the weight vector $\boldsymbol{\lambda}$, the scalar optimization problem form according to the Tchebycheff approach is defined by:

$$\text{minimize} \quad \boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x})|\boldsymbol{\lambda}, \boldsymbol{z}^*) = \max_{1 \leq i \leq m} \left\{ \lambda_i |f_i(\boldsymbol{x}) - z_i^*| \right\} \qquad \text{subject to } \boldsymbol{x} \in \boldsymbol{\Omega} \tag{5.10}$$

in which $\boldsymbol{z}^* = [z_1, \dots, z_m]^T$ is a reference point defined as $z_i^* = \min \left\{ \boldsymbol{f}_i(\boldsymbol{x}) | \boldsymbol{x} \in \Omega \right\}$ for each $i = 1, \dots, m$. According to Zhang and Li (2007), for each Pareto optimal point $\boldsymbol{x}^*$ there exists a weight vector $\boldsymbol{\lambda}$ such that $\boldsymbol{x}^*$ is the optimal solution of Eq. (5.10). Furthermore, each optimal solution of Eq. (5.10) is a Pareto optimal solution of the multi-objective problem.

The MOEA/D implementation in PaGMO has three different approaches of generating the weight vectors $\boldsymbol{\lambda}$. These three methods are described by Mambrini and Izzo (2014):

- **GRID**: The weight vectors are generated to optimally maximize their spreads. However, the number of weight vectors that can be generated is constrained, which yields that the population size cannot be selected freely.

- **RANDOM**: The weight vectors are generated randomly. Although for this method the amount of weight vectors that cen be selected is free, the spread of the weight vectors is not as optimal as the GRID method.

- **LOW-DISCREPANCY**: This method can generate any amount of weight vectors and ensures a good spread of these vectors. It makes use of a Halton sequence to generate the weight vectors (see Halton (1964)).

Given a set of of weight vectors $\boldsymbol{\lambda}^1,\dots,\boldsymbol{\lambda}^N$ and a reference point $\boldsymbol{z}^*$, the multi-objective optimization problem can be decomposed into N single-objective or scalar optimization problems. Using the Tchebycheff approach, the objective function of the jth subproblem is given by Eq. (5.11).

$$\boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x})|\boldsymbol{\lambda}^j,\boldsymbol{z}^*) = \max_{1\le i\le m}\left\{\lambda_i^j|f_i(\boldsymbol{x})-z_i^*|\right\} \tag{5.11}$$

The MOEA/D algorithm optimize these N objective functions in parallel. If the $\boldsymbol{\lambda}$ vectors used to obtain two objective functions, lets say $\boldsymbol{\lambda}^i$ and $\boldsymbol{\lambda}^j$ are relatively close, it follows that that the optimal solutions of $\boldsymbol{g}(\boldsymbol{x}|\boldsymbol{\lambda}^i,\boldsymbol{z}^*)$ and $\boldsymbol{g}(\boldsymbol{x}|\boldsymbol{\lambda}^j,\boldsymbol{z}^*)$ are close together as well. As a result, any information about $\boldsymbol{g}$ with a weight vector close to $\boldsymbol{\lambda}_i$ can be usefull for optimizing $\boldsymbol{g}(\boldsymbol{x}|\boldsymbol{\lambda}^i,\boldsymbol{z}^*)$, which is one of the main motivations behind MOEA/D (Zhang and Li, 2007). The weight vectors used to optimize $\boldsymbol{g}(\boldsymbol{x}|\boldsymbol{\lambda}^i,\boldsymbol{z}^*)$, are determined by the neighborhood, which is a set of $\boldsymbol{\lambda}$'s that is closest to $\boldsymbol{\lambda}^i$.

To initialize the MOEA/D algorithm, a population size N is selected, consisting of $\left\{\boldsymbol{x}^1,\dots,\boldsymbol{x}^N\right\}\in\Omega$. Furthermore, each of the individuals in the population is used to evaluate the objective function $\boldsymbol{f}(\boldsymbol{x})$ such that $\boldsymbol{FV}^i=\boldsymbol{f}(\boldsymbol{x}^i)$. Next, a vector $\boldsymbol{z}=[z_1,\dots,z_N]^T$ is defined where z_i is the best value found for objective f_i. Finally, an external population $\boldsymbol{EP}$ is defined that is used to store non-dominated solutions. Then, the algorithm performs the steps shown in Fig. 5.3. The problem-specific methods and heuristics that Zhang and Li (2007) refers to in Fig. 5.3 are based on a Differential Evolution algorithm in its implementation in PaGMO.

```
1. Initialization
   1.1. Set $\boldsymbol{EP}=\emptyset$.
   1.2. Compute the Euclidean distances between any two weight vectors and then work out the $T$ closest weight vectors to each weight vector. For each $i=1,\dots,N$, set $B(i)=\{i_1\dots,i_T\}$, where $\boldsymbol{\lambda}^{i_1},\dots,\boldsymbol{\lambda}^{i_T}$ are the $T$ closest weight vectors to $\boldsymbol{\lambda}^i$.
   1.3. Generate an initial population $\boldsymbol{x}^1,\dots,\boldsymbol{x}^N$ randomly or by a problem-specific method. set $\boldsymbol{FV}^i=\boldsymbol{f}(\boldsymbol{x}^i)$.
   1.4. Initialize $\boldsymbol{z}=[z_1,\dots,z_m]^T$ by a problem-specific method.
2. Update
        For $i=1,\dots,N$, do
   2.1. Reproduction: Randomly select two indexes $k$,$l$ from $B(i)$, and then generate a new solution $y$ from $\boldsymbol{x}^k$ and $\boldsymbol{x}^l$ using genetic operators.
   2.2. Improvement: Apply a problem-specific repair/improvement heuristic on $\boldsymbol{y}$ to produce $\boldsymbol{y}'$.
   2.3. Update of $\boldsymbol{z}$ For each $j=1,\dots,m$ if $z_j<f_j(\boldsymbol{y}')$, then set $z_j=f_j(\boldsymbol{y}')$.
   2.4. Update of Neighboring Solutions: For each index $j\in B(i)$, if $g(\boldsymbol{y}'|\lambda^j,\boldsymbol{z})\le g(\boldsymbol{x}^j|\lambda^j,\boldsymbol{z}$, then set $\boldsymbol{x}^j=\boldsymbol{y}'$ and $FV^j=f(\boldsymbol{y}')$.
   2.5. Update of $\boldsymbol{EP}$: Remove from $\boldsymbol{EP}$ all the vectors dominated by $\boldsymbol{f}(\boldsymbol{y}')$ and add $\boldsymbol{f}(\boldsymbol{y}')$ to $\boldsymbol{EP}$ if no vectors in $\boldsymbol{EP}$ dominate $\boldsymbol{f}(\boldsymbol{y}')$.
3. Stopping Criteria: If stopping criteria is satisfied, then stop and output $\boldsymbol{EP}$. Otherwise, go to Step 2.
```

Figure 5.3: MOEA/D algorithm pseudo code (Zhang and Li, 2007)

5.4.2 Parameter selection

The Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) implementation in PaGMO has a total number of seven parameters that can be adjusted to change the performance of the algorithm. Ideally, these parameters should be selected such that the optimization converges to the global optimum with a minimum number of function evaluations to reduce the computational effort. In other words, the effect of the different algorithm parameters on the following properties should be evaluated.

Two parameters that are related to the Differential Evolution algorithm are the crossover probability C_r and the mutation factor F. The mutation factor determines how much an individual of a population is influenced by other individuals of the population while the crossover probability determines the chance of this mutation actually occurring. Small values of the mutation factor reduces the exploration of the search space and can result in premature convergence. If $C_r = 1$, the individuals of a population will be close to each other since all individuals will inherit the properties of the other individuals in the population.

Values for F and C_r were found using the following approach. While leaving all optimization variables at their default values, the values for F and C_r were varied independently. The optimization for each combination of parameters was repeated three times using three different random seeds in an attempt to eliminate the random effect. Decent results were found for $C_r = 0.5$ and $F = 1.0$.

The population size N_p and the neighborhood size N_b are also two important parameters of the optimization algorithm. Increasing the neighborhood size increases the search space, but also increases the computational effort. According to Liu et al. (2010) and Ishibuchi et al. (2013), the neighborhood size should be between 2% and 10% of the population size. However, this often resulted in premature convergence. Instead, this percentage was increased to 20%, although this did not avoid the problem of premature convergence completely.

The population size directly influences the computational effort since it relates to the number of times the objective function should be evaluated. To determine the required population size, the optimization was executed with populations that contained 40, 60, 80, 100, 120, 140 and 160 individuals. As will be described in Section 7.1, the two objectives that should be optimized are related to the offset in inclination and the atmospheric bending angle. The offset in inclination should be minimized to guarantee that the vehicle remains in the same orbital plane. The bending angle can either be maximized or be optimized to achieve a certain target bending angle. The number of guidance nodes, which are part of the decision variable as discussed in Section 7.1, also influence the optimization. For the population size selection, six guidance nodes were used. In addition, two more optimization parameters were selected: the initial velocity and the initial flight-path angle, resulting in a global dimension of eight for the optimization problem.

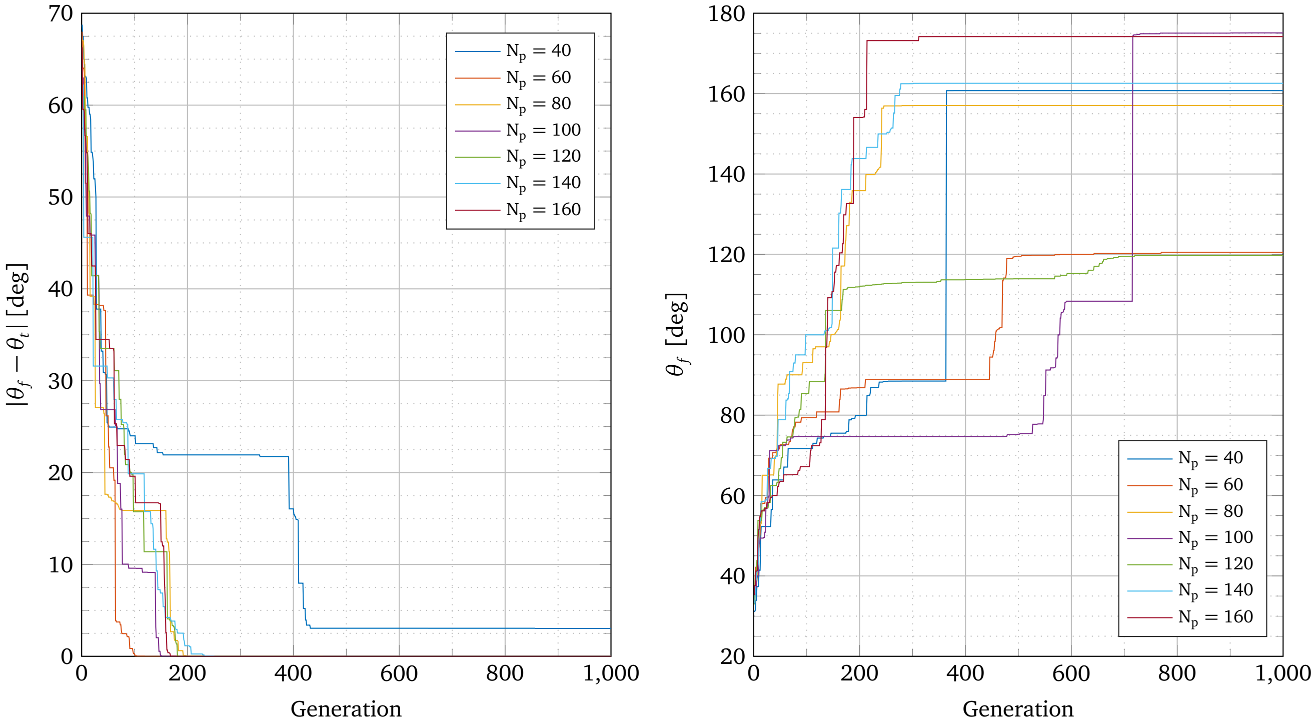
Fig. 5.4a shows the offset in actual bending angle from the target bending angle $|\theta_f - \theta_t|$ as a function of the number of generations for different population sizes. The second fitness value, the offset of the final inclination from the target inclination, is not shown but it was checked that this value would not become too large (e.g. $> 3°$) such that the departing segment of the trajectory would be more or less in the same orbital plane as the arriving segment.

From Fig. 5.4a it can be observed that for a population consisting of 40 individuals, there is a chance that the optimization will not converge to a solution. As a result, the population size should not be too small to avoid premature convergence. For populations consisting of 60 individuals or more, the optimization does converge to an solution where $|\theta_f - \theta_t|$ is approaching zero. For a population consisting of 60 individuals, the optimization converges the quickest to an optimum after approximately 100 generations while a population consisting of 100 individuals will converge after 150 generations and a population consisting of 80, 120 and 140 individuals will converge after approximately 200 generations.

In addition to achieving a target bending angle, an optimization was performed for the case where the final bending angle θ_f was unconstrained. In other words, the largest possible bending angle needed to be found. These results are shown in Fig. 5.4b. The optimization runs with a population consisting of 100 and 160 individuals achieve a bending angle of approximately 175°, while the simulation runs with a population size of 40,80 and 140 obtain a final bending of approximately 160°. The runs with a population size of 60 and 120 obtain approximately 120°.

Several conclusions can be drawn from the result in Figs. 5.4a and 5.4b. First of all, even though multiple runs with different random seeds were performed with the same populations, the optimization still can converge to a local optimum. Furthermore, although one could expect that a larger population size would result in faster convergence, this is not the case. Again, it is expected that this is the result of the random effect in the simulation.

Since the aim of trading-off the different population sizes is to find a decent performance with relatively low computational effort, it has been decided to use a population size of 100 individuals. In Fig. 5.4b, this population size resulted in the maximum value of the atmospheric bending angle and converged properly to the target value as shown in Fig. 5.4a. To eliminate the random effect, multiple optimization runs will be performed where each run will have a different random seed. This will reduce the probability of finding a less optimal solution due to the random effect.



(a) Effect of population size on convergence to optimum

(b) Effect of population size on convergence to maximum bending angle

Figure 5.4: Effect of population size on optimization

6 Software Design and Verification

The theory explained in the previous chapters was combined into a simulator capable of simulating aerogravity assist trajectories. The first section of this chapter will explain the major components of the simulation software including the external libraries. Next, the interface between the different components work together will be discussed. Finally, the used verification methods are discussed and shown in the final section of this chapter.

6.1 Software components

To find the trajectories of an aerogravity assist, several different software libraries and packages were used. By using existing software libraries, the development time could be reduced. The list of packages is shown below:

- **Tudat**[1], abbreviation for the TU Delft Astrodynamics Library, is a set of C++ libraries and functions containing many different software routines useful for astrodynamics applications, such as reference-frame transformations, numerical integrators and interpolators. The library is completely developed by staff and students from Delft University of Technology.
- **SPARTA**[2] is the external software package used for the calculation of the aerodynamic coefficients in the rarefied flow regime. The output of SPARTA consists of an ASCII file containing the normal pressure and shear-stress components per face of the vehicle shape.
- The **aerodynamic database generator** consists of three major components. The first component consists of reading the output from SPARTA. The normal pressure and shear stress components from the SPARTA output file are integrated and rotated to the correct frame to obtain the aerodynamic lift and drag coefficients. The second component consists of using the modified Newtonian method as implemented by Ronse (2013) to determine the aerodynamic coefficients for the continuum flow regime. Finally, using `MATLAB`, the aerodynamic coefficients for both flow regimes are stored in two separate data files, which are of the same format. The coefficients in the transition flow regime are a weighted sum of the coefficients in the continuum flow regime and rarefied flow regime and are therefore determined within the simulator itself.
- **SPICE**[3], a software package developed by the Jet Propulsion Laboratory, contains routines to retrieve the ephemeris and properties of planetary bodies, as well as the possibility of retrieving trajectories of existing spacecraft. Tudat has an interface to communicate with SPICE
- **GRAM**[4], the Global Reference Atmospheric Model is an atmosphere model written in FORTRAN for both Mars and Venus developed by the National Aeronautics and Space Administration. Since it was written in FORTRAN, a wrapper had to be written to ensure these models could be incorporated in the simulator which is written in C++.

[1] Tudat, *http://tudat.tudelft.nl/*, accessed 10 December 2015

[2] SPARTA, *http://sparta.sandia.gov/*, accessed 25 January 2016

[3] SPICE, *https://naif.jpl.nasa.gov/naif/toolkit.html*, accessed 10 December 2015

[4] GRAM, *https://software.nasa.gov/*, accessed 25 January 2016

- **NRLMSISE**[5], an Earth atmosphere model written in C developed by the Naval Research Laboratory. Since this model was written in C, a wrapper had to be written to ensure this model could be incorporated in the simulator, which is written in C++.
- **Simulator**, where the equations of motion, environmental models and aerodynamic databases are combined and used to numerically propagate the initial state. Due to its many components, it is not possible to explain the design of the simulator briefly. Therefore, Section 6.2 is dedicated to the design of the simulator.
- **PaGMO**[6], an optimization toolbox developed by ESA's Advanced Concepts Team. It is a very extensive library containing over 50 different optimization algorithms. The MOEA/D algorithm, used for the trajectory planning, is also part of PaGMO.
- **Libconfig**[7], an external library to interpret configuration files. These files contain input variables such as the initial conditions, integration method and the used models.

6.2 Simulator design

As was explained in Section 6.1, the simulator is quite extensive and consists of a large number different elements. In Fig. 6.1, the main elements of the simulator are shown.

- **Initialize simulation**. The configuration file is loaded and read using the Libconfig library and the required models are loaded. The models that are loaded are the gravity model, atmosphere model, ephemeris, rotational ephemeris and the aerodynamic databases of the vehicle. In addition, the state derivative model (Tudat) containing the equations of motion is initialized such that the numerical integrator can propagate the initial state.
- **Initialize integrator**. In the input file, the desired integrator model can be selected, as well as the desired tolerances, integration start and end time, guidance interval and the data writing interval. Using these values, the numerical integrator, obtained from Tudat, is initialized.
- **Get control variable**. Based on the choice made in the configuration file, two different methods of obtaining the control variable can be selected: using the node-control algorithm explained in Section 7.1, or the tracking algorithm in Chapter 9. If the former is selected, a guidance matrix should be provided.
- **Propagate state**. Here, the actual numerical integration is performed. Since the numerical integration methods supported are all adaptive step-size methods, no step-size needs to be provided. However, the guidance interval, which is the interval at which the control algorithm should provide an updated value, and the data writing interval should be provided. The step-sizes determined by the numerical integration algorithm are reduced when necessary to ensure an integration step is performed at each guidance- and data writing interval.
- **Get flight parameters**. Once the integration step is performed, a new state is obtained in Cartesian coordinates. Using these Cartesian coordinates, variables such as the longitude, latitude, heading angle and flight-path angle are calculated, as shown in Section 2.3.
- **Store current state**. The current Cartesian state, as well as the flight parameters that are computed, are stored in a container, such that they can be written to file once the integration is completed.
- **Update control variable**. The control variables are updated to a new value by the guidance algorithm.
- **Write output**. At constant intervals, which do not have to match the step sizes used by the numerical integrator, the state is be stored. At the end of this simulation, the stored data points are written to an ASCII file. A `MATLAB`-script has been written and is capable of reading the output file. It converts the columns of the output file to vectors with a variable name corresponding to the value they represent.

[5] NRLMSISE, *http://www.brodo.de/space/nrlmsise/*, accessed 3 March 2015

[6] PaGMO, *https://esa.github.io/pagmo/*, accessed 25 January 2016

[7] C/C++ Configuration File Library: *http://www.hyperrealm.com/libconfig/*, accessed: 10 December 2015

Several decisions can be found in Fig. 6.1. The first decision that is made is whether the current time step coincides with the data writing interval. If this is the case, the current state will be stored such that it can be written to a data file later. The second decision that is made is related to the guidance interval, which is the interval at which the guidance algorithm can provide a new value for the control variables. It might be necessary that the numerical integrator requires time steps smaller than this guidance interval. This decision ensures that the control variable remains constant during each guidance interval. The final decision made determines when to stop the simulation. Three stop conditions are incorporated: stopping at the final integration time, stopping when the altitude becomes smaller than zero and stopping when the altitude becomes larger than the starting altitude. This latter condition is implemented to avoid unnecessary integration steps outside of the atmosphere, since there the orbit can very well be described by a Kepler orbit. Both the output interval and the guidance interval, as well as the stop conditions are parameters in the configuration file, as is shown in Appendix C.

Figure 6.1 does not provide a clear connection between the different elements of the software, and how different variables flow from one software unit to another. This is, however, shown in the architectural design in Fig. 6.2. Three major units can be identified that correspond to a Guidance, Navigation and Control (GNC) system. As it was not part of the thesis work, the navigation and control system were not implemented, but are shown in Fig. 6.2 for the sake of completeness.

The starting point of Fig. 6.2 is the reference trajectory. This trajectory is fed to the guidance algorithm, and is used by this algorithm to select the control variables such that the actual trajectory matches the reference trajectory as closely as possible. Once the guidance algorithm has determined the control variables, the current state is used to determine the flight conditions, such as the longitude, altitude and relative velocity. These flight conditions are used the obtain the correct environmental parameters from the atmosphere and gravity models, and to determine the aerodynamic coefficients. This will be used by the flight dynamics units, which contains the equations of motion. Finally, two constraints are calculated: the heat flux and the g-load. These constraints are used by the optimization algorithm to determine if trajectories are physically possible.

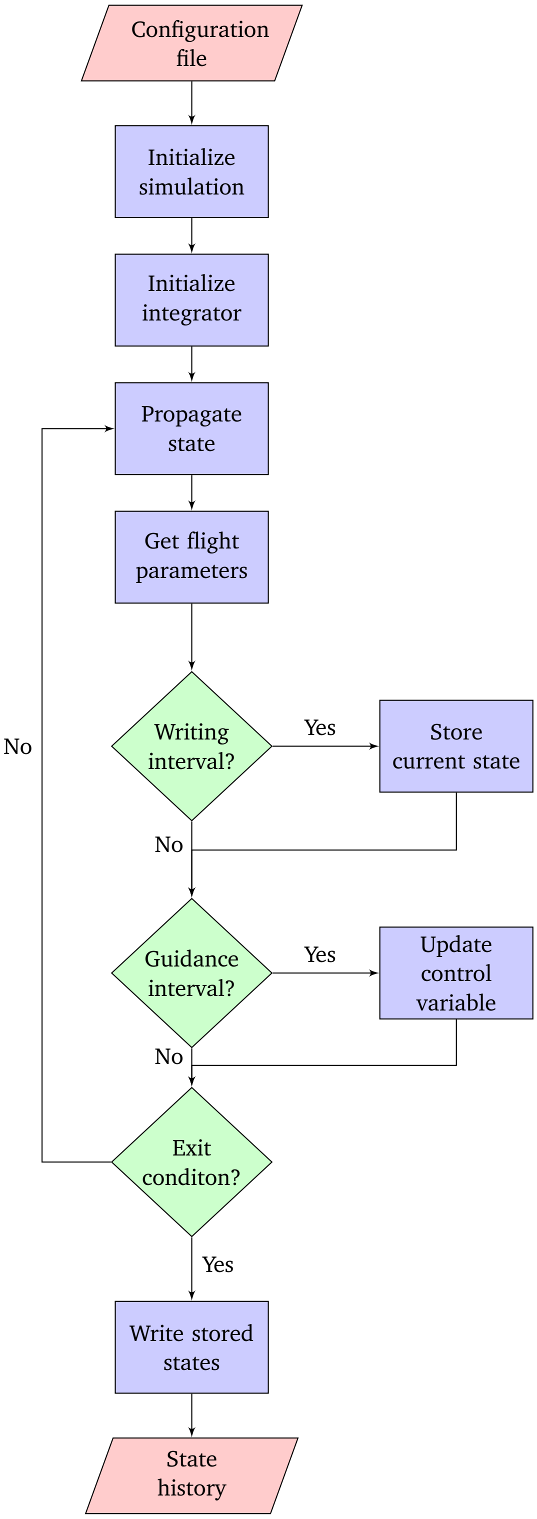


Figure 6.1: Software flow chart of simulator

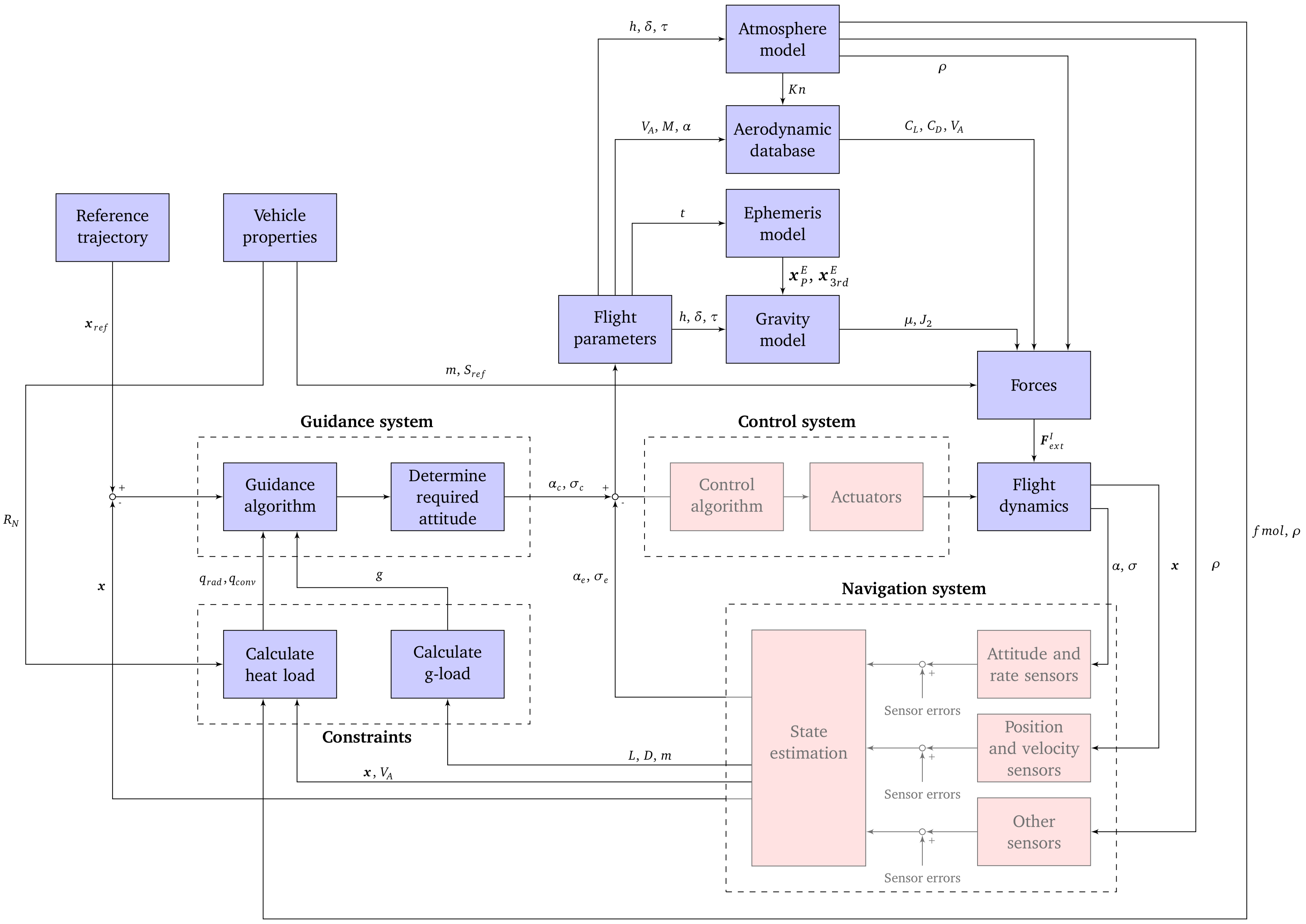


Figure 6.2: Software architecture of the developed simulator

6.3 Verification

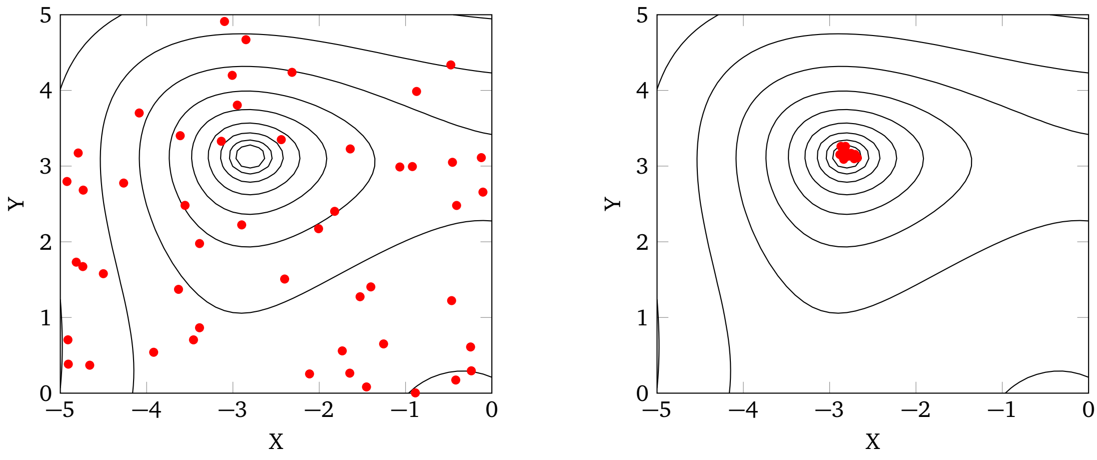
One should be able to rely on the results of the simulator. Each of the individual components, module or subroutine of the simulator should be producing the results they are supposed to produce, and the different elements should work together properly. In this section, the verification of the different subroutines will be described in Section 6.3.1 and the verification of the complete simulator will be discussed in Section 6.3.2.

6.3.1 Verification of subroutines

To ensure the obtained results are correct, the different subroutines created were verified. Such a verification step consists of one or more simple tests of which the result can be found by other means (e.g., from literature or analytical). For the simulator developed for this thesis, a number of these unit tests were executed, which are listed below.

- The **Flight parameters module** was verified by comparing its output with the hand-calculated evaluations the equations presented in Section 2.3. The results were identical.
- The **Atmosphere model module**, consists of two major elements: the detailed atmosphere model wrappers and the exponential atmosphere models. The wrappers written for the detailed atmosphere model were verified by evaluating the atmosphere models at a specific altitude, longitude, latitude and time and compare them to the results from the original model (either in C or in FORTRAN). As the wrapper should only be an interface, the output values of the wrapper and the actual model should be identical, which was indeed the case. The exponential atmosphere model that was also included in the *Atmosphere model* unit were obtained from Tudat, and were already verified.
- The gravitational properties in the **Gravity model module** were obtained from Tudat. Although already verified, the properties such as the gravitational parameter and the J_2 coefficient were compared with Tables 3.1 and 3.2 such that the correctness of the implementation could be confirmed. The output of this module matched the results of Tables 3.1 and 3.2 exactly.
- The magnitude of the gravitational and aerodynamic forces were verified by comparing the output of the **Forces module** with the analytical equations presented in Section 2.2. This resulted in an exact match. To verify the direction of these forces, they were plotted in three dimensions in the inertial planetocentric reference frame by means of frame transformations. As these reference frame transformations were obtained from Tudat, no additional verification steps were required. The gravitational acceleration should, in case no perturbing accelerations are taken into account, point towards the center of the central body, while the lift and drag force should respectively be perpendicular and parallel to the velocity vector. It was found that the direction of the forces that resulted from this module matched these directions.
- In the **Flight dynamics module**, all the forces are combined and used to evaluate the equations of motion. The Runge-Kutta-Fehlberg (RKF) numerical integration scheme that was used for the propagation of the equations of motion was obtained from Tudat, in which unit tests are included to ensure the correctness of the integrator. No additional verification of this numerical integrator has been been performed, but the correctness of its implementation will be verified in more detail in Section 6.3.2.
- The heat fluxes and g-load obtained from the **Constraints module** were compared with hand-calculated evaluations of the equations in Section 4.5.1 and Eq. (2.26). The results were identical.
- The **Guidance algorithm module** consists of two options: either it uses the guidance nodes or the drag tracking algorithm to obtain a value for the control variables. For the guidance nodes, a Hermite spline interpolator was developed, which was verified by interpolating the data points from Fig. 5.1 with both the C++ and `MATLAB` version of the interpolator, after which the resulting interpolants were compared. The obtained interpolants were identical. The drag tracking algorithm uses a drag reference to determine the required values for the control variables. Assume that only the bank angle is used as a control variable. If the same initial conditions are used as during the generation of the reference trajectory, the bank angles commanded by the drag tracking algorithm should be identical to the bank angle history of the reference trajectory. This was indeed the case.

- Although the optimization algorithm is not a module in the simulator, the implementation should be verified nonetheless. The implementation of the simulator in the MOEA/D algorithm from PaGMO was verified by substituting the Himmelblau function $f(x) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2$ as its objective function, while using the algorithm to find the minimum of this function. The search space was restricted to $-5 \leq x \leq 0$ and $0 \leq y \leq 5$. It is known that the Himmelblau function has a minimum at $f(-2.805118, 3.131312) = 0.0$. From Fig. 6.3, it can be seen that the optimization algorithm has indeed found this minimum.

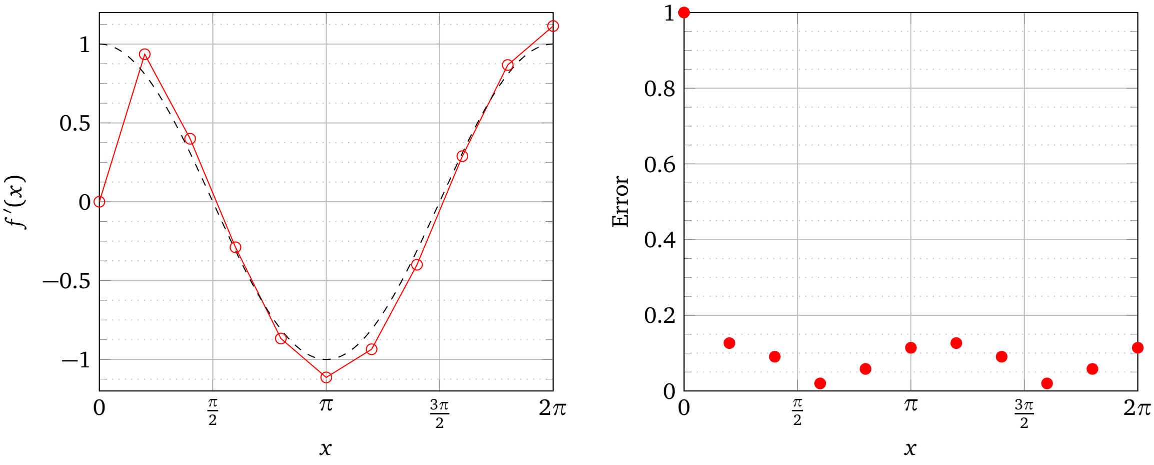


(a) Initial population

(b) Population after 10 generations

Figure 6.3: Verification of the implementation of MOEA/D in PaGMO

- The central differences methods were part of Tudat and were therefore already verified. The backward differences methods, required to obtain the derivatives of various variable in the guidance algorithm, were verified by determining the derivatives $f(x) = \sin(x)$ numerically and comparing the results to the analytical derivative. In Fig. 6.4a, the numerical derivative (red) is compared with $f'(x) = \cos(x)$ (black), while the error is shown in Fig. 6.4b. Initially, error is 1.0 as the derivative is set to 0.0 as backward differences scheme does not have sufficient data points to calculate the derivative. As more data points become available, the error is $O(h^2)$. It has been confirmed that for decreasing step-size, the error is decreased accordingly. Although not shown, the second derivative that was obtained using backward differences was compared with $f''(x) = -\cos(x)$, where also an error of $O(h^2)$ was observed.



(a) First derivative

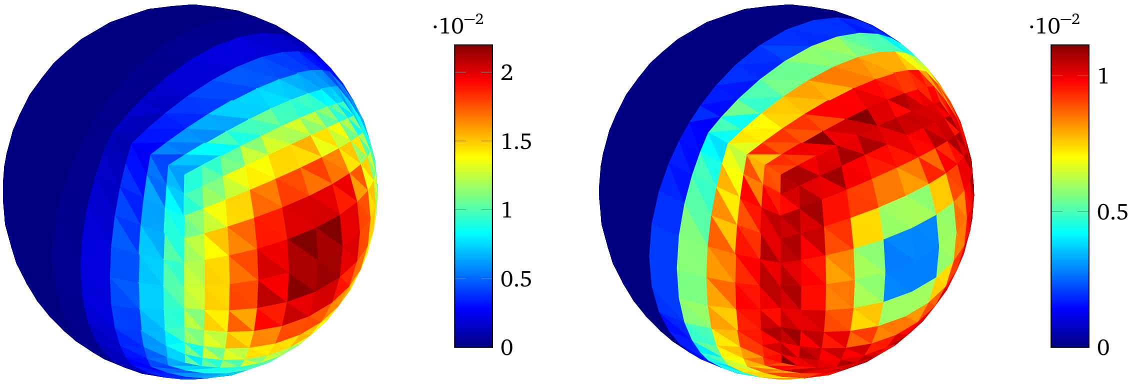
(b) Error

Figure 6.4: Verification of backward differences derivative for $f(x) = \sin(x)$

- The software that was obtained from Ronse (2013) to determine the aerodynamic coefficients in continuum flow, was already verified and the verification was therefore not repeated. SPARTA was used to determine these coefficients in rarefied flow. Although it is supposed to be verified by Sandia National Laboratories, this verification was not publicly available. Therefore, two tests were performed to verify the results obtained from SPARTA.

  First of all, the magnitude of the normal pressure and shear stress were projected onto the surface of a sphere. It was expected that the magnitude of the normal pressure would be maximum at the stagnation point of the flow and would decrease radially, while the opposite behavior was expected for the shear stress. In Figs. 6.5a and 6.5b, this behavior is indeed observed.

  For the second test, the aerodynamic coefficients of a sphere were determined using SPARTA. For a sphere, it is known that the lift coefficient $C_L = 0$ due to symmetry. The drag coefficient of a sphere, according to Sengers et al. (2014), should become $C_D = 2$ as the molecular speed ratio $s \to \infty$. Indeed, from Fig. 6.6, the drag coefficient approaches a value of $C_D = 2$ as the free-stream velocity increases.



(a) Normal pressure [N/m^2]

(b) Shear stress [N/m^2]

Figure 6.5: Normal pressure and shear stress distribution for a sphere in rarefied flow

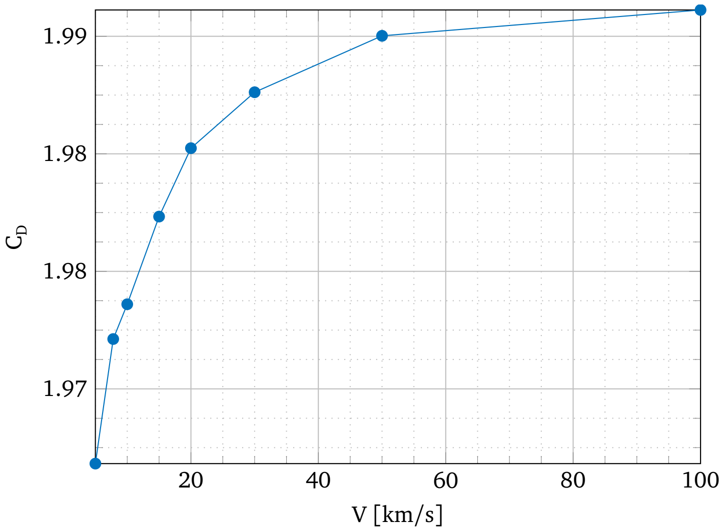


Figure 6.6: Drag of a sphere in rarefied flow determined using SPARTA

6.3.2 Verification of simulator

The simulator combines the equations of motion as well as the derived values such as heading angle and relative velocity. To confirm these were implemented correctly, a different set of equations of motion were used. While the implemented equations of motion are defined in the inertial planetocentric reference frame using the Cartesian coordinates, the equations of motion in Eqs. (6.1) to (6.6) (Mooij, 1997) are defined in the rotating planetocentric reference frame using spherical coordinates.

$$\dot{\tau} = \frac{V\cos\gamma\sin\chi}{R\cos\delta} \tag{6.1}$$

$$\dot{\delta} = \frac{V\cos\gamma\cos\chi}{R} \tag{6.2}$$

$$\dot{r} = V\sin\gamma \tag{6.3}$$

$$\dot{V} = -D - g\sin\gamma \tag{6.4}$$

$$\dot{\gamma} = \frac{1}{V}\left[L\cos\sigma - \left(g - \frac{V^2}{R}\right)\cos\gamma\right] + 2\omega_p\cos\delta\sin\chi \tag{6.5}$$

$$\dot{\chi} = \frac{1}{V}\left[\frac{V^2\cos\chi\tan\delta\cos\gamma}{R} + \frac{L\sin\sigma}{\cos\gamma} - 2\omega_p V\left(\tan\gamma\cos\delta\cos\chi - \sin\delta\right)\right] \tag{6.6}$$

Given a trajectory, all the variables required to evaluate Eqs. (6.1) to (6.6) are available or can be calculated using the equations in Section 2.3. When evaluated at each time step, Eqs. (6.1) to (6.6) will provide the derivative of the radial position R, the planetocentric velocity V, the longitude τ, the latitude δ, the flight-path angle γ and the heading angle χ with respect to time. These derivatives can also be calculated numerically using simple central differencing. If implemented correctly, these numerical derivatives should be identical to the results of Eqs. (6.1) to (6.6).

The derivatives of the six variables calculated using both methods are shown in Fig. 6.7. As can be observed from these figures, the numerical derivatives of the six state variables are identical to the ones obtained from Eqs. (6.1) to (6.6). From this, it can be concluded that the equations of motion in the simulation are implemented correctly.

The verification performed so far in this section corresponds to the equations of motion relative to the center of the planet. The position and velocity of a vehicle in the ecliptic reference frame determine inclination and velocity change that were achieved due to the aerogravity assist maneuver, and therefore need to be verified as well. The SPICE toolkit, for which an interface is available in the Tudat, also has the option to load trajectories of existing missions and obtain the Cartesian state at any epoch.

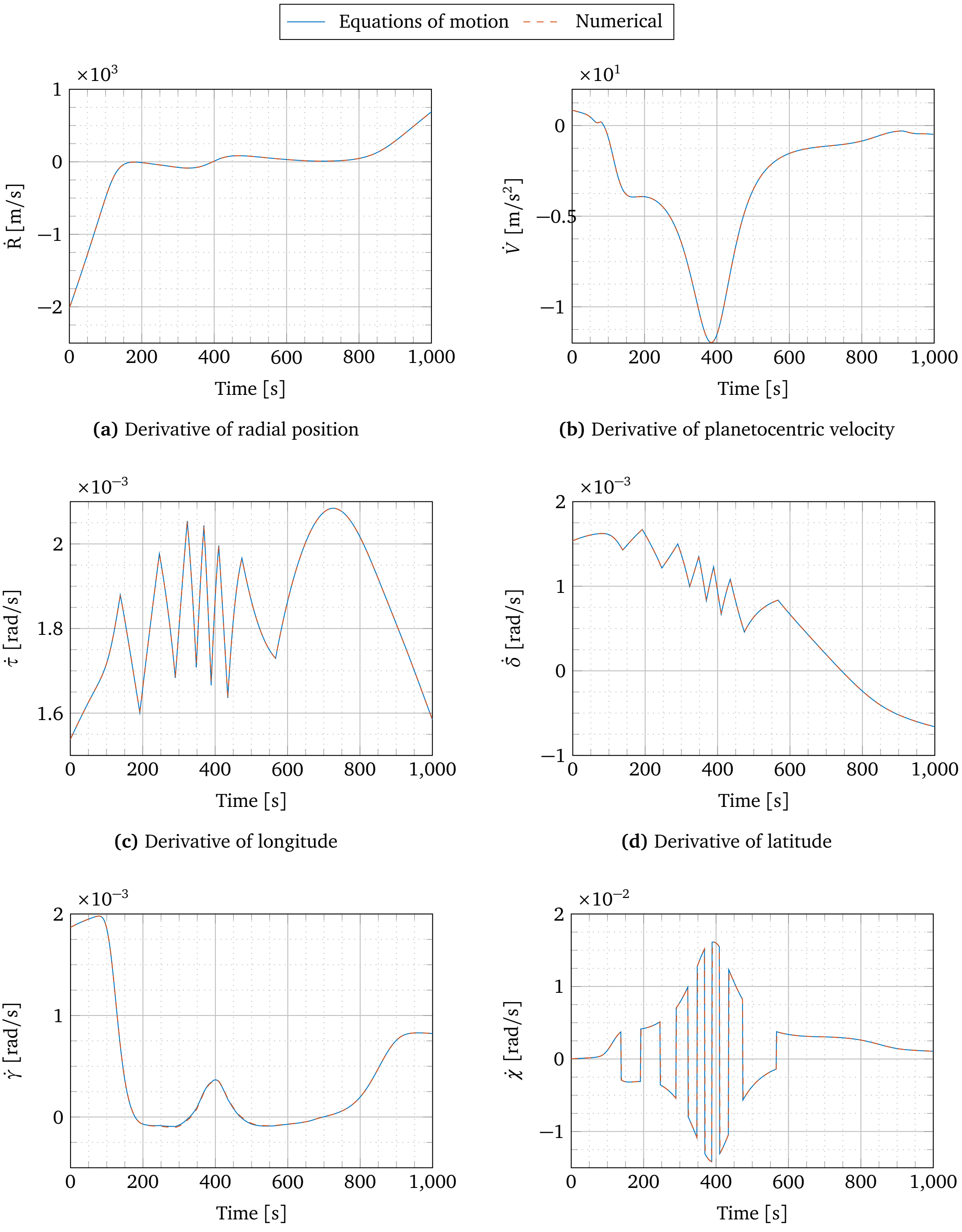
The Rosetta gravity assist around Mars was used as a reference case, and was downloaded from the ESA website[8]. The epoch of the closest approach occurs at 225640800 seconds past J2000, and a period of 5 hours around this epoch was extracted from Rosetta's SPICE kernel to capture the gravity assist maneuver. At the initial epoch, the initial conditions of the Rosetta spacecraft with respect to Mars were determined in the vertical reference frame, which are:

Variable	**Value**
Altitude [km]	7.379×10^5
Longitude [deg]	69.83
Latitude [deg]	-0.3055
Velocity [km/s]	10.06
Flight-path angle [deg]	-56.68
Heading angle [deg]	-86.77

These initial conditions were used as an input to the simulator, and the orbit was propagated for 5 hours. The individual components of the Cartesian state vector that were an output of the simulation are compared in Fig. 6.8. The position of the vehicle matches exactly the position of the Rosetta that was obtained from the SPICE kernel. However, there is a slight offset in the velocity.

[8] SPICE for Rosetta: `http://www.cosmos.esa.int/web/spice/spice-for-rosetta`, accessed: 4 December 2015

It was found that this offset is the result of a deviation in the used rotational rate of the rotating planetocentric reference frame. The angle between the rotating planetocentric reference frame and the inertial planetocentric reference frame is defined as the rotational rate of the body times the number of seconds that have passed since J2000. Since the gravity assist performed around Mars gravity assist occurred at 225640800 seconds past J2000, a small deviation in the rotational rate results in different orientation of the rotating planetocentric reference frame compared to the position of the rotating planetocentric reference. This offset in the frames result in different velocity vectors, as can be observed from Fig. 6.8. By increasing the orbital period of Mars by 0.2 seconds, which decreases the rotational rate of planet, the velocity vector in the ecliptic reference frame matched the one from the SPICE kernel. This confirms the offset in the components of the velocity vector is the result of the difference in rotational rate between the simulation and the SPICE kernel.



(a) Derivative of radial position

(b) Derivative of planetocentric velocity

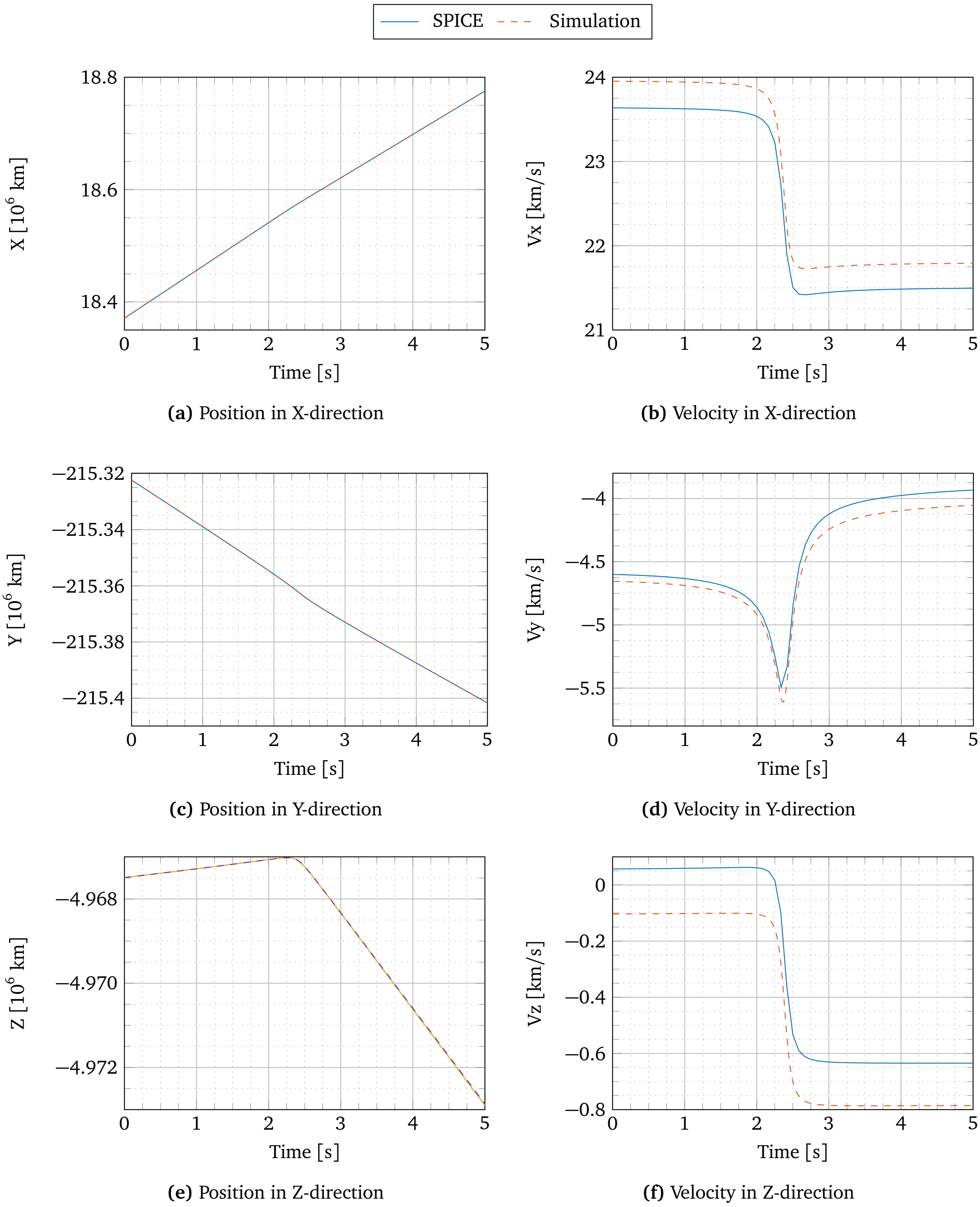
(c) Derivative of longitude

(d) Derivative of latitude

(e) Derivative of flight-path angle

(f) Derivative of heading angle

Figure 6.7: Verification of the equations of motion



(a) Position in X-direction

(b) Velocity in X-direction

(c) Position in Y-direction

(d) Velocity in Y-direction

(e) Position in Z-direction

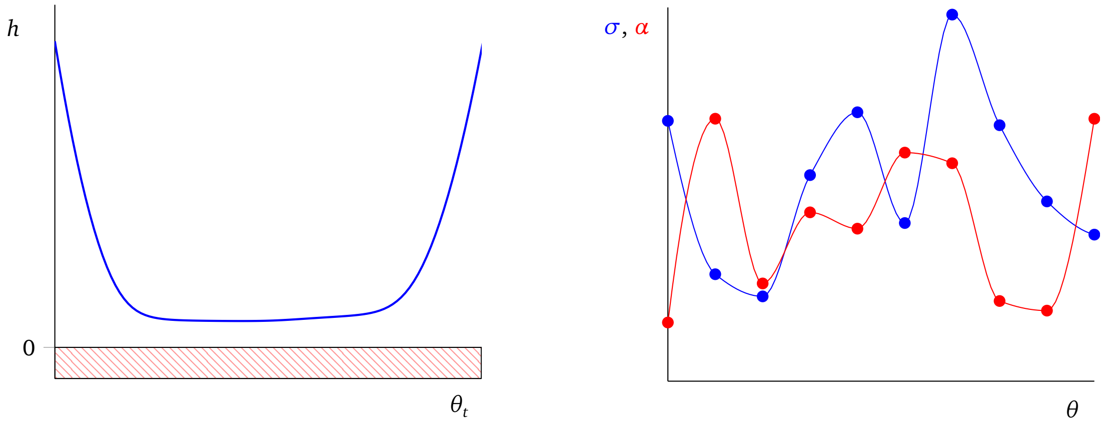
(f) Velocity in Z-direction

Figure 6.8: Cartesian components in ecliptic reference frame

7
Reference Trajectory Planning

The trajectory of an aerogravity assist is not trivial. Although essential, entering a planetary atmosphere is not sufficient for an aerogravity assist to be successful. As the goal of an aerogravity assist is to bend the trajectory over a specific angle, the vehicle should be steered such that this achieved. This is the reason the reference trajectory planner is developed.

When only considering aerodynamic forces as means of a control, the angle of attack and bank angle are the two main control variables. Assume that the trajectory in Fig. 7.1a, where the altitude is shown as a function of atmospheric bending angle, is necessary to rotate the velocity vector over a specific angle θ_t. Then, the control variables as a function of an independent variable, in this case the atmospheric bending angle θ, should be known. Such a control variable history, as depicted in Fig. 7.1b, will be defined by a number of nodes. This concept is discussed in Section 7.1.



(a) Altitude

(b) Control variables

Figure 7.1: Schematic overview of an aerogravity assist

The history of the control variables is not the only thing that should be taken into account. It is easy to imagine that a specific control variable history can result in a trajectory that enters the atmosphere quite deep, but will still achieve a specific bending angle. This will result in large aerodynamic forces, which increase the mechanical load on the vehicle. Furthermore, entering a more dense part of the atmosphere will increase the heat load on the vehicle. As the structure of the vehicle can only withstand a limited mechanical and heat load, not all control histories can be used to achieve the required bending angle. By defining a number of constraints, the numerical optimization algorithm will be able to find control histories that will be physically feasible. The constraints used in the reference trajectory planner are discussed in Section 7.2.

To find a control history by means of numerical optimization, the concept of node control should be converted to a number of decision variables, which are used by numerical optimization algorithm. Not only the control nodes, but also the initial conditions have a large impact on the results, of which some can be included in the list of decision variables. These decision variables variables and their impact on the trajectories will be analyzed in Section 7.3.

As the atmospheric bending angle is an indication of the performance of an aerogravity assist, maximizing this angle will be used as one of the objectives. However, as bank angle control can result in lateral maneuvers, the final offset in inclination should also be included in a second objective. How the atmospheric bending angle and orbital inclination relate to the objectives of the optimization is shown in Section 7.4.

In Section 7.5, an overview of the complete optimization problem is given. This planner is used to investigate the maximum possible atmospheric bending angles at both Mars and Jupiter for different arrival velocities in Sections 7.6 and 7.7. The impact of tighter constraints on the found trajectory is investigated in Section 7.8. Finally, Section 7.9 will discuss one of the found problems related to the orientation of the orbital plane.

7.1 Node control

Once a vehicle enters the atmosphere, the trajectory that follows is depending on the used control commands. For the aerogravity assist considered in this research, the control commands are limited to the angle of attack α and the bank angle σ. A method of obtaining a control history for the angle of attack and the bank angle is called node control, and was used by, for example, Dijkstra (2012) and Papp (2014). In this concept, a total of N nodes $\mathscr{N}$ are considered as a function of a independent variable. For each node, there is a corresponding control variable value. To obtain a continuous control history, these nodes are connected using an interpolation method. To prevent overshooting the guidance nodes, a Hermite spline interpolator, as was discussed in Section 5.2, will be used.

Dijkstra (2012), but also Papp (2014), consider the normalized specific energy $\hat{E}$ as an independent variable. This is, since both considered a re-entry problem and the specific energy reduces from $\hat{E} = 1$ to $\hat{E} = 0$, not an appropriate independent variable for an aerogravity assist. At the beginning of a trajectory, one does not know the amount of energy reduction due to the atmosphere. Consider the following example: the control nodes are spaced between $\hat{E} = 1$ and $\hat{E} = 0$, where $\hat{E} = 1$ is the specific energy at the start of the trajectory and $\hat{E} = 0$ is the energy level where the vehicle has no potential or kinetic energy. As the goal of an aerogravity assist is to leave the planet again, the energy level at the end of the trajectory will not be zero, which means that a part of the defined nodes are not used. It would of course be possible to space the control nodes between $\hat{E} = 1$ and the energy level at the end of the trajectory $\hat{E}_f$, but this would require to know the specific energy after the maneuver before the optimization is started. Even though one could estimate $\hat{E}_f$ by simple approximations, this was found to be difficult nonetheless. Furthermore, if $\hat{E}$ would be estimated incorrectly, the specific energy could decrease below $\hat{E}_f$, for which no nodes are defined. It is therefore decided to not use specific energy as an independent variable.

A more appropriate independent variable is the atmospheric bending angle θ. Even in case of an entry, this independent variable will not change as drastically as the specific energy. The atmospheric bending angle is considered to be the angle between the position vector at t_0, given by $\boldsymbol{X}_0^I$, and the position vector at a time $t > t_0$, given by $\boldsymbol{X}^I$, and is calculated by:

$$\cos\theta = \frac{\boldsymbol{X}_0^I \cdot \boldsymbol{X}^I}{||\boldsymbol{X}_0^I||\,||\boldsymbol{X}^I||} \tag{7.1}$$

One could argue that the problem when selecting the specific energy as an independent variable also exist for the atmospheric bending angle: for a given set of nodes, one does not know the resulting atmospheric bending angle. Therefore, some of the defined nodes could also remain unused. However, it can be said that an aerogravity assist will never have an atmospheric bending angle larger than 180°. If this would be necessary, it would be better to approach the planet from the other side. Therefore, using the atmospheric bending angle as independent variable, the end node is better defined.

The nodes that, once interpolated, form the control history, can be grouped into a guidance matrix $\boldsymbol{\Gamma}$:

$$\boldsymbol{\Gamma} = \begin{bmatrix} \mathscr{N}_0 \\ \mathscr{N}_1 \\ \vdots \\ \mathscr{N}_{N-1} \end{bmatrix} = \begin{bmatrix} \theta_0 & \alpha_0 & \sigma_0 \\ \theta_1 & \alpha_1 & \sigma_1 \\ \vdots & \vdots & \vdots \\ \theta_{N-1} & \alpha_{N-1} & \sigma_{N-1} \end{bmatrix} \tag{7.2}$$

where in the first column, the atmospheric bending angle is given, whereas in the second and third column, the angle of attack and bank angle for the corresponding bending angle is given, respectively. The atmospheric bending angle has been linearly spaced between 0° and 180°, and therefore the values of θ_i in the guidance matrix depends on the number of guidance nodes.

The only thing that currently remains is the actual value of the guidance nodes. To obtain these values, the numerical optimization algorithm presented in Section 5.4 will be used. The decision values in the optimization problem will be the guidance nodes. For a set of guidance nodes selected by the optimization algorithm, the resulting trajectory can be calculated.

A large number of guidance nodes give the opportunity to create more detailed control histories, at the cost that this also increases the number of decision variables. To find a good balance between the number of nodes and the optimization time, a trajectory optimization for 4, 6, 8 and 10 guidance nodes was performed with only the bank angle as control variable; the angle of attack has been set to its value at maximum lift-to-drag ratio. In an attempt to eliminate the effect of the randomness of the problem, each optimization was performed for three different random seeds. Although a multi-objective optimization is used to maximize the atmospheric bending angle and to minimize the offset from the target inclination (which will be discussed in Section 7.4), only the atmospheric bending angle will be shown in Fig. 7.2, since this is the most important objective. For each generation, the maximum bending angle will with an offset from the target inclination smaller than 1° i sshown.. The values of the maximum bending angle as a function generation number for the four different numbers of nodes are depicted in Figs. 7.2a to 7.2d.

In the case where $N_n = 4$, it takes longer to increase the bending angle than for the case where $N_n = 10$, which seems at first sight slightly odd: it would be expected that for less guidance nodes, the number of combinations is less and would therefore result in a faster convergence, while a larger number of guidance nodes has a much larger number of combinations possible, increasing the time for convergence. However, when looking at Figs. 7.2a and 7.2d, this does not appear to be the case.

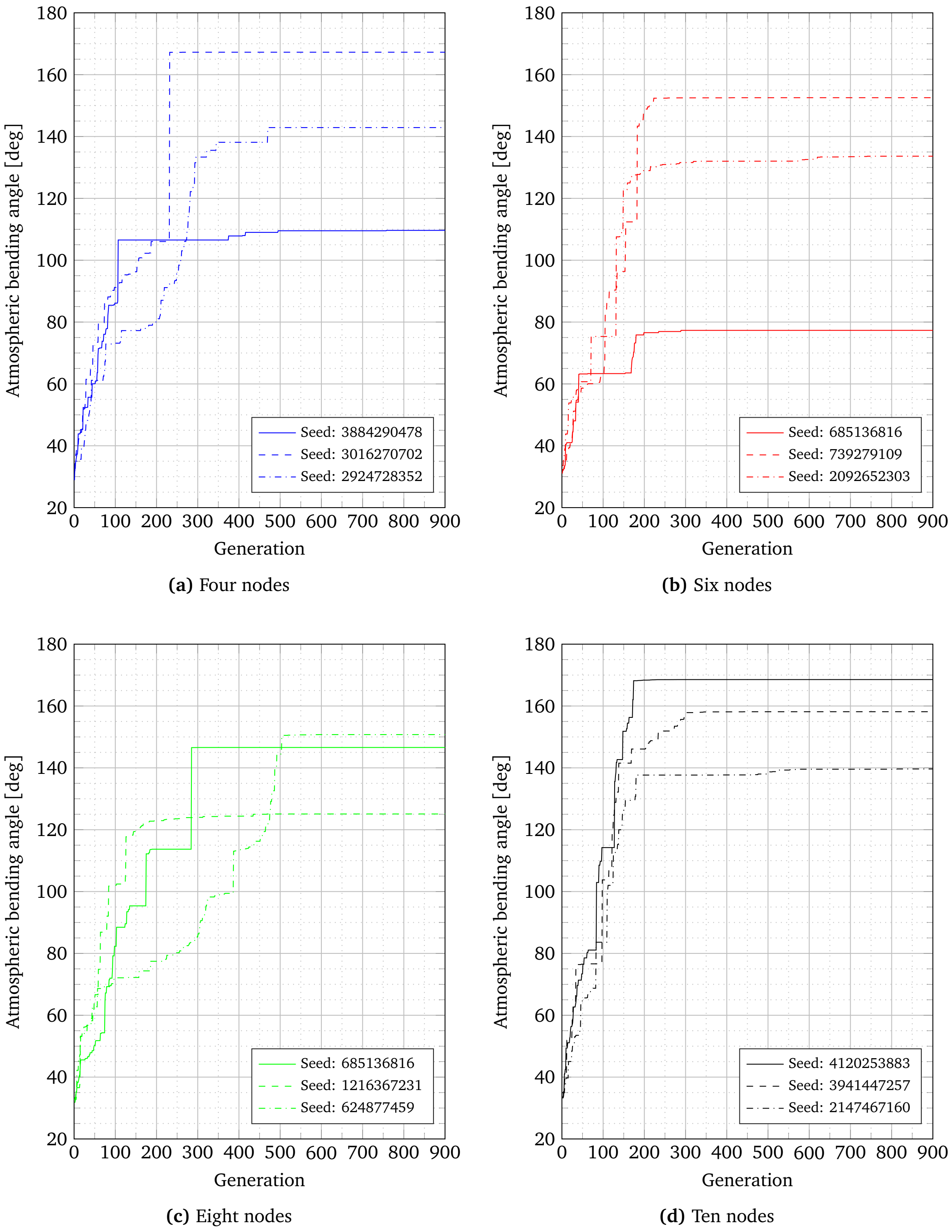
Although the results are different from the expectations, there is an explanation for this behavior. Consider the beginning of the trajectory optimization. At the beginning, the optimization algorithm has to make a guess of the initial nodes. This is either a bad choice, resulting in an entry or the trajectory only enters the atmosphere briefly before leaving it again. In this last case, the atmospheric bending angle is small. Lets say this bending angle is 45°. That means that in case of $N_n = 4$, the nodes at $\theta_0 = 0°$ and $\theta_1 = 60°$ determine the bank angle profile for this initial trajectory, while for $N_n = 10$, the nodes at $\theta_0 = 0°$, $\theta_1 = 20°$, $\theta_2 = 40°$ and $\theta_3 = 60°$ can form the the bank angle profile. This means that for more guidance nodes, it is possible to form a more detailed bank angle profile at the start of the optimization. Fig. 7.2d also shows more guidance nodes will result in a faster convergence.

Apart from the fact that the optimization for $N_n = 10$ converges faster, it also achieves on average a large value of the atmospheric bending angle than the other cases. This does not mean that it is not possible to obtain the same optimum value for the atmospheric bending angle with less nodes. The third run for $N_n = 4$ has also achieved maximum value of the bending angle similar to the first run of $N_n = 10$. However, from the results in Fig. 7.2, it appears that a larger number of nodes results in a higher probability of finding a maximum value with fewer generations. For this reason, it has been decided to use a total of 10 guidance nodes for the trajectory optimization.

Figure 7.2 also shows that the selected random seed has a large influence on the achievable atmospheric bending angle. Even though for $N_n = 10$, on average, the largest atmospheric bending angle can be achieved, the difference between the best case (seed 4120253883) and worst case (seed 2147467160) is still 30°. This indicates that the trajectory optimization is highly dependent on how the initial population is generated.

This dependence can also be seen for $N_n = 6$ (seed: 685136816) and $N_n = 8$ (seed: 624877459): the first one achieves only a very low final bending angle ($\theta_f = 77°$), while the second one achieves the largest bending angle of all three evaluated seeds for $N_n = 8$, but takes much longer to converge. No conclusive explanation can be given for this behavior, other than that the initial population is generated in a part of the design space that makes it difficult to find a global optimum.

To ensure a global optimum is achieved, the parameters of the optimization algorithm should be tuned more carefully, something which will be left for further research. To increase the probability of achieving a global optimum while maintaining the current parameters of the optimization algorithm, the number of runs could be increased. For the remainder of this thesis, an optimization run will be repeated for four instead of three times. Although this does not guarantee the global optimum will be found, it is assumed that the best optimization run is closest to the global optimum.



(a) Four nodes

(b) Six nodes

(c) Eight nodes

(d) Ten nodes

Figure 7.2: Influence on the number of nodes on the trajectory optimization

7.2 Constraints

If no constraints would be added to the trajectory optimization, faster convergence would be possible. However, this could also lead to trajectories that are physically unfeasible or undesirable. To prevent the trajectory optimization from finding these solutions, constraints are introduced. Four different constraints are used, which will be discussed below.

The first constraint that will be used is related to the maximum heat flux on the vehicle. As was discussed in Chapter 4, both radiative and convective heat transfer occur at high velocities. When these become too large, the vehicle will basically burn up. To avoid trajectories where this occurs, a heat flux constraint is imposed during the optimization. This heat-flux constraint is determined by the properties of the thermal protection system that is used. As the design of the thermal protection system is not part of this thesis work (see Section 4.5.3), literature is used to find reasonable estimates of the heat flux constraint. For the Mars entry, a heat-flux constraint of $500\,\mathrm{W/cm^2}$ was used, which is in agreement Dijkstra (2012) used as the magnitude of a heat flux constraint for the trajectory optimization of the Hyperion-II, a vehicle that has geometric characteristics similar to a waverider (e.g., a sharp nose). This value for the heat-flux constraint would be too strict for the trajectory optimization for a Jovian aerogravity assist. At Jupiter, the entry velocities are larger than on Mars, which results in larger heat fluxes. Based on Aso et al. (2006), where an aerocapture at Jupiter was studied, a heat flux constraint of $6000\,\mathrm{W/cm^2}$ is proposed for the trajectory optimization at Jupiter.

In addition to the heat flux, the aerodynamic force on the vehicle will result in a mechanical load. If too excessive, this could lead to structural damage or even disintegration. It is therefore necessary to include a constraint in the trajectory optimization on the maximum value of the mechanical load. Based on the analysis performed in Casoliva et al. (2008), a mechanical load of 15g was selected as a constraint.

The final two constraints are used to avoid solutions that physically would not result in an aerogravity assist. If the eccentricity of the orbit after the atmospheric phase is less than 1, the vehicle is captured by the planet and has entered an elliptical orbit around the planet. Since for an aerogravity assist it is necessary to leave sphere of influence of the planet, the final eccentricity should remain larger than 1. The last constraint is that the altitude along the trajectory should never become negative. This is considered to be an entry and/or crash on to the surface, and is incorporated in the simulator it self. If the trajectory is flagged by the simulator as an entry, the optimizer knows this constraint is violated.

7.3 Decision variables

The decision variables in the trajectory optimization are the variables that are varied by the optimization algorithm to find a trajectory that does not violate the constraints. This section will discuss the selected decision variables used in the trajectory optimization.

The guidance nodes, which represent the used control variable profile, form a major part of the decision variables. For the control variables, the angle of attack and the bank angle, constraints are also imposed. The aerodynamic database was generated between $-20° \leq \alpha \leq 20°$, so this constraint should be imposed to prevent selecting an angle-of-attack value outside of the range of the aerodynamic database if the angle of attack would be taken into account. Although the angle of attack can be used, it was decided to only use the angle of attack at maximum lift-to-drag ratio and only use the bank angle to modulate the vertical lift component. To limit the search space, it was decided to only optimize the magnitude of the bank angle, $|\sigma|$. The sign of the bank angle was determined by the amount of offset from the desired inclination angle. If the inclination angle would overshoot this offset, which was set to 3°, the sign of the bank angle would be reversed, leading to a so-called bank reversal.

The guidance nodes, as was explained in Section 7.1, form a set of the decision variables. However, the initial flight-path angle and velocity are also two parameters that have a large influence the resulting trajectory. After some trial optimization runs, it was observed that some of the solutions resulted in a bank angle that would only remain 180°. Since only the magnitude of the bank angle will be optimized, its value should be between 0° and 180°. So a bank-angle profile where the bank angle remains 180°, the vehicle basically travels upside-down. If due to uncertainties in any of the used models, the guidance algorithm would require an increase in the vertical force, this cannot be achieved and the vehicle would skip out of the atmosphere.

The behavior explained in the previous paragraph was investigated in more detail. Using a bank angle of

180°, the initial velocity and initial flight-path angle were varied, such that their influence on the resulting atmospheric bending angle could be observed. Figures 7.3 and 7.4 show the influence of the initial velocity and initial flight-path angle on the atmospheric bending angle, where each individual dot is one evaluated trajectory. An asymptotic behavior can be observed in this figure around a flight-path angle of approximately −15° for Mars and −11° for Jupiter. If a vehicle enters at a shallow flight-path angle, the vehicle will not penetrate the atmosphere enough to be captured, so the vehicle will skip out of the atmosphere and consequently the atmospheric bending angle will be relatively small. This corresponds to the right side of the figures. At larger flight-path angles, the effect of the atmosphere becomes more noticeable, which results in larger atmospheric bending angles. This bending angle increases asymptotically to a line projected on the horizontal plane. At flight-path angles steeper than indicated by this line on the horizontal plane, the atmosphere will be penetrated so deep that a too large amount of energy is dissipated, which results in an entry and a decrease in atmospheric bending angle. In the figures, this can be observed on the left side of the figure, where the possible bending angle starts decreasing again.

This division between entry and an atmospheric flight is also shown in Figs. 7.5 and 7.6, and was obtained by fitting a polynomial through the points around the asymptotic behavior (indicated in red in Figs. 7.3 and 7.4). The area that would result in an entry for $\sigma = 180°$ are shaded green, as these combinations of flight-path angle and velocity would guarantee a trajectory that is controllable. The combinations of flight-path angle and velocity that would result in a bank-angle profile of $\sigma = 180°$, and therefore would be uncontrollable, are shaded red.

Intuitively, one would want to avoid the area below the curve in Figs. 7.5 and 7.6, since this would result in an entry. However, the analysis on which Figs. 7.5 and 7.6 is based upon was performed with a bank angle of 180°. This means that in the area below the curve, the downward component of the lift vector pulls the vehicle to the surface of the planet, implying that the magnitude of this downward component is too large. By decreasing the magnitude of the bank angle, the downward component of the lift vector is also decreased, which can avoid the entry. Therefore, when for a given initial velocity, a flight-path angle is selected below the curve in Figs. 7.5 and 7.6, the magnitude of the bank angle profile that will be provided by the optimization algorithm will always be less than 180°, which is exactly the sort of bank angle profile that is desired.

From Figs. 7.3 and 7.4, it can also be obtained that the used initial flight-path angle has a large impact on the resulting trajectory. Entering the atmosphere too shallow and the vehicle will skip out of the atmosphere, while entering the atmosphere at a too steep angle will result in an entry. It is also expected that small variations in the initial flight-path angle can greatly influence the trajectory. This expectation is also supported by the results from Figs. 7.3 and 7.4, where for initial flight-path angles around −15° (Mars) or −11° (Jupiter) the gradient with respect to the bending angle is quite steep. The initial flight-path angle is for this reason selected as a decision variable, with an upper limit imposed by the curve shown in Fig. 7.5 for Mars and Fig. 7.6 for Jupiter.

The velocity at which a vehicle approaches a planetary atmosphere is an additional initial parameter. The initial velocity could also be added to the list of decision variables, such that the optimization algorithm determines its value. However, the initial velocity, which is determined to be the velocity at which the simulation starts, is a function of the hyperbolic excess velocity V_{∞}^{-}, which in turn is an output of the interplanetary mission planning. Accelerating or decelerating to obtain a velocity to start an aerogravity assist with is not desired, since this would require, for example, the use of propulsive maneuvers. Therefore, the initial velocity will be used as a free variable and will be varied such that the influence of the initial velocities can be studied.

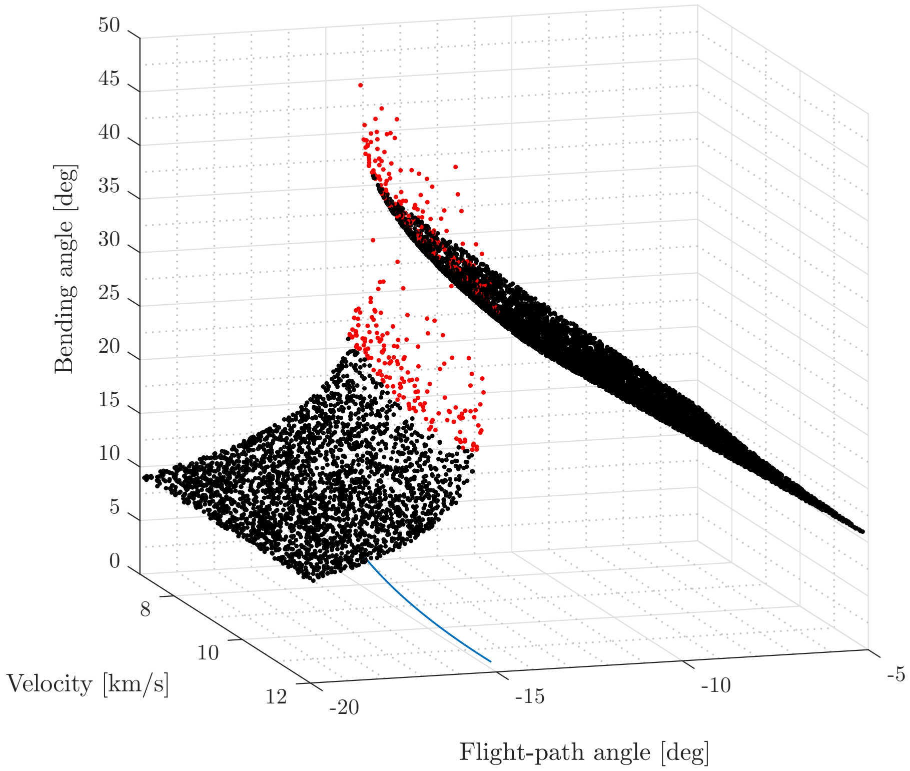


Figure 7.3: Influence of flight-path angle on the resulting trajectory for Mars

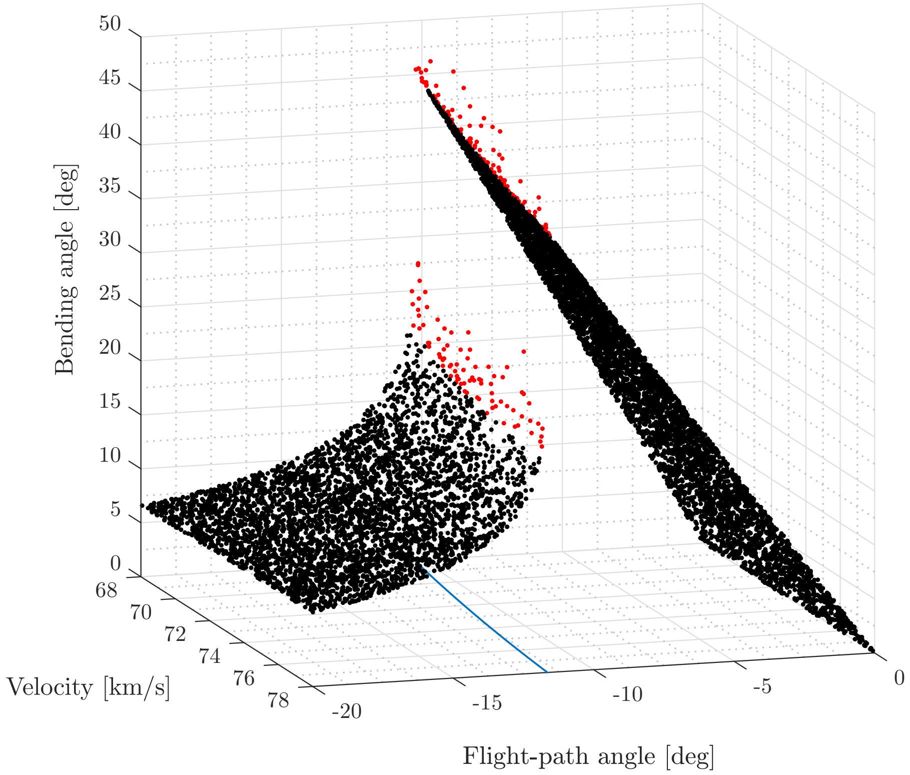


Figure 7.4: Influence of flight-path angle on the resulting trajectory for Jupiter

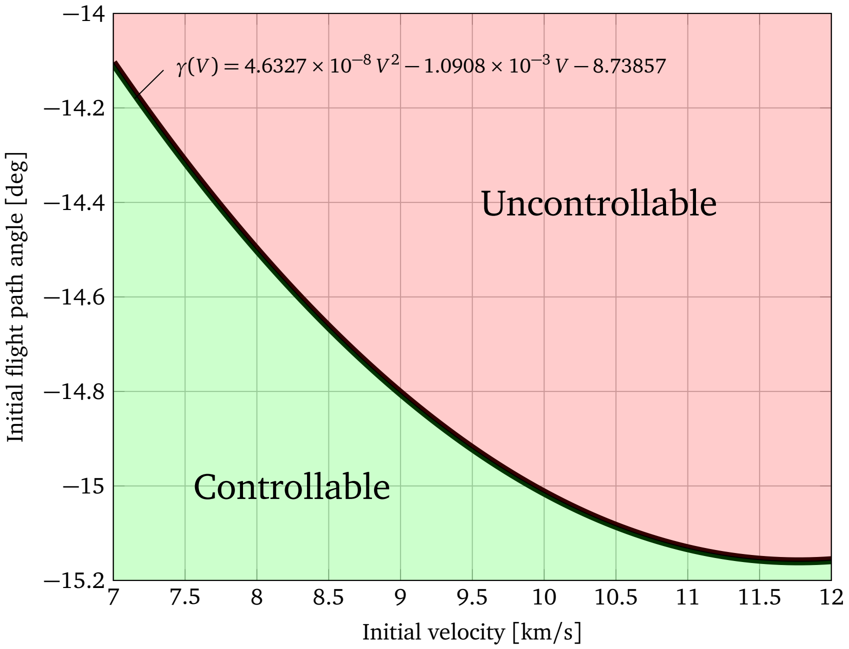


Figure 7.5: Allowable velocity and flight-path angle combinations based on a Monte Carlo analysis with $\sigma = 180°$ and an initial altitude of 160 km for Mars

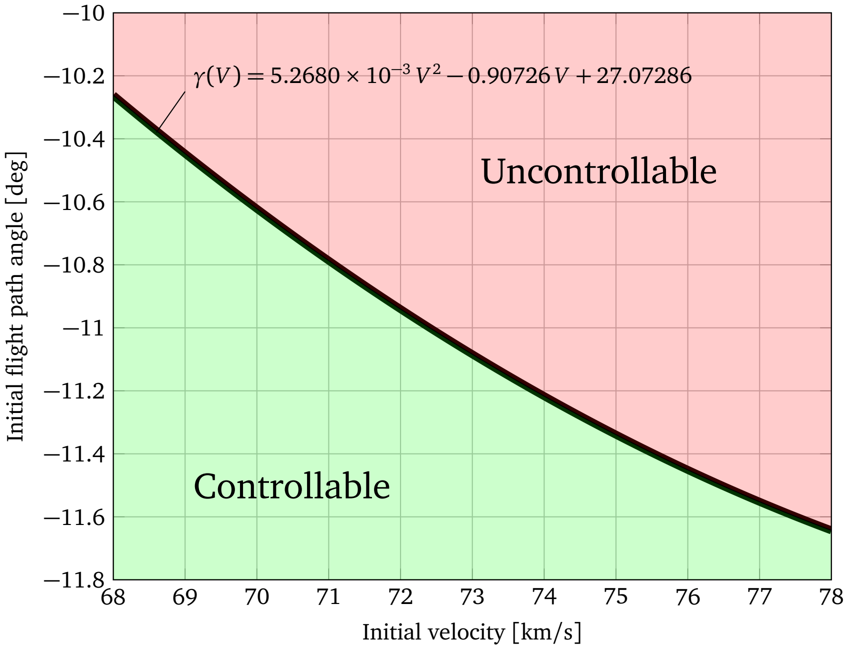


Figure 7.6: Allowable velocity and flight-path angle combinations based on a Monte Carlo analysis with $\sigma = 180°$ and an initial altitude of 3200 km for Jupiter

7.4 Objectives

The objectives of an optimization are the goals that one wants to achieve. Since for regular gravity assists, the the angle over which the velocity is bend is a measure of the effectiveness of the maneuver, a similar metric is necessary to determine the effectiveness of an aerogravity assist. Before the first objective is introduced, it is important to make a distinction between two different bending angles: the *atmospheric* bending angle θ and the *velocity* bending angle δ, as was depicted in Fig. 1.2. The atmospheric bending angle is the angle between the position vector at the start of the trajectory and the end of the trajectory. Since the trajectory starts and ends at the edge of the atmosphere, this angle indicates for how long the trajectory is in the atmosphere. The velocity bending angle is the angle between the hyperbolic excess velocity of the arrival trajectory $\boldsymbol{V}_{\infty}^{-}$ and the hyperolic excess velocity of the departure trajectory $\boldsymbol{V}_{\infty}^{+}$. These two parameters can be determined from the first and last state of the trajectory: since the trajectories will start and end at the edge of the atmosphere, the orbital elements will hardly be effected once the vehicle is out of the atmosphere. These orbital parameters can then be used to determine $\boldsymbol{V}_{\infty}^{-}$ and $\boldsymbol{V}_{\infty}^{+}$.

For the first objective, the maximization of the atmospheric bending angle θ has been selected. A larger value of the atmospheric bending angle directly relates to the angle over which the velocity can be rotated. A larger angle also means more freedom in the planning of interplanetary missions. In addition, the value for θ was available as it was used for the definition of the guidance matrix in Eq. (7.2). Since the optimization algorithm in PaGMO can only minimize variables, the variable that should be minimized is $-\theta$.

Since the aerodynamic force can have a cross-track component because of non-zero bank angles, the inclination of the trajectory can adjusted. At the beginning of the optimization, a target inclination value has to be provided. This can either be the same as the initial inclination if no orbital plane change is required or a target value if this plane change is desired. Even though bank reversals are introduced to avoid a too large offset in inclination, it might still be possible that the final inclination is slightly off from the target inclination. By defining the second objective as the minimization of the difference between the target inclination and the final inclination of the trajectory, the bank reversals can be scheduled such that the final inclination is indeed close to the required target inclination.

7.5 Optimization overview

In summary, the following decision variables and constraints will be used to define the optimization problem:

- **Objective**
  - Minimize negative atmospheric bending angle $-\theta$
  - Minimize offset between final inclination and target inclination Δi
- **Decision variables**
  - Initial flight-path angle: $\gamma_{init} < f\,(V_{init})$
  - Guidance nodes $\sigma_0 \ldots \sigma_{N-1}$
- **Operational constraints**
  - Mechanical load: $\leq 15\text{g}$
  - Heat flux: ≤ 500 W/cm^2 (Mars) / ≤ 600 W/cm^2 (Jupiter)
  - Bank reversal: $\Delta i > 3°$
- **Control constraints**
  - Angle of attack: α at $(L/D)_{\max}$
  - Bank angle: $0° \leq \sigma \leq 180°$

Due to time constraints, it was only possible to find reference trajectories where the bank angle was used as a control variable. As was explained earlier, this would result a cross-track component of the lift, which would in turn result in a lateral maneuver. Using the angle of attack as control variable, the lift vector will remain in the plane of motion and no lateral guidance has to be used. Disadvantage of angle-of-attack control, however, is that the drag coefficient will increase compared to the maximum lift-to-drag ratio, which will lead to larger energy dissipation in the atmosphere. It might also be possible to combine both angle-of-attack and bank-angle control, in which the benefits of the two methods could be combined. However, angle-of-attack control, and angle-of-attack and bank-angle control combined will be left for future research.

Using these definitions of the optimization problem, the Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) in PaGMO can be initialized. The objective function consists of the evaluation of a trajectory given the decision variables generated by the MOEA/D algorithm. From the evaluated trajectory, the maximum heat flux and mechanical load can be obtained, as well as the final inclination, eccentricity and altitude. If these values violate the constraints, the found solution is flagged as an impossible trajectory.

Apart from the initial flight-path angle and velocity that are defined by the optimization algorithm, the remaining initial conditions that were used for the aerogravity assist at Mars and Jupiter are summarized in Table 7.1.

Table 7.1: Initial conditions used in trajectory optimization

Parameter	Mars	Jupiter
Altitude [km]	162.545	3200.0
Latitude [deg]	0.0	0.0
Longitude [deg]	0.0	0.0
Heading angle [deg]	45.0	37.6
Mass [kg]	800.0	800.0
Reference area [m^2]	66.67	6.67
Nose radius [m]	1.0	1.0

7.6 Influence of initial velocity - Mars

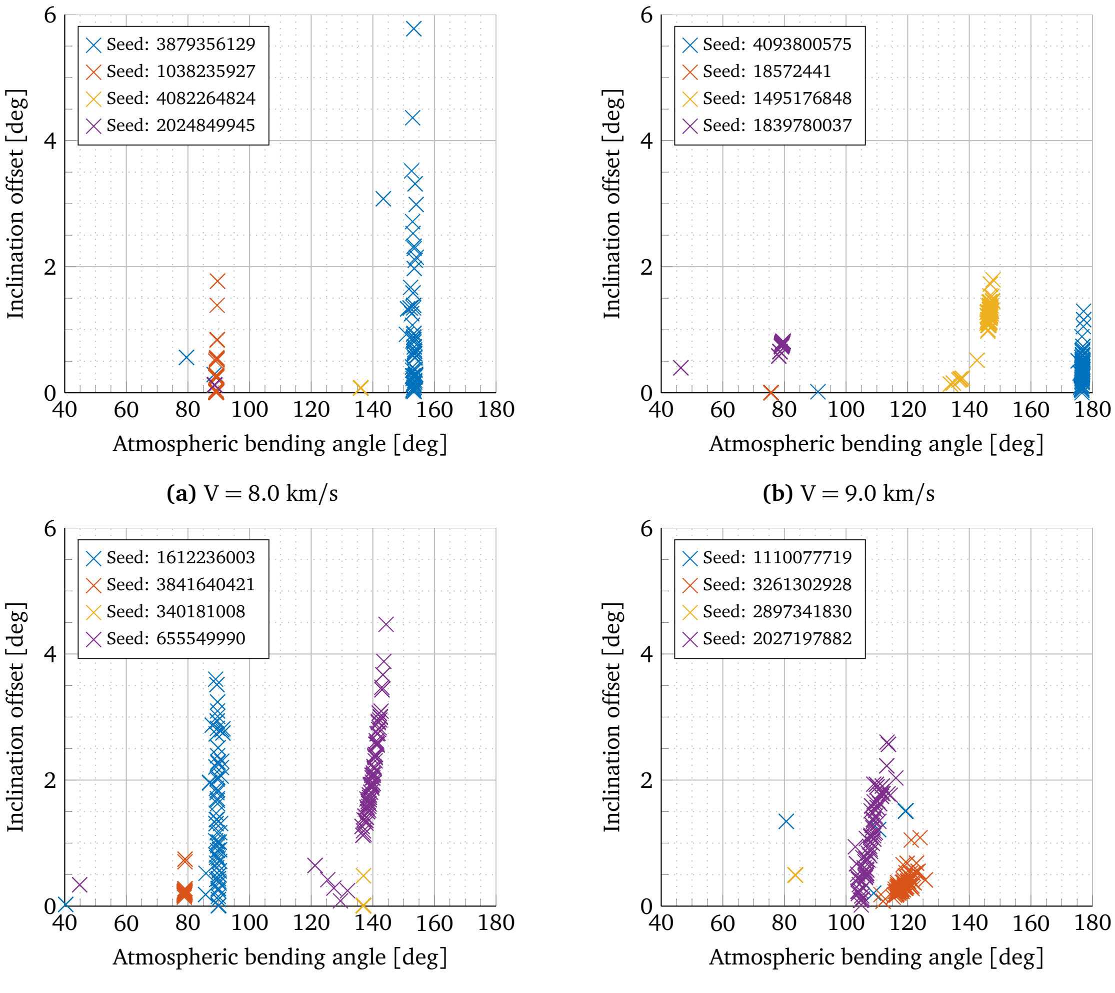
From Section 7.2, it became apparent that the initial velocity has a large influence on the resulting trajectory. To investigate the effect of the velocity on the resulting trajectory, the trajectory optimization was performed for four different initial velocities: 8.0 km/s, 9.0 km/s, 10.0 km/s and 11.0 km/s. The optimization run was performed multiple times with four different random seeds to prevent that a solution gets stuck in a local optimum. The populations of the best optimization runs for each of the four velocities is shown in Fig. 7.7.

Figure 7.7 shows that all the four analyzed initial velocities have random seeds that converge prematurely. This is in agreement with what was observed in Section 7.1. For each of the seeds that performed the best, being seeds 3879356129, 4093800575, 655549990 and 3261302928 for initial velocities of 8.0 km/s, 9.0 km/s, 10.0 km/s and 11.0 km/s respectively, one individual has been selected that achieved the maximum atmospheric bending angle for that velocity. For these individuals, the actual trajectories are shown in Fig. 7.8.

The optimization algorithm found trajectories that have a more or less constant altitude, as is shown in Fig. 7.8a. However, some dips in the atmosphere can be observed, especially for initial velocities 8.0 km/s and 11.0 km/s. These dips automatically results in a change in flight-path angle, as is shown in Fig. 7.10e. In addition, these dips will also result in an increase of the g-load and heat flux as shown in Figs. 7.8c and 7.8d. This is the result of the increase in atmospheric density, which occurs when the atmosphere is penetrated deeper. As these dips are undesirable, one could impose tighter constraints during the cruise phase to prevent this from happening.

For an initial velocity of 8.0 km/s, the maximum atmospheric bending angle is approximately 153°. After the atmospheric maneuver, the eccentricity of the orbit is 1.029, as can be seen in Fig. 7.8b. This means that the orbit is still hyperbolic and therefore leaves still leaves the planet. For low arrival velocities, it can be concluded that the limiting constraint is $e > 1.0$, and not the heat-flux or g-load constraints.

As the initial velocity increases, the experienced g-load and heat flux increases, which is visible in Figs. 7.8c and 7.8d. Due to the higher velocities a vehicle enters the atmosphere, the aerodynamic forces on



(a) V = 8.0 km/s

(b) V = 9.0 km/s

(c) V = 10.0 km/s

(d) V = 11.0 km/s

Figure 7.7: Pareto fronts after optimization for Mars

the vehicle become larger. The eccentricities of the orbits after the atmospheric flight become larger (2.260, 3.052 and 3.885 for 9.0 km/s, 10.0 km/s and 11.0 km/s respectively), which yields that the orbital energy is still sufficiently large to leave the planet. However, the increase in g-load and heat flux become a more important constraint. Especially for the largest initial velocity, 11.0 km/s, the maximum heat flux approaches, but does not reach the constraint of 500 W/cm^2, while the maximum experienced g-load is approximately 10g.

The maximum bending angle was obtained for a trajectory with an initial velocity of 9.0 km/s. For this trajectory, neither the g-load constraint nor the heat flux constraint is violated. At this point, the velocity is decreased by such an amount that the lift force, which is a function of the velocity, is no longer large enough to keep the vehicle close to the planetary surface.

Using the same reasoning, one would expect that for an initial velocity of 10.0 km/s, which also does not violate the heating and g-load constraints, a bending angle similar to the trajectory with the initial velocity of 9.0 km/s could be achieved. However, the achieved bending angle for an initial velocity of 10.0 km/s is significantly smaller than the bending angle that was obtained for a initial velocity of 9.0 km/s. Since none of these constraints are violated for an initial velocity of 10.0 km/s and the eccentricity does not approach 1.0, it can be concluded that the found solutions are local optima, even though the optimization was repeated multiple times.

By looking at Fig. 7.8f, one can observe that the four trajectories end in different orbital planes although one of the objectives of the optimization was to minimize the offset between final inclination and the initial inclination. One of the reasons that the orbital plane was changed was that not only the orbital inclination

Table 7.2: Comparison of gravity assist (GA) and aerogravity assist (AGA) at Mars

	Variable	Unit	GA	AGA	% Difference
8.0 km/s	δ	deg	26.15	154.39	490.48
	V_∞^+	km/s	6.54	0.61	−90.70
	i	deg	43.71	42.85	−1.97
9.0 km/s	δ	deg	19.83	178.53	800.51
	V_∞^+	km/s	7.75	3.96	−48.86
	i	deg	43.85	44.66	1.84
10.0 km/s	δ	deg	15.63	140.57	799.32
	V_∞^+	km/s	8.91	5.06	−43.20
	i	deg	43.96	41.12	−6.47
11.0 km/s	δ	deg	12.67	116.31	818.20
	V_∞^+	km/s	10.04	6.00	−40.26
	i	deg	44.05	45.42	3.09

was changed, but also the right ascension of the ascending node (RAAN). Even though the final inclination is more or less equal to the initial inclination, the change in RAAN results in a rotation of the orbital plane, which explains the different orbital planes that can be seen in Fig. 7.8f. It can therefore be concluded that only using the Keplerian inclination as a means of controlling the orbital plane is not sufficient.

The results found during the trajectory optimization are compared with a gravity assist for the same initial conditions in Table 7.2. In reality, a gravity assist with the same conditions could not be executed as the atmosphere will be entered. For the sake of analysis, the atmosphere model was disabled such that the contribution of the atmosphere only could be investigated. Furthermore, in Table 7.2, the angle over which the velocity is bent, δ, rather than the atmospheric bending angle θ is shown, since the rotation of the velocity vector is the major purpose of an (aero)gravity assist.

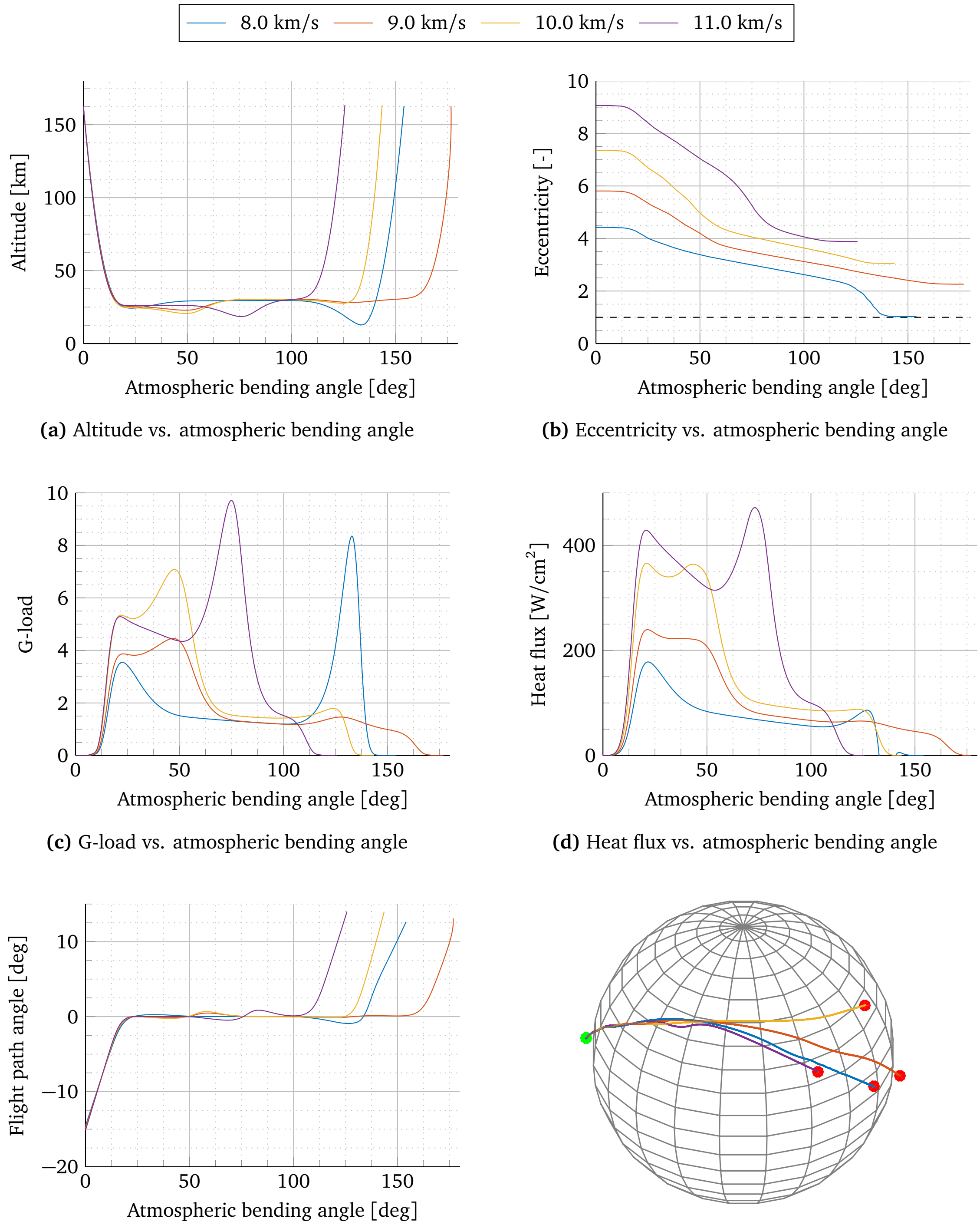
For all four cases in Table 7.2, the hyperbolic excess velocity is decreased with respect to a gravity assist due to energy dissipation as a result of drag. However, for an entry velocity of 8.0 km/s, the hyperbolic excess velocity is just over 600 m/s, which is barely enough to depart from the planet. This is in agreement with the found eccentricity of the departure branch mentioned earlier in this section. In the worst case, the velocity bending angle is increased by 490%, while for an entry velocity of 11 km/s, this angle is increased by 818%. Deviations up to 6.5% in the final inclination of the trajectory can be observed. An explanation for this will be given in Chapter 9.

7.7 Influence of initial velocity - Jupiter

The analysis performed for Mars in Section 7.6 has also been performed for an aerogravity assist around Jupiter. Due to its large mass, the initial velocities that were used for Mars would result in elliptical orbits at Jupiter. Since aerogravity assists require hyperbolic arrival and departure legs, larger initial velocities were required to ensure hyperbolic orbits. The initial, planetocentric velocities that were selected for the analysis on a Jovian gravity assist were 68.0 km/s, 72.0 km/s, 76.0 km/s and 80.0 km/s, at a starting altitude of 3200.0 km, where $h = 0$ m is defined as the point where the atmospheric pressure is 1 bar. Furthermore, as the heat flux constraint used for the Martian aerogravity assist proved to be too strict, it was increased to $6000\,\mathrm{W/cm^2}$, which was discussed in Section 7.2.

Again, for each initial velocity, the trajectory optimization was performed multiple times with different seeds in an attempt to avoid local optima. The Pareto fronts for each of the four different initial velocities are shown in Fig. 7.9. A number of optimization runs has converged prematurely. For an initial velocity of 80.0 km/s, the achieved atmospheric bending angle did not exceed 40°. To investigate the trajectories in more detail, for each of the random seeds that achieved the largest bending angle (seeds 2352524897, 188331177, 1993949069 and 114798285), one individual has been selected. For these individuals, the corresponding trajectory is shown in Fig. 7.10.

The trajectory with the largest atmospheric bending angle is the trajectory with an initial velocity of 68.0



(a) Altitude vs. atmospheric bending angle

(b) Eccentricity vs. atmospheric bending angle

(c) G-load vs. atmospheric bending angle

(d) Heat flux vs. atmospheric bending angle

(e) Flight-path angle vs. atmospheric bending angle

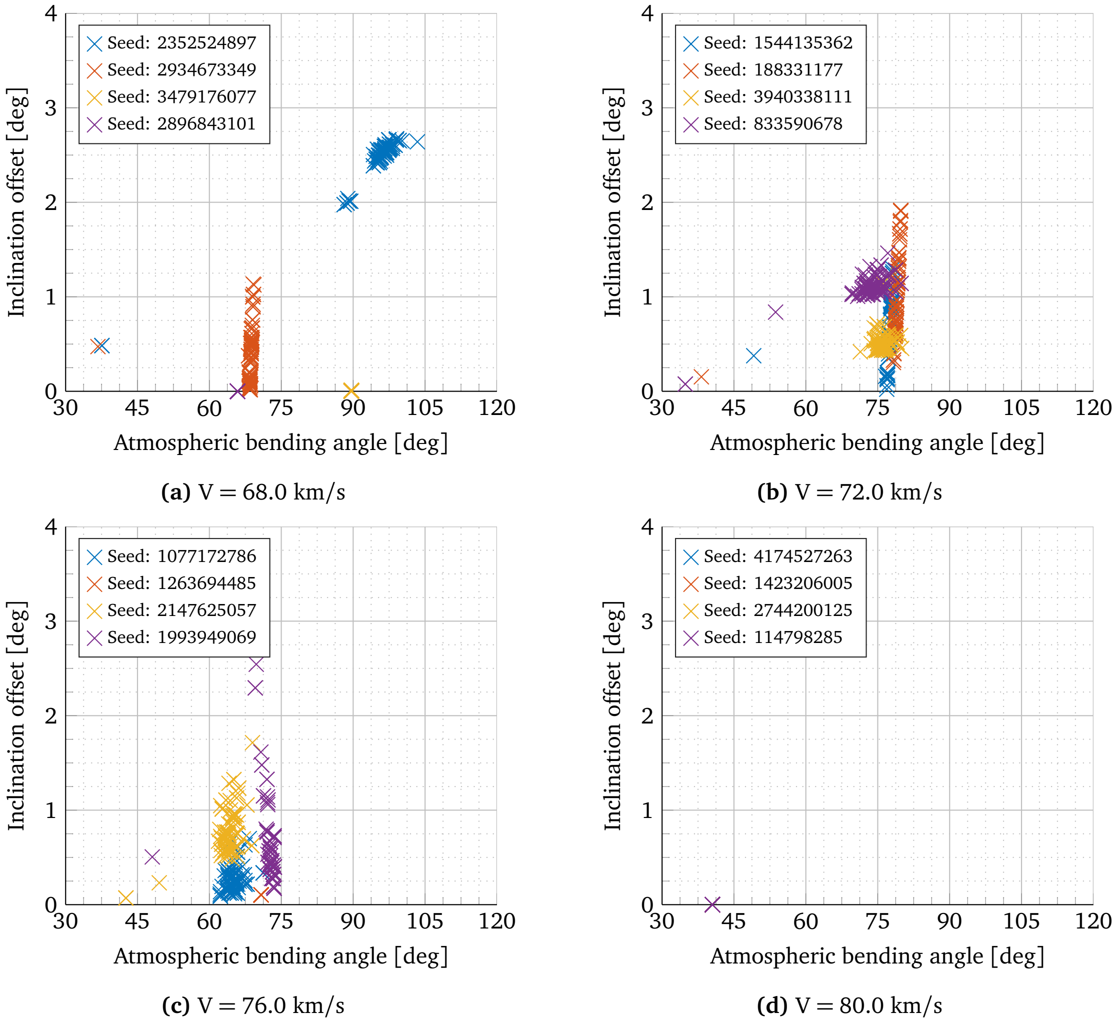
(f) 3D trajectory

Figure 7.8: Found trajectories for four different initial velocities around Mars

km/s; the lowest initial velocity. Due to the lower velocities involved in such a trajectory, the initial heat flux peak that will be experienced is lower as well, as can be seen in Fig. 7.10d. However, at an atmospheric bending angle of approximately 40° for all four random seeds, a second heat flux peak occurs due to a decrease in altitude, which is constraining the trajectory. At the same moment, a peak in the g-load occurs as well as shown in Fig. 7.10c, although this peak is still below the constraint of 15g. The final eccentricity of this trajectory is 1.5, as shown in Fig. 7.10b, which sufficiently large to leave the planetary sphere of influence.

To increase the maximum atmospheric bending angle that can be achieved at Jupiter, the initial velocity could be decreased even further such that the heat flux is decreased. However, the expected gain from decreasing the initial velocity, however, is minimal since the eccentricity of an orbit with a lower initial velocity will be lower as well. This will mean that the optimization will be constraint by the requirement for an aerogravity assist to have a final eccentricity larger than 1. Since the final eccentricity of the trajectory with an initial velocity of 68.0 km/s is 1.5 and therefore close to an highly elliptical orbit, it is expected that lowering the initial initial velocity will, if any, have only a marginal advantage.

As the initial velocity increases, the initial heat flux the vehicle experiences when the atmosphere is entered increases as well. From Fig. 7.9, it can be observed that the atmospheric bending angle for an initial velocity of 80.0 km/s does not exceed 40°. This was the case for all the performed optimization runs with this initial velocity. The reason the atmospheric bending angle remains rather small becomes obvious from Fig. 7.10d: the heat flux peak that occurs at this initial velocity is violating the heat flux constraint of 6000 W/cm^2. Trajectories that have a larger bending angle, would require a steeper entry, which in



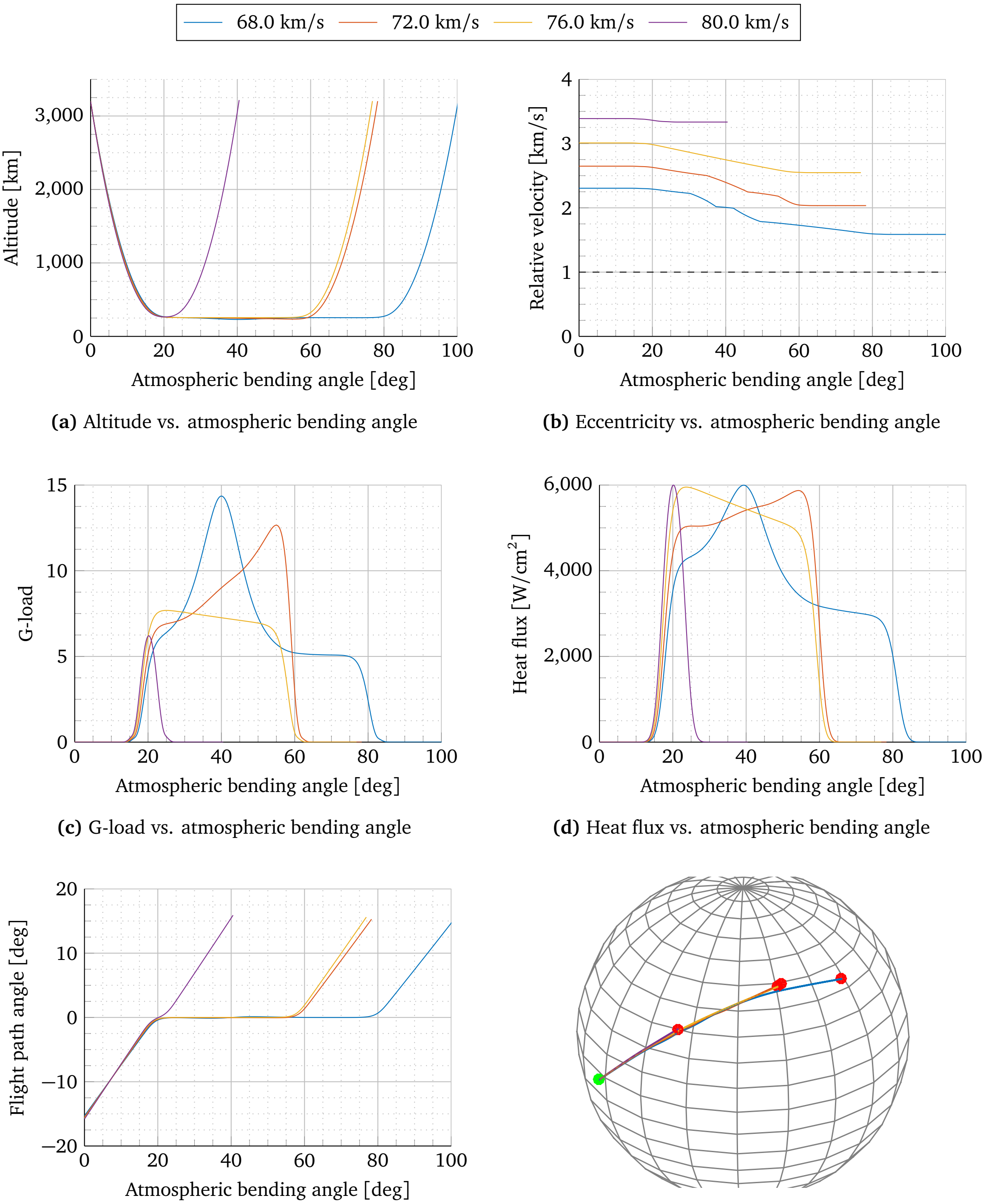
(a) V = 68.0 km/s

(b) V = 72.0 km/s

(c) V = 76.0 km/s

(d) V = 80.0 km/s

Figure 7.9: Pareto fronts after optimization for Jupiter



(a) Altitude vs. atmospheric bending angle

(b) Eccentricity vs. atmospheric bending angle

(c) G-load vs. atmospheric bending angle

(d) Heat flux vs. atmospheric bending angle

(e) Flight-path angle vs. atmospheric bending angle

(f) 3D trajectory

Figure 7.10: Found trajectories for four different initial velocities around Mars

turn results in a larger initial heat flux peak. Since this peak occurs at a point along the trajectory where the atmospheric density is not large enough the provide sufficient lift, the trajectory is constraint to an atmospheric bending angle of 40°.

The g-load constraint for all trajectories is not a constraining factor; the largest g-load occurred at an initial velocity of 68.0 km/s, where the g-load is just over 14g. This is the result of a slightly deeper dip into the atmosphere compared to the other trajectories, where the atmospheric density and therefore the aerodynamic forces are larger. Such peaks can be reduced by imposing tighter constraints, which is discussed in Section 7.8.

As can be seen in Fig. 7.10f, most of the four trajectories do not remain in the same orbital plane, even though the final inclination is almost equal to the arrival inclination. Compared to the trajectories found for Mars in Fig. 7.8f, the trajectories remain much closer to the original orbital plane than the trajectories at Mars. The origin of this different departure orbital plane is the same: the aerodynamic force results in a shift of the RAAN. In Section 7.9, a method which ensures that the trajectory remains in the same orbital plane will be discussed.

The results found during the trajectory optimization are compared with a gravity assist for the same initial conditions in Table 7.3. It was possible to increase the atmospheric bending angle by 143% for the lowest entry velocity, but this gain decreases as the entry velocity increases. For the largest entry velocity, 80.0 km/s, the velocity bending angle could only be increased by a mere 10% as this trajectory was constrained by the large heat fluxes. This entry velocity has only one peak at the maximum heat flux, while a trajectory that follows this heat-flux constraint could be expected. The reason that trajectories which would follow the maximum heat-flux constraint were not found, is that once a trajectory would violate the heat-flux constraint, it was flagged and it was excluded from the optimization. This means that any information from trajectory that would exceed the heat-flux constraint cannot be used to find this trajectory along this constraint. In hindsight, it would have been better to just penalize trajectories that would exceed the heat-flux or g-load constraints instead of removing the results from the optimization. This way, the optimization algorithm could still use the results from "unfeasible" trajectories to find a better optimum.

Compared to the Mars trajectories found in Table 7.2, the observed bending angles for gravity assists is larger, which is what is expected as Jupiter's mass is much larger than the mass of Mars. On Mars, however, atmospheric maneuvering allowed to increase the velocity bending angle by a much larger amount than on Jupiter. Therefore, when taking into consideration the feasible velocity bending angles and the heat loads, Mars is a much better candidate for performing an aerogravity assist than Jupiter.

Table 7.3: Comparison of gravity assist (GA) and aerogravity assist (AGA) at Jupiter

	Variable	Unit	GA	AGA	% Difference
	δ	deg	51.43	125.1	143.3
68.0 km/s	V_∞^+	km/s	48.52	32.53	−32.95
	i	deg	44.50	42.35	−4.83
	δ	deg	44.37	90.86	104.8
72.0 km/s	V_∞^+	km/s	54.53	43.21	−20.76
	i	deg	44.88	45.31	−1.89
	δ	deg	38.82	81.43	109.77
76.0 km/s	V_∞^+	km/s	60.21	52.84	−12.25
	i	deg	45.23	44.38	−1.89
	δ	deg	34.33	38.06	10.86
80.0 km/s	V_∞^+	km/s	65.64	64.90	−1.14
	i	deg	45.55	45.00	−1.21

7.8 Influence of tighter constraints

The trajectory optimization that was performed so far in this section, had a maximum g-load of 15g and a maximum heat flux of 500 W/cm^2 for Mars, and 6000 W/cm^2 for Jupiter. To investigate the impact of an adjustment of the constraints, the constraints were made tighter for the trajectory optimization at Mars with an initial velocity of 9.0 km/s. The trajectory optimization was performed with a g-load constraint of 5g and a heat flux constraint of 200 W/cm^2. In Fig. 7.11, the original Pareto front of the trajectory optimization with an initial velocity of 9.0 km/s is compared with the Pareto front for the optimization with tighter constraints. Again, the optimization was repeated multiple times (seeds 534155203, 3712450573, 3549753428, 1625810129), and the solution with the largest atmospheric bending angle has been plotted in Fig. 7.11.

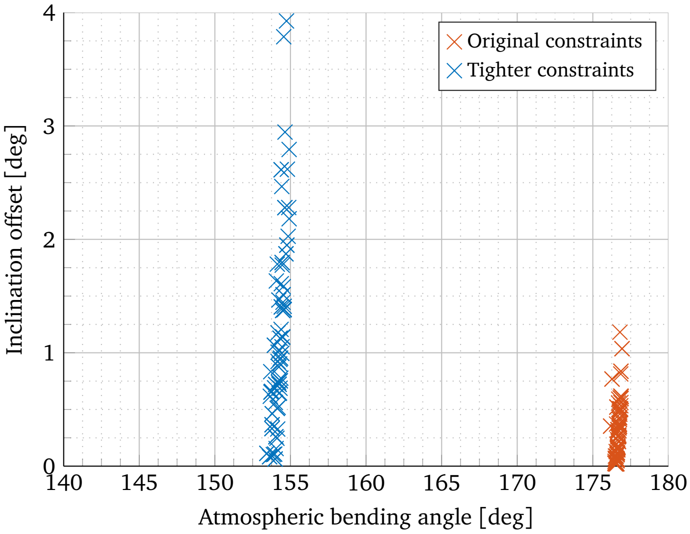


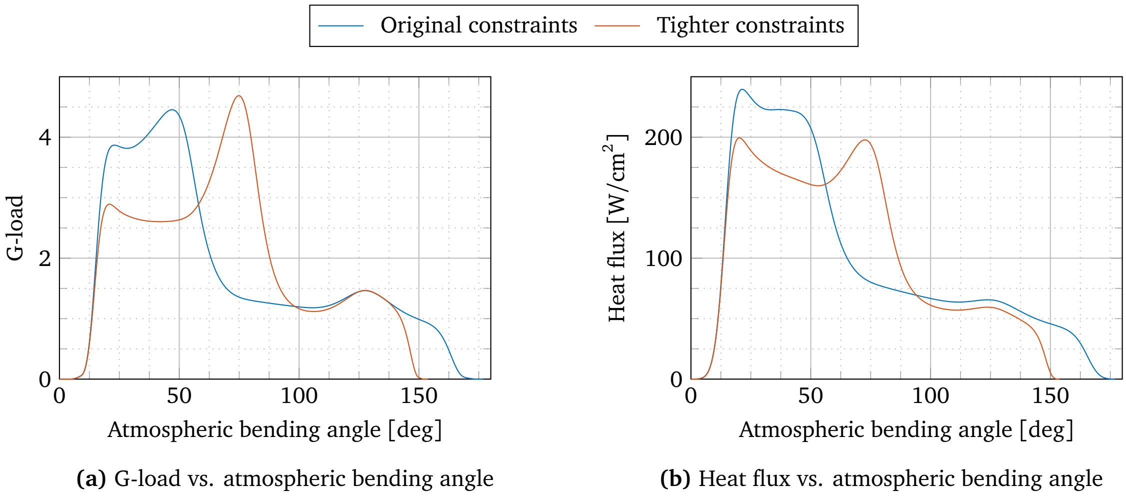
Figure 7.11: Pareto front comparison for a Mars trajectory with an initial velocity of 9.0 km/s with original constraints (seed: 4093800575) and tighter constraints (seed: 3712450573)

From Fig. 7.11, it becomes clear that when tighter constraints are used, the atmospheric bending angle is less than when the original constraints were used. To asses the origin of this reduction in atmospheric bending angle, the heat flux and g-load of one of the trajectories from the Pareto front using the tighter constraints is compared with the solution for an initial velocity of 9.0 km/s from Fig. 7.8. The heat flux and g-load for both trajectories are shown in Fig. 7.12

The reduction of the g-load constraint did not have an influence on the optimization, since the g-load of the original trajectory was well below the constraint value of 5g. In fact, the g-load becomes at one point slightly larger than the original trajectory. This indicates the the original g-load constraint is not affecting the trajectory. In Fig. 7.8, especially for higher velocity, it can be seen that the g-load peaks correspond to somewhat deeper plunges in the atmosphere. Lowering the g-load constraint can prevent the trajectory optimization from finding solutions where these plunges occur.

The heat flux is the reason that the atmospheric bending angle of the new trajectory is smaller: at two points along the trajectory, the magnitude of the heat flux becomes equal to the heat flux constraint of 200 W/cm^2. From Fig. 7.8d, it can be seen that the peak heat flux often occurs during the capture phase where the vehicle enters the atmosphere. Reducing this peak heat flux by imposing a tighter constraint required the vehicle to enter the atmosphere at a more shallow flight-path angle (−14.806° vs. −14.825° for the trajectory original trajectory), which in turn results in a lower atmospheric bending angle.

Next to the physical impact of imposing a tighter constraint, it also has influence on the trajectory optimization. By reducing the heat flux constraint, the optimization algorithm has more difficulty finding trajectories with a larger atmospheric bending angle. The trajectories where the peak heat flux during the capture phase exceed the heat flux constraint are flagged as an unfavorable solution, so the optimization algorithm will not use this trajectory to find solutions in the next generation. In other words, the optimization ignores some information that could lead to a better solution, which result in slower or premature convergence. This



(a) G-load vs. atmospheric bending angle

(b) Heat flux vs. atmospheric bending angle

Figure 7.12: Comparison of heat flux and g-load for original and tighter constraints

is indeed what was observed; only one of the five evaluated random seeds resulted in a reasonable bending angle, while the other four converged to a solution with a very small atmospheric bending angle. As was already concluded in Section 7.7, it would be better to add a penalty to solutions that exceed the constraints such that the information from these "unfeasible" solutions can still be used for the next generation.

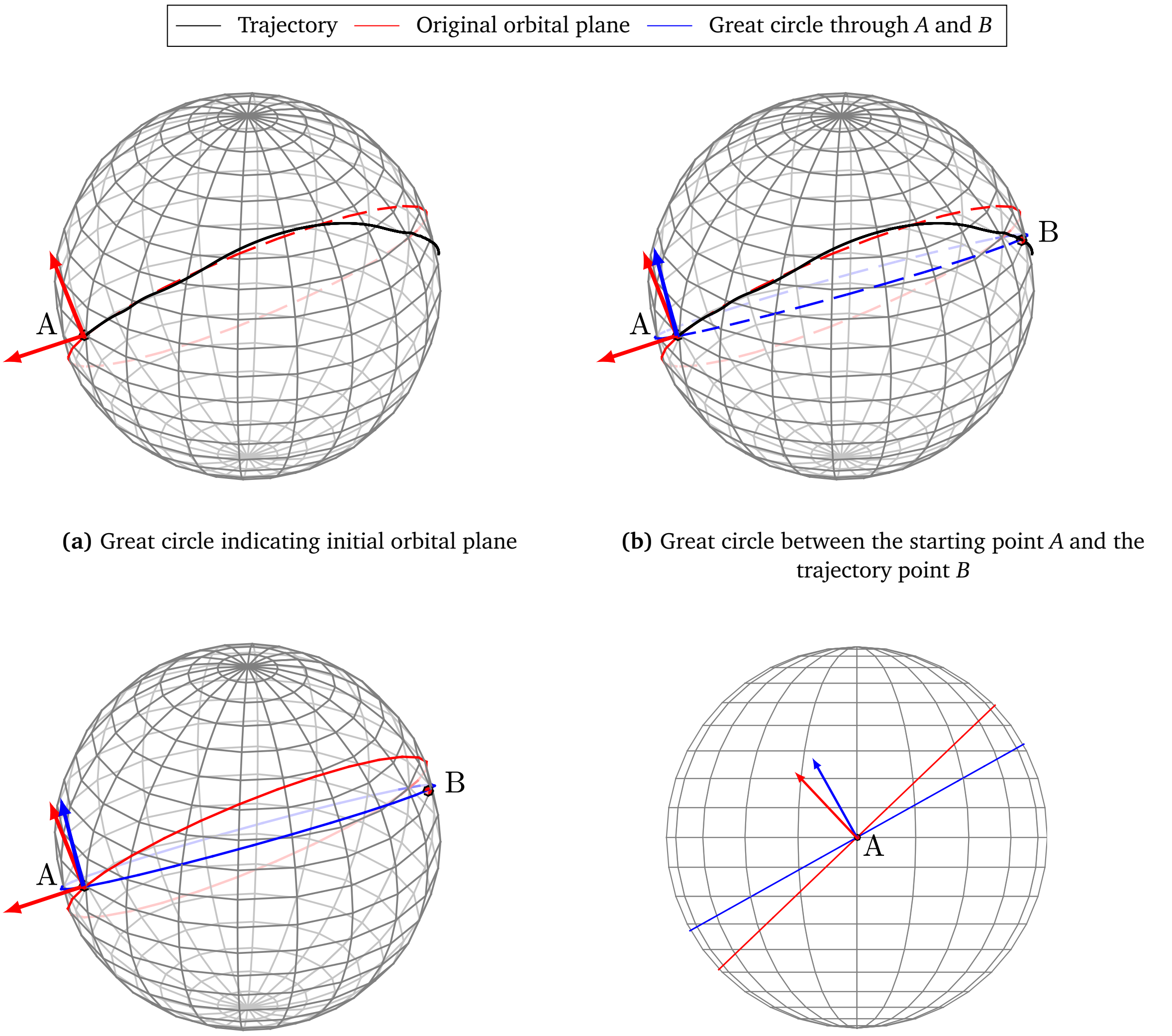
7.9 Orbital plane control

From the analysis in Section 7.6, it became clear that steering on the orbital inclination only does not ensure a trajectory after the atmospheric maneuver that is in the same orbital plane as the arrival orbital plane due to the rotation of the ascending node. Since steering on inclination is insufficient, a different approach should be used to keep the departing branch of the trajectory in the same orbital plane as the arriving branch of the trajectory. Two proposals were made to achieve this.

- The first method makes use of great circles. A great circle is defined as a circle on the surface of a sphere that divides this sphere in two equal hemispheres. Consider the trajectory with an initial velocity of 8.0 km/s from Fig. 7.8f, denoted by the black line in Fig. 7.13a. The angular momentum vector ($\boldsymbol{r}_A \times \boldsymbol{V}_a$) at the starting point A of this trajectory defines the initial orbital plane, indicated by the red great circle. Now a second point along the ground track of the trajectory is selected, denoted by the label B. The points A and B can be used to draw a second great circle, indicated by Fig. 7.13b. For clarity, the two great circles are also shown in Fig. 7.13c without the trajectory. In Fig. 7.13d, the angle between the two different planes becomes obvious. This angle is defined as the pseudo-inclination. If instead of the Keplerian inclination, the angle between the two planes in Fig. 7.13d will be used as steering parameter to control the orbital plane should assure that the departing branch of the trajectory is in the same orbital plane as the arrival branch of the trajectory.

- The second method is based on adding a third variable to the fitness variable. This variable is the offset angle in the right ascension between the ascending node of the arriving branch of the trajectory and the departing branch of the trajectory. When this offset is zero, the departing branch of the trajectory lies exactly in the same plane as the arriving branch. The steering parameter in this approach is still the orbital inclination. Although this might appear an easier approach to implement that the first method, it requires the implementation of a third fitness variable, which causes the optimization to slow down. As the trajectory optimization is already time-consuming, slowing down the optimization even further is not preferred.

For both methods, several optimization runs were performed to check the if the proposed methods do actually work. For the first method, a trajectory resulting from trajectory optimization is shown in Fig. 7.14a; for the second method, the trajectory is shown in Fig. 7.14b.



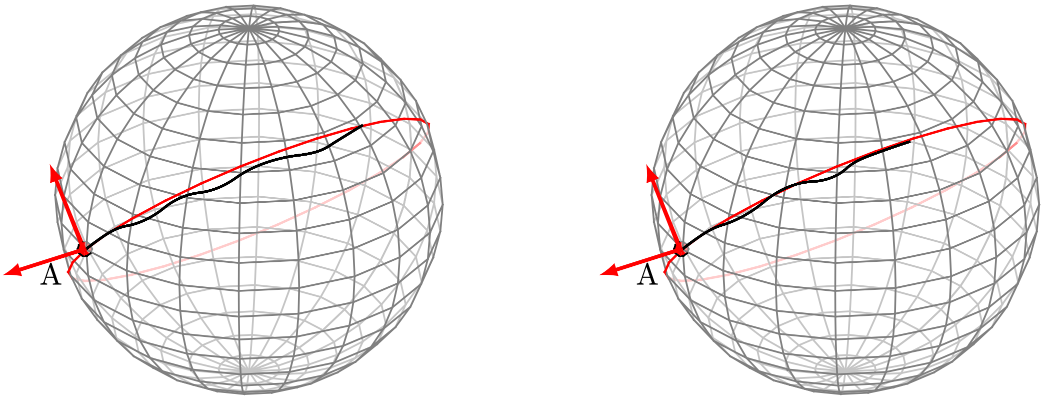
(a) Great circle indicating initial orbital plane

(b) Great circle between the starting point A and the trajectory point B

(c) Unobstructed view of the two great circles

(d) Side view of the two great circles

Figure 7.13: Proposed definition of the pseudo-inclination



(a) Pseudo-inclination control

(b) RAAN control

Figure 7.14: Results of the two different orbital plane control methods

A great circle made between the end point and the starting point of the trajectory that made use of the first method is indeed the same as the initial orbital plane. However, it is clear from Fig. 7.14a that if the trajectory would be propagated for a longer period of time, the orbital plane would not be identical to the initial orbital plane. It was also observed that when a bank reversal was performed once the angle between the two great circles became too large, this angle would not decrease afterwards. This was the case when only the inclination was used as a steering parameter: after a bank reversal, the inclination difference would decrease again. From the results in Fig. 7.14a, it can be concluded that the method based on great circles does not work as expected. The reason for this is that the pesudo-inclination method only takes into account the position of the vehicle, while the velocity of the outgoing branch of the trajectory should also be in the same plane the incoming branch.

The method where the offset in right ascension of the ascending node was minimized as a third optimization objective, does appear to keep the trajectory in the same orbital plane. This is also something one could expect since the ascension of the ascending node and the inclination determine the orientation of the plane. However, adding the third objective resulted in worse convergence. It is therefore recommended to reevaluate the population size and the tuning parameters of the algorithm for the optimization problem with three objectives. However, due to time constraints, this will be left for further research and in the remainder of this thesis, the inclination will still be considered as the only control parameter for the control of the orbital plane.

8 Mission Analysis

To determine the effect an aerogravity assist could have, a mission has to be defined. As became clear from the analysis performed in Chapter 7, there are many different reference trajectories possible. It would be impossible to explore every possible trajectory through an atmosphere, and its impact on the heliocentric velocity and inclination. To still be able to make a quantitative analysis of the influence of an aerogravity assist, a mission was defined. Since an aerogravity assist is considered to be an improvement to a regular gravity assist, two gravity assists that have been performed during a real interplanetary missions have been selected: Rosetta's swing-by at Mars and Ulysses' swing-by at Jupiter.

8.1 Rosetta swing-by at Mars

On 2 March 2004, the European Space Agency (ESA) launched the Rosetta probe. The key objective of the Rosetta mission was to approach the comet 67P/Churyumov-Gerasimenko. The journey took ten years, and required three gravity assist at Earth and one at Mars to obtain sufficient orbital energy to rendezvous with the comet. Once arrived at the comet, a lander called Philae descended towards the surface of the comet to perform scientific measurements (Munoz et al., 2012).

To determine how the entire Rosetta mission could have benefit from the use of aerogravity assists would be beyond the scope of this thesis. However, to investigate the possible benefit of an aerogravity assist, one of the gravity assists performed by Rosetta is used, i.e., the one around Mars, will be used as a baseline. Using almost the same initial conditions as the gravity assist, slightly modified to ensure the vehicle enters the Martian atmosphere, the reference trajectory planner will generate a bank-angle profile that results in an aerogravity-assist trajectory. Using this generated reference trajectory, the tracking algorithm is used to determine if the found trajectory is indeed achievable.

The trajectory of Rosetta can be downloaded from the ESA website[1], and is contained in a SPICE kernel. The TU Delft Astrodynamics Toolbox (Tudat) contains an interface with SPICE, which makes it possible to determine the Cartesian state of a planetary body or vehicle with respect to any body in the Solar System, given that the appropriate kernels are loaded. In addition to Rosetta's SPICE kernel, the DE413 kernel was required to determine the position of Mars.

The Mars gravity assist took place on 25 February 2007, at which Rosetta approached the Martian surface to approximately 250 km. The Martian gravity assist was used to decelerate Rosetta: the heliocentric velocity was reduced by 2.32 km/s. The heliocentric inclination remained more or less constant: 26.97° before the gravity assist and 25.13° after the gravity assist.

Now for the aerogravity assist mission that will be based on this Rosetta swing-by, many different objectives are possible. For example, one could want to achieve a specific atmospheric bending angle, adjust the heliocentric inclination to a particular target, or increase or decrease the heliocentric velocity. Since investigating all these different objectives is not feasible within the allocated time for this thesis work, one objective was selected to assess the influence an aerogravity assist. For Rosetta, the goal of the gravity assist at Mars was to decrease its heliocentric velocity. Using slightly modified initial conditions, that ensures the periapsis of the trajectory is lowered into the atmosphere, it will be investigated what the velocity decrease could have been if use was made of an aerogravity assist. In addition, the waverider shape that was used

[1] SPICE for Rosetta: `http://www.cosmos.esa.int/web/spice/spice-for-rosetta`, accessed: 2 December 2015

for the analysis for Mars in Chapter 7 has also been used for the analysis of Rosetta. This means that the actual Rosetta mission could not have flown this proposed aerogravity assist, even if its arrival conditions at Mars were modified. The initial condition that will be used are defined in the local horizontal reference frame and are listed in Table 8.1.

Table 8.1: Modified initial conditions for Rosetta's swing-by around Mars

Parameter	Adjusted	Original
Altitude [km]	162.6	421.7
Velocity [km/s]	10.27	10.27
Longitude [deg]	−37.90	−33.50
Latitude [deg]	45.07	41.99
Heading angle [deg]	−78.05	−77.04

Using the reference trajectory planner, discussed in Chapter 7, an aerogravity assist trajectory was found that minimized the heliocentric velocity after the maneuver. It has to be noted that this objective is different from the one in Chapter 8, where the main objective was to maximize the atmospheric bending angle. Where the maximum atmospheric bending angle gives the trajectory that travels the largest distance through the atmosphere, this does not ensure that the heliocentric velocity is indeed minimized.

Since the initial conditions of the Rosetta trajectory are modified slightly, comparing the aerogravity assist trajectory with the original Rosetta trajectory might result in some differences that are not caused by the atmospheric maneuver. To make this a fair comparison, a gravity assist trajectory is computed with the initial conditions from Table 8.1, but with no atmosphere. By disabling the atmosphere, a gravity assist is the result since the trajectory is not affected by any atmospheric forces.

Figure 8.1 shows how the trajectory, and therefore also the velocity vector, is bent by both the gravity assist and the aerogravity assist. Where for a gravity assist, the velocity vector is rotated over an angle of almost 16°, the atmospheric maneuver increased this angle to 65° for an aerogravity assist. The atmospheric maneuver, however, comes at a price: the atmospheric drag results in a decrease in magnitude of the departure velocity. The magnitude of the hyperbolic excess velocity of the arriving trajectory is $V_\infty^- = 8.829\,\text{km/s}$, whereas this magnitude is decreased to $V_\infty^+ = 7.558\,\text{km/s}$

The reduction in drag, however, is not the reason the reduction of the heliocentric velocity is increased compared to a regular gravity assist. Without using the atmosphere, the heliocentric velocity was reduced by 2.308 km/s for the gravity assist, while for the aerogravity assist, the heliocentric velocity was decreased by 6.160 km/s, an increase of 167%.

The atmospheric phase of the aerogravity assist is highlighted in Fig. 8.2. Fig. 8.2a shows that the vehicle

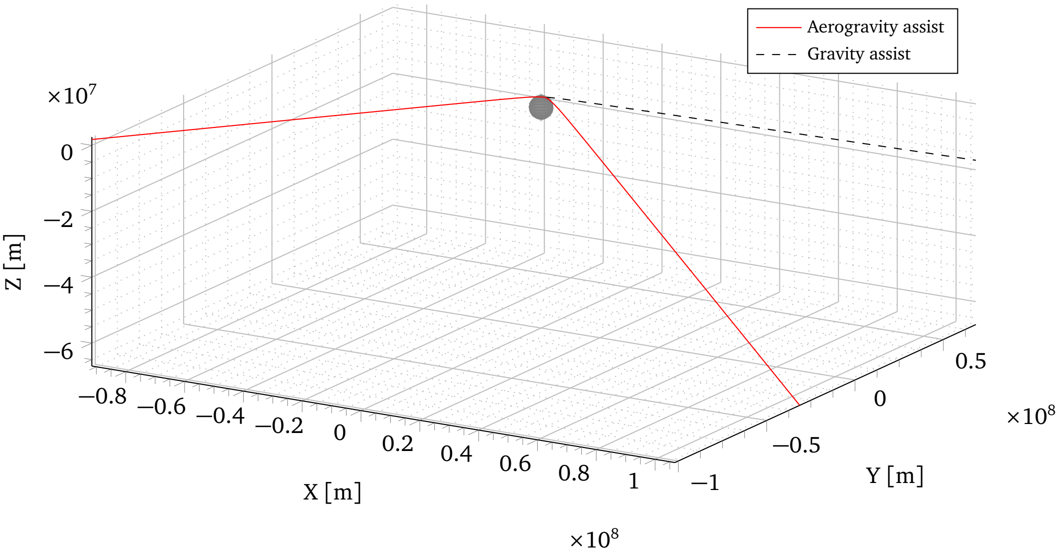
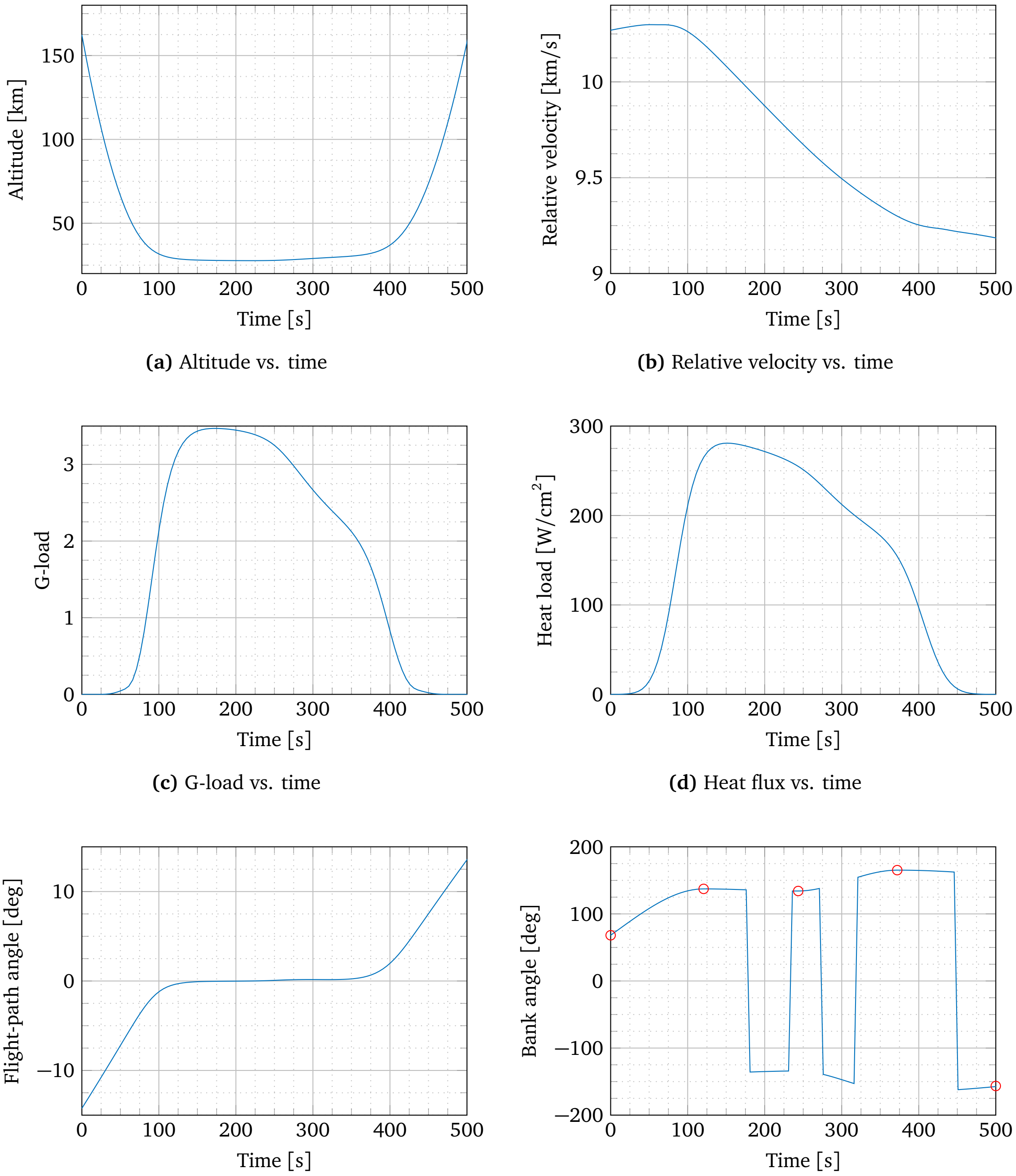


Figure 8.1: Trajectories of Rosetta for both a gravity assist (black) and an aerogravity assist (red) around Mars

will travel at a more or less constant altitude for approximately 300 s. This is also visible in Fig. 8.2e, where the flight-path angle remains close to 0° for the same interval. The velocity (Fig. 8.2b) decreases as a result of energy dissipation due to drag. With a maximum mechanical load of 3.5g (Fig. 8.2c) and a maximum heat load of 280 W/cm^2 (Fig. 8.2d), the found trajectory is well within the imposed constraints of 500 W/cm^2 and 15g. Furthermore, it can be seen in Fig. 8.2f that the optimization algorithm used five bank reversals to keep the trajectory in the same orbital plane.



(a) Altitude vs. time

(b) Relative velocity vs. time

(c) G-load vs. time

(d) Heat flux vs. time

(e) Flight-path angle vs. time

(f) Bank angle vs. time

Figure 8.2: Aerogravity assist trajectory for Rosetta around Mars

8.2 Ulysess swing-by at Jupiter

Ulysses was launched in October 1990, with the goal to measure the heliospheric field and particles perpendicular to the solar equatorial plane (Smith and Marsden, 1995). The mission was a collaboration between the European Space Agency (ESA) and the National Aeronautics and Space Administration (NASA). ESA was responsible for the design and construction of the spacecraft, while NASA was responsible for the launch and tracking. Since this scientific goal requires that the probe performs measurements at higher heliographic latitudes, an heliocentric inclination change was necessary.

The trajectory for the Ulysses data can be downloaded from NASA's Jet Propulsion Laboratory website[2]. The Jupiter swing-by occurred at 8 February 1992. In addition to Ulysses SPICE kernel, the DE421 kernel was loaded to determine the position of Jupiter with respect to the Sun.

Heliocentric inclination changes, especially the ones required for the Ulysses mission, are not possible to obtain by the launch vehicle. Instead, a gravity assist was performed at Jupiter. Due to the large mass, it was possible to increase the heliocentric inclination by 80°, which made it possible to observe heliocentric latitudes from $-80°$to $+80°$. The gravity assist, in addition to changing the heliocentric inclination, decreased the heliocentric velocity of Ulysses by 7.4 km/s.

The Ulysses mission has been selected as a baseline for a Jupiter gravity assist because the heliocentric inclination change was the main objective of the maneuver. To see how well the orbital inclination of a vehicle could be adjusted using an aerogravity assist, a trajectory based on the Ulysses mission is proposed. First of all, consider the original trajectory of Ulysses, as depicted by the blue line in Fig. 8.3. Ulysses' hyperbolic orbit has an inclination of approximately 140° with respect to Jupiter, which resulted heliocentric inclination change of 80°. The mission that is defined assumes that the arrival trajectory of Ulysses lies in the equatorial plane of Jupiter, such that the influence of the swing-by on the heliocentric inclination is minimal. This orbit is represented by the red trajectory in Fig. 8.3. Now this trajectory does not enter the atmosphere, so atmospheric maneuvering will not be possible. Instead, the semi-major axis was increased from -67615 km to -10239 km, such that the periapse of the trajectory would be within the atmosphere, illustrated by the yellow trajectory in Fig. 8.3.

The initial conditions of the modified orbit are listed in Table 8.2. The initial velocity of this trajectory is larger than 80.0 km/s. From the analysis in Section 7.7, it was concluded that at an initial, planetocentric velocity of 80.0 km/s, the reference trajectory planner was not able to find trajectories due to the high peak heat flux that occurred during the entry phase. To avoid this, the heat flux constraint is removed for the trajectory planning of Ulysses, while maintaining the mechanical load constraint of 15g. Although removing this constraint will result in heat loads that cannot be withstood by any thermal protection system, the change of orbital inclination due to lateral atmospheric maneuvers can still be studied.

The trajectory obtained from the trajectory planning, where achieving the maximum inclination change was set as the objective, is shown in Fig. 8.4. It is clear from this figure that the aerodynamic force is capable of lifting the vehicle out of its original orbital plane. The dashed, black line shows trajectory in absence of

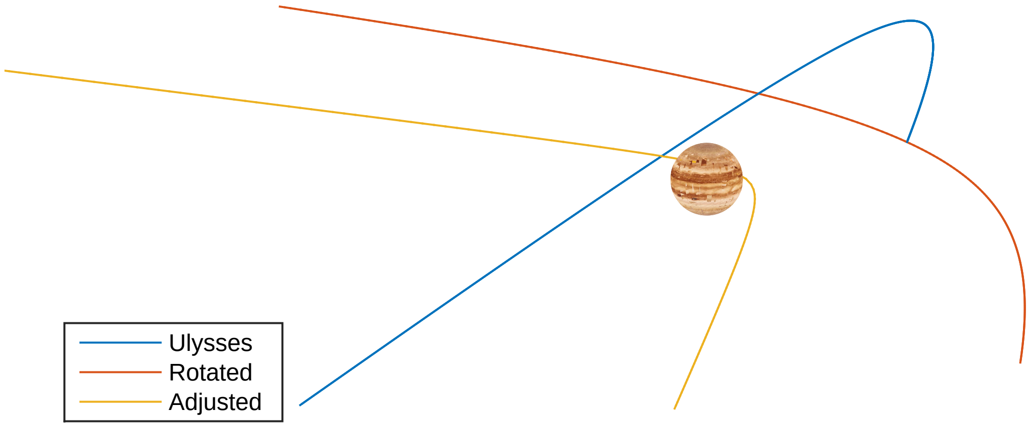


Figure 8.3: Adjustments to Ulysses' orbit

[2] SPICE for Ulysses: `http://naif.jpl.nasa.gov/pub/naif/ULYSSES/kernels/spk/`, accessed: 2 December 2015

Table 8.2: Modified initial conditions for Ulysses' swing-by around Jupiter

Parameter	**Adjusted**	**Original**
Altitude [km]	3200	3.112×10^6
Velocity [km/s]	81.16	553.1
Longitude [deg]	109.1	−21.81
Latitude [deg]	0.00	10.75
Heading angle [deg]	−90.00	−89.75

an atmosphere, where the vehicle remains in the equatorial plane. Due to the aerodynamic forces, however, the inclination of the orbital plane was changed by approximately 50°, indicated by the red trajectory.

The atmospheric phase of the found trajectory is shown in Fig. 8.5. The vehicle will travel at constant altitude for approximately 700 s as is shown in Fig. 8.5a. As a result of the aerodynamic drag, the vehicle's velocity is decreased, which is shown in Fig. 8.5b. The mechanical load constraint was a limiting factor during the reference trajectory planning, which can be seen in Fig. 8.5c. After approximately 900 s, the maximum mechanical load of 15g is reached. As expected, the head load very large: 40, 620 W/cm^2 (Fig. 8.5d). Even though the loads on the vehicle are large, an inclination change of 54.2° was achieved by reducing the orbital inclination from 180° to 125.8° (Fig. 8.5e).

In contrast to other reference trajectories, no bank reversals are used as can be seen in Fig. 8.5f. The bank angle starts at approximately 150°, where the vertical lift component is the largest. As the trajectory enters the atmosphere, energy is dissipated due to drag and as a result, the velocity decreases as well (see Fig. 8.5b). Although a lower velocity directly results in a lower lift force, the required lift force to increase the centripetal force also become less due to the dissipation of energy as a result of the atmospheric drag. As the magnitude of the required bank angle is lowered, it can be concluded that less vertical lift is required to maintain a level flight. The added benefit of reducing the vertical lift is that the cross-track component of the lift increases, which allows an increase in the rate of change of the inclination. This is confirmed by the slope of Fig. 8.5e, which slowly increases over time.

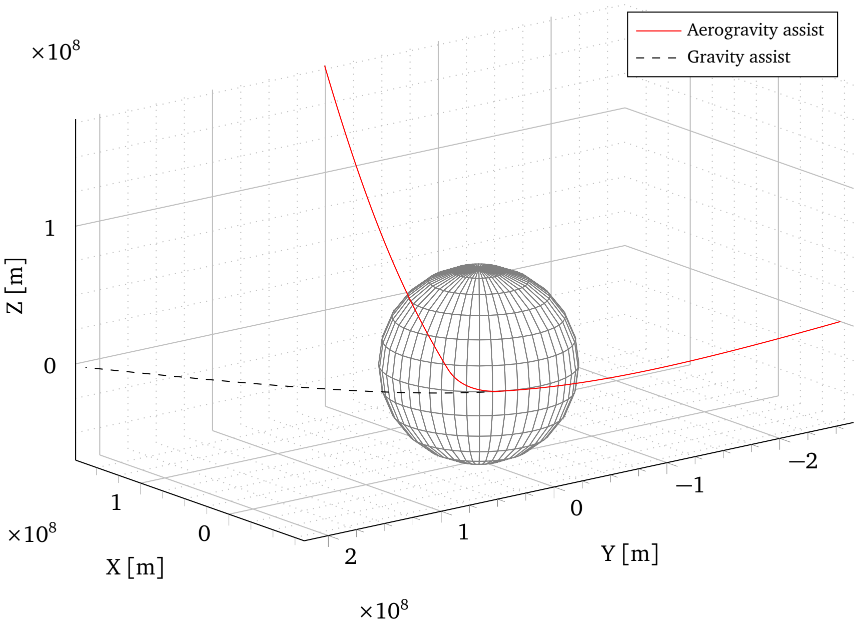
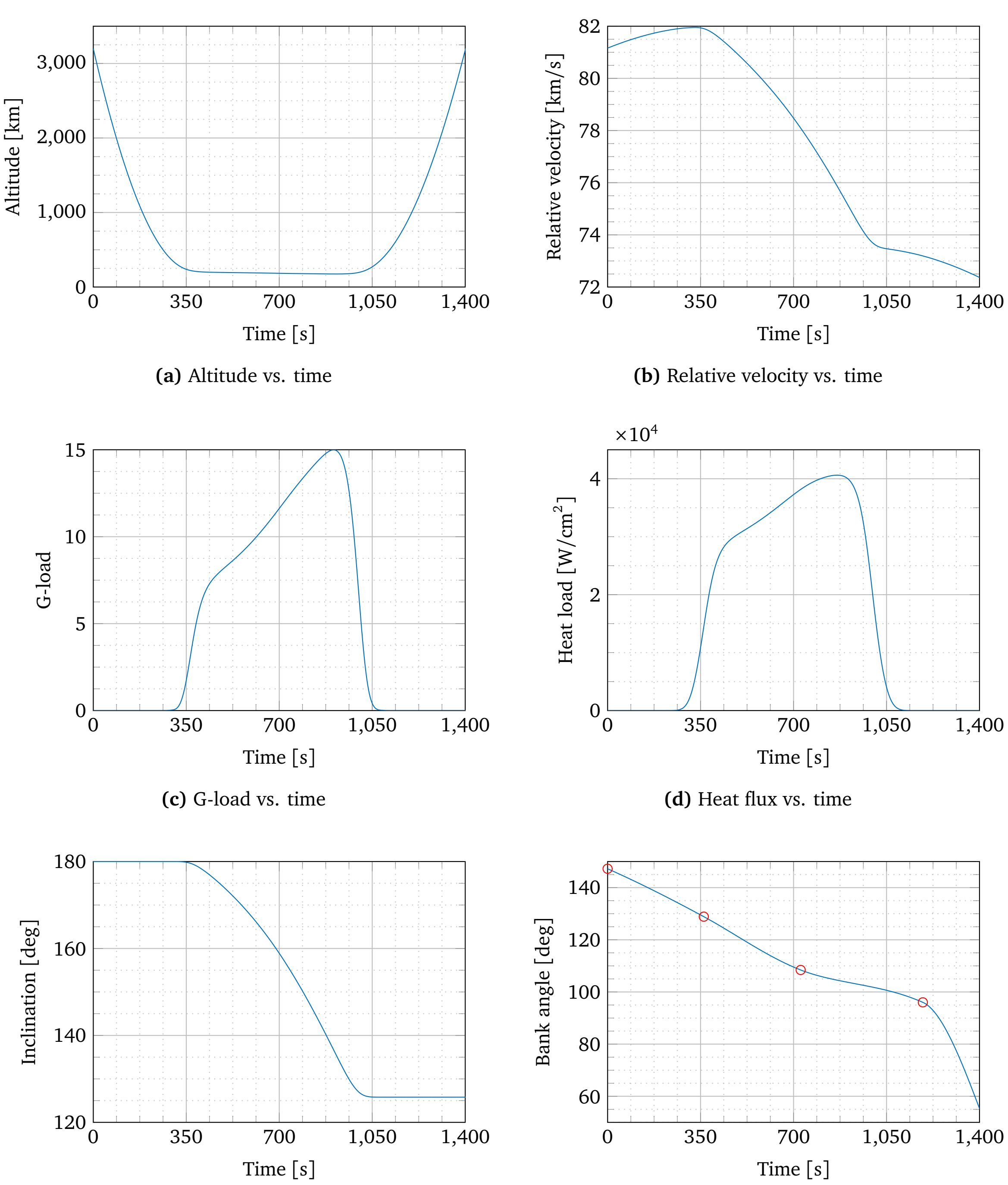


Figure 8.4: Trajectories of Ulysses for both a gravity assist (black) and an aerogravity assist (red) around Jupiter



(a) Altitude vs. time

(b) Relative velocity vs. time

(c) G-load vs. time

(d) Heat flux vs. time

(e) Flight-path angle vs. time

(f) Bank angle vs. time

Figure 8.5: Aerogravity assist trajectory for Ulysses around Jupiter

9 Guidance Algorithm

The reference trajectories found in Chapter 7 are based on nominal conditions. Due to the high velocities involved in an aerogravity assist maneuver, any uncertainties in the atmospheric density, aerodynamic coefficients or initial conditions can result in a very different trajectory. To prevent this, a robust guidance algorithm should be designed to ensure the vehicle will follow the proposed reference trajectory as good as possible. The guidance algorithm proposed for an aerogravity assist presented in this chapter consists of two major elements: longitudinal guidance (Section 9.1) and lateral guidance (Section 9.2). Finally, in Section 9.3, the guidance algorithm is applied to a reference trajectory to study the effect of varying initial conditions and perturbations.

9.1 Longitudinal guidance

For the longitudinal guidance, a drag-tracking algorithm is considered. Such an algorithm, as was proposed by Saraf et al. (2004), uses a reference trajectory from which a drag reference profile D_{ref} is obtained. The drag-tracking algorithm, which will be shown below, tries to match this drag reference profile by determining a value for the bank angle. The required reference trajectory that will be used, was generated using the reference trajectory planning discussed in Chapter 7 and is the Rosetta aerogravity assist discussed Section 8.1.

Let $e = D_{ref} - D$. Saraf et al. (2004) modeled the guidance law such that, in absence of modeling errors, second-order linear drag error dynamics would be achieved. A standard second-order model for the error $e = D_{ref} - D$ is:

$$\left(D''_{ref} - D''\right) + 2\zeta\omega\left(D'_{ref} - D'\right) + \omega^2\left(D_{ref} - D\right) = 0 \tag{9.1}$$

where ω is the natural frequency and ζ is the damping ratio of the second order system. In Eq. (9.1), the derivatives are defined with respect to energy instead of time, as shown by:

$$D' = \frac{dD}{dE} \tag{9.2}$$

where the total energy E of the vehicle is defined as:

$$E = \frac{V_I^2}{2} - \frac{\mu}{r_I} \tag{9.3}$$

in which r_I and V_I are the radial distance and inertial velocity of the vehicle in the inertial planetocentric reference frame, and μ is the gravitational parameter of the central body.

To be able to use Eq. (9.1) to determine the bank angle, all variables should be known, such that this equation can be solved for the bank angle σ. Obviously, ζ and ω are design variables that control the response of the error dynamics. The variables D'_{ref} and D''_{ref} can be obtained from the reference trajectory. This means that D, D, and D'' need to be known. Both the drag D and the derivative of the drag with respect to energy $\dot{D}$ can be determined from analytical expressions that are derived from the equations of motion, which is shown in Appendix B. The second derivative of drag with respect to energy D'' is a function of the bank angle, which is the variable that needs to be estimated. Saraf et al. (2004) provides an expression for D'':

$$D'' = a + bu_D \tag{9.4}$$

in which

$$u_D = (L/D)\cos\sigma \tag{9.5}$$

where L and D are the lift and drag accelerations, σ is the required bank angle, and a and b are two variables that are a function of the current flight parameters. The variables a and b can be obtained by taking the derivative of D (given by Eq. (2.12)) with respect to energy twice. This will result in the following expressions for a and b:

$$a = D\left(\frac{C_D''}{C_D} - \frac{C_D'^2}{C_D^2}\right) + \dot{D}\left(\frac{\dot{C}_D}{C_D} + \frac{2}{V^2}\right) - \frac{4D}{V^4} + \frac{1}{DV^2}\left(\frac{1}{h_s} + \frac{2g}{V^2}\right)\left(g - \frac{V^2}{r}\right) + \left(\frac{1}{h_s} + \frac{2g}{V^2}\right)\mathscr{C}_\gamma \tag{9.6}$$

$$b = -\frac{1}{V^2}\left(\frac{1}{h_s} + \frac{2g}{V^2}\right) \tag{9.7}$$

in which

$$\mathscr{C}_\gamma = -\left[2\omega_p/(VD)\right]\cos\chi\cos\delta$$

The derivation of the variables a and b is quite extensive, and is therefore shown in Appendix B. By substituting Eq. (9.4) in Eq. (9.1), an expression can be found for the control variable u_D:

$$u_D = \frac{1}{b}\left[-a + D_{ref}'' + 2\zeta\omega\left(D_{ref}' - D'\right) + \omega^2\left(D_{ref} - D\right)\right] \tag{9.8}$$

Once u_D has been determined, the required magnitude of the bank angle can be determined using Eq. (9.5). For a trajectory with the same initial conditions as the reference trajectory, the differences in Eq. (9.8) are both zero, which result in an identical bank angle profile as was used in the reference trajectory. For deviations from the nominal trajectory, due to, for example, uncertainties in initial conditions or atmospheric density, a value for the bank angle is determined that takes these deviations into accounts.

There are a few notes that have to be made about this guidance law. For very small values of the aerodynamic drag, this law does not work properly. If $\|u_D/(L/D)\| > 1$, no solution can be found for the bank angle, since argument of the arccosine should be in the interval $[-1,1]$. This problem can be resolved by using open-loop control for small values of D, and use the bank angle computed using Eq. (9.8) once this drag threshold has been passed. It was found that a good value of the drag threshold $D_T = 0.6\,\mathrm{N/kg}$. For the reference trajectory of Rosetta, this drag threshold was reached at an altitude of approximately 50 km at 80 s after the start of the simulation. At drag accelerations smaller than this drag threshold, deviations of the bank angle from the reference bank angle had hardly any effect on the trajectory. Therefore, during the arrival phase, the bank angle will be set to the value of the reference bank angle. For the exit phase, a different strategy will be used, as will be discussed in Section 9.2.

The problem described in the previous paragraph can also occur when the deviation from the reference trajectory becomes too excessive. If this occurs, it basically means that the vehicle is not capable of handling this deviation. To let the guidance algorithm still provide a solution, the bank angle will be either set to 0° if $u_D/(L/D) > 1$, or be set to 180° if $u_D/(L/D) < -1$.

Equation (9.1) requires values for the damping ratio ζ and a natural frequency ω. To find ζ and ω, randomly selected values were used to see how the system would responded. Manually, this would be rather tedious work, so use has been made of the adaptive Differential Evolution algorithm in PaGMO. This is the same algorithm MOEA/D uses for decomposed multi-objective optimization problems. Goal of this optimization is to find a combination of ζ and ω that would match the atmospheric bending angle of the reference trajectory. A population consisting of 15 individuals was evolved for 100 generations, which resulted in $\zeta = 0.662$ and $\omega = 0.126\,\mathrm{rad/s}$.

9.2 Lateral guidance

For bank angles not equal to 0° or 180°, there is a component of the lift-force pointing in cross-track direction, which results in an out-of-plane maneuver, and in turn results in a change of the orbital inclination. This can be desirable if one of the goals is to achieve an inclination change, or undesirable if only an increase in bending angle is needed. In either case, the lateral maneuvers should be controlled, such that a certain target inclination is achieved.

The method proposed to achieve this target inclination is the following: if the inclination starts to deviate too much, the sign of the bank angle is reversed. This so-called bank reversal inverts the component of the lift-force pointing in the cross-track direction, while the magnitude of the lift force in radial direction remains the same. By performing one or more of these reversals, the deviation of the inclination from the target inclination can be controlled.

Now what has to be determined is when a bank reversal is started. For example, one could start a bank reversal once the difference between the current inclination and the target inclination is 10°. This is the method of bank reversals implemented in the reference trajectory planning. However, it could occur that this difference is 10° at the end exit phase, when there is no sufficient lift to adjust the inclination anymore. Therefore, the selected value of the inclination threshold should not become too large. Values between 1° and 5° were found to give acceptable results, depending on the required bending angle.

Finally, once the drag threshold is passed and the exit phase is initiated, a small inclination offset can still remain, but one cannot rely on the drag tracking algorithm to find a solution for the magnitude of the bank angle. In addition, once the drag threshold is passed, just as during the entry phase, the selected bank angle does not influence the bending angle in any significant way, but still has an influence on the inclination. The magnitude of the bank angle during the exit phase is determined using (Casoliva et al., 2008):

$$\sigma = \text{sign}(i - i_t)\arcsin\left[\text{sat}\left(\kappa_0(i - i_f)\right)\right] \tag{9.9}$$

Eq. (9.9) contains a saturation function sat(x), which saturates the variable x if its value is outside of the interval $0 \leq x \leq 1$. The constant κ_0 is a proportional gain. If the difference $\kappa_0(i - i_f) \geq 1$, Eq. (9.9) will command a bank angle of 90°, which is the value for the bank angle that has the largest cross-track component. For values of $0 \leq \kappa_0(i - i_f) \leq 1$, the commanded bank angle will decrease until $i - i_f = 0°$, where the commanded bank angle will be 0°, such that the lift vector will not have a cross-track component, which in turn will ensure that the inclination is not changed.

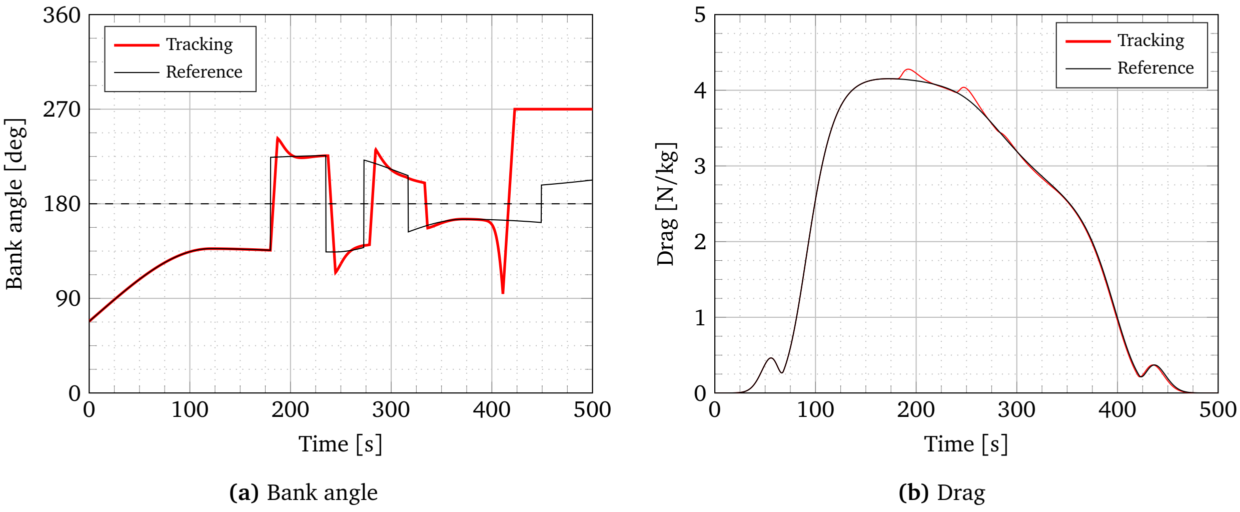
Even though the fixed reversal threshold will result in a minimal offset in the inclination for the nominal case, for off-nominal cases, as will be discusses in the following sections, a fixed threshold can still result in a large offset in inclination. For further development of the guidance algorithm, which is beyond the scope of this thesis, it is recommend to implement a different reversal threshold definition. For example, one could allow a large inclination offset at the beginning of the aerogravity assist, but this allowable offset decreases as the atmospheric bending angle increases. An example of a function that has this type of behavior is a hyperbolic tangent function.

So far, it has been assumed that once a bank reversal was required, the sign of the commanded bank angle would be changed. In reality, however, an instantaneous change in bank angle is not possible, as this would imply an infinite bank angle-rate and acceleration. To overcome this simplification, a constant but finite bank-angle rate $\dot{\sigma}_c$ will be used for a bank reversal. Assume the guidance algorithm operates on the guidance interval Δt_g. Once a bank reversal is initiated by the guidance algorithm, instead of changing the sign, the bank angle is increased or decreased by:

$$\sigma_{i+1} = \sigma_i + \dot{\sigma}_c \Delta t_g \tag{9.10}$$

The bank angle is increased or decreased during a reversal using Eq. (9.10) until the difference between the target bank angle and the actual bank angle is less than $\dot{\sigma}_c \Delta t_g$. Once this condition is true, the bank angle will be determined using Eq. (9.5) until the next reversal.

The result of a bank reversal using a constant bank-angle rate is shown in Fig. 9.1, where the reference trajectory found for Rosetta's swing-by at Mars is tracked. The bank reversals in the reference trajectory consist of discrete jump in time, whereas the bank reversals performed by the tracking algorithm consist of a more smooth change of the bank angle. It is clear that some overshoot occurs during the tracking of the bank reversal, but this overshoot is not too large. The large offset in the commanded bank angle from the



(a) Bank angle

(b) Drag

Figure 9.1: Example of the PID control during bank reversals

reference bank-angle profile at the end of the trajectory is due to the different control strategy used for the exit phase, as was described by Eq. (9.9).

9.3 Mars analysis: Rosetta

Now that the guidance algorithm is defined, it is applied to the reference trajectory found for Rosetta in Section 8.1. It will be investigated how the guidance algorithm responds to variations in initial flight-path angle, and to perturbations and variations in atmospheric density.

9.3.1 Influence of varying initial flight-path angle

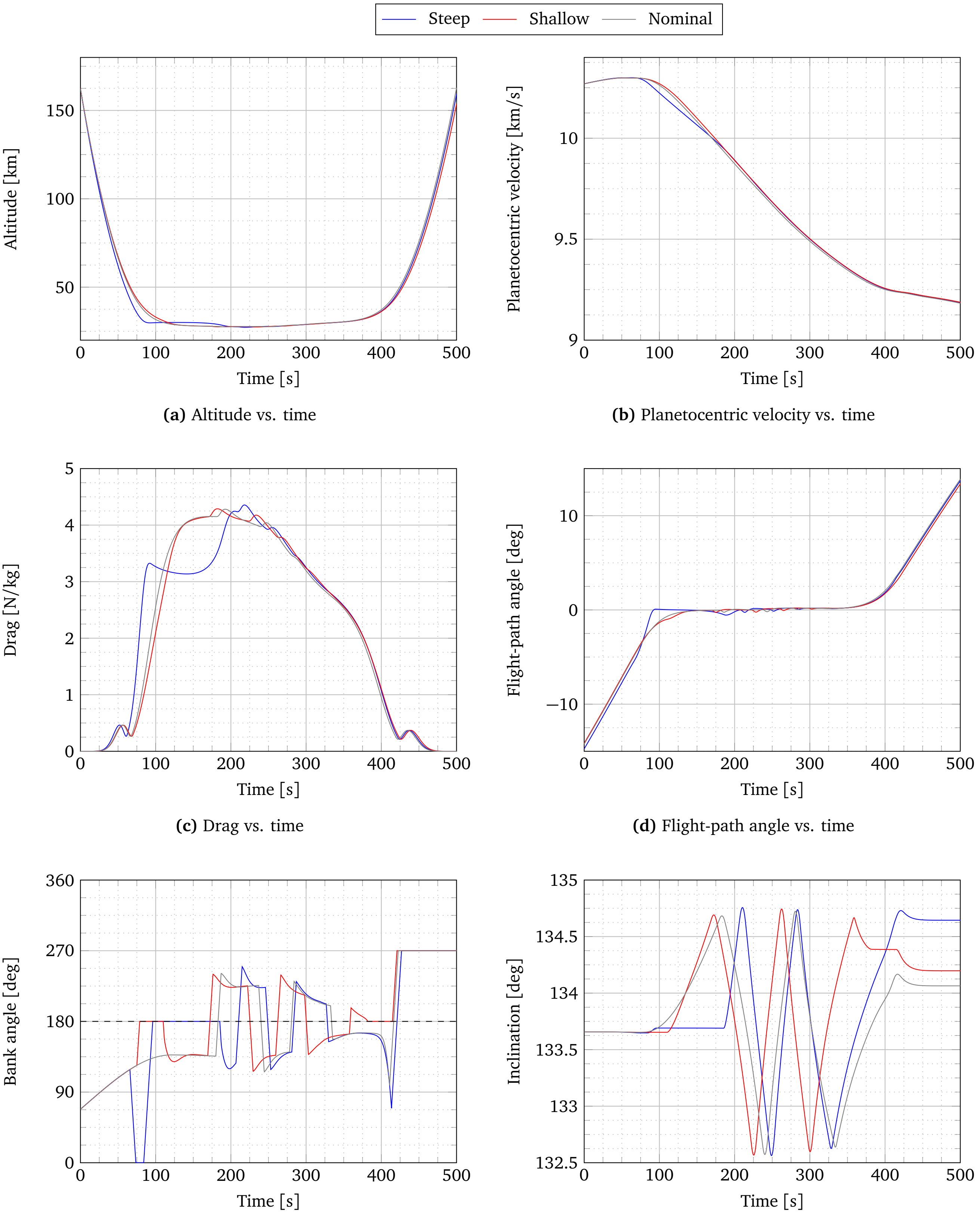
In Section 7.3, it was shown that the initial flight-path angle has a major impact on the trajectory. Too steep entries can result in large mechanical or thermal loads, and could even result in a crash on the planetary surface. To shallow entries will not penetrate the atmosphere deep enough, such that the aerodynamic force will not become sufficiently large enough to have any effect on the trajectory.

To asses the impact of the varying initial flight-path angle and how the guidance algorithm responds the these, a the nominal trajectory is compared with a steeper and a shallower entry. As the nominal trajectory was already quite shallow, it was not possible to decrease the magnitude of initial flight-path angle such that a more shallow trajectory was formed. The reason for this was already explained in the previous paragraph; during a too shallow entry, the lift force that can be generated will not be able to increase the centripetal force sufficiently and will as a result skip out of the atmosphere almost immediately.

For the more shallow entry, the magnitude of the nominal flight-path angle was decreased by 0.10°, whereas for the steeper entry, the magnitude of the nominal flight-path angle was increased by 0.50°. For trajectories outside the range $|\gamma_0|-0.10° \leq |\gamma_0| \leq |\gamma_0|+0.50°$, the guidance algorithm was not able to follow the reference trajectory. The response of the guidance algorithm to the variation of initial flight-path angle is shown in Fig. 9.2.

Figures 9.2a and 9.2b show that both the position and velocity of the off-nominal cases follow the nominal case quite well. Due to the variations of the initial flight-path angle, the drag experienced during the capture phase is different, which is why the guidance algorithm commands different bank angles compared to the nominal trajectory. This is also visible in Fig. 9.2c. In the initial phase of the trajectory, the experienced drag is not equal to the drag of the nominal trajectory. However, from approximately 200 seconds onward, the drag of both the shallow and steep entry starts approaching the drag of the nominal trajectory, exactly what the purpose of the guidance algorithm was. During the bank reversals, the flight-path angle depicted in Fig. 9.2d will not remain constant during bank reversals.

In Fig. 9.2e, it can be observed that for the steep entry, the guidance algorithm requires a smaller bank angle than was required in the nominal case once the drag threshold is passed. At a bank angle of 0°, the



(a) Altitude vs. time

(b) Planetocentric velocity vs. time

(c) Drag vs. time

(d) Flight-path angle vs. time

(e) Bank angle vs. time

(f) Inclination vs. time

Figure 9.2: Response of the guidance algorithm to variations in initial flight-path angles for an aerogravity assist at Mars

lift vector is pointing away from the planet. In this way, the guidance algorithm prevents the vehicle from entering the atmosphere too deep. The opposite is observed for the more shallow entry. For a shallow entry, the altitude will be higher, which results in turn in a smaller density and a lower lift force. Therefore, more negative lift is required to capture the vehicle, which is why the guidance algorithm commands a bank angle of 180° once the drag threshold is passed.

Finally, Fig. 9.2f shows the inclination as a function of time. Due to the different drag accelerations, the bank reversals are needed either sooner or later compared to the reference case. Furthermore, it can be seen that during the exit phase, the guidance algorithm tries to reduce the inclination offset to zero, but does not entirely succeed. Still, a small offset remains. The origin of this offset was already discussed in Section 9.2.

9.3.2 Atmospheric variations and perturbations

To decrease the computation time during the reference trajectory planning, use was made of an exponential atmosphere model. The guidance algorithm also assumes an exponential atmosphere, which made the expression of the derivative of density with respect to altitude a simple function of the scale height. In reality, spatial and temporal variations in the atmospheric density exist, to which the guidance algorithm should respond accordingly.

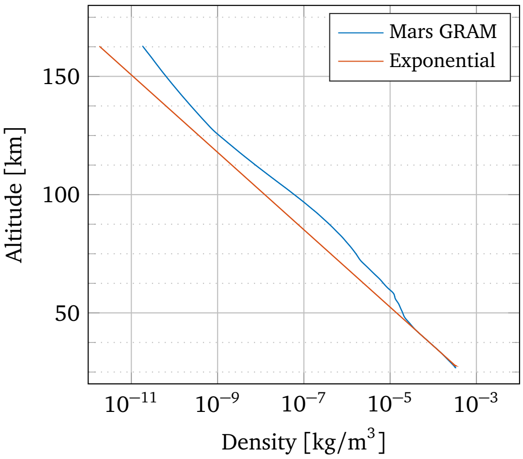


Figure 9.3: Comparison of exponential atmosphere and the Mars Global Reference Atmospheric Model

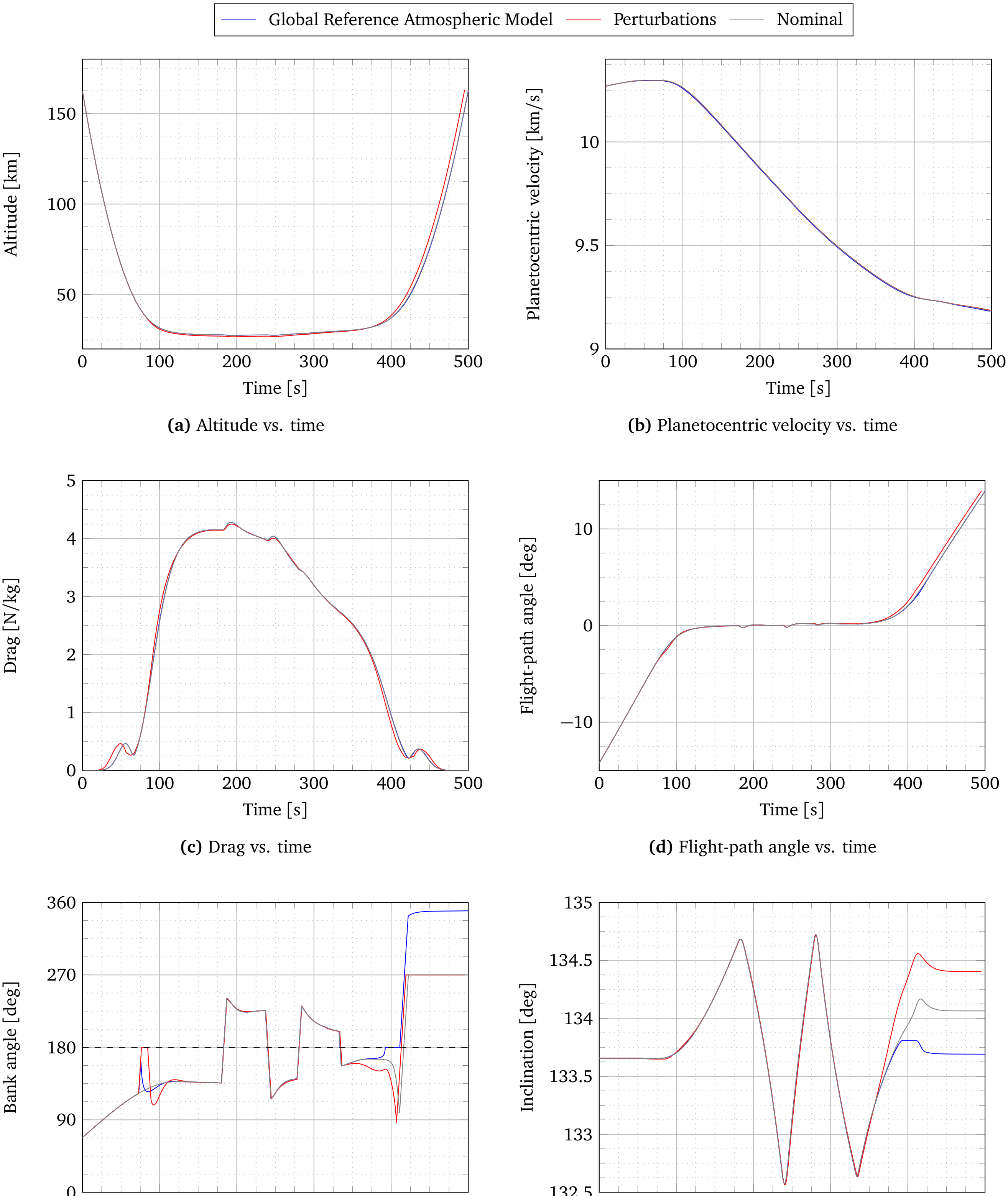
Fig. 9.3 shows the difference in density between the two models. At lower altitudes, the atmospheric density modeled by the exponential atmosphere matches the GRAM model rather well. From 50 kilometers and higher, the exponential atmosphere underestimates the density. This will increase the drag on the vehicle during the capture phase.

Next to variations in the atmospheric densities, the effect perturbations was investigated. As was discussed in Section 3.1, the perturbations that were taken into account were the oblateness of the planet (J_2) and the perturbing acceleration of the Sun. Although the direct impact on the trajectory is difficult to identify, it is clear that the guidance algorithm should be able to react to these perturbing accelerations.

Figure 9.4 shows the response of the guidance algorithm to perturbing accelerations and variations of the atmospheric density due to the use of Mars-GRAM. In the altitude and velocity, shown in Figs. 9.4a and 9.4b, only minor difference can be found. Compared to the influence of the flight-path angle on the altitude and velocity of the vehicle, the perturbations do not appear to have a large influence.

The increased drag due to the larger density during the capture phase (Fig. 9.3) is clearly visible in Fig. 9.4c, where a larger drag acceleration is experienced. Although this does not seem to be much, the guidance algorithm requires to correct for this variation in density by adjusting the bank angle, visible in Fig. 9.4e. At the cruising altitude, there are still some variations in the density compared to the exponential model. This requires the guidance algorithm to make some small adjustments in the bank angle profile.

The effect of the J_2 acceleration and the third-body perturbation is less than the effect of the variation of the density. This is visible in Fig. 9.4e, where the required correction of the bank angle once the drag threshold is passed after approximately 80 s, is smaller than the correction required for the correction for atmospheric density. For the remainder of the trajectory, the drag acceleration is tracked very well.



(a) Altitude vs. time

(b) Planetocentric velocity vs. time

(c) Drag vs. time

(d) Flight-path angle vs. time

(e) Bank angle vs. time

(f) Inclination vs. time

Figure 9.4: Response of the guidance algorithm to perturbations and variations in atmospheric density for an aerogravity assist at Mars

9.3.3 General remarks

From the analysis in Sections 9.3.1 and 9.3.2, it became clear that the principle of tracking the drag acceleration profile of a reference trajectory is working. This is confirmed by the overview in Table 9.1, where the final bending angle of the velocity vector, the magnitude of the hyperbolic excess velocity after the maneuver and the final orbital inclination are presented for the different cases that were evaluated. The difference between these three variables and the ones from the reference case remains limited to a few percent.

Table 9.1: End conditions for different off-nominal cases

Case	Variable	Reference	Achieved	Difference	%
Nominal	Bending angle [deg]	64.98	64.45	−0.54	−0.824
	Velocity [m/s]	7558.42	7558.89	0.47	0.006
	Inclination [deg]	133.66	134.06	0.41	0.305
Steep entry	Bending angle [deg]	64.98	63.92	−1.06	−1.638
	Velocity [m/s]	7558.42	7558.71	0.29	0.004
	Inclination [deg]	133.66	134.64	0.99	0.739
Shallow entry	Bending angle [deg]	64.98	65.10	0.12	0.190
	Velocity [m/s]	7558.42	7560.30	1.88	0.025
	Inclination [deg]	133.66	134.20	0.54	0.405
GRAM	Bending angle [deg]	64.98	63.74	−1.24	−1.910
	Velocity [m/s]	7558.42	7556.59	−1.84	−0.024
	Inclination [deg]	133.66	134.40	0.75	0.560
Perturbations	Bending angle [deg]	64.98	64.36	−0.62	−0.957
	Velocity [m/s]	7558.42	7560.57	2.14	0.028
	Inclination [deg]	133.66	133.69	0.03	0.026

Now the question is whether these few-percent differences are acceptable. This would depend on the interplanetary mission planning, something which is beyond the scope of this thesis. If this difference is not acceptable, using an aerogravity assist alone is not sufficient. Use could be made of propulsive maneuvers to make corrections once the vehicle is out of the atmosphere. In addition to these corrections that can be made, propulsive maneuvers could also be used to increase the atmospheric bending angle. In Section 7.6, the atmospheric bending angle was limited by the amount of energy lost; the eccentricity would approach 1.0, yielding that any additional energy loss would result in a highly elliptical orbit around the planet. One or more propulsive maneuvers could compensate for the energy loss due to drag, which would increase the atmospheric bending angle. One approach to implement this in the current trajectory planning is to define a thrust profile with a number of nodes similar to the currently used guidance nodes. However, time limitations do not allow to investigate the impact of propulsive maneuvers, and will be left for further research.

The allowable variation in initial flight-path angle gives an interesting insight in the feasibility of the aerogravity assist in combination with the currently proposed guidance algorithm. As the measured vertical velocity is used to determine the flight-path angle by the navigation system, an unrealistically low value of the vertical velocity due to the allowable variation in flight-path angle would prove that an aerogravity assist is a too sensitive maneuver. For the smallest allowable offset in flight-path angle, −0.10 deg, and an arrival velocity of 10.247 km/s would result in a offset in vertical velocity of 17.33 m/s. This means that if the velocity in vertical direction would be off by 17.33 m/s, the nominal trajectory could not be followed exactly.

Based on the results presented by Vaughan et al. (1998), the Mars Pathfinder was capable of achieving a navigation accuracy for the vertical velocity of 17.29 m/s. The on-board computer of the Apollo spacecraft could determine the vertical velocity with an accuracy of 13.81 m/s during re-entry (Duncan, 1966). Given the achieved navigation uncertainties of Mars Pathfinder and Apollo, it considered to be possible to design a navigation system with a sufficient accuracy to make a Mars aerogravity assist feasible.

Even though the guidance algorithm has not been tested due to the excessive heat loads that can be expected, there is different reason an aerogravity assist at Jupiter is not feasible. By increasing or decreasing the initial flight-path angle by 0.0001 deg for the found Ulysses trajectory, the resulting trajectory would either result in an early skip or an entry. With an arrival velocity of 81.150 km/s, this variation in initial flight-path angle would change vertical component of the velocity by 14 cm/s. This would mean that the vertical component of the velocity for a Jupiter aerogravity assist should be accurate up to 14 cm/s. When looking at the navigation uncertainties of Mars Pathfinder and Apollo discussed in the previous paragraph, an navigation accuracy of 14 cm/s is currently not possible.

For the results in Fig. 9.4, the first passage through the free-molecular flow regime and the transition flow regime consists of approximately the first 75 sec. When looking at Fig. 9.4f, the inclination during these flow regimes does not hardly change, which indicated that the atmospheric density in these flow regimes is too low to generate and noticeable drag. A large amount of computational effort was required to obtain the aerodynamic coefficients in the free-molecular flow regime using DSMC. As this has very small impact on the results, it could be considered to only calculate the aerodynamic coefficients in continuum flow and assume the same coefficients for the free molecular flow regime.

Although the aerodynamics in the free-molecular flow regime has very little impact on the found trajectories, the effort that was put into obtaining these coefficients was not useless. SPARTA has proven to be an easy to use tool that can handle complex geometries and could be used in, for example, orbital decay analysis. For satellites at low altitudes, a small amount of aerodynamic drag results over time in the decay of the orbit. By using a DSMC tool such as SPARTA, one could predict the aerodynamic characteristics of a satellite more accurately. Of course, currently, this is not relevant for the research presented in this thesis.

Another point of attention is the lateral guidance. As can be seen in Figs. 9.2f and 9.4f, a bank reversal is started once the inclination is 1.0° larger than the inclination of the arriving trajectory. Larger values of this inclination threshold, especially for this trajectory that has a relatively short atmospheric cruise phase compared to the results in Section 7.6, would result in a bank reversal that happens at a point where there is not enough atmospheric flight time left to bring the inclination back to the original inclination. Even with this low value for the inclination threshold, a difference remains once the vehicle leaves the atmosphere. The exit-phase lateral guidance, as was described by Eq. (9.9), attempts to reduce the inclination error, but for most of the cases, a small error remains. This could be solved by using a hyperbolic tangent function as an inclination threshold instead of a fixed value, although this is difficult to implement in the trajectory optimization method as was already discussed in Section 9.2.

10
Conclusions and Recommendations

This chapter contains the major conclusions that can be drawn from the work presented in this thesis. For future work or related studies, a number of recommendations are given that could improve the found results.

10.1 Conclusions

The focus of this thesis was to investigate the possibilities of aerogravity assists and how such a maneuver could affect interplanetary missions. The main research of this thesis work was formulated as follows:

> **How efficient can the interplanetary orbital inclination and velocity be changed with an aerogravity assists?**

To find an answer to this research question, four different tasks were initially identified:

- Development of a simulator capable of simulating an aerogravity assist.
- Determination of appropriate vehicle shape for an aerogravity assist maneuver, and determining the aerodynamic characteristics of such a vehicle.
- Development of a reference trajectory planner.
- Design and testing of a guidance algorithm.

The execution of these four tasks would allow us to find an answer to the main research question. In the following sections, the major conclusions from this thesis are presented.

Aerodynamics

Based on found literature, it was concluded that a waverider was the most suitable vehicle shape for an aerogravity assist. However, the term waverider describes a whole range of vehicle shapes, and a trade-off between different shapes was made. From the found waverider shapes, the geometry of the waverider with the largest lift-to-drag ratio was used to generate an aerodynamic database for the continuum flow regime and the rarefied flow regime. This waverider had the following properties: $w = 2\,\text{m}$, $l = 5\,\text{m}$, $n = 0.5$, $\theta = 5°$ and $\delta = 9°$.

The found waverider design was used during the reference trajectory planning to find some possible reference trajectories. This waverider did not provide a sufficiently large lift force in the thin Martian atmosphere to increase the centripetal acceleration. For Jupiter, having a much thicker atmosphere compared to Mars, this problem did not occur. By increasing the reference area by a factor 10 for Mars, which relates directly to the magnitude of the lift and drag force, the reference trajectory planner was able to find reference trajectories around Mars, while being able to have a maximum lift-to-drag ratio. For an aerogravity assist, it can be concluded that the lift-to-drag ratio is not the only important parameter, but also the magnitude of the lift force. Given that the surface area of a waverider is properly scaled such that it is able to increase the centripetal forces sufficiently, it has been shown that an aerogravity assist is possible using a waverider.

In literature, for example in Casoliva et al. (2008), a waverider with a reference area of $19.21\,\mathrm{m}^2$ is used for an aerogravity assist at Mars. If it is assumed that this value for the reference area is technically feasible, then used reference area for Mars ($66.7\,\mathrm{m}^2$) is approximately three times too large. The reference area was increased to be able to perform the aerogravity assist at maximum lift-to-drag ratio. However, to have the same magnitude of the lift force with a smaller reference area, the angle of attack should be increased. Although this will increase the magnitude of the lift, it will decrease the lift-to-drag ratio as the drag coefficient increases at larger angles of attack. As larger drag will increase the energy dissipation, a higher angle of attack will reduce the achievable bending angles.

For a typical aerogravity assist trajectory, the vehicle starts in the free molecular flow regime, enters the transition flow regime, cruises in the continuum flow regime and then leaves the planetary body through the transitional flow regime and the free molecular flow regime. In the free molecular flow regime and transition flow regime, where the atmospheric density is very small, it was observed that effect of the aerodynamic force is very small. As DSMC analysis by means of the SPARTA for the determination of the aerodynamic coefficients of the free molecular flow regime is very computationally expensive, the use of this method is not justified given the small impact on the results.

Reference trajectory planning

To obtain an aerogravity assist trajectory, it was chosen to use node control, a method where a finite number of guidance nodes are connected using an interpolant to obtain a continuous control history. Connecting the guidance nodes was done by means of a Hermite spline interpolant. This type of interpolant was preferred over the more commonly used cubic spline, where more overshoot can be observed than for a Hermite spline. The reference trajectory planner, which made use of this Hermite spline, determined the required guidance nodes to achieve a number of objectives while keeping the vehicle within operational constraints. It was found that a larger number of guidance nodes (10) resulted in better convergence.

To assess the possibilities of the aerogravity assist maneuver, the maximum atmospheric bending angle that could be achieved was investigated for both Mars and Jupiter as a function of the initial velocity. For both planets, it holds that a heat flux peak occurs during the entry that increases with increasing arrival velocity. Since the atmospheric density in the region where this peak occurs is relatively low, the aerodynamic lift cannot be used to reduce this peak. Especially for high velocities, this initial heat flux peak exceeds the imposed constraint, which makes it impossible for the optimization algorithm to find any feasible trajectories.

For Mars, the largest velocity bending angle (178.5°) was obtained for an initial velocity of 9.0 km/s, while for the aerogravity assist at Jupiter, the largest atmospheric bending angle (125.1°) was achieved for an initial velocity of 68.0 km/s. During the analysis of the reference trajectories around both planets, it was found that lower initial velocities resulted in larger velocity bending angles, as long as these initial velocities were not too small such that the energy dissipation would prevent the vehicle from leaving the planetary sphere of influence.

Two gravity assists that were part of actual missions were used to see the impact an aerogravity assist could have had. For Mars, the gravity assist of Rosetta was used, for Jupiter, the gravity assist of Ulysses. The vehicles used to simulate the trajectories were not Rosetta and Jupiter's actual vehicles, but the same waveriders as used in the trajectory planning. Assuming the waverider with $10S_{ref}$ for Mars is feasible, the heliocentric velocity decrease for Rosetta could be increased by 167%, from $\Delta V = 2.31\,\mathrm{km/s}$ to $\Delta V = 6.16\,\mathrm{km/s}$, confirming the possibilities of an aerogravity assist. For Ulysses, an inclination change of 54.2° was achieved using aerodynamic forces only. However, the heat loads were extremely large ($40,620\,\mathrm{W/cm}^2$), which suggests that aerogravity assists at Jupiter are not feasible using the currently possible thermal protection system designs.

Guidance algorithm

The trajectories found with the reference trajectory planner are nominal trajectories. By implementing a guidance algorithm, it should still be possible to achieve the same exit conditions as the nominal trajectories under the influence of uncertainties or perturbations. A guidance algorithm based on the tracking of atmospheric drag was found to be a successful. Values for the damping ratio and natural frequency were found by means of a differential evolution optimization scheme, and were $\zeta = 0.662$ and $\omega = 0.126\,\mathrm{rad/s}$ respectively.

The guidance algorithm was tested by varying the initial flight-path, by using a different, more accurate, atmosphere model and by adding perturbing accelerations. In all cases, the guidance algorithm was able to respond accordingly by tracking the drag acceleration. Although deviations were observed in the exit conditions for the different cases, they remain small.

10.2 Recommendations

As it was difficult to find convergence during the trajectory optimization, it might be better to divide the optimization in different segments of the trajectory, for example a capture, cruise and exit phase. As there is less freedom for each of the phases, it is expected that the optimization algorithm could reach convergence faster.

Even if the optimization does not divide the trajectory into different segments, the used optimization algorithm should be tuned more carefully. Often, the optimization algorithm converged prematurely, which required to repeat the optimization multiple times with different random seeds. Especially if the offset in right-ascension of the ascending node would be included in the optimization to ensure the trajectory remains in the same orbital plane, a total of three objectives needs to be optimized. As it was found that convergence was more difficult with three objectives, a more extensive tuning of the optimization algorithm is recommended.

The main motivation for performing an aerogravity assist is to decrease flight-time, decrease propellant mass and therefore increase payload capacity. Given the results from the trajectory planning in terms of (atmospheric) bending angle that could be achieved, the implementation of such a maneuver in interplanetary mission planning could provide great insight in the function of an aerogravity assist.

As atmosphere and gravitational models for Earth and Venus are included in the developed simulator, aerogravity assist at these two planets could be investigated in future research. Both Earth and Venus are larger than Mars in terms of mass and diameter, larger velocities will be required to fly a hyperbolic orbit around these planets. In addition, the atmospheric density is larger for both Earth and Venus compared to Mars. It is expected that this would allow cruise flight at higher altitudes than on Mars. As the composition of the atmospheres Earth and Venus are different from the Martian atmosphere, and therefore different heat flux relations exist, it is difficult to predict what the heat flux for aerogravity assists around Earth and Venus will be. As it was concluded that Jupiter is unsuitable for an aerogravity assists, investigation of aerogravity assists at other gas giants is not recommended as similar results can be expected.

Currently, the bank angle was chosen as a control variable such that the maximum lift-to-drag ratio could be maintained. This required bank reversal implementation to prevent deviation from the target inclination. By implementing angle of attack control, which is possible in the current simulator design, the need of bank reversals would be avoided as the vertical lift component can be modulated by adjusting the angle of attack. This most likely will limit the possible bending angles that can be achieved due to the larger drag that occurs at larger angles of attack, but it might still be possible to increase the bending angle compared to gravity assists.

Using angle-of-attack control will also allow to decrease the reference area. For Mars, the reference area was increased from $S_{ref} = 6.67\,\text{m}^2$ to $10S_{ref}$, such that the aerogravity assist could be performed at maximum lift-to-drag ratio. However, a reference area of $10S_{ref}$ is difficult to launch given current launch capabilities. At larger angles of attack, the lift coefficient of the vehicle will increase, which, similar to increasing the reference area, will also result in a larger lift force. Therefore, performing an aerogravity assist at a larger angle of attack will make it possible to decrease the required reference area. However, the increased drag that is a result of the larger angles of attack will also increase the energy dissipation, in turn reducing the possible bending angles.

If maximum lift-to-drag ratio remains necessary, ways of increasing the reference area should be investigated. A method of increasing the reference area is deploying an inflatable heat shield in the shape of the waverider. So-called inflatable aeroshells have often the shape of blunt entry capsules. It is recommended to investigate whether the concept of inflatable aeroshells can also be applied to waverider-geometries.

The guidance algorithm showed some minor deviations in the exit conditions for off-nominal cases. Although these deviations are small, for interplanetary mission planning it might be required that these deviations are reduced. This could be achieved by the use of propulsive maneuvers. In addition, propulsive maneuvers could be used to compensate for energy dissipation in the atmosphere due to drag, which could increase the atmospheric bending angle.

In an actual aerogravity assist, sensors and actuators are implemented such that the state can be estimated and maneuvers can be performed. By implementing a navigation and control system, the implementation of sensors and actuators can be modeled, as well as a state estimation algorithm and a control allocation algorithm. As modeling these systems introduce uncertainties, it can be tested if the guidance algorithm presented in this thesis is capable of handling these uncertainties. This is crucial for the feasibility of an aerogravity assist maneuver: if no guidance algorithm is capable of handling the uncertainties due to the navigation and/or control system, performing an aerogravity assist will not be possible.

The heat loads during an aerogravity assist are very high. To handle these heat loads, a thermal protection system should be designed that is capable of handling these high loads. As too excessive heat loads will result in vehicle damage or disintegration, a robust thermal protection system is essential to the viability of an aerogravity assist.

References

Alexander, F. and Garcia, A. (1997). "The Direct Simulation Monte Carlo method". *Computers in Physics*, 10(6):558–593.

Anderson, J. (2006). *Hypersonic and High-Temperature Gas Dynamics, Second Edition*. AIAA, 2 edition.

Armellin, R., Lavagna, M., Starkey, R. P., and Lewis, M. J. (2007). "Aerogravity-Assist Maneuvers: Coupled Trajectory and Vehicle Shape Optimization". *Journal of Spacecraft and Rockets*, 44(5):1051–1059.

Arora, N. and Russell, R. (2010). "A fast, accurate, and smooth planetary ephemeris retrieval system". *Celestial Mechanics and Dynamical Astronomy*, 108:107–124.

Aso, S., T, Y., Hirayama, H., Poetro, R., and Hatta, S. (2006). "Preliminary studies on the planetary entry to Jupiter by aerocapture technique". *Acta Astronautica*, 59:651–660.

Bansal, N. and Lamon, J. (2014). *Ceramix Matrix Composites: Materials, ModModel and Technology*. John Wiley & Sons.

Bayen, A. and Siauw, T. (2015). *An introduction to MATLAB Programming and numerical method for engineers*. Academic Press.

Bird, G. (2013). *The DSMC Method*. CreateSpace Independent Publishing Platform.

Biscani, F., Izzo, D., and Yam, C. (2010). "A Global Optimisation Toolbox for Massively PArallel Engineering Optimisation". In *4th International Conference on Astrodynamics Tools and Techniques*.

Carbonne, D., Lorenzoni, L., Huot, J., Desjean, M., Forget, F., and Portigliotti, S. (2011). "Comparisons of Mars Climate Database v4.3.1 and Mars-GRAM 2005 atmosphere models on Exomars 2016 landing site". In *Fourth international workshop on the Mars atmosphere: Modelling and observations*.

Casoliva, J., Lyons, D., and Mease, A. W. W. (2008). "Robust Guidance via a Predictor-Corrector Algorithm with Drag Tracking for Aero-Gravity Assist Maneuvers". In *AIAA Guidance, Navigation and Control Conference and Exhibit*. American Institute of Aeronautics and Astronautics. AIAA-2008-6816.

Chapra, S. and Canale, R. (2009). *Numerical Introduction for Engineers*. McGraw-Hill Science/Engineering/-Math.

Curtis, H. (2014). *Orbital Mechanics for Engineering Students*. Butterworth-Heinemann.

Dijkstra, M. (2012). Trajectory optimization of Hyperion-II for the study of hypersonic aerothermodynamic phenomena. Master's thesis, Delft University of Technology.

Duncan, R. (1966). "Apollo Navigation, Guidance and Control". In *Apollo Lunar Landing Mission Symposium*. National Aeronautics and Space Administration. NASA TM X-58006.

Gabeaud, A. (1950). "Base pressures at supersonic velocities". *Journal of the Aeronautical Sciences*, 17(8):525–526.

Gallis, M., Torczynski, J., Plimpton, S., Rader, D., and Koehler, T. (2014). "Direct Simulation Monte Carlo: The Quest for Speed". In *Proceedings of the 29th Rarefied Dynamics Symposium*. AIP Conf. Proc. 1628, 27-36.

Halton, J. (1964). "Radical-inverse quasi-random point sequence". *Communications of the ACM*, 7(12):701–702.

Hart, K., Dutta, S., Simonis, K., Steinfeldt, B., and Braun, R. (2014). "Analytically-derived Aerodynamic Force and Moment Coefficients of Resident Space Objects in Free-Molecular flow". In *AIAA Atmospheric Flight Mechanics Conference*. AIAA-2014-0728.

Ishibuchi, H., Akedo, N., and Nojima, Y. (2013). "Relation between neighboorhood size and MOEA/D performance on many-objective problems". In *Proceedings of the 7th International Conference on Evolutionary Multi-Criterion Optimization*, pages 459–474.

J.J. Bertin, J. Periaux, J. B. (1992). *Advances in Hypersonics*, volume 2. Birkhauser.

Justh, H., Justus, C., and Keller, V. (2006). "Global reference atmospheric models, including thermospheres, for Mars, Venus and Earth". In *Guidance, Navigation, and Control and Co-located Conferences*. American Institute of Aeronautics and Astronautics. AIAA 2006-6394.

Justus, C., Duvall, A., and Johnson, D. (2004). "Earth global reference atmospheric model (GRAM-99) and trase constituents". *Advances in Space Research*, 34:1731–1735.

Justus, C., James, B., Bougher, S., Bridger, A., Haberle, R., Murphy, J., and Engel, S. (2002). "Mars-GRAM 2000: A Mars atmospheric model for engineering applications". *Advances in Space Research*, 29(2):193–202.

Kemble, S. (2006). *Interplanetary Mission Analysis and Design*. Springer.

Kremer, G. (2010). *An Introduction to the Boltzmann Equation and Transport Processes in Gases*. Springer.

Laub, B. (2003). "Thermal protection concepts and issues for aerocapture at Titan". In *Joint Propulsion Conference*. American Institute of Aeronautics and Astronautics. AIAA 2003-4954.

Lavagna, M., Povoleri, A., and Finzi, A. (2005). "Interplanetary mission design with aero-assisted manoeuvers multi-objective evolutive optimization". *Acta Astronautica*, 57:498–509.

Lissauer, J. and De Pater, I. (2013). *Fundamental Planetary Science*. Cambrdige University Press.

Liu, B., Fernandez, F., Zhang, Q., Pak, M., Sipahi, S., and Gielen, G. (2010). "An Enhanced MOEA/D-DE and Its Application to Multiobjective Analog Cell Sizing". In *Proceeding of the IEEE Congress on Evolutionary Computation*, pages 1–7.

Lockwood, M. (2004). "Neptune aerocapture systems analysis". In *Atmospheric Flight Mechanics Conference and Exhibit*. American Institute of Aeronautics and Astronautics, Providence, Rhode Island.

Lohar, F., Mateescu, D., and Misra, A. (1994). "Optimal atmospheric trajectory for aero-gravity assist". *Acta Astro*, 32(2):89–96.

Lyons, D. (2002). "Mars Reconnaissance Orbiter: Aerobraking Reference Trajectory". In *Guidance, Navigation, and Control and Co-located Conferences*. American Institute of Aeronautics and Astronautics.

Lyons, D. T., Saunders, R., and Griffith, D. G. (1995). "The Magellan Venus mapping mission: Aerobraking operations". *Acta Astronautica*, 35(9-11):669 – 676.

Mambrini, A. and Izzo, D. (2014). "PaDe: A Parallel Algorithm Based on the MOEA/D Framework and Island Model". In *Proceedings of the 13th International Conference on Parallel Problem Solving from Nature*, pages 711 – 720.

Mazzaracchio, A. and Marchetti, M. (2010). "A probabilistic sizing tool and Monte Carlo analysis for entry vehicle ablative thermal protection systems". *Acta Astronautica*, 66(5-6):821–835.

McRonald, A. and Randolph, J. (1990). "Hypersonic maneuvering to provide planetary gravity assist". In *Proceedings of Aerospace Sciences Meetings*. American Institute of Aeronautics and Astronautics.

Millour, E., Forget, F., Spiga, A., Navarro, T., Madeleine, J.-B., Montabone, L., Lefevre, F., Chaufray, J.-Y., Lopez-Valverde, M., Gonzalez-Galindo, F., Lewis, S., Read, P., Desjean, M.-C., and Huot, J.-P. (2014). "The Mars Climate Database (MCD Version 5.1)". In *8th International Conference on Mars*.

Mooij, E. (1997). *The motion of a vehicle in a planetary atmosphere*. Delft University Press, The Netherlands.

Mooij, E. (2013). AE4870B - Re-entry systems - lecture notes. Internal publication. Delft University of Technology.

Mooij, E., Huot, J.-P., and Ortega, G. (2006). "Entry Trajectory Simulation Using ESA Mars Climate Database Version 4.1". In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*.

Munoz, P., Budnik, F., Godard, B., Morley, T., Companys, V., Herfort, U., and Cacas, C. (2012). "Preparations and strategy for navigation during Rosetta comet phase". In *Proceedings 23rd International Symposium on Space Flight Dynamics*.

Papp, Z. (2014). Mission planner for heating-optimal re-entry trajectories with extended range capabilities. Master's thesis, Delft University of Technology. http://repository.tudelft.nl/view/ir/uuid%3A2355ab39-550c-4447-8c56-61ffabd53353/.

Picone, J., Drob, D., Meier, R., and Hedin, A. (2003). NRLMSISE-00: A new empirical model of the atmosphere. http://www.nrl.navy.mil/research/nrl-review/2003/atmospheric-science/picone/. Accessed on: March 3, 2015.

Player, C., Cheatwood, F. M., and Corliss, J. (2005). "Development of Inflatable Entry Systems Technologies". In *In Proceedings of the 3rd International Planetary Probe Workshop*. ESA SP-607.

Regan, F. and Anandakrishnan, S. (1993). *Dynamics of Atmospheric Re-entry*. American Institute of Aeronautics and Astronautics.

Ritter, H., F. Mazoue, A. S., and Atzei, A. (2006). "Jupiter Entry Probe Feasibility Study from the ESTEC CDF Team: Heat Flux Evaluation and TPS Definition". In *Proceedings of the 5th European Workshop on Thermal Protection Systems and Hot Structures*. SP-631.

Ronse, A. (2013). Statistical impact prediction of decaying objects. Master's thesis, Delft University of Technology.

Sandlin, D. and Pessin, D. (1993). "Aerodynamic analysis of hypersonic waverider aircraft". Technical report, NASA Ames Research Center. NASA-CR-192981.

Saraf, A., Leavitt, J., Chen, D., and Mease, K. (2004). "Design and Evaluation of an Acceleration Guidance Algorithm for Entry". *Journal of Spacecraft and Rockets*, 41(6):986–996.

Sengers, J., Wang, Y.-Y. L., Kamgar-Parsi, B., and Dorfman, J. (2014). "Kinetic theory of drag on objects in nearly free molecular flow". *Physica A: Statistical Mechanics and its Applications*, 413:409–425.

Smith, E. and Marsden, R. (1995). "Ulysses Observations from Pole-to-Pole: An Introduction". *Geophysical research letters*, 22(23):3297–3300.

Spilker, T. R. (2005). "Significant science at Titan and Neptune from aerocaptured missions". *Planetary and Space Science*, 53(5):606–616.

Standish, E. (2011). "Keplerian Elements for Approximate Positions of the Major Planets". Technical report, Jet Propulsion Laboratory / Caltech. http://ssd.jpl.nasa.gov/txt/aprx_pos_planets.pdf.

Starkey, R. and Lewis, M. (1999). "Simple Analytical Model for Parametric Studies of Hypersonic Waveriders". *Journal of Spacecraft and Rockets*, 36(4):516–523.

Sutton, K. and Graves, R. (1971). "A general stagnation-point convective-heating equation for arbitrary gas mixtures". Technical report, National Aeronautics and Space Administration. NASA TR R-376.

Svehla, R. (1962). "Estimated Viscosities and Thermal Conductivities of Gases at High Temperatures". Technical report, National Aeronautics and Space Administration. NASA TR R-132.

Tauber, M., Palmer, G., and Prabhu, D. (2012). "Stagnation point radiative heating relations for Venus entry". Technical report, National Aeronautics and Space Administration. ARC-E-DAA-TN2887.

Tauber, M. and Sutton, K. (1991). “Stagnation-point radiative heating relations for Earth and Mars entries”. *Journal of Spacecraft and Rockets*, 28(1):40–42.

Tauber, M., Wercinski, P., Yang, L., and Chen, Y.-K. (1999). “A Fast Code for Jupiter Atmospheric Entry Analysis”. Technical report, National Aeronautics and Space Administration. NASA/TM-1999-208796.

Tewari, A. (2007). *Atmospheric and Space Flight Dynamics*. Birkhäuser Boston.

Uesugi, K., Hayashi, T., and Matsuo, H. (1988). “MUSES-A double lunar swingby mission”. *Acta Astronautica*, 17(5):495–501.

Vaughan, R., P.H. Kallemeyn, J., Spencer, D., and Brauwn, R. (1998). “Navigation flight operations for Mars Pathfinder”. Technical report, National Aeronautics and Space Administration. AAS 98-145.

Venkatapathy, E., Szalai, C., Laub, B., Hwang, H., Conley, J., and Arnold, J. (2010). “White Paper: Thermal protection system technologies for enabling future sample return missions”. Technical report, NASA.

Villefranche, P., Evans, J., and Faye, F. (1997). “Rosetta: The ESA comet rendevous mission”. *Acta Astronautica*, 40(12):871–877.

Wakker, K. (2010). AE4874 - Astrodynamics lecture notes. Internal publication. Delft University of Technology.

Wie, B. (2008). *Space Vehicle Dynamics and Control*. American Institute of Aeronautics and Astronautics.

Williams, D. (2014). *Planetary Fact Sheet*. `http://nssdc.gsfc.nasa.gov/planetary/factsheet/`, accessed May 20, 2014.

Wilmott, P., Howison, S., and Dewynne, J. (1995). *The Mathematics of Financial Derivatives: A Student Introduction*. Cambridge University Press.

Zhang, Q. and Li, H. (2007). “MOEA/D: A Multi-objective Evolutionary Algorithm based on Decomposition”. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731.

A

SPARTA Tips & Tricks

Although it was shown in this thesis that the aerodynamics in the free molecular flow regime did not have a large impact on an aerogravity assist, SPARTA has proved to be an easy-to-use and versatile implementation of the Direct Simulation Monte Carlo (DSMC) method. For other applications, such as orbit decay of satellites in Low Earth Orbit, SPARTA could be used to estimate the aerodynamic properties of a vehicle in the rarefied flow regime. An example, as given by the developers of SPARTA, is shown in Fig. A.1, where the rarefied flow around the Mir space station is shown.

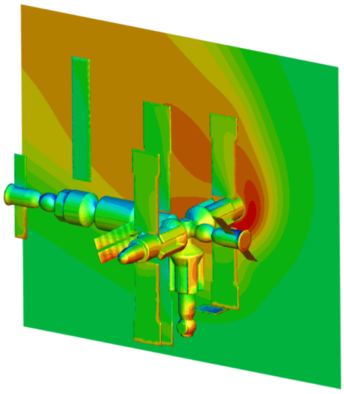


Figure A.1: Flow around Mir space station (Courtesy: Sandia/SPARTA)

SPARTA is well documented, but as it is a very extensive software package, one might encounter some pitfalls that I have experienced myself. This chapter will give a few basic tips on how to get started with SPARTA and how to avoid common errors and mistakes.

- Before SPARTA can be used, the source code (C++) should be compiled. By default, a large number of Makefiles are included in `/sparta/src/MAKE/`. As DSMC is computational intensive, it supports the Message Passing Interface (MPI). By default, SPARTA compiles using a dummy MPI library, but I recommend investing some effort in installing an MPI library such as openMPI[1] or MPICH[2], as it will allow you split the computation over multiple processes and decrease the required computational time.
- If the compilation of the source code was successful, an executable will be obtained. This executable, which I renamed `sparta`, can be used from the command line, and makes use of an input file. A large number of examples of input files can be found in `/sparta/examples`. These examples are really useful for the development of your own input file, and I highly recommend playing around with them to learn how to define flows, gas compositions and collision models. Once you have obtained or created your input file, for example, `in.waverider`, it can be evaluated by SPARTA using the following command:

[1] openMPI: https://www.open-mpi.org/

[2] MPICH: https://www.mpich.org/

```
./sparta -in in.waverider
```

If you have an MPI library and compiled SPARTA accordingly, it is possible to split the computation over multiple processes. For example, if the computation needs to be split over 4 processes, the following command can be used:

```
mpirun -n 4 ./sparta -in in.waverider
```

- An important parameter in the input file which will allow to reduce the computation time is the ratio of physical particles to simulation particles `fnum`. Decreasing this parameter will increase the number of particles simulated which will give a more accurate result, but it also increases the computation time. Increasing this number will decrease the number of particles simulated and reduce the simulation time, but the accuracy is decreased. It is recommended to vary the value for `fnum` such that a balance is found between computation time and accuracy.

- It is not possible to specify the atmospheric density. Instead, a number density should be specific, which basically represents the number of particles per unit volume. The number density n and atmospheric density ρ are related by:

$$n = \frac{N_A}{M}\rho \tag{A.1}$$

  where N_A is Avogadro's number ($N_A = 6.0221409 \times 10^{23}$) and M is the molar mass of the gas.

- The geometry will be one of the inputs of the input file. A geometry in SPARTA should be defined by a triangular mesh. This means that the entire geometry is described by a list of triangles. A text file, that contains this geometry, consists of a list of points and a list that define which three points are used in each triangle. Below, an example of such a geometry file is shown.

```
Sphere

602 points
1200 triangles

Points

1 -0.57735026919 -0.57735026919 -0.57735026919
2 -0.615457454897 -0.492365963917 -0.615457454897
3 -0.662266178533 -0.529812942826 -0.529812942826
...

Triangles

1 1 3 2
2 1 4 3
...
```

  It is important that the defined geometry is watertight. This means that each point should only be defined once and that every triangle edge is part of exactly two triangles. Delaunay triangulation in `MATLAB`, which can convert a quadrilateral mesh to a triangular mesh, does **not** guarantee a watertight geometry. This should always be checked to avoid errors in SPARTA.

- It is possible to rotate a geometry in the input file. This prevent generating different geometry files for, for example, different angles of attack. Assuming the geometry file is called `data.waverider`, the following entry will allow to rotate the geometry over 45° around the Y-axis:

```
read_surf data.waverider rotate 45 0 1 0
```

- To determine the aerodynamic coefficients, one needs the components of the normal pressure and shear stress for each of the triangles of the geometry. This can be achieved by using the following command:

  ```
  compute 3 surf all all px py pz shx shy shz
  ```

  after which it can be written to a file by using:

  ```
  dump 3 surf all 200 press_shear.* id f_3
  ```

  which means that every 0.200 seconds simulated, a text file will be written which contains the individual components of the normal pressure and shear stress per face. I personally used `MATLAB` to load these text files and integrate the normal pressure and shear stress over the entire surface of the geometry to determine the aerodynamic force on the vehicle. This force can then be decomposed to a component perpendicular and parallel to the velocity vector to determine the lift and drag, which in turn can be used to find the lift and drag coefficients.

- If it is necessary to repeat the DSMC analysis for, for example, multiple angles of attack or velocities, it is possible to automate this in the input file by defining loops. This will avoid manually adjusting the input file for each combination of these parameters. These loops are explained in more detail on page 265 of the SPARTA manual (15 May 2015 version).

B

Drag derivatives with respect to energy

This appendix contains the derivations for the derivatives of the drag with respect to energy, used in the guidance algorithm. First of all, an alternative set of equations of motion will be presented, as well as the conversion from time as independent variable to energy as independent variable. Next, the first- and second derivative of the drag with respect to energy will be derived.

Equations of motion

To obtain the values for a and b (see Eqs. (9.6) and (9.7)), one has to take two times the derivative of the drag with respect to the energy, after which expressions with known variables obtained from a set of equations of motion is substituted. The set of equations of motion that are considered were shown in Eqs. (6.1) to (6.6) and are listed again below. These derivatives in these equations are with respect to time.

$$\dot{\tau} = \frac{V\cos\gamma\sin\chi}{R\cos\delta} \tag{6.1}$$

$$\dot{\delta} = \frac{V\cos\gamma\cos\chi}{R} \tag{6.2}$$

$$\dot{r} = V\sin\gamma \tag{6.3}$$

$$\dot{V} = -D - g\sin\gamma \tag{6.4}$$

$$\dot{\gamma} = \frac{1}{V}\left[L\cos\sigma - \left(g - \frac{V^2}{R}\right)\cos\gamma\right] + 2\omega_p\cos\delta\sin\chi \tag{6.5}$$

$$\dot{\chi} = \frac{1}{V}\left[\frac{V^2\cos\chi\tan\delta\cos\gamma}{R} + \frac{L\sin\sigma}{\cos\gamma} - 2\omega_p V\left(\tan\gamma\cos\delta\cos\chi - \sin\delta\right)\right] \tag{6.6}$$

Now consider the total energy of a vehicle E. An expression for this total energy is give in Eq. (B.1), where V is the inertial velocity, μ is the gravitational parameter of the central body, R_p is the radius of the planetary body and R is the radial distance with respect to the center of the planetary body (Saraf et al., 2004).

$$E = \frac{V^2}{2} + \frac{\mu}{R} \tag{B.1}$$

Using energy as independent variable allows to reduce the six equations presented earlier in this section to a total of five differential equations. If $(\cdot)$ is an arbitrary variable, then its derivative with respect to energy can be obtained by Eq. (B.2).

$$\frac{d(\cdot)}{dE} = (\cdot)' = \frac{1}{\dot{E}}\frac{d(\cdot)}{dt} = -\frac{1}{DV}\frac{d(\cdot)}{dt} \tag{B.2}$$

Using energy as independent variable, and using the differentiation rule presented in Eq. (B.2), the equations of motion become the expressions shown in Eqs. (B.3) to (B.7).

(B.3)

(B.4)

(B.5)

$$+\frac{1}{V^2\cos\gamma}\left(\frac{L}{D}\sin\sigma\right)+\mathscr{C}_\chi \tag{B.6}$$

$$\gamma'=\left(g-\frac{V^2}{R}\right)\frac{\cos\gamma}{V^2}\left(\frac{1}{D}\right)-\frac{1}{V^2}\left(\frac{L}{D}\cos\sigma\right)+\mathscr{C}_\gamma \tag{B.7}$$

in which $\mathscr{C}_\chi$ and $\mathscr{C}_\gamma$, also known as the Coriolis terms, are given by Eqs. (B.8) and (B.9).

$$\mathscr{C}_\chi=-\left[2\omega_p/(VD)\right](\tan\gamma\cos\chi\cos\delta-\sin\delta) \tag{B.8}$$

$$\mathscr{C}_\gamma=-\left[2\omega_p/(VD)\right]\sin\chi\cos\delta \tag{B.9}$$

Eqs. (6.1) to (6.5) are a set of differential equations in which the independent variable is energy. The drag tracking algorithm presented by Saraf et al. (2004) uses these expressions in combination with the second derivative of the drag acceleration.

First derivative

Consider the drag acceleration D, which is given by Eq. (2.12) and is repeated below:

$$D=\frac{1}{2}\rho V^2 C_D S/m \tag{2.12}$$

First, the first derivative of the drag with respect to the energy D' is given by:

$$D'=\frac{dD}{dE}=\underbrace{\frac{\partial D}{\partial\rho}\frac{\partial\rho}{\partial E}}_{\text{Term 1}}+\underbrace{\frac{\partial D}{\partial V}\frac{\partial V}{\partial E}}_{\text{Term 2}}+\underbrace{\frac{\partial D}{\partial C_D}\frac{\partial C_D}{\partial E}}_{\text{Term 3}} \tag{B.10}$$

As Eq. (B.10) contains many partial derivatives, these will first be individually determined to maintain readability, after which they are substituted into this equation to obtain an expression for D'.

Term 1

The derivative of the drag D with respect to the atmospheric density ρ is given by Eq. (B.11).

$$\frac{\partial D}{\partial\rho}=\frac{1}{2}V^2 C_D S/m \tag{B.11}$$

The derivative of the atmospheric density ρ with respect with respect to the energy is given by Eq. (B.12).

$$\frac{\partial\rho}{\partial E}=\frac{\partial\rho}{\partial R}\frac{\partial R}{\partial E}+\frac{\partial\rho}{\partial V}\frac{\partial V}{\partial E} \tag{B.12}$$

The derivative of the density with respect to the velocity in Eq. (B.12) is known to be zero. From Eq. (B.5), the partial derivative $\partial r/\partial E$ can be obtained, resulting in expression Eq. (B.13) for the derivative of the density with respect to the energy.

$$\frac{\partial \rho}{\partial E} = -\frac{\partial \rho}{\partial R}\frac{\sin\gamma}{D} \tag{B.13}$$

Term 2

The derivative of the drag D with respect to the velocity V is given by Eq. (B.14).

$$\frac{\partial D}{\partial V} = \rho V C_D S/m \tag{B.14}$$

The derivative of velocity with respect to energy is given by Eq. (B.15).

$$\frac{\partial V}{\partial E} = \frac{\partial V}{\partial t}\frac{\partial t}{\partial E} \tag{B.15}$$

From Eq. (6.4), the partial derivative $\partial V/\partial t$ can be obtained. Combining this with Eq. (B.2), the expression of the partial derivative of the velocity with respect to energy becomes Eq. (B.16)

$$\frac{\partial V}{\partial E} = \frac{D + g\sin\gamma}{DV} \tag{B.16}$$

Term 3

The partial derivative of the drag D with respect to the drag coefficient C_D is given by Eq. (B.17).

$$\frac{\partial D}{\partial C_D} = \frac{1}{2}\rho V^2 S/m \tag{B.17}$$

Finally, the partial derivative of the drag coefficient C_D with respect to energy E remains. Since there is no analytical expression for this derivative, its notation will only be simplified, as shown in Eq. (B.18).

$$\frac{\partial C_D}{\partial E} = C_D' \tag{B.18}$$

Substituting Eqs. (B.11), (B.13), (B.14) and (B.16) to (B.18) into Eq. (B.10) and combining terms, will result in Eq. (B.19).

$$D' = D\left(\frac{2}{V^2} + \frac{C_D'}{C_D}\right) + \sin\gamma\left(-\frac{1}{\rho}\frac{\partial \rho}{\partial R} + \frac{2g}{V^2}\right) \tag{B.19}$$

Second derivative

To obtain the second derivative of the drag with respect to energy, one starts with differentiating Eq. (9.2) with respect to energy. This second derivative is shown in Eq. (B.20).

$$\begin{aligned} D'' = \frac{\partial D}{\partial E}\left(\frac{2}{V^2} + \frac{C_D'}{C_D}\right) + \overbrace{\frac{\partial}{\partial E}\left(\frac{2}{V^2} + \frac{C_D'}{C_D}\right)}^{\text{Term 1}} D + \\ + \underbrace{\frac{\partial \sin\gamma}{\partial E}}_{\text{Term 2}}\left(-\frac{1}{\rho}\frac{\partial \rho}{\partial R} + \frac{2g}{V^2}\right) + \underbrace{\frac{\partial}{\partial E}\left(-\frac{1}{\rho}\frac{\partial \rho}{\partial R} + \frac{2g}{V^2}\right)}_{\text{Term 3}} \sin\gamma \end{aligned} \tag{B.20}$$

Three terms from Eq. (B.20) need to be expanded, which will be performed below.

Term 1

The first term that needs to be expanded can be divided into two separate terms, as shown in Eq. (B.21).

$$\frac{\partial}{\partial E}\left(\frac{2}{V^2}+\frac{C'_D}{C_D}\right)=\underbrace{\frac{\partial}{\partial E}\left(\frac{2}{V^2}\right)}_{A}+\underbrace{\frac{\partial}{\partial E}\left(\frac{C'_D}{C_D}\right)}_{B} \tag{B.21}$$

Subterm A

Using Eq. (B.16), subterm A can be expressed as Eq. (B.22).

$$\frac{\partial}{\partial E}\left(\frac{2}{V^2}\right)=\frac{\partial}{\partial V}\left(\frac{2}{V^2}\right)\frac{\partial V}{\partial E}=\frac{-4}{V^4}\left(\frac{D+g\sin\gamma}{D}\right) \tag{B.22}$$

Subterm B

Using the quotient rule, subterm B can be expressed as Eq. (B.23).

$$\frac{\partial}{\partial E}\left(\frac{C'_D}{C_D}\right)=\frac{C''_D C_D-C'^2_D}{C_D^2}=\frac{C''_D}{C_D}-\frac{C'^2_D}{C_D^2} \tag{B.23}$$

Term 2

The second term that needs to be expanded contains a partial derivative of the flight-path angle γ with respect to energy. By applying the chain rule, term 2 can be expressed in terms of known quantities, as shown in Eq. (B.24).

$$\frac{\partial \sin\gamma}{\partial E}=\frac{\partial \sin\gamma}{\partial \gamma}\frac{\partial \gamma}{\partial E}=\cos\gamma\frac{\partial \gamma}{\partial E} \tag{B.24}$$

Substituting Eq. (B.7) in Eq. (B.24) results in

$$\frac{\partial \sin\gamma}{\partial E}=\cos\gamma\left[\left(g-\frac{V^2}{r}\right)\frac{\cos\gamma}{V^2}\left(\frac{1}{D}\right)-\frac{1}{V^2}\left(\frac{L}{D}\cos\sigma\right)+\mathscr{C}_\gamma\right] \tag{B.25}$$

Term 3

The third term that needs to be expanded can be divided into two separate terms, as is shown in Eq. (B.26).

$$\frac{\partial}{\partial E}\left(-\frac{1}{\rho}\frac{\partial \rho}{\partial R}+\frac{2g}{V^2}\right)=\underbrace{\frac{\partial}{\partial E}\left(-\frac{1}{\rho}\frac{\partial \rho}{\partial R}\right)}_{\mathrm{A}}+\underbrace{\frac{\partial}{\partial E}\left(\frac{2g}{V^2}\right)}_{\mathrm{B}} \tag{B.26}$$

Subterm A

The expansion of subterm A is shown in Eqs. (B.27) and (B.28).

$$\frac{\partial}{\partial E}\left(-\frac{1}{\rho}\frac{\partial \rho}{\partial R}\right)=\frac{\partial}{\partial E}\left(-\frac{1}{\rho}\right)\frac{\partial \rho}{\partial R}-\frac{1}{\rho}\frac{\partial}{\partial E}\left(\frac{\partial \rho}{\partial R}\right) \tag{B.27}$$

$$=\frac{\partial}{\partial R}\left(-\frac{1}{\rho}\right)\frac{\partial \rho}{\partial E}\frac{\partial \rho}{\partial R}-\frac{1}{\rho}\frac{\partial^2 \rho}{\partial R^2}\frac{\partial R}{\partial E} \tag{B.28}$$

Expressions for the terms $\partial\rho/\partial E$ and $\partial r/\partial E$ can be found in Eq. (B.13) and Eq. (B.5) respectively.

Subterm B

The expansion of subterm B is show in Eq. (B.29).

$$\frac{\partial}{\partial E}\left(\frac{2g}{V^2}\right)=\frac{\partial}{\partial V}\left(\frac{2g}{V^2}\right)\frac{\partial V}{\partial E} \tag{B.29}$$

Using Eq. (B.16), Eq. (B.29) can be expressed as Eq. (B.30)

$$\frac{\partial}{\partial E}\left(\frac{2g}{V^2}\right)=\frac{-4g}{V^3}\frac{D+g\sin\gamma}{DV} \tag{B.30}$$

Assuming that $\sin^2\gamma\approx 0$ and $D+g\sin\gamma\approx D$, combining terms A, B, and C, and substituting them in Eq. (B.20) results in an expression for D'', as shown in Eq. (B.31).

$$\begin{aligned}D''=D\left(\frac{C_D''}{C_D}-\frac{C_D'^2}{C_D^2}\right)+D'\left(\frac{C_D'}{C_D}+\frac{2}{V^2}\right)-\frac{4D}{V^4}+\frac{1}{DV^2}\left(-\frac{1}{\rho}\frac{\partial\rho}{\partial R}+\frac{2g}{V^2}\right)\left(g-\frac{V^2}{r}\right)\\+\left(-\frac{1}{\rho}\frac{\partial\rho}{\partial R}+\frac{2g}{V^2}\right)\mathscr{C}_\gamma+\frac{1}{V^2}\left(-\frac{1}{\rho}\frac{\partial\rho}{\partial R}+\frac{2g}{V^2}\right)\frac{L}{D}\cos\sigma\end{aligned} \tag{B.31}$$

Eq. (B.31) can be written as

$$D''=a+b\left(\frac{L}{D}\cos\sigma\right) \tag{B.32}$$

where

$$a=D\left(\frac{C_D''}{C_D}-\frac{C_D'^2}{C_D^2}\right)+D'\left(\frac{C_D'}{C_D}+\frac{2}{V^2}\right)-\frac{4D}{V^4}+\frac{1}{DV^2}\left(-\frac{1}{\rho}\frac{\partial\rho}{\partial R}+\frac{2g}{V^2}\right)\left(g-\frac{V^2}{r}\right)+\left(-\frac{1}{\rho}\frac{\partial\rho}{\partial R}+\frac{2g}{V^2}\right)\mathscr{C}_\gamma$$

and

$$b=-\frac{1}{V^2}\left(-\frac{1}{\rho}\frac{\partial\rho}{\partial R}+\frac{2g}{V^2}\right)$$

C Example configuration file

```
# Example configuration file

// Name of the planet
planetaryBody                           = "Mars";

// Output directory
dataOutput =
{
        outputDirectory                 = "/OutputDirectory/";
        outputFilename                  = "stateHistory.dat";
};

// Numerical integration properties
integrationProperties =
{
        integrationMethod               = "RungeKuttaFehlberg78";
        absoluteTolerance               = 1.0e-10;
        relativeTolerance               = 1.0e-10;
        dataWritingInterval             = 0.25;         // seconds
        guidanceInterval                = 0.25;         // seconds
        integrationStartTime            = 0.0;          // seconds
        integrationEndTime              = 1000.0;       // seconds
        stopAtInitialAltitude           = true;
};

// Initial conditions of the trajectory
initialConditions =
{
        altitude                        = 162545.6;     // meters
        longitude                       = 0.0;          // degrees
        latitude                        = 0.0;          // degrees
        velocity                        = 7580.3;       // meters per second
        flightPathAngle                 = -14.975;      // degrees
        headingAngle                    = 45.0;         // degrees
};

// Vehicle properties
vehicleProperties =
{
        referenceArea                   = 66.67;        // square meters
        mass                            = 800.0;        // kilograms
        hypersonicDatabase              = "continuumFlowDB.dat";
        freeMolecularDatabase           = "freeMolecularFlowDB.dat";
};

// Environmental properties
environmentalProperties =
{
        atmosphereModel                 = "exponential";
        tabulatedAtmosphereFile         = "tabulatedAtmosphereMars.dat";
        J2acceleration                  = false;
        thirdBodyAcceleration           = false;
};

// Tracking algorithm properties
trackingProperties =
{
        useTracking                     = true;
        referenceTrajectory             = "stateHistoryReference.dat";
};
```