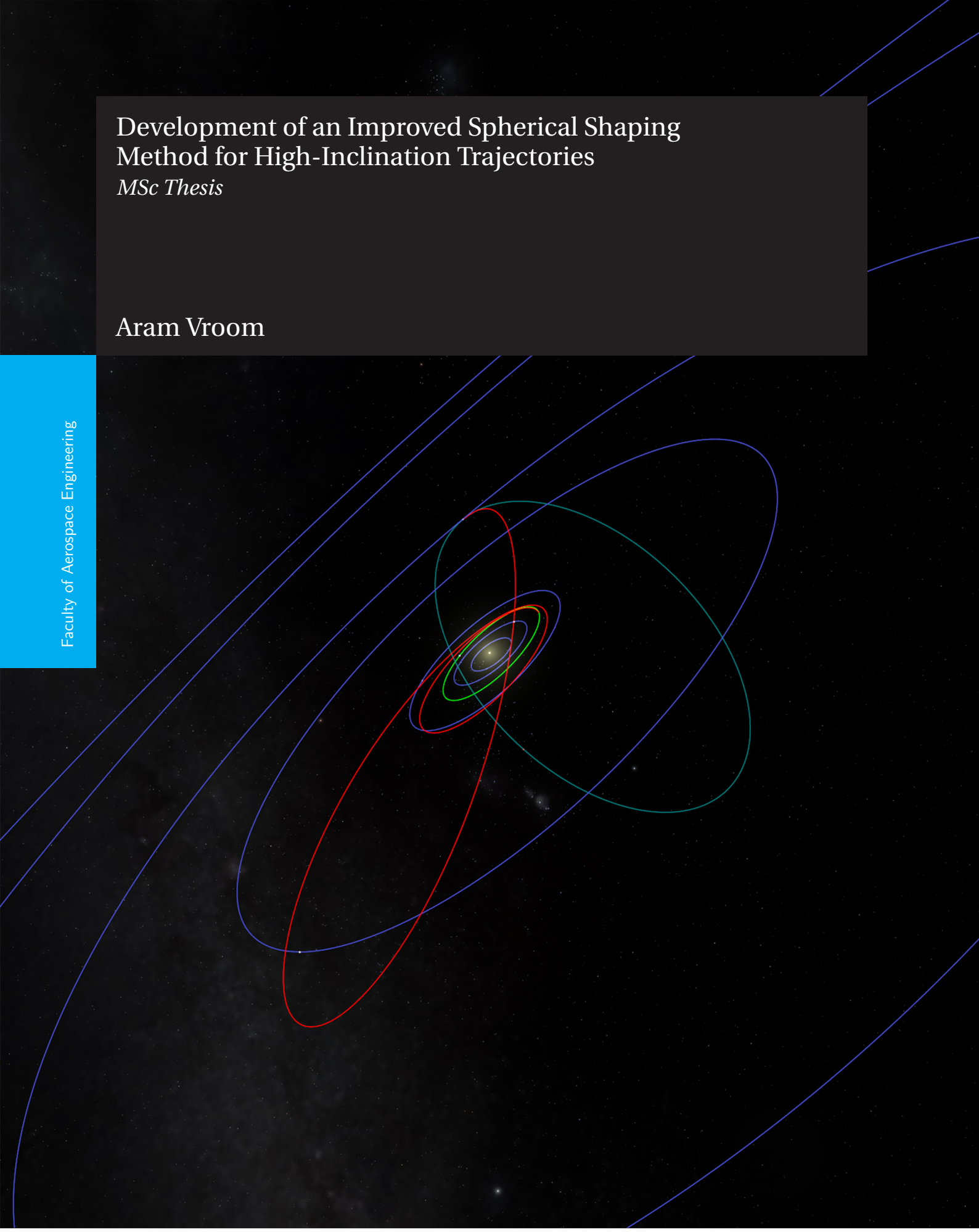
Development of an Improved Spherical Shaping Method for High-Inclination Trajectories

MSc Thesis

Aram Vroom

Faculty of Aerospace Engineering



TU Delft Delft University of Technology

Challenge the future

The image on the title page shows a transfer from Earth to the comet 2003 EH1. The image was generated using the outcome of this thesis and the Celestia software that is available from `https://celestiaproject.net/`.

Development of an Improved Spherical Shaping Method for High-Inclination Trajectories

MSc Thesis

by

Aram Vroom

in partial fulfillment of the requirements for the degree of

Master of Science
in Aerospace Engineering

at Delft University of Technology,
to be defended publicly on Wednesday August 30, 2017 at 2:00 PM.

Student number:	4140192	
Thesis committee:	Prof. dr. ir. P. N. A. M. Visser,	TU Delft
	ir. R. Noomen,	TU Delft, supervisor
	ir. B. T. C. Zandbergen,	TU Delft

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft Delft University of Technology

PREFACE

This thesis was written as part of the Space Flight master at the Aerospace Engineering Faculty of Delft University of Technology and discusses my graduation project to become a Master of Science.

During this master's thesis, highly interesting results were obtained with regards to further improving the spherical shaping method's applicability to high-inclination trajectories. While admittedly the results were perhaps not as good as I had initially hoped, a very significant improvement in terms of accuracy was still achieved. I greatly enjoyed working on this project, and for that I would like to thank a number of people.

First of all, I want to thank my supervisor Ron Noomen. During our weekly meetings, he provided invaluable feedback and tips for this research. His enthusiasm and thoughts on matters helped me immensely.

Furthermore, I want to thank my parents and my sister. Without their love and support, this thesis would have been significantly more difficult.

Aram Vroom - July 2, 2017

LIST OF ABBREVIATIONS

AU	Astronomical Unit
BC	Boundary Condition
DITAN	Direct Interplanetary Trajectory Analysis
JPL	Jet Propulsion Laboratory
LQ	Linear Quadratic
MC	Monte-Carlo
MJD2000	Modified Julian Date 2000
NM	Nelder-Mead
RAAN	Right Ascension of the Ascending Node
RK4	Fourth-order Runge-Kutta integrator
RMSE	Root-Mean-Square Error
TOF	Time-of-Flight
TU Delft	Delft University of Technology
Tudat	Technical University of Delft Astrodynamics Toolbox

LIST OF SYMBOLS

LATIN SYMBOLS

a	Acceleration	[m s^{-2}]
a	Semi-major axis	[m]
a	Shaping function coefficient	[-]
$\mathbf{a}$	Acceleration vector	[m s^{-2}]
A	Cross-sectional area	[m^2]
$\mathbf{A}$	Matrix used to find the coefficient values	[rad]
$\mathbf{A}_{r_2}$	Vector containing the terms related to r_2	[rad]
b	Shaping function coefficient	[-]
$\mathbf{b}_j$	Vector containing the coefficients of the elevation shaping function	[rad]
$\mathbf{B}$	Vector used to find the coefficient values	[rad]
c	Non-linear least-squares coefficient	[-]
c	Shaping function coefficient	[-]
c	Speed of light	[m s^{-1}]
$\mathbf{c}$	Centroid of the Nelder-Mead simplex	[-]
C	Auxiliary variable used in the spherical shaping method	[m rad^{-2}]
C_3	Injection energy	[m^2 s^{-2}]
D	Parabolic anomaly	[rad]
D	Time equation scalar function	[m rad^{-2}]
e	Eccentricity	[-]
$\mathbf{e}_h$	Unit vector in the direction of the angular momentum	[-]
$\mathbf{e}_r$	Unit vector in the direction of the radius vector	[-]
$\mathbf{e}_z$	Unit vector perpendicular to the equatorial plane	[-]
E	Eccentric anomaly	[rad]
f	Summation of the argument of periapsis and the true anomaly	[rad]
f_a	Factor by which the theoretical value of a^2 is multiplied	[-]
F	Auxiliary variable used in the spherical shaping method	[various]
F	Hyperbolic anomaly	[rad]
$\mathbf{F}$	Force vector	[N]
G	Gravitational constant	[m^3 s^{-2} kg^{-1}]

h	Angular momentum per unit of mass	[$m^2\ s^{-1}\ kg^{-1}$]
h	Step size	[-]
H	Angular momentum	[$m^2\ s^{-1}$]
i	Inclination	[rad]
l	Auxiliary variable used in the conversion to Cartesian coordinates	[-]
m	Auxiliary variable used in the conversion to Cartesian coordinates	[-]
m	Mass	[kg]
m_0	Initial mass	[kg]
m_e	Empty mass	[kg]
M	Mean anomaly	[rad]
n	Auxiliary variable used in the conversion to Cartesian coordinates	[-]
n	Mean motion	[rad s^{-1}]
n	Number of points in the Nelder-Mead simplex	[-]
N_r	Number of revolutions	[-]
p	Semi-latus rectum	[m]
p	Shaping function coefficient	[-]
q	Periapsis distance	[m]
r	Shaping function coefficient	[-]
$\mathbf{r}$	Radius vector	[m]
$\hat{\mathbf{r}}$	Radial unit vector	[-]
$\mathbf{r}_k$	Vector containing the coefficients of the radial shaping function	[rad]
$R(\theta)$	Radial shaping function	[rad]
$\mathbf{R}_k$	Vector containing the sub-functions of the radial shaping function	[various]
t	Time	[s]
t_0	Reference time	[s]
T	Orbital period	[s]
$T(\theta)$	Time evolution function	[s]
$\mathbf{T}$	Thrust vector	[N]
T_{syn}	Synodic period	[s]
$\mathbf{u}$	Control acceleration vector	[m s^{-2}]
$\mathbf{u}_i$	Unit vector in i^{th} coordinate in the Nelder-Mead method	[-]
$\mathbf{v}$	Velocity vector	[m s^{-1}]
V_e	Exhaust velocity	[m s^{-1}]
w	Shaping function coefficient	[-]

W	Solar constant	[W m^{-2}]
x	Cartesian x-coordinate	[m]
$\mathbf{x}_c$	Contraction point in the Nelder-Mead method	[-]
$\mathbf{x}_e$	Expansion point in the Nelder-Mead method	[-]
$\mathbf{x}_h$	Worst point in the Nelder-Mead simplex	[-]
$\mathbf{x}_l$	Best point in the Nelder-Mead simplex	[-]
$\mathbf{x}_r$	Reflection point in the Nelder-Mead method	[-]
$\mathbf{x}_s$	Second worst point in the Nelder-Mead simplex	[-]
y	Cartesian y-coordinate	[m]
z	Cartesian z-coordinate	[m]
Z	Auxiliary variable used in the spherical shaping method	[m^{-1}]

GREEK SYMBOLS

α	Angle between the radius vector and the spacecraft's velocity	[rad]
α	Auxiliary variable used in the spherical shaping method	[m rad^{-1}]
α	Azimuthal angle in spherical triangles	[rad]
α	Reflection parameter in the Nelder-Mead method	[-]
β	Contraction parameter in the Nelder-Mead method	[-]
γ	Flight path angle	[rad]
δ	Shrinkage parameter in the Nelder-Mead method	[-]
δ_{pos}	Variation of the position boundary conditions in each direction	[m]
δ_{vel}	Variation of the velocity boundary conditions in each direction	[m s^{-1}]
ΔV	Change in velocity	[m s^{-1}]
$\Delta\phi$	Difference in elevation angle	[rad]
ϵ	Machine precision	[-]
γ	Expansion parameter in the Nelder-Mead method	[-]
Γ	Auxiliary variable used in the spherical shaping method	[-]
η	Auxiliary variable used in reference frame transformations	[m]
η	Auxiliary variable used in the conversion to Cartesian coordinates	[m]
θ	Azimuthal angle in the spherical coordinate system	[rad]
θ	True anomaly	[rad]
λ	Auxiliary variable used in the spherical shaping method	[rad]
μ	Standard gravitational parameter	[m^3 s^{-2}]
ξ	Auxiliary variable used in the conversion to Cartesian coordinates	[m]
ρ	Reflectivity	[-]

$\boldsymbol{\pi}$	Unit vector in the direction of the orthogonal projection of $\mathbf{e}_r$	[-]
σ	Auxiliary variable used in the spherical shaping method	[rad]
τ	Auxiliary variable used in the spherical shaping method	[-]
ϕ	Elevation angle in the spherical coordinate system	[rad]
$\Phi(\theta)$	Elevation shaping function	[rad]
$\mathbf{\Phi}_j$	Vector containing the sub-functions of the elevation shaping function	[various]
ω	Argument of periapsis	[rad]
Ω	Right ascension of the ascending node	[rad]
$\mathbf{\Omega}$	Unit vector pointing towards the right ascension of the ascending node	[-]

SUBSCRIPTS

f	Final value
h	Out-of-plane component
i	Index in iteration
i	Initial value
k	Index in iteration
n	Normal component
new	Newly calculated value
old	Earlier obtained value
$previous$	Value calculated using the previous elevation shaping function
$prop$	Propagated value
r	Radial component
req	Required
$Roegiers$	Value shown in Roegiers' MSc thesis
t	Tangential component
θ	Component along the vector corresponding to the azimuthal angle
ϕ	Component along the vector corresponding to the elevation angle

SUPERSCRIPTS

$\dot{\square}$	First derivative with respect to time
$\ddot{\square}$	Second derivative with respect to time
$\square'$	First derivative with respect to the azimuthal angle
$\square''$	Second derivative with respect to the azimuthal angle
$\square'''$	Third derivative with respect to the azimuthal angle
$\tilde{\square}$	Parameterised by the azimuthal angle and along the trajectory
$\square^{-1}$	Inverse of a matrix
$\square^{T}$	Transpose of a matrix

ABSTRACT

Over the past years, numerous missions for spacecraft with low-thrust propulsion have been planned to include large orbital plane changes. In order efficiently generate and evaluate orbits for such missions, use can be made of the spherical shaping method proposed by Novak [1]. When inclinations larger than 15 degrees are present however, the error in the ΔV found by the method increases significantly. At an inclination of 50 degrees, the spherical shaping method is no longer capable of modelling the trajectory [2]. It was found in the literature review that the most promising method of increasing the shaping method's applicability at large inclination, is by developing an improved elevation shaping function [3].

In order to obtain an improved elevation shaping function, a number of test cases are defined first. These test cases include both unperturbed and perturbed orbits, the latter in turn being comprised of both cases with a discontinuous and a continuous thrust. Unperturbed orbits at inclinations of 0 and 89.99 degrees, as well as orbits of which the semi-major axis changed or the inclination varied by more than 89 degrees are among others included. As an additional problem encountered in literature is the inability to accurately model changes in the right ascension of the ascending node (RAAN) when the arrival and departure orbits are inclined, two additional cases that model this variation are generated as well.

As a first step in the development process itself, the ability of three potential methods to obtain a better elevation shaping function is evaluated with the use of the unperturbed cases. These methods include the usage of spherical triangles, Fourier series and an alternative function that was found during the development. In this first step, the usage of spherical triangles is discarded as no shaping function with coefficients is obtained and complications are expected when they are used to model non-circular orbits.

It is found that, to be able to accurately describe perturbed orbits, the coefficients of the Fourier series and the alternative function should change as a function of the azimuthal angle. To find these functions, each perturbed orbit is split into numerous unperturbed orbits and the MATLAB curve fitting application is used to fit both the Fourier and the alternative function to each of these orbits. By then observing the variation of each coefficient as a function of the azimuthal angle for each case, potential functions that can describe the variations of the coefficients are found. In this step, the Fourier series is discarded due to the larger number of required coefficients and the significantly worse fit at high inclinations.

Combinations of the aforementioned candidate functions for each coefficient of the alternative function are compared using the root-mean-square error (RMSE) found by the MATLAB curve fitting application when the combinations are used to approximate the previously defined test cases. With this methodology, a new elevation shaping function is found.

The new elevation shaping function can be implemented into the spherical shaping method by keeping the original methodology but replacing the current elevation shaping function and its derivatives. As less boundary conditions are available for the elevation shaping function than the number of coefficients that need to be solved, values for the additional coefficients are obtained through optimisation. This optimisation is done by first performing a global search with the Monte-Carlo method, after which a predefined number of the best solutions found are further optimised with the local Nelder-Mead method.

The validation of the new elevation shaping function is done with a number of internal and external validation cases. In order to be able to compare the elevation shaping functions, test cases previously defined by Roegiers are used [2]. The internal test cases include both planar and non-planar transfers, as well as cases in which the RAAN changes. The external validation includes an Earth - Mars and an Earth - Neptune transfer. From these cases, it is found that the new elevation shaping function is much more capable of modelling transfers with high inclinations. However, a misfit at the level of 7.5 degrees is still observed when a Keplerian orbit with an inclination of 50 degrees is modelled. Similar conclusions are also drawn when a transfer with a

change in RAAN is simulated.

The RMSE observed during the validation is significantly larger than the RMSE predicted during the development. This difference lies in the fact that the MATLAB curve fitting toolbox used during the development does not strictly enforce the boundary conditions, whereas the spherical shaping method does. For this reason, it is investigated whether using more flexible position and velocity boundary conditions improves the solutions.

By modelling Keplerian orbits at an inclination of 50 degrees using varying amounts of flexibility for the boundary conditions, it is concluded that the ΔV and the RMSE can be reduced from 31.6 km/s and 7.5 degrees to 21.8 km/s and 2.9 degrees. It is also noted that the new elevation shaping function is much more capable of modelling half a revolution than a full revolution. Depending on the initial true anomaly, a ΔV of 0.7 km/s and a misfit of 0.1 degrees can be found. However, the ΔV does not always decrease when more flexible boundary conditions are used and half a Keplerian orbit is approximated. As loosening the boundary conditions should result in a ΔV that is lower than or equal to the ΔV found when more strict boundary conditions are used, it is found that this is caused by the optimisation procedure having difficulty finding the global minimum. By increasing the number of Monte-Carlo samples and the number of points further optimised with the Nelder-Mead algorithm, the ΔV found can be decreased further. However, as this causes the computation time to become impractical, this is not feasible considered for further usage.

After the validation, the spherical shaping method with the new elevation shaping function is applied to the design of two missions. The first of these includes a transfer from Earth to the dwarf planet Makemake and the second mission includes a transfer from Earth to the comet 2003 EH1. These two targets are chosen due to their high inclinations and because neither of these celestial bodies has been visited yet.

For each of these missions, a grid search is first used together with a small number of Monte-Carlo and Nelder-Mead samples to find the optimal departure time, time-of-flight and the number of revolutions. Once the optimal point in the grid is found, the departure date, number of revolutions and time-of-flight of this point are used to further optimise the trajectory with a larger number of Monte-Carlo and Nelder-Mead samples. Using this methodology, it is found that the spherical shaping method is capable of producing smooth trajectories to both targets, even though the inclination of 2003 EH1 is approximately 70 degrees. The required ΔVs of the trajectories are 16.5 km/s and 23.0 km/s respectively, whereas using the old elevation shaping function results in trajectories with ΔVs of 24.0 km/s and 50.8 km/s. Highly promising is that in the optimal Earth - 2003 EH1 transfer found, a large portion of the inclination change is performed at a large distance from the Sun; this is generally considered as the most efficient way of changing the inclination.

The aforementioned concept of using flexible boundary conditions is also applied to the Makemake and 2003 EH1 transfers found by the grid search. However, this does not improve the ΔV beyond the smaller amount needed to meet the change in velocity boundary conditions. Nonetheless, it is seen that the new elevation shaping function is capable of producing transfers to these targets when flexible boundary conditions are used, as the trajectories found are all also smooth, highly similar and as expected.

Finally, the reliability of the solutions is evaluated by performing the optimisation of the best point found by the grid search five additional times. The same departure time, time-of-flight and the number of revolutions is used for each of these runs, but different seeds are set for the pseudorandom number generator used by the Monte-Carlo method. It is observed that the maximum difference in ΔV is approximately 4 m/s, that the maximum difference in the out-of-plane control accelerations is less than $8 \cdot 10^{-5}$ m/s^2 and that the trajectories are also highly similar. Furthermore, it is seen that one of these additional runs finds a ΔV lower than the one initially obtained and that the values of the shaping function's coefficients are significantly different for each run. While the small differences in ΔV, control accelerations and trajectories indicate that the solutions are reliable, the lower ΔV found in the additional runs and the different values of the coefficient support the aforementioned conclusion that the optimisation procedure has difficulty finding the global minimum. Therefore, in order to obtain a full analysis of the solution reliability, the grid search itself should also be run multiple times and the results should be compared. As this could not be done during this thesis, it is highly recommended that future research evaluates and potentially further improves the optimisation procedure.

CONTENTS

1

INTRODUCTION

Numerous missions featuring large orbital plane changes and spacecraft with low-thrust propulsive means have been planned over the past years. During the preliminary assessment of such mission concepts however, parameters such as the launch and arrival windows can have very large search ranges. As this results in the evaluation of a large number of feasible trajectories, an efficient method that is capable of generating and evaluating such a large number of potential trajectories is needed [1].

One way to efficiently generate and evaluate trajectories for such missions, is by analytically modelling the trajectories with the use of so-called shape-based methods. One of these shape-based methods is the spherical shaping method proposed by Novak [1]. This method is capable of finding rendezvous trajectories that meet the initial and final position and velocity constraints, as well as the required time-of-flight. However, the error in the ΔV found by this method increases significantly when trajectories have a large constant inclination or when they include a plane change larger than 15 degrees. At an inclination of 50 degrees, the spherical shaping method breaks down [2]. It can thus be concluded that when such inclinations are involved, the current spherical shaping method lacks an accurate description of the orbit's out-of-plane component. Nonetheless, it was found in the literature review that the spherical shaping method can be improved in order to include a more accurate formulation for the out-of-plane motion [3].

The purpose of this MSc thesis is therefore to do research on the development of a more accurate elevation shaping function and to incorporate this function into the spherical shaping method. To this end, three methods of finding this elevation shaping function are investigated, the most promising of these methods is further developed and the resulting function is incorporated into the spherical shaping method. In short, this thesis can be described using the following research question:

Can an alternative elevation shaping function be developed for the spherical shaping method to improve the shaping method's applicability and accuracy at high orbital inclinations?

This thesis report shall first elaborate upon the problem statement and a number of background concepts related to orbital dynamics and low-thrust propulsion in Chapter 2. Afterwards, the development of the elevation shaping function is discussed in Chapter 3. The implementation of the alternative elevation shaping function into the spherical shaping method is then discussed in Chapter 4. This implementation is internally and externally validated in Chapter 5 and the resulting spherical shaping method is applied to missions to the dwarf planet Makemake and the comet 2003 EH1 in Chapter 6. Lastly, the numerous conclusions drawn throughout the development process of this more accurate elevation shaping function for the spherical shaping method are summarised and a number of recommendations for future research are given in Chapter 7.

2

BACKGROUND

Before the new elevation shaping function can be developed and applied, a number of background topics need to be discussed. In this chapter, the problem statement is first discussed in Section 2.1, after which various concepts related to orbital dynamics are elaborated upon in Section 2.2. Lastly, the concept of low-thrust propulsion will be addressed in Section 2.3.

2.1. PROBLEM STATEMENT

As mentioned in the introduction, the current spherical shaping method is incapable of accurately modelling (transfer) orbits with large inclinations. Therefore, the goal of this thesis is to improve the existing spherical shaping method such that large inclinations do not impose problems.

As the spherical shaping method is not the only shape-based method with this problem, various candidate shape-based methods were evaluated in the literature review to determine in which method a more accurate description of the out-of-plane motion would be implemented. It was found that a large amount of research has already been done on Novak's spherical shaping method within Delft University of Technology (TU Delft) and that the software written by Roegiers for part of this research is readily available. Furthermore, it was found that Roegiers also proposed a solution to the inability of the spherical shaping method to model highly inclined orbits in her MSc thesis, but that this method only worked under certain conditions [2]. As the spherical shaping method also finds ΔVs that are either lower than or comparable to those found by other methods, it was decided that a more accurate description of the out-of-plane motion would be developed for the spherical shaping method. It was found that by reformulating the part of the shaping method that models the change in elevation angle, this being the elevation shaping function, the accuracy of the spherical shaping method can be improved. As such, this thesis will aim to develop an improved spherical shaping method by developing and implementing an alternative elevation shaping function.

2.2. ORBITAL DYNAMICS

In order to find a more accurate elevation shaping function, one may start by looking at a number of concepts in the field of orbital dynamics. This section will first discuss the basics such as Newton's laws of motion and Newton's law of universal gravitation in Sections 2.2.1 and 2.2.2, after which it will elaborate on the concept of orbital elements and Kepler orbits in Sections 2.2.3 and 2.2.4 respectively. Lastly, the various interplanetary perturbations are discussed in Section 2.2.5. The information presented here has been retrieved from the book *Fundamentals of Astrodynamics* by Wakker, unless specified otherwise [4].

2.2.1. NEWTON'S LAWS

Of course, one of the main fundamentals of today's mathematical methods is Newton's *Philosophiae Naturalis Principia Mathematica* of 1687. In these three books, the following laws of motion were formulated:

First law:
Every particle continues in its state of rest or uniform motion in a straight line relative to an inertial reference frame, unless it is compelled to change that state by forces acting upon it.

Second law:
The time rate of change of linear momentum of a particle relative to an inertial reference frame is proportional to the resultant of all forces acting upon that particle and is collinear with and in the direction of the resultant force.

Third law:
If two particles exert forces on each other, these forces are equal in magnitude and opposite in direction.

As the linear momentum of a particle can be written as $m\mathbf{v}$, it can be noted that the second law can also be written as shown in Equation 2.1.

$$\mathbf{F} = \frac{d}{dt}(m\mathbf{v}) \tag{2.1}$$

in which m is the mass, $\mathbf{v}$ is the particle's velocity vector and $\mathbf{F}$ is the resultant force vector.

2.2.2. NEWTON'S LAW OF GRAVITATION

A fourth law that Newton defined in his Principia, is his law of universal gravitation. The definition of this is as follows:

Law of universal gravitation:
Two particles attract each other with a force directly proportional to their masses and inversely proportional to the square of the distance between them.

Mathematically, this law can be written as shown in Equation 2.2.

$$\mathbf{F}_{12} = G\frac{m_1 m_2}{|\mathbf{r}_{12}|^2}\hat{\mathbf{r}}_{12} \tag{2.2}$$

in which G is the gravitational constant, $\mathbf{F}_{12}$ is the force that object 1 experiences due to object 2, m_1 and m_2 are the masses of objects 1 and 2 respectively, $|\mathbf{r}_{12}|$ is the distance between the two objects and $\hat{\mathbf{r}}_{12}$ is the unit vector pointing from object 1 to object 2.

When the number of particles is increased to n, this formulation can be rewritten to Equation 2.3 to obtain an expression for the force on a body i in a many-body problem.

$$\mathbf{F}_i = \sum_{\substack{j=1 \\ j\neq i}}^{j=n} G\frac{m_i m_j}{|\mathbf{r}_{ij}|^2}\hat{\mathbf{r}}_{ij} \tag{2.3}$$

Equations 2.1 and 2.3 lay the foundation of present-day dynamics and are used during the derivation of the more accurate elevation shaping function.

2.2.3. ORBITAL ELEMENTS

In order to describe an orbit and the position of a body in its orbit, six parameters can be used. These parameters, also known as orbital elements, are the semi-major axis a, eccentricity e, inclination i, right ascension of the ascending node Ω, argument of periapsis ω and the true anomaly θ. A graphical representation of the inclination, right ascension of the ascending node, true anomaly and argument of periapsis can be seen in Figure 2.1. The semi-major axis and eccentricity define the size and shape of the orbit respectively.

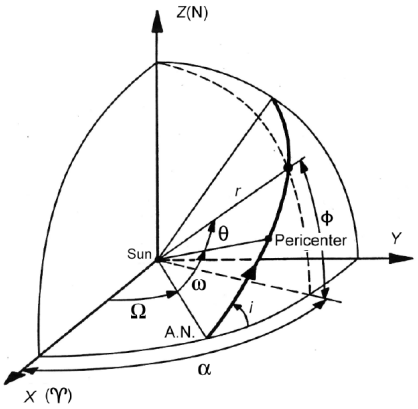
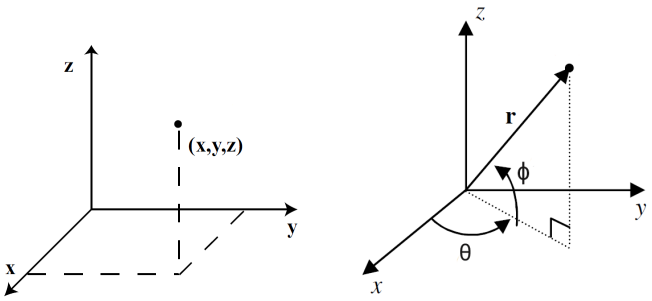


Figure 2.1: A graphical representation of among others the inclination, right ascension of the ascending node, true anomaly and argument of periapsis [4].

The six orbital elements can be converted into the Cartesian or spherical coordinate systems with the use of a set of unambiguous transformations that can be found in Appendix A. The definitions of these coordinate systems can be seen in Figure 2.2, where **r** is the radius vector, θ is the azimuthal angle and ϕ is the elevation angle [5]. While primarily the spherical coordinate system will be used in this thesis, the Cartesian coordinate system will be utilised as well. The various reference frames in which these coordinates can be used are elaborated upon in Appendix B.



(a) Cartesian coordinate system. (b) Spherical coordinate system.

Figure 2.2: The two coordinate systems used in this thesis [5].

2.2.4. KEPLER ORBITS

When only the gravitational attraction discussed in Section 2.2.2 is considered and it is assumed that the mass of the central body is much larger than that of the satellite, so-called Kepler orbits are the solutions found to the two-body problem. These Kepler orbits can be parametrised by the orbital elements discussed previously and comprise all conic sections. The four conic sections and their orbit shapes can be seen in Figure 2.3.

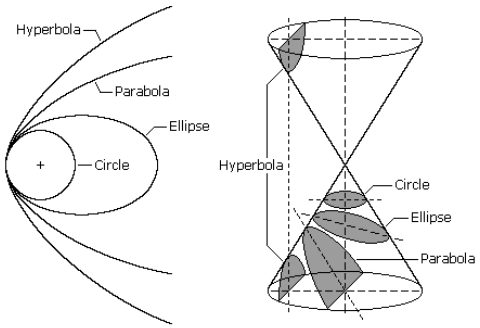


Figure 2.3: The four conic sections that form the Kepler orbits and their orbit shapes [4].

Further properties of the elliptical, parabolic and hyperbolic orbits can be found in Appendix C. As it is assumed that the reader is familiar with Kepler orbits, no further details will be discussed here. For more information, the reader is referred to *Fundamentals of Astrodynamics* by Wakker [4].

2.2.5. PERTURBATIONS

In the interplanetary medium, a number of perturbations are present. These perturbations include among others the solar radiation pressure and third-body perturbations.

The effective acceleration due to the solar radiation pressure can be found with Equation 2.4. In this equation, ρ is the reflectivity, W is the solar constant, c is the speed of light, r is the distance to the Sun in astronomical units, A is the cross-sectional area and m is the mass of the spacecraft.

$$a_{rad} = (1+\rho)\frac{W}{c \cdot r^2}\frac{A}{m} \tag{2.4}$$

To obtain the accelerations on the spacecraft caused by third-body perturbations, Equation 2.2 can be used.

As an example, one can consider a solar sail with a mass of 300 kg, a cross-sectional area of 900 m^2, a reflectivity of 0.2 and an additional low-thrust engine providing 1 N of thrust. By using the aforementioned equations, it can be found that the effective acceleration due to the celestial bodies, solar radiation and the spacecraft's thrust varies over the distance to the Sun as shown in Figure 2.4 [6].

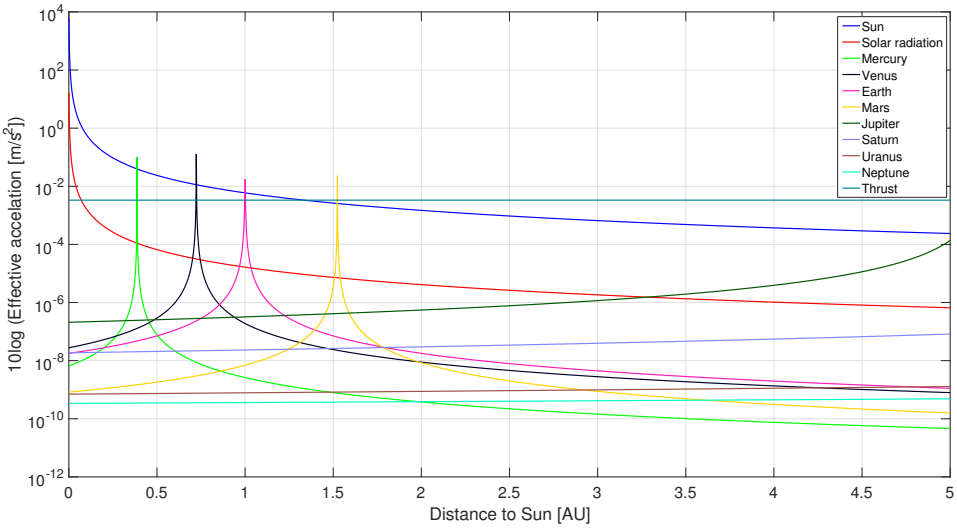


Figure 2.4: The perturbing accelerations in m/s^2 of a solar sail with a mass of 300 kg, a cross-sectional area of 900 m^2, a reflectivity factor of 0.2 and a low-thrust engine providing 1 N of thrust [6].

From this figure, it can be concluded that the perturbing accelerations are significantly smaller than the accelerations due to the thruster and the Sun's gravitational force. As such, the orbit planning problem can be considered as a two-body problem in which only thrust acts as an additional force on the spacecraft.

2.2.6. SYNODIC PERIOD

When a trajectory is generated from a certain celestial body to another, the concept of a synodic period can prove to be useful. The synodic period is defined as the time it takes for two bodies that are orbiting a third body to return to their original relative positions. It can be calculated using Equation 2.5.

$$\frac{1}{T_{syn}} = \left|\frac{1}{T_1} - \frac{1}{T_2}\right| \tag{2.5}$$

where T_{syn} is the synodic period, T_1 is the orbital period of the departure body and T_2 is the orbital period of the target body.

2.3. LOW-THRUST PROPULSION

As mentioned in the introduction of this report, the main focus of this thesis will include the orbit design of low-thrust spacecraft. This section will therefore discuss the principle and fundamental equations of low-thrust propulsion and orbits, as well as provide an introduction to shape-based methods.

2.3.1. PRINCIPLE

The basis of spacecraft propulsion lies in the preservation of momentum. Generally, matter is expelled from the engine at a high velocity and the rocket or satellite is accelerated in the opposite direction as a consequence.

From Tsiolkovsky's rocket equation, it can be found that the exhaust velocity of a spacecraft's engine should be as high as possible to decrease the amount of propellant needed to achieve a certain amount of ΔV. When considering chemical rocket propulsion systems, these exhaust velocities can be as high as 4.6 km/s with the use of liquid hydrogen and liquid oxygen. However, when one would compare this to a low-thrust ion thruster, exhaust velocities of 31.5 km/s can be found, resulting in a large performance increase of the spacecraft's propulsion system [7]. For reference, Tsiolkovsky's rocket equation has been shown here as Equation 2.6. In this equation, V_e is the exhaust velocity, m_0 is the initial mass and m_e is the empty mass.

$$\Delta V = V_e ln\left(\frac{m_0}{m_e}\right) \tag{2.6}$$

2.3.2. EQUATIONS OF MOTION

The motion of the spacecraft is considered using the aforementioned assumptions that the mass of the central body is much larger than the spacecraft's and that this motion can be approximated by a two-body problem where only the thrust acts as an additional force upon the spacecraft. With these assumptions, the spacecraft's equations of motion are as shown in Equation 2.7.

$$\ddot{\mathbf{r}} + \frac{\mu}{r^3}\mathbf{r} = \frac{\mathbf{T}}{m} = \mathbf{u} \tag{2.7}$$

where $\mathbf{r}$ is the position vector, μ is the gravitational parameter of the central body, $\mathbf{T}$ is the thrust vector, m is the mass of the spacecraft and $\mathbf{u}$ is the control acceleration exerted upon the spacecraft. If a certain thrust profile $u(t)$ is known, Equation 2.7 can be integrated numerically to find the corresponding trajectory of the spacecraft. As mentioned previously, the solutions found to this equation when $\mathbf{u}$ is set to zero are the Kepler orbits discussed in Section 2.2.4.

2.3.3. SHAPE-BASED METHODS

An alternative to numerically integrating the equations of motion to obtain a spacecraft's trajectory, is the use of shape-based methods. By assuming a certain shape for the trajectory and deriving the control accelerations needed to follow this orbit from this shape, the computationally expensive integration can be avoided. This shape can be changed such that it matches the boundary conditions by solving the coefficients present in the function that describes the shape. However, as only a limited number of coefficients are present in these functions, there is also a limit on the number of boundary conditions that can be met.

In the past, a number of shape-based methods have been developed. An overview of various shape-based methods, their capabilities and their limitations can be seen in Table 2.1 [5].

Table 2.1: A comparison between the various shape-based methods and their capabilities. In this table, "BC" is an abbreviation for "boundary condition".

Shape-based method	Position BC	Velocity BC	TOF BC	3D	Thrust limit	n_{rev}	Ref.
Spherical	Yes	Yes	Yes	Yes	No	Multi	[1]
Exposin	Yes	No	Yes	Approx.	No	Multi	[8]
Inverse Polynomial	Yes	Yes	Yes	Yes	No	Few	[9]
Improved Inverse Polynomial	Yes	Yes	Yes	Yes	Yes	Multi	[10]
Pseudo-equinoctial	Yes	Yes	Yes	Yes	Yes	Multi	[11]
Hodographic	Yes	Yes	Yes	Yes	No	Multi	[12]

In this table, it can among others be seen that all shape-based methods shown, with the exception of the exposin method, are able to match the position, velocity and time-of-flight boundary conditions. It can also be seen that the thrust acceleration can only be limited when the either the improved inverse polynomial or pseudo-equinoctial shaping method is used and that the inverse polynomial shaping method is not suitable for a large number of revolutions. It should furthermore be mentioned that, while all shape-based methods shown here are capable of generating non-planar trajectories and are thus three-dimensional, no literature was found in which one of the aforementioned methods was shown to be accurate when large orbital inclination were present. For example, only a rough approximation of the out-of-plane motion can be used in the exposin method.

As mentioned in Section 2.1, a large amount of research has already been done on the spherical shaping method within TU Delft and the software written by Roegiers for a portion of this research is readily available. As a partial solution to the aforementioned inaccuracy was furthermore proposed but found to only work in certain cases, it was decided in the literature review that a more accurate description of the out-of-plane motion will be implemented in the spherical shaping method [3].

3

DEVELOPMENT

Now that the problem has been defined and the background knowledge has been presented, the development of the new elevation shaping function can be discussed. This chapter shall first in Section 3.1 elaborate upon the test cases that are used during this development process, after which the development process itself is discussed in Sections 3.2 and 3.3.

3.1. TEST CASES

In order test the developed elevation shaping function, a number of test cases should be defined. This section describes the three type of orbits that the aforementioned elevation shaping function should be capable of reproducing.

3.1.1. UNPERTURBED ORBITS

Naturally, the developed shaping function should be capable of reproducing an unperturbed orbit as a minimum. In order to ensure that the function is capable of reproducing a large variety of mission scenarios, it will be tested whether it can reproduce a number of extreme cases. The first of these two cases, being a near-circular orbit with a semi-major axis of 6778.1 kilometres and an inclination of 0 degrees, can be seen in Figure 3.1. As the elevation shaping function should describe the variation of the elevation angle as a function of the azimuthal angle, this variation has been plotted in the right-hand graph shown in this figure.

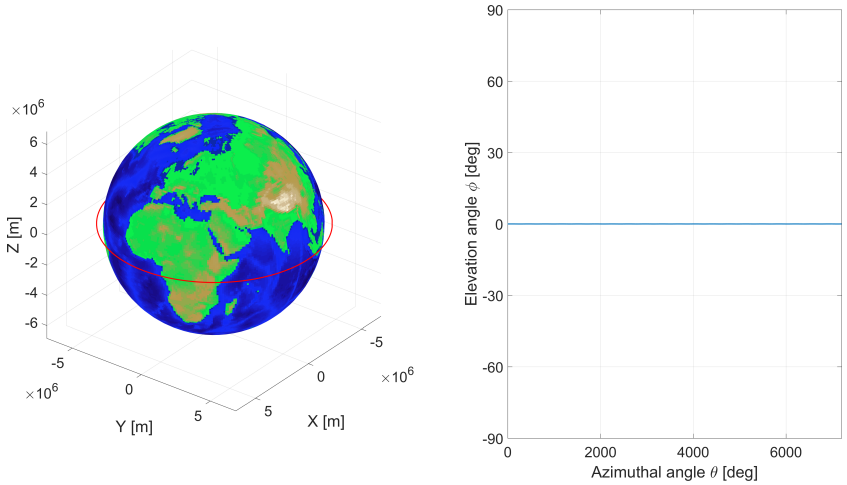


Figure 3.1: Left: simulation of an unperturbed orbit with an inclination of 0 degrees. Right: the variation of the elevation angle as a function of the azimuthal angle.

It should be mentioned here that in each of the test cases, multiple revolutions are used in order to ensure that the shaping function is unaffected by the number of revolutions. Due to certain limitations within the program used to generate the test cases, a small eccentricity of 0.01 is furthermore used instead of a circular

orbit. To determine the trajectory, a Runge-Kutta 4 (RK4) integrator was used. By setting the initial orbital elements and integrating this position over time using a time-step of 2 seconds, the spacecraft's position can be determined over time and the elevation angle as a function of the azimuthal angle can be found. This time-step and integrator was chosen in order to ensure a highly accurate propagation of the trajectory. For example, the error in the semi-major axis at the end of the integration is approximately $1 \cdot 10^{-6}$ meters in the test case shown in Figure 3.1. While this accuracy seems much higher than required, the small value ensures that no significant errors are introduced in the development of the shaping function by the test cases.

The second case is a near-circular orbit with an inclination of 89.99 degrees and an eccentricity of 0.01. The variation of the elevation angle that the shaping function should be capable of reproducing can be found in Figure 3.2.

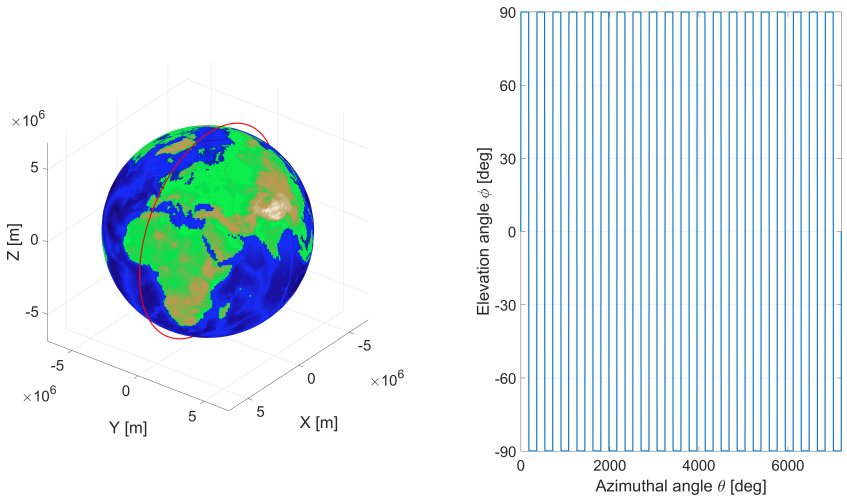


Figure 3.2: Left: simulation of an unperturbed orbit with an inclination of 89.99 degrees. Right: the variation of the elevation angle as a function of the azimuthal angle.

In this figure, it can be observed that the elevation angle as a function of the azimuthal angle closely resembles a set of rectangular functions. This can be seen more clearly in Figure 3.3, where a close-up of Figure 3.2 is shown.

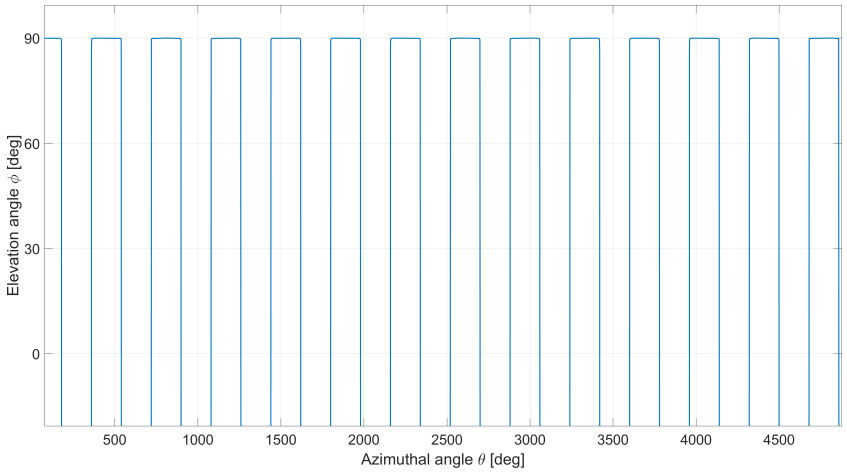


Figure 3.3: Close-up of the change in elevation angle as a function of the azimuthal angle shown in Figure 3.2.

3.1.2. DISCONTINUOUS THRUST

Aside from an unperturbed orbit, the developed elevation shaping function should also be capable of reproducing an orbit of which the orbital elements change due to a discontinuous thrust force.

Firstly, it can be noted that a change in the semi-major axis of the orbit does not influence the elevation angle. This can be seen in Figure 3.4, where the semi-major axis of an orbit with an inclination of 45 degrees is changed over time. This change in semi-major axis is performed by applying a transverse acceleration of 1 m/s^2 on the spacecraft during the time that its true anomaly is within 5 degrees of the perigee. The additional acceleration has been highlighted in this and the subsequent figures with a black line in the left-hand figure. For this integration, the aforementioned RK4 integrator and a time-step of 2 seconds were used again to integrate the spacecraft's position and velocity over 20 revolutions.

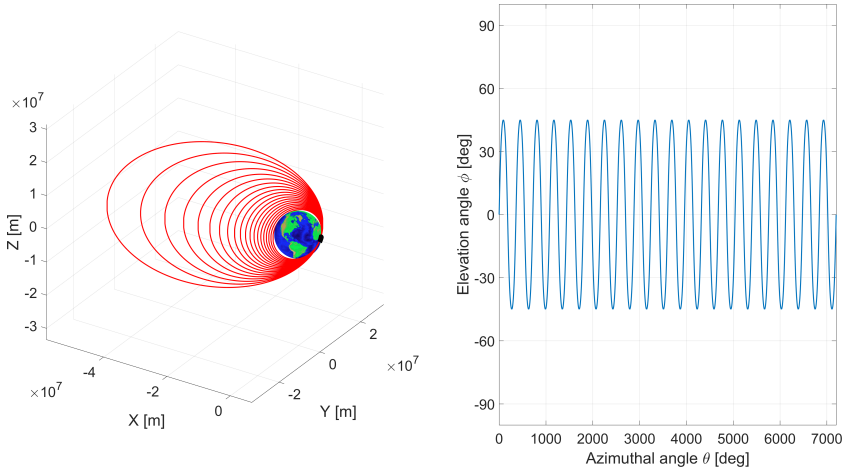


Figure 3.4: Left: the orbit found when an inclination of 45 degrees is used and a transverse acceleration is applied on the spacecraft around the perigee. Right: the change in elevation angle as a function of the azimuthal angle.

As the shaping function uses the azimuthal angle as a variable instead of time, the graph shown on the right in Figure 3.4 does not change as the semi-major axis and the eccentricity increase. This means that the eccentricity and semi-major axis corresponding to a certain variation of the elevation angle are irrelevant to the elevation shaping function. Instead, these two Keplerian elements are described by the radial shaping function. Therefore, even though this thesis focuses on interplanetary trajectories, the orbits around Earth shown in the section can be used in the process of testing the shaping function.

In order to generate the fourth test case, an acceleration of 1 m/s^2 in the out-of-plane direction was exerted on the spacecraft when it was within 5 degrees of the ascending and descending nodes and the thrust direction was reversed when the z-coordinate of the satellite was negative. By doing so until the inclination has changed from 0 to 45 degrees, the variation in the elevation angle shown in Figure 3.5 can be observed.

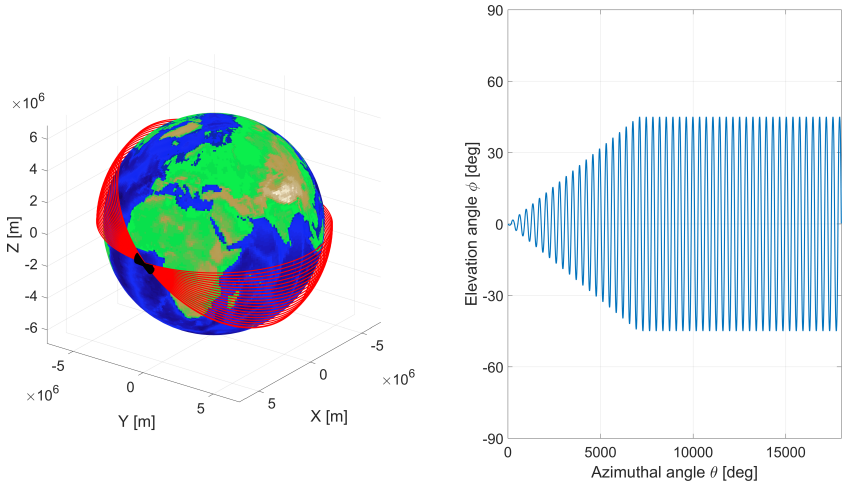


Figure 3.5: Left: the spacecraft's orbit over time when the inclination is changed from 0 to 45 degrees by applying an acceleration of 1 m/s^2 around the ascending and descending nodes. Right: the change in elevation angle as a function of the azimuthal angle.

When the final inclination is required to reach the extreme case of 89.99 degrees, Figure 3.6 can be obtained.

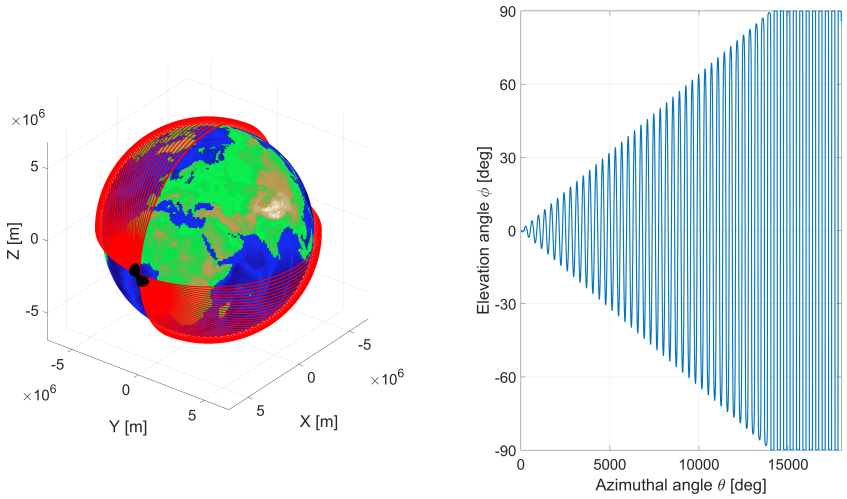


Figure 3.6: Left: the spacecraft's orbit over time in the Cartesian reference frame. Right: the variation of the elevation angle when the inclination is changed from 0 to 89.99 degrees and an acceleration of 1 m/s^2 is applied around the ascending and descending nodes.

By looking closely at the change in the elevation angle, one can observe that the shape of the function's peaks and valleys changes as a function of the azimuthal angle. It can be seen in Figure 3.7 that, as the inclination of the spacecraft approaches 90 degrees, the function describing the elevation angle becomes more rectangular. For the developed shaping function to be able to reproduce this orbit, it should be capable of displaying this change as well.

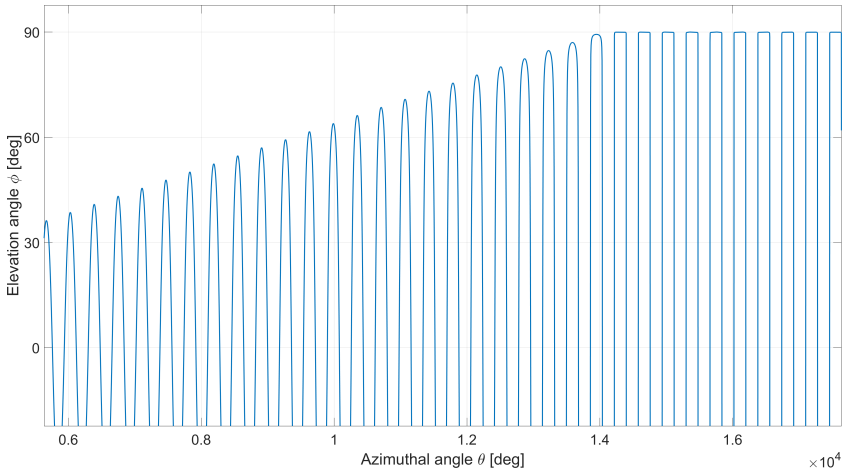


Figure 3.7: Close-up of the change in elevation angle as a function of the azimuthal angle when the inclination is changed from 0 to 89.99 degrees.

So far, only trajectories with a constant or increasing inclination have been tested. However, the shaping function should naturally also be capable of reproducing trajectories with a decreasing inclination. For this reason, a test case with a decreasing inclination is included. The test case consists of a transfer from an orbit with an inclination of 89.98 degrees to an orbit with an inclination of 0.1 degrees. This orbit was generated by applying an out-of-plane acceleration of 1 m/s^2 within a true anomaly range of 5 degrees to both sides of the ascending and descending nodes. The orbit itself can be seen in Figure 3.8.

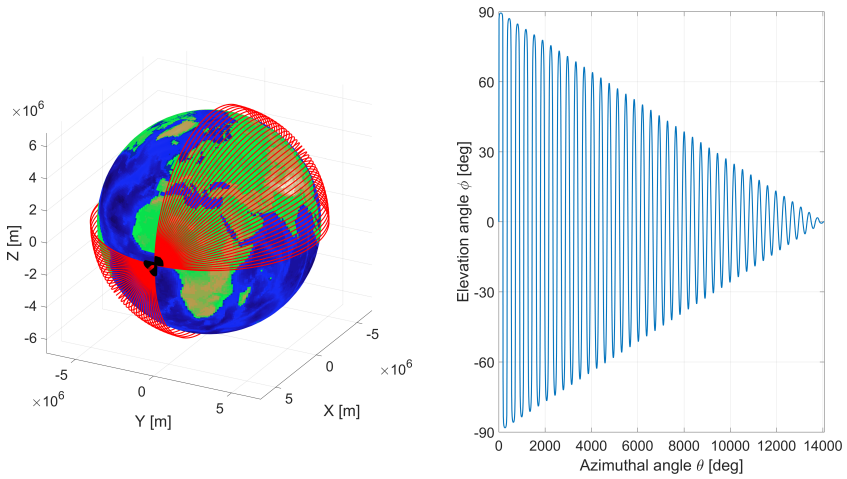


Figure 3.8: Left: the spacecraft's orbit over time when the inclination is decreased from 89.98 to 0.1 degrees by applying an out-of-plane acceleration of 1 m/s^2 around the ascending and descending nodes. Right: the variation of the elevation angle.

3.1.3. CONTINUOUS THRUST

The third type of orbit that the shaping function should be able to reproduce, is an orbit that changes due to a continuous thrust force. An extreme example of such an orbit can be seen in Figure 3.9.

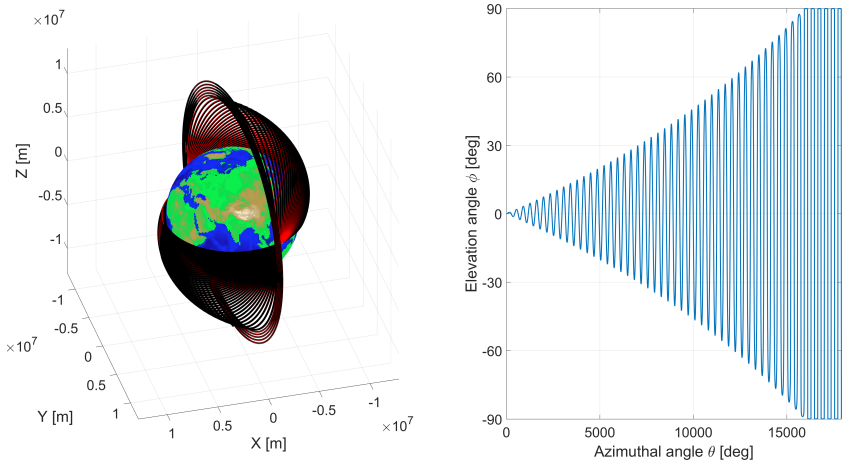


Figure 3.9: Left: the spacecraft's orbit over time in the Cartesian reference frame. Right: the elevation angle as a function of the azimuthal angle when a continuous thrust is applied in both the transverse and out-of-plane direction, increasing the inclination from 0 to 89.99 degrees and varying the semi-major axis.

In order to obtain this trajectory, a transverse acceleration of $5 \cdot 10^{-3}$ m/s^2 and an out-of-plane acceleration of $5 \cdot 10^{-2}$ m/s^2 were applied, where the direction of the out-of-plane acceleration was reversed when the z-coordinate was negative and its value was set to 0 m/s^2 when the final inclination of 89.99 degrees was reached. Interesting to note is that, whereas Figures 3.5 and 3.6 seem to have a linear increase of the maximum elevation angle, an exponential increase can be seen in Figure 3.9.

The test cases shown so far have all included a large number of revolutions and had an initial azimuthal angle of 0 degrees. However, the developed shaping function should also be capable of reproducing solutions to problems that do not have an initial azimuthal angle of 0 degrees and have a small number of revolutions. For this reason, two solutions found by Roegiers will be used as test cases as well. The first case is a transfer to Neptune within a single revolution. The change of the elevation angle as a function of the azimuthal angle can be seen in Figure 3.10 [2].

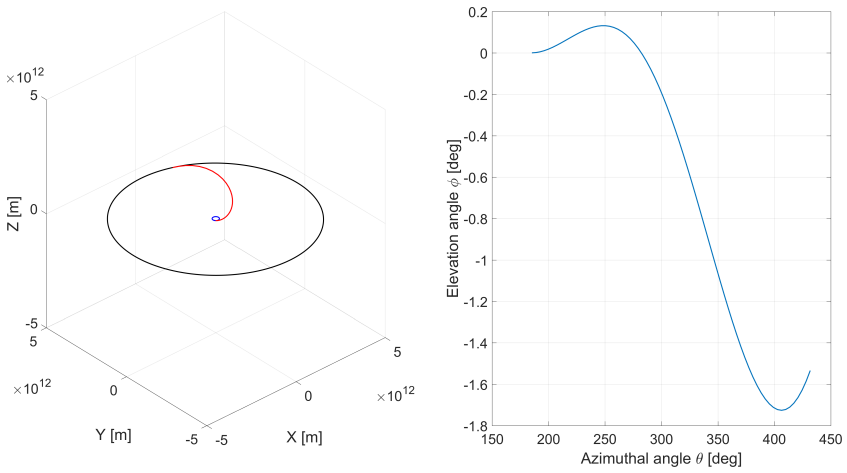


Figure 3.10: Left: the transfer to Neptune in the Cartesian reference frame as found by Roegiers. Right: the elevation angle of the transfer orbit as a function of the azimuthal angle [2].

Furthermore, a 10-revolution transfer to Neptune that was also found by Roegiers will be used as a test case as well. While, as can be seen in Figure 3.11, the rate of change of the elevation angle is less than that of the cases presented earlier, the developed elevation shaping function should not lose functionality over the currently existing function. As such, it is important that these two test cases are also evaluated.

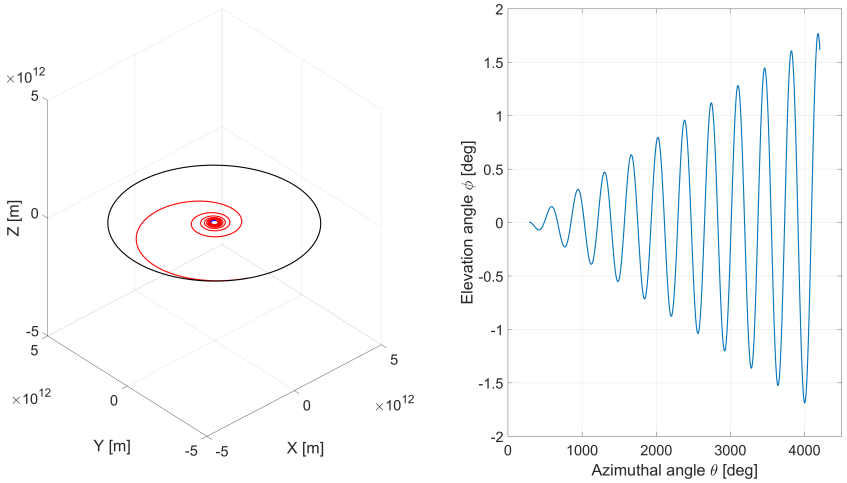


Figure 3.11: Left: the transfer to Neptune that uses 10 revolutions and that was found by Roegiers. Right: the elevation angle as a function of the azimuthal angle [2].

Aside from large inclination changes, the shaping function currently used in the spherical shaping method is unable to accurately represent changes in the RAAN. Therefore, once a potential elevation shaping function capable of describing the aforementioned test cases has been found, it will be investigated whether the function can be further altered such it can describe orbits of which the RAAN changes.

In order to evaluate the ability of the shaping function to model these changes, two test cases are used. The first of these test cases consists of an orbit with an inclination of 45 degrees of which the RAAN changes from 58.3 to 346.8 degrees. This is done by using a constant out-of-plane acceleration of 0.1 m/s^2. When the z-coordinate of the satellite is negative, the thrust direction is reversed. The resulting orbit can be seen in Figure 3.12.

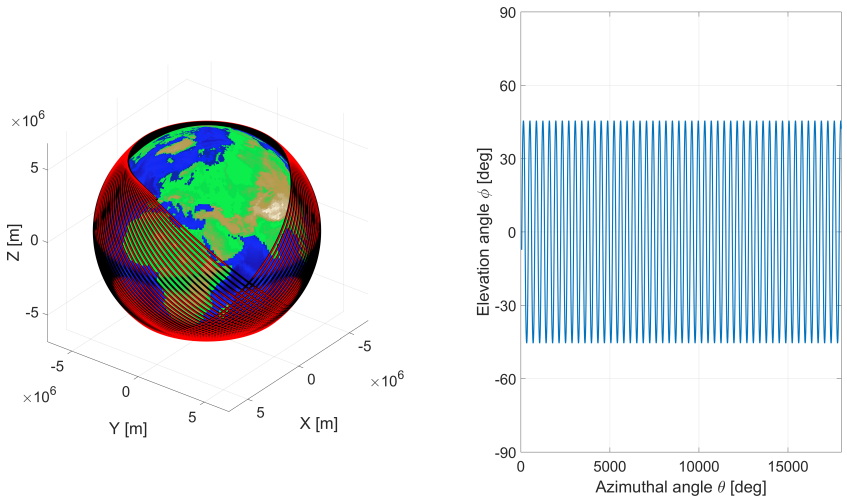


Figure 3.12: Left: the spacecraft's orbit over time when the RAAN is changed from 58.3 to 346.8 degrees through the application of a constant out-of-plane acceleration of 0.1 m/s^2. Right: the elevation angle as a function of the azimuthal angle.

The last test case used has the same initial Keplerian elements as the previous case. However, in this case, the acceleration is 1 m/s^2 and its direction is not reversed when the z-coordinate of the satellite becomes negative. This causes a change in all Keplerian elements. The resulting orbit can be seen in Figure 3.13.

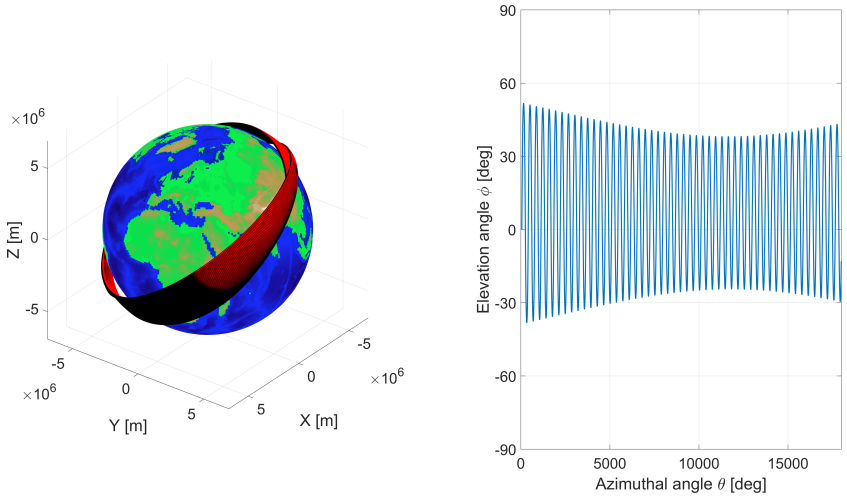


Figure 3.13: Left: the spacecraft's orbit over time when a constant out-of-plane acceleration of 1 m/s^2 is applied of which the direction is not reversed when the z-coordinate of the satellite is negative. Right: the elevation angle as a function of the azimuthal angle.

3.2. UNPERTURBED ORBITS

As a first step in the development process, a number of methods were tested in an attempt to find a shaping function capable of describing an unperturbed orbit. This section shall elaborate on the two methods that were found during the literature study, being the use of spherical triangles and the Fourier series. Aside from these two methods, an additional description that was found during the MSc thesis shall be discussed and evaluated as well.

3.2.1. SPHERICAL TRIANGLES

In the literature review, several relations between the orbital elements that were found by Vinti et al. using spherical triangles were discussed. These relations can be seen in Equation 3.1 [13]. The verification of these

equations can be found in Appendix D.

$$\begin{aligned} sin(\phi) &= sin(i)\, sin(\omega+\theta) \\ cos(\phi) &= cos(\alpha-\Omega)\, cos(\omega+\theta) + sin(\alpha-\Omega)\, sin(\omega+\theta)\, cos(i) \\ cos(\omega+\theta) &= cos(\phi)\, cos(\alpha-\Omega) \\ cos(\phi)\, sin(\alpha-\Omega) &= sin(\omega+\theta)\, cos(i) \end{aligned} \tag{3.1}$$

where the definition of the variables can be seen in Figure 2.1.

Two approaches to finding an elevation shaping function using spherical triangles have been tested, both of which use the relations shown above. The following sections shall elaborate on these approaches and their corresponding results.

FIRST APPROACH

This approach uses the first relation shown in Equation 3.1 in a loop and includes splitting up the orbit into a pre-defined number of sections, each forming a spherical triangle. The definition of the angles is shown in Figure 2.1.

By assuming that the initial elevation angle ϕ_1, the true anomaly change throughout the orbit θ_{orb} and the initial azimuthal angle α_1 are known, the projected initial coordinates of the object on the celestial sphere can be found using Equation 3.2. This projection onto the celestial sphere is required for non-circular orbits as spherical triangles can only be defined on spherical surfaces.

$$\begin{aligned} x_{proj,j} &= r_{cel} cos(\alpha_j)\, cos(\phi_j) \\ y_{proj,j} &= r_{cel} sin(\alpha_j)\, cos(\phi_j) \\ z_{proj,j} &= r_{cel} sin(\phi_j) \end{aligned} \tag{3.2}$$

where r_{cel} is the constant radius of the celestial sphere. For convenience, this can be set to 1.

Once the current position has been projected onto the celestial sphere, the first relation shown in Equation 3.1 can be used to find the elevation angle ϕ_{j+1} of the next point in the orbit's projection on the celestial sphere [3]. The rewritten form of this relation can be seen in Equation 3.3.

$$\phi_{j+1} = asin\left(sin(i)\, sin\left(\omega+\theta_j\right)\right) \tag{3.3}$$

in which i is the inclination, ω is the argument of periapsis and θ_{orb} is the true anomaly.

Furthermore, by using the elevation angle ϕ and true anomaly θ of the next point, as well as the argument of periapsis ω of the orbit, the azimuthal angle of the next point can be found as well with Equation 3.4.

$$\alpha_{j+1} = acos2\left(\frac{cos\left(\omega+\theta_j\right)}{cos\left(\phi_{j+1}\right)}, H\left(\theta_j\right)\right) \tag{3.4}$$

In this equation H is a auxiliary function that is used for the selection of the proper quadrant for α_{j+1}. Its value is 1 if $0 \geq \theta_j < \pi$ and -1 if $\pi \geq \theta_j < 2\pi$. If θ_j is equal to or larger than 2π, the remainder after division by 2π should be used instead to determine the value of H. If the value of H is -1, π is added to the result of the inverse cosine. Otherwise, the unmodified result is used.

By substituting the value obtained from Equation 3.4 and the elevation angle ϕ found from Equation 3.3 into Equation 3.2, the aforementioned loop can be closed and the next point along the orbit can be evaluated.

SECOND APPROACH

This approach uses the same assumptions as the first approach and starts with the calculation of the spacecraft's elevation angle throughout the orbit from the inclination, argument of periapsis and true anomaly using Equation 3.5.

$$\phi_j = asin\left(sin(i)\, sin\left(\omega+\theta_j\right)\right) \tag{3.5}$$

Using the elevation angle together with the third relation shown in Equation 3.1, the longitude can now be found [3]. The rewritten form of the third relation can be seen in Equation 3.6 .

$$\alpha_j = acos2\left(\frac{cos(\omega+\theta_j)}{cos(\phi_j)}, H(\theta_j)\right) + \Omega \tag{3.6}$$

Lastly, the spherical coordinates can be converted into the Cartesian frame with Equation 3.7.

$$\begin{aligned} x &= r_{cel} cos(\alpha)\, cos(\phi) \\ y &= r_{cel} sin(\alpha) \cos(\phi) \\ z &= r_{cel} sin(\phi) \end{aligned} \tag{3.7}$$

RESULTS

The two approaches described previously were tested by simulating the two unperturbed cases discussed in Section 3.1. The results obtained when 100 steps were used to cover a true anomaly change of 2π radians (i.e one revolution) can be seen in Figure 3.14.

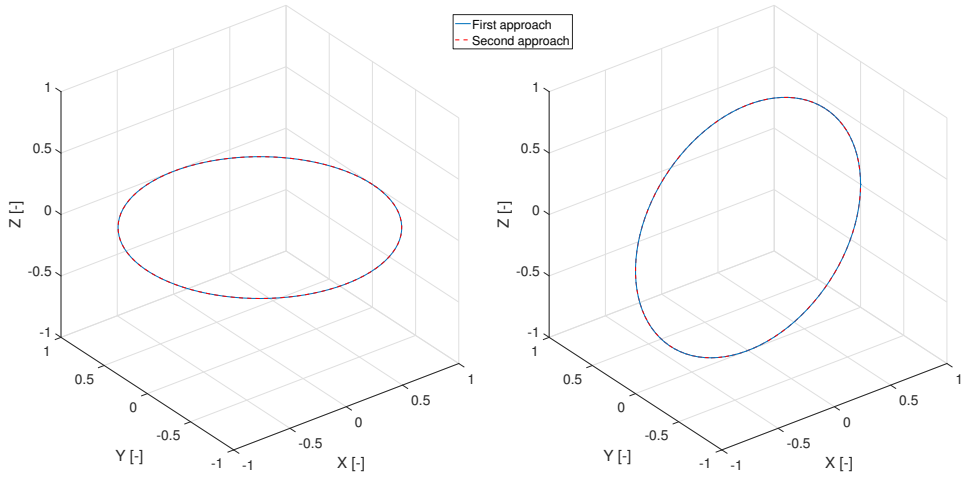


Figure 3.14: Left: the results found when 100 steps were used to model an orbit at an inclination of 0 degrees. Right: the results found when an orbit with an inclination of 89.99 degrees is modelled using 100 steps.

As can be seen in these plots, the two approaches described earlier are capable of representing orbits at inclinations of 0 and 89.99 degrees. It is found for both cases that there is no error in the elevation angle when the inclination is 0 degrees and that when the inclination is 89.99 degrees, the error in the elevation angle at the point where it should theoretically be 89.99 degrees is approximately 1.81 degrees. The latter error can be decreased to approximately 0.17 degrees when 1000 steps are used instead. It should also be noted that coefficients are not yet present in these two approaches and that using these approaches to find a more accurate elevation shaping function is therefore not straightforward. As the result of these two approaches is furthermore a projection of the orbit onto the celestial sphere, an additional step would be needed in order to obtain the trajectory itself. This deprojection can be done if the trajectory's radius as a function of the azimuthal angle is known. However, as this variation of the radius is described by the radial shaping function and as this function is solved at the same time as the elevation shaping function, this need to deproject the orbit could make further development more complicated as well.

3.2.2. FOURIER SERIES

As mentioned in the introduction of this chapter, the second option that was identified for further investigation during the literature study, was the use of a Fourier series. This section shall therefore elaborate on the usage of the Fourier series to find an elevation shaping function capable of simulating an unperturbed orbit.

The Fourier series can be used in various manners in order to simulate the variation of the elevation angle as a function of the azimuthal angle. These various manners include among others the use of different amounts

of coefficients, or combining a shaping function based on the Fourier series with the original shaping function used by Novak [1].

In order to determine the optimal number of coefficients that the shaping function based on the Fourier series should have and whether or not it should be based on the original shaping function, a number of options will be tested in various cases. These options include two shaping functions based on the original shaping function, a shaping function based on the Fourier series and a shaping function based on a polynomial. As the accuracy of each of these options will be evaluated for different amounts of coefficients, the last of these can be used to compare the shaping functions that are based on the Fourier series with a function based on a different mathematical concept. It should be noted that, as increasing the number of coefficients increases the computation time, the number of coefficients should be minimised.

The first shaping function which shall be tested, consists of the original shaping function with additional $cos(\theta)$ and $sin(\theta)$ terms. The function is shown below:

$$\phi = (a_0 + a_1\theta)cos(\theta) + (a_2 + a_3\theta)sin(\theta) + \sum_{i=2}^{n}\left(a_{i+2} + a_{i+3}\theta)cos^n(\theta) + (a_{i+4} + a_{i+5}\theta)sin^n(\theta)\right) \tag{3.8}$$

In this equation, the "priority" of the coefficients is from left to right. Thus, if one for example only desires to have six coefficients, only the coefficients a_0 to a_5 are present and only the additional $cos^n(\theta)$ term is added. By defining this priority, a comparison can be made with another shaping function even if it uses only six coefficients. This priority was based upon the PhD thesis of Novak [1].

If the additional sine and cosine terms are multiplied instead of added, a second option for a shaping function can be obtained. This option is shown in Equation 3.9.

$$\phi = (a_0 + a_1\theta)cos(\theta) + (a_2 + a_3\theta)sin(\theta) + \sum_{i=1}^{n}\left((a_{i+3} + a_{i+4}\theta)cos^n(\theta)sin^n(\theta)\right) \tag{3.9}$$

where the priority of the coefficients is the same as for Equation 3.8.

A third option that is considered, is the use of a standard Fourier series to describe the change of the out-of-plane component as a function of the azimuthal angle. This shaping function is shown in Equation 3.10.

$$\phi = a_0 + \sum_{i=1}^{n}(a_i cos(n \cdot w \cdot \theta) + b_i sin(n \cdot w \cdot \theta)) \tag{3.10}$$

Lastly, the use of a polynomial to describe the out-of-plane component will be evaluated. As mentioned previously, this is done in order to compare the shaping functions that are based on the Fourier series with a function that is based on a different mathematical concept. The general form of the polynomials can be seen in Equation 3.11. As n is the degree of the polynomial in this equation, it can be noted that a polynomial of degree n uses $n+1$ coefficients.

$$\phi = \sum_{i=0}^{n}\left(a_i\theta^i\right) \tag{3.11}$$

The four options described shall be tested using four cases, being a single revolution at an inclination of 45 degrees, three revolutions at an inclination of 45 degrees, a single revolution at an inclination of 89.99 degrees and three revolutions at an inclination of 89.99 degrees. The four options shall approximate the change in elevation angle as a function of the azimuthal angle and the optimal coefficient values are found with the use of MATLAB's curve fitting toolbox and its non-linear least-squares algorithm. As the correct value of the elevation angle can be calculated using Equation 3.12, the root-mean-square error (RMSE) can be determined and the various options can be compared [1]. The verification of this equation can be found in Appendix D.

$$sin\left(\phi\right) = \frac{sin\left(\theta - \Omega\right)sin\left(i\right)}{\sqrt{sin^2\left(\theta - \Omega\right) + cos^2\left(\theta - \Omega\right)cos^2\left(i\right)}} \tag{3.12}$$

in which i is the inclination, Ω is the right ascension of the ascending node (RAAN), θ is the azimuthal angle and the ϕ is the elevation angle. In the following four test cases, the RAAN is arbitrarily set to 0 degrees.

A first test case that will be evaluated, is the simulation of a single revolution at an inclination of 45 degrees. The RMSE as a function of the number of coefficients for the four options can be seen in Figure 3.15. In this figure, the dotted line indicates a reasonable target RMSE of 3 degrees.

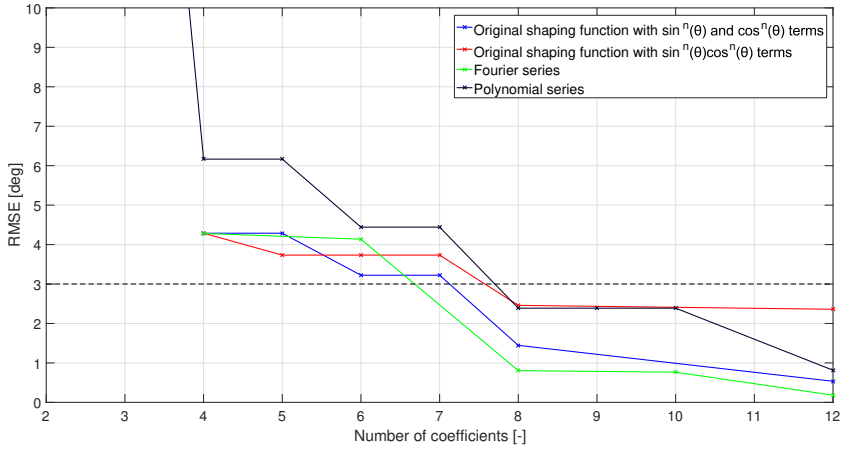


Figure 3.15: The RMSE as a function of the number of coefficients for the four options when a single revolution at an inclination of 45 degrees is simulated. The results for the polynomial series with two and three coefficients have been omitted for clarity. The RMSE is 26.92 degrees in both cases.

Interesting to note here, is that with the use of a polynomial, at least eight coefficients are needed to obtain an RMSE of less than 3 degrees, but that adding additional coefficients has barely any affect. This can also be seen in Figure 3.16, where a comparison between the variations of the elevation angle as a function of the azimuthal angle has been plotted.

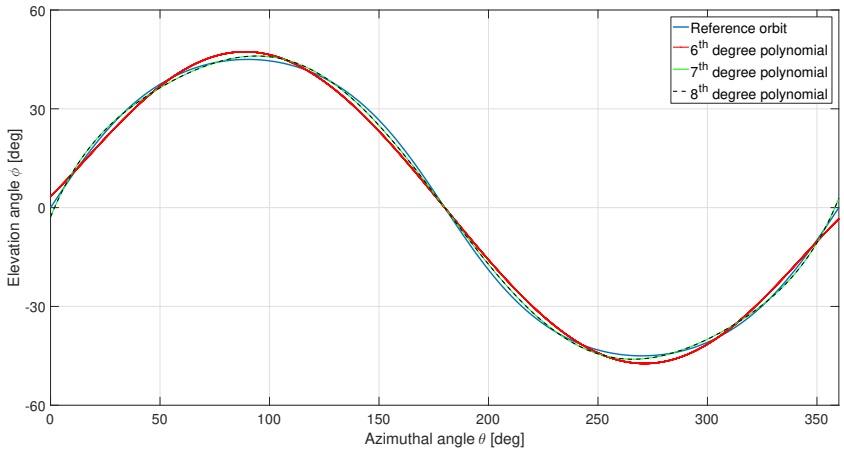


Figure 3.16: A comparison between the use of a sixth, seventh and eighth-degree polynomial.

It can be seen here that a sixth-degree polynomial does not accurately follow the variation of the elevation angle and that seventh-degree polynomial is capable of doing this more accurately. The eighth-degree polynomial overlaps the seventh-degree polynomial. This can be confirmed with the fact that the coefficient of the θ^8 term is approximately $-1 \cdot 10^{-15}$, resulting in a shaping function highly similar to a seventh-degree polynomial.

As the shaping function should also be capable of reproducing the change in elevation angle over multiple revolutions, the second case that is used consists of three revolutions at an inclination of 45 degrees. The variation of the RMSE over the number of coefficients for the four options can be seen in Figure 3.17.

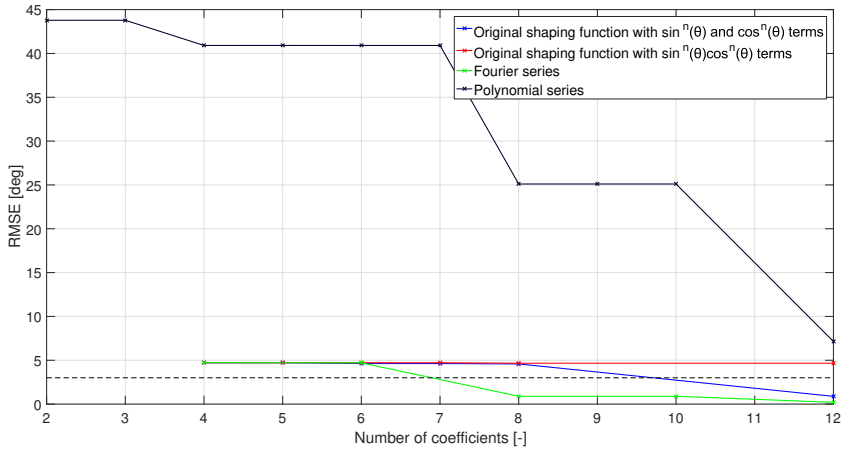


Figure 3.17: The variation of the RMSE over the number of coefficients for the four options when three revolutions at an inclination of 45 degrees are simulated.

In this figure, it can be seen that the polynomial performs considerably worse than all other options, while the Fourier series requires the least amount of coefficients to obtain an RMSE less than 3 degrees and is unaffected by the number of revolutions. A comparison between these two options when eight coefficients are used can be seen in Figure 3.18.

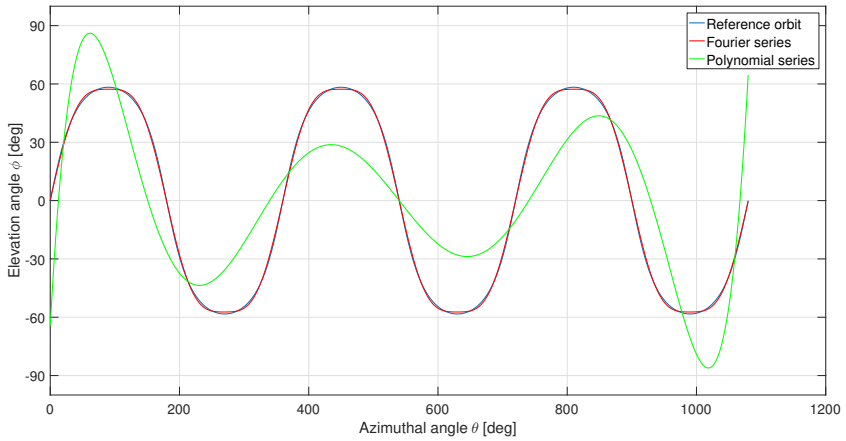


Figure 3.18: A comparison between the use of a Fourier series and a polynomial when eight coefficients are used in their description.

Important to note from this figure, is that the increase in the RMSE for the Fourier series is mostly caused around the peaks and the valleys instead of the roots, as the difference with the reference orbit is larger here. As the errors near the roots are small, it is expected that the Fourier series is also capable of representing more than three revolutions due to the repeating pattern. The polynomial series on the other hand, is clearly incapable of doing so.

In order to further compare the four options, the inclination is increased to the 89.99 degrees of the second test case shown in Section 3.1. The variation of the RMSE for the four options when a single revolution is used as reference case can be found in Figure 3.19.

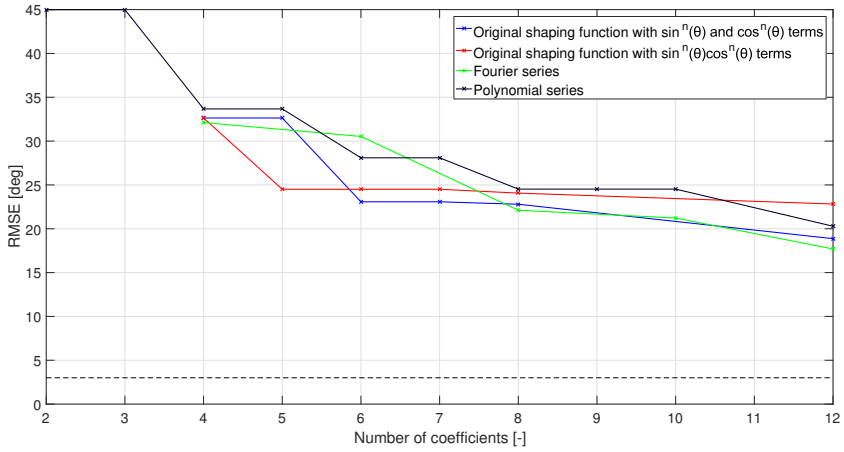


Figure 3.19: The RMSE as a function of the number of coefficients when one revolution at an inclination of 89.99 degrees is simulated.

It can be seen in this figure that even the use of 12 coefficients is no longer sufficient for any of the options. This is caused by the sharp corners in the reference orbit's elevation change (cf. Figure 3.2).

Lastly, the case of three revolutions at the aforementioned inclination of 89.99 degrees is tested. The RMSE as a function of the number of coefficients can be seen in Figure 3.20.

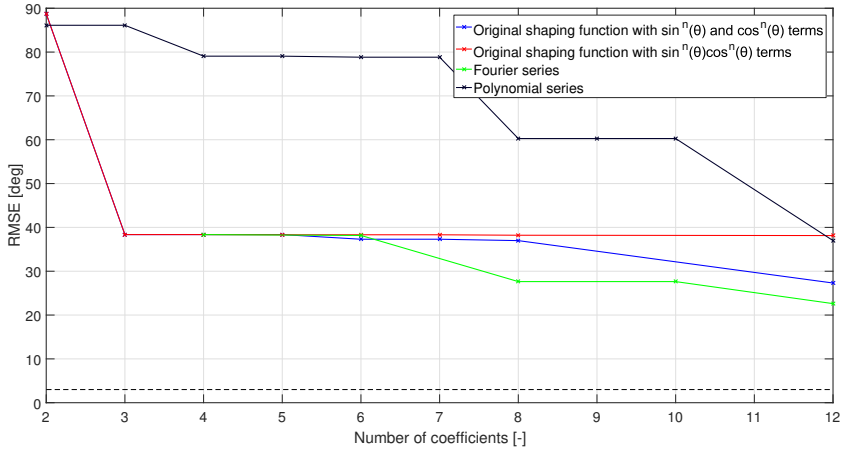


Figure 3.20: The RMSE as a function of the number of coefficients when three revolutions at an inclination of 89.99 degrees are simulated.

In this plot, a clear difference between the polynomial, the Fourier series and the shaping functions based on the original shaping function can be seen. While none of the shaping functions is capable of accurately describing the orbit, it can be seen in particular that the polynomial has a very large misfit.

To summarise the plots shown, a comparison between the four options can be found in Table 3.1. In this table, the green background indicates the option with the lowest RMSE when the specified number of coefficients is used for each case. From this summary, it can be concluded that the Fourier series consistently performs well compared to the other shaping functions. This can be seen most clearly when three revolutions are simulated. In particular, it can be seen that the third-order Fourier series performs better than the other shaping functions when eight coefficients are used and that increasing the number of coefficients to 12 does not provide as much of an improvement as increasing the number of coefficients from six to eight does. Due to the fact that the number of coefficients should be minimised in order to minimise the computation time, the usage of 12 coefficients is deemed to be inefficient. It is therefore concluded that, of the shaping functions discussed in this section, the third-order Fourier series has the most potential of resulting in an accurate description of the elevation angle. However, it should be noted that the shaping function's accuracy at high inclinations still needs to be improved.

Table 3.1: A comparison between the RMSE in degrees of the four options for the cases presented.

	1 Rev. at 45 deg	3 Revs. at 45 deg	1 Rev. at 89.99 deg	3 Revs. at 89.99 deg
Shaping F1 6 coeff	3.2	4.6	23.1	37.3
Shaping F2 6 coeff	3.7	4.7	24.5	38.3
Fourier Series 6 coeff	4.1	4.7	30.5	38.2
Polynomial 6 coeff	4.4	40.9	28.1	78.8
Shaping F1 8 coeff	1.4	4.6	22.8	37.0
Shaping F2 8 coeff	2.5	4.7	24.1	38.2
Fourier Series 8 coeff	0.8	0.9	22.1	27.6
Polynomial 8 coeff	2.4	25.1	24.5	60.3
Shaping F1 12 coeff	0.5	0.9	18.9	27.3
Shaping F2 12 coeff	2.4	4.7	22.8	38.1
Fourier Series 12 coeff	0.2	0.2	17.7	22.6
Polynomial 12 coeff	0.1	0.9	15.5	27.6

3.2.3. ALTERNATIVE SHAPING FUNCTION

Aside from the spherical triangles and Fourier series, an alternative description is tested as well. While researching the various functions that could be used to describe the unperturbed orbit, Equation 3.13 was found [14]. This section shall elaborate upon the application of it to the unperturbed orbit problem.

$$\phi = \sqrt{\frac{1+a^2}{1+a^2 cos^2(\theta)}} cos(\theta) \tag{3.13}$$

where a is a coefficient that can be tuned.

In order to illustrate the coefficient a, the resulting curves when various values of a are used have been plotted and can be seen in Figure 3.21.

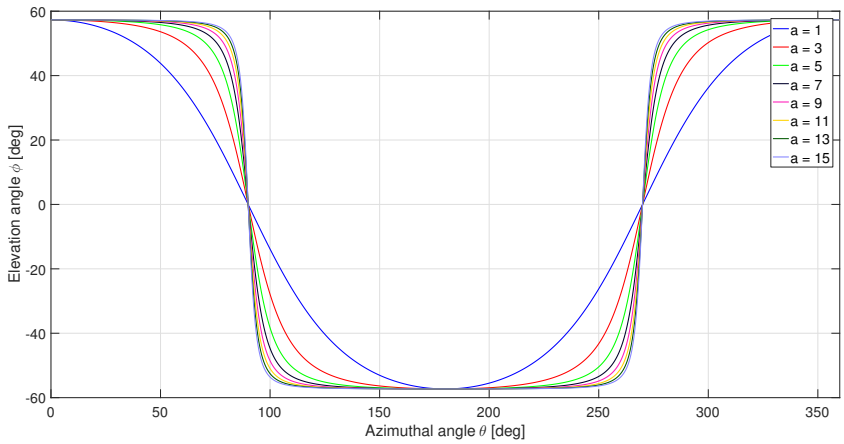


Figure 3.21: The variation of the elevation angle as a function of the azimuthal angle for various various values of a.

In this figure, it can be seen that an increase in the coefficient a results in the curve approaching a rectangular function and that the function has a maximum of $\phi = 1$ rad (this being approximately 57 degrees). The latter can easily be confirmed by substituting an azimuthal angle of 0 degrees into Equation 3.13.

As the function should also be able to represent cases where the initial elevation angle is for example 0 radians, a phasing coefficient p is added. This results in the following equation:

$$\phi = \sqrt{\frac{1+a^2}{1+a^2 cos^2(\theta+p)}} cos(\theta+p) \tag{3.14}$$

Furthermore, a third coefficient is added in order to enable the function to simulate various orbital inclinations. The resulting function is shown in Equation 3.15.

$$\phi = \sqrt{\frac{1+a^2}{1+a^2 cos^2(\theta+p)}} cos(\theta+p) \cdot b \tag{3.15}$$

To test the accuracy of this function, its ability to simulate the cases shown in Section 3.2.2 is evaluated with the use of the MATLAB's curve fitting toolbox. The RMSE as found for these four cases is shown in Table 3.2.

Table 3.2: The RMSE in degrees of the proposed shaping function for four cases.

1 Rev. at 45 deg	3 Revs. at 45 deg	1 Rev. at 89.99 deg	3 Revs. at 89.99 deg
0.03	0.03	0.2	0.2

By comparing these values with the ones shown in Table 3.1, it can be seen that this alternative formulation performs significantly better than the options discussed in Section 3.2.2 and that it is unaffected by the number of revolutions. The performance of the proposed shaping function can also be observed in Figure 3.22, where the function is used to simulate three revolutions at an inclination of 89.99 degrees.

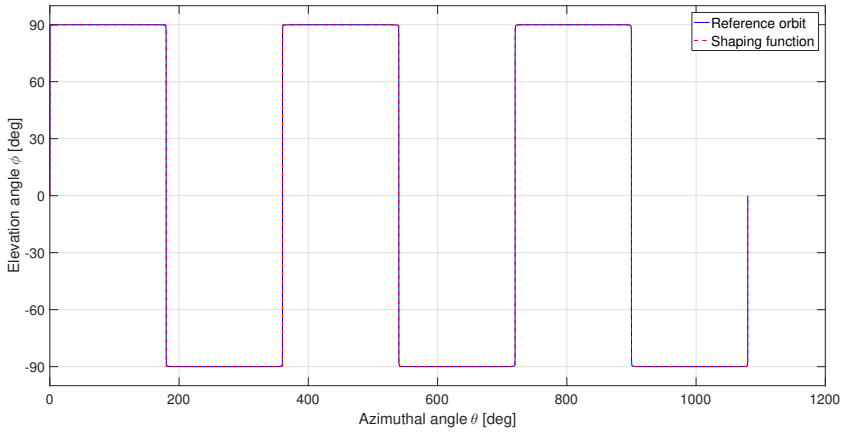


Figure 3.22: The performance of the proposed shaping function when it is used to simulate three revolutions at an inclination of 89.99 degrees.

It can be seen in this figure that the proposed shaping function is capable of following the reference orbit with a much higher accuracy, even though it uses less coefficients than the shaping functions described previously.

3.3. PERTURBED ORBITS

As a next step in the development process, the shaping functions suggested in Section 3.2 will be used to describe a perturbed orbit. In the test cases evaluated in this chapter, the perturbation is solely caused by the thrust force of the spacecraft.

By comparing the proposed shaping functions, it can be seen that only the spherical triangles do not include coefficients that allow the shaping function to be adapted to the boundary conditions. As there are also potential issues with the required projection of the orbit onto the celestial sphere and the required deprojection, it can be concluded that the spherical triangles have the least potential. Due to the limited time frame of the master's thesis, it was decided that the potential of the alternative shaping function and the Fourier series was first evaluated. If neither of these shaping functions could be used, the elevation shaping function that uses the spherical triangles would be developed further. This turned out to be unnecessary.

3.3.1. ALTERNATIVE SHAPING FUNCTION

From Section 3.2.3, it can be noted that the alternative shaping function is currently only capable of following orbits with a constant inclination due to the constant value of the coefficients a and b. As such, it can be concluded that the values of these coefficients need to change with the azimuthal angle for the function to be capable of accurately following perturbed orbits.

In order to determine the potential of the alternative shaping function, the change of the equations' coefficients is modelled using the test cases shown in Section 3.1. This is done by dividing the perturbed orbit into a number of unperturbed orbits, each having a constant inclination. By using the non-linear least-squares method and MATLAB's curve-fitting toolbox to fit the function to these orbits, the change of the coefficients can be plotted as a function of the azimuthal angle.

COEFFICIENT a

When the aforementioned methodology is applied to cases 1, 3 and 4, Figure 3.23 can be found.

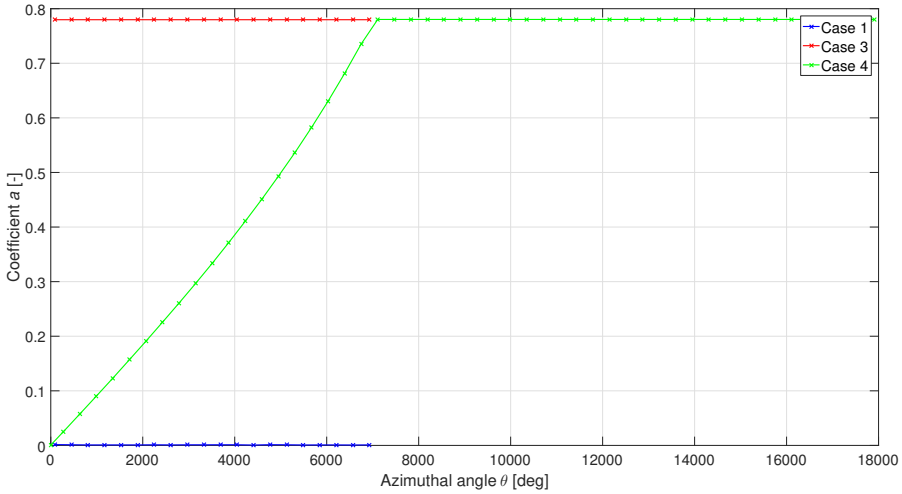


Figure 3.23: The variation of the coefficient a for cases 1, 3 and 4.

As the coefficient a is a function of inclination, the curves for cases 1 and 3 are as expected, since the inclination is constant in these test cases. Interesting to note, is that while the change in inclination over the azimuthal angle appears to be linear in Figure 3.5, the curve describing the coefficient a is not.

When the change of a is plotted for the cases 2, 5, 6 and 7, the following curves are obtained:

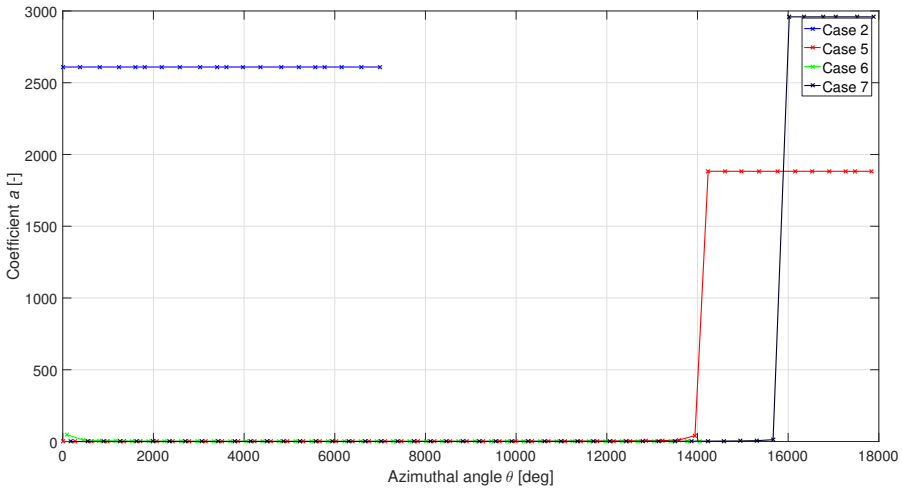


Figure 3.24: The change of the coefficient a for cases 2, 5, 6 and 7.

From this figure, one can conclude that at high inclinations, the value of a rather suddenly becomes significantly larger, while a constant high inclination results in a higher constant value of a.

By looking more closely at the smaller values of a for cases 5, 6 and 7, Figure 3.25 can be obtained.

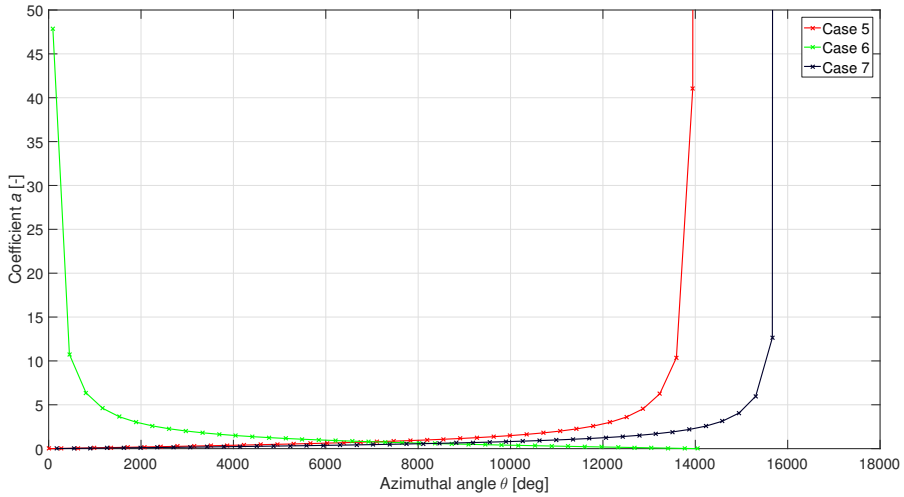


Figure 3.25: Close-up of the variation of the coefficient a for cases 5, 6 and 7.

Here it can clearly be seen that the coefficient's value exponentially increases and decreases as the orbit's inclination increases and decreases as well.

For cases 8 and 9, the following curves are obtained:

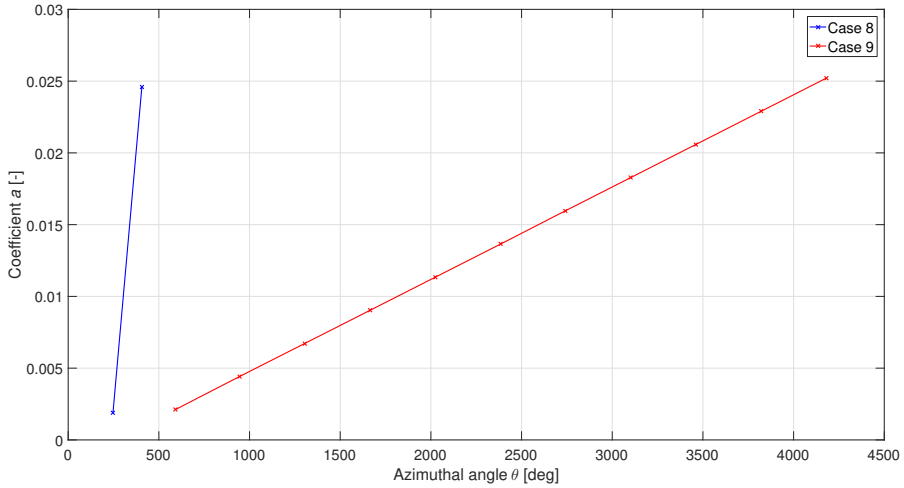


Figure 3.26: The variation of the coefficient a for cases 8 and 9.

In order to obtain more information on the behaviour of a as a function of inclination, the coefficient was determined for various inclinations using non-linear least-squares fitting. By determining the coefficient a 10 times for each inclination while using random starting values for the non-linear least-squares fitting algorithm, one can obtain additional insight in the variation of a. The mean value of a that was found can be seen in Figure 3.27 as a function of inclination.

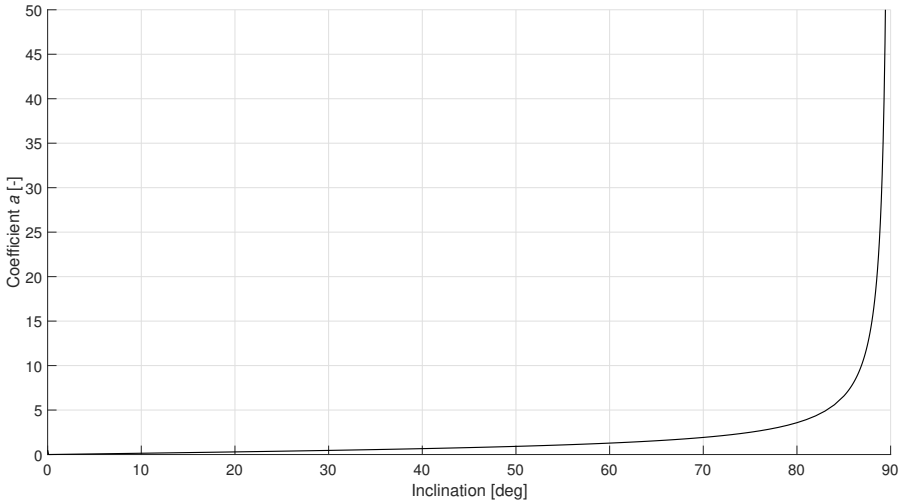


Figure 3.27: The value of the coefficient a as a function of inclination.

From this, it can be seen that when the inclination increases linearly, the coefficient a increases exponentially. However, it should be mentioned that in order to create this plot, the absolute value of a was used. As a^2 is used in the shaping function, the non-linear least-squares algorithm also found $-a$ as a solution. Because this only changes the sign of a, it is expected that this does not impact the accuracy of the shaping function. After taking the absolute values, the maximum standard deviation of a was found to be 0.27 at an inclination of 0 degrees. This indicates that the same value for a will be found regardless of the starting point of the non-linear least-squares algorithm when the absolute value is used.

COEFFICIENT b

By using the same methodology used for the coefficient a, the change of the coefficient b as a function of the azimuthal angle can be found as well. By doing so for cases 1 to 7, Figure 3.28 is obtained.

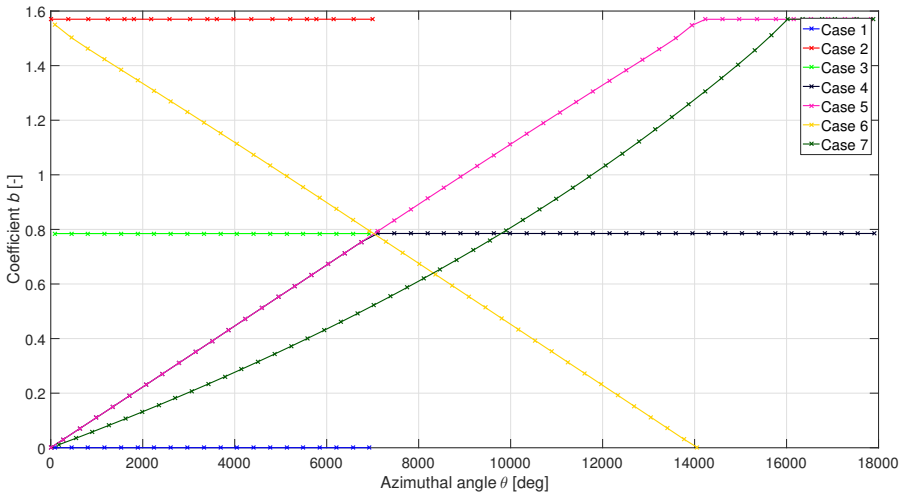


Figure 3.28: The variation of the coefficient b for cases 1 to 7.

It can be seen in this figure that, as expected, b changes linearly as a function of inclination. The variation of the coefficient b as a function of the azimuthal angle for cases 8 and 9 can be seen in Figure 3.29.

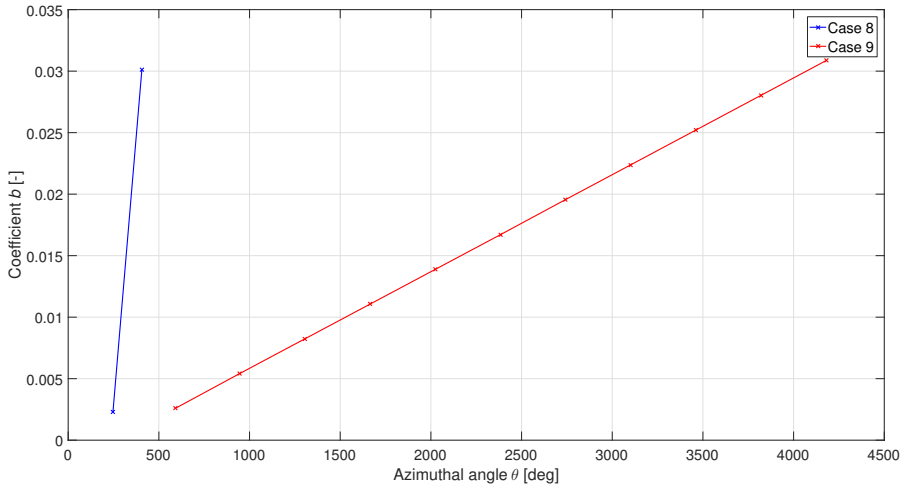


Figure 3.29: The variation of the coefficient b for cases 8 and 9.

In order to confirm the linear behaviour of the coefficient b, its value was determined as a function of inclination using the same methodology used for a. The resulting curve can be seen in Figure 3.30.

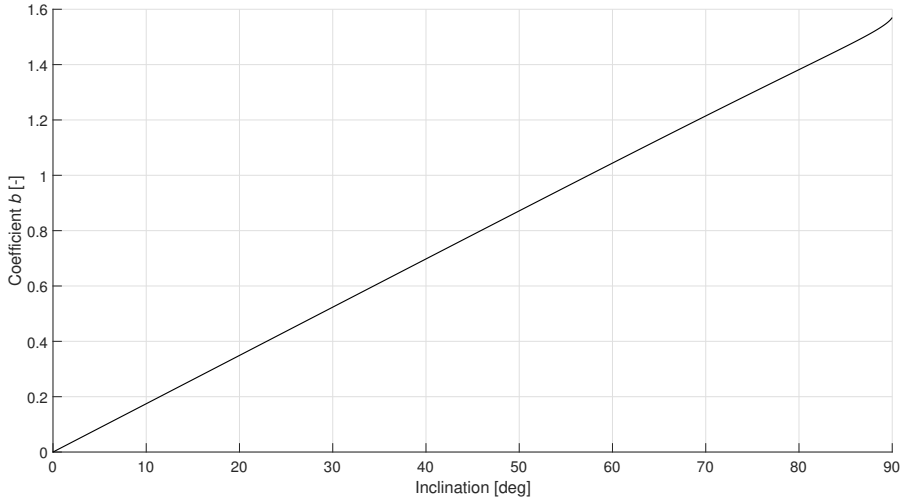


Figure 3.30: The value of the coefficient b as a function of inclination.

In this figure, it can be seen that b indeed has a linear relation with inclination. The reason for this is of course that b solely changes the amplitude of the curve, as can be seen in Equation 3.15. Because a only changes the shape of the curve instead of its amplitude, b should be equal to the inclination in radians. However, it can be seen in the top right corner of Figure 3.30 that the curve is not quite linear. To inspect the cause of this, the difference between the theoretical value of b, being the inclination in radians, and the obtained value of b was calculated. The result of this is a skewed curve with a peak of 1.06 degrees. The shape of this curve is highly similar to the variation of the shaping function's RMSE as a function of inclination, shown here in Figure 3.31. This suggests that the aforementioned non-linearity of b is caused by inaccuracies of the proposed shaping function itself. It can be seen in this figure that the RMSE stays below 1.4 degrees at all inclinations. Furthermore, it should be noted from this figure that the RMSE does not continuously increase as the inclination approaches 90 degrees. Instead, the maximum RMSE occurs at an inclination of 87.7 degrees.

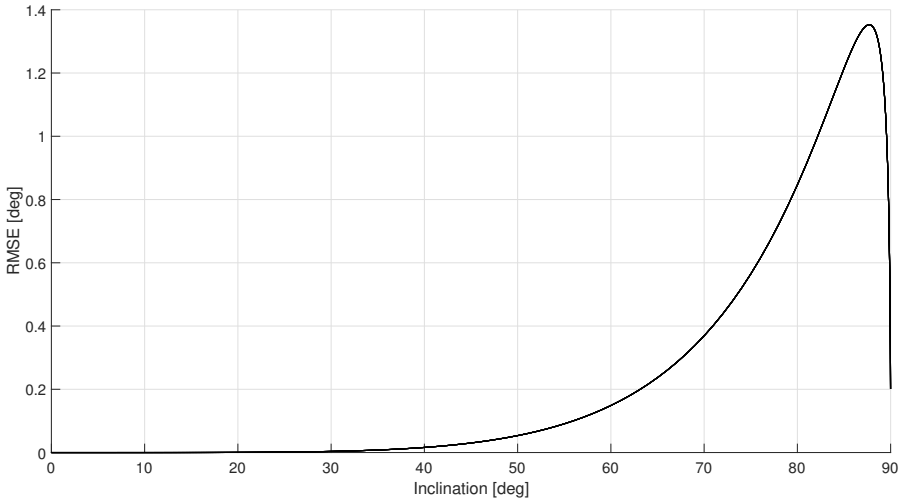


Figure 3.31: The RMSE of the alternative shaping function versus the inclination.

3.3.2. FOURIER SERIES

Using a similar methodology to the one described previously, the variation of the coefficients of the third-order Fourier series can also be found as a function of the azimuthal angle. As the proposed function has eight coefficients and because not all curves can fit in a single axis interval, a large number of figures are needed to describe the behaviour of the coefficients. Therefore, the majority of the figures have been added to Appendix E and only the plots characterising the overall behaviour of the coefficients are shown here.

Firstly, the behaviour of the coefficient a_0 can be evaluated. The variation of the coefficient's value as a function of the azimuthal angle for all eight cases can be seen in Figure 3.32.

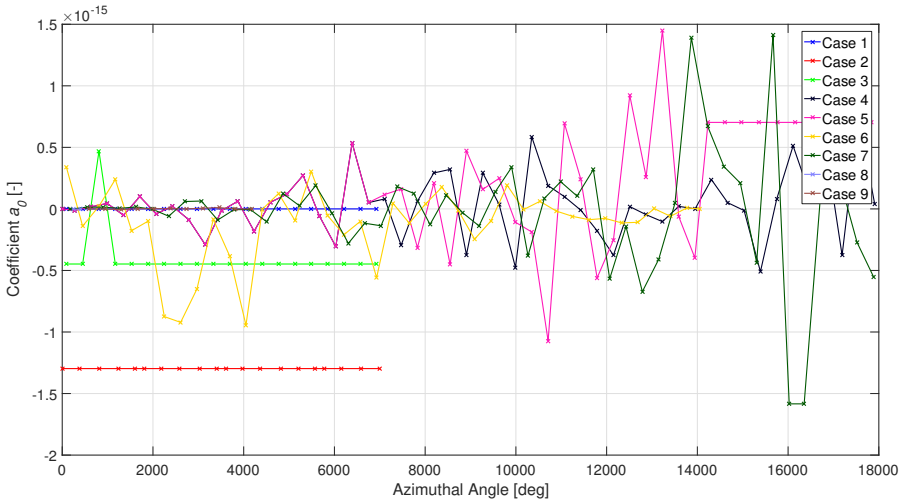


Figure 3.32: The variation of the coefficient a_0 for cases 1 to 9.

From this figure, it can be observed that the variation of the coefficient's value is extremely small and highly random. As all values are of the order 10^{-15}, it can be assumed that they are approximately zero.

A second figure that characterises the behaviour of the coefficients, is the variation of a_1 for cases 1, 3 and 4. Their respective curves can be seen in Figure 3.33.

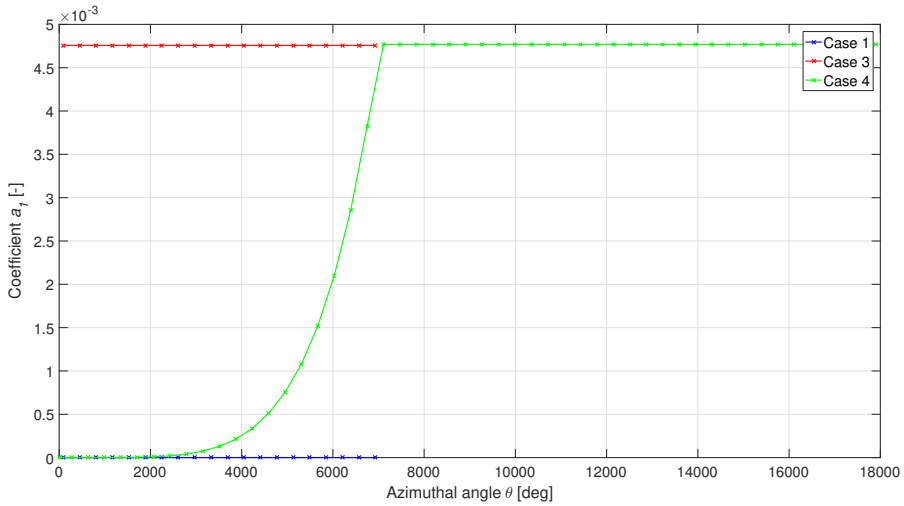


Figure 3.33: The variation of the coefficient a_1 for cases 1, 3 and 4.

Interesting to note here is that for case 4, a_1 starts to increase exponentially but continues to increase linearly from a certain point onward. To obtain an accurate representation of the trajectory, both of these sections should be reproduced. It can also be seen in Figure 3.33 that the values for cases 1 and 3 stay constant as expected.

Thirdly, one can note from Figure 3.34 that the variation of a_1 for cases 8 and 9 is highly similar to the ones shown in Section 3.3.1.

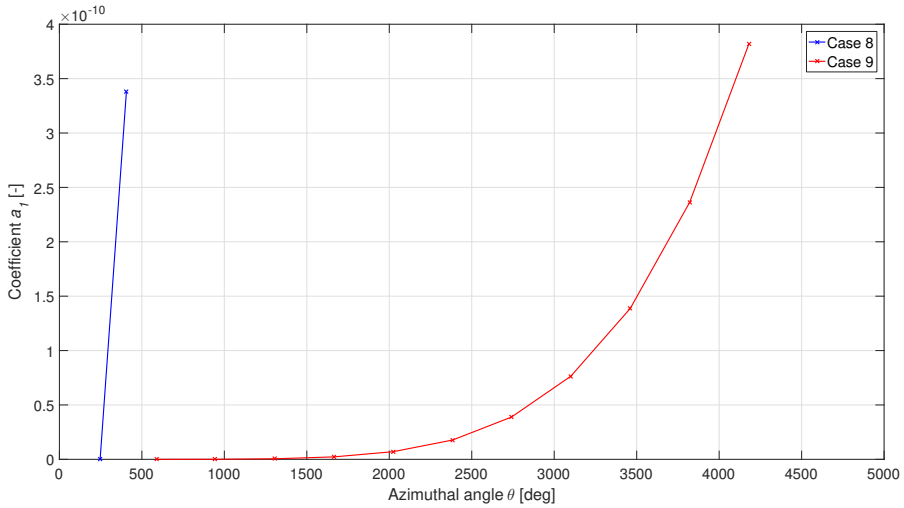


Figure 3.34: The variation of a_1 for cases 8 and 9.

As a last indication of the overall behaviour of the Fourier coefficients, the variation of the coefficient b_3 as a function of the azimuthal angle is shown for cases 1 to 7 in Figure 3.35.

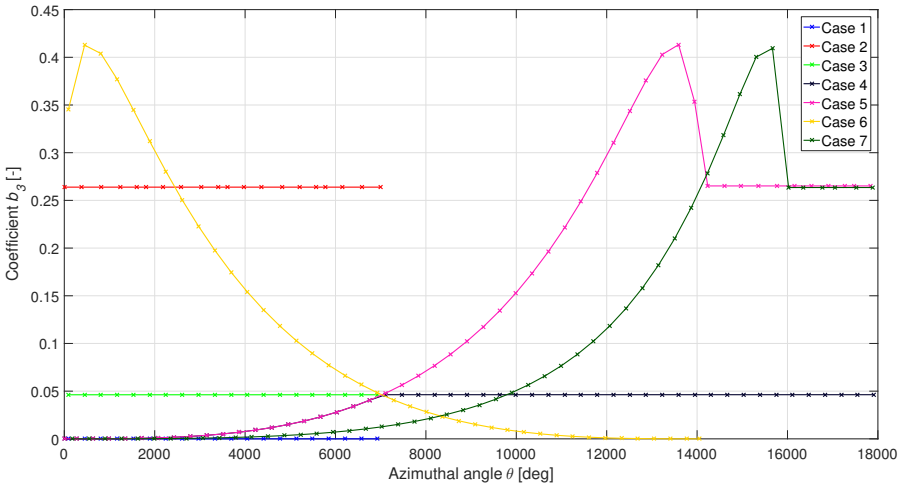


Figure 3.35: The variation of the coefficient b_3 for cases 1 to 7.

In this figure, the curves of cases 5, 6 and 7 stand out in particular, as they display a different type of behaviour from the coefficients shown so far. In order to determine the source of this behaviour, the variation of the coefficients has also been plotted as a function of inclination. The result for the coefficient b_3 can be seen in Figure 3.36 and the curves of the other coefficients can be found in Appendix F.

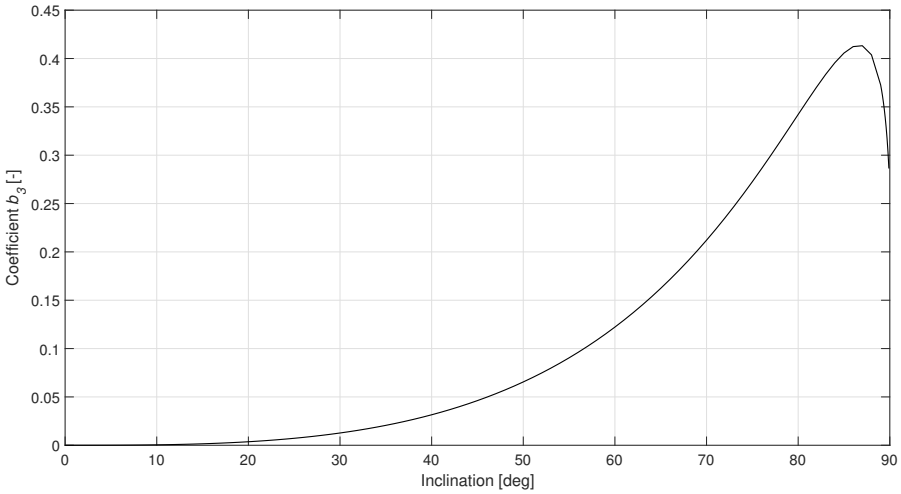


Figure 3.36: The value of the coefficient b_3 as a function of inclination.

As the curve shown here is highly similar to the one shown in Figure 3.35, it can be concluded that the behaviour shown in Figure 3.35 is a result of the behaviour of b_3 itself as a function of inclination. It should be mentioned that the curve shown in Figure 3.36 was obtained by estimating the coefficient 10 times for each inclination using non-linear least-squares and plotting the variation of the mean value. The maximum standard deviation of b_3 was found to be $5.9 \cdot 10^{-17}$ at an inclination of 73 degrees, thus indicating that the overall estimation of the coefficient was highly accurate.

Nonetheless, it can be seen in Figure 3.36 that there is not always a unique relation between the value of the coefficient b_3 and the inclination, as multiple inclinations use the same value for b_3. If the third-order Fourier series is used in the development of the alternative elevation shaping function, the aforementioned characteristic could impact the robustness of the resulting spherical shaping method. Aside from that, the rapidly increasing RMSE shown in Table 3.1 also has a significant impact on the quality of the results produced by the spherical shaping method. While the significance of the RMSE was also discussed in Section 3.2, only the RMSE at an inclination of 45 and 89.99 degrees was discussed in this section. As such, the variation of the RMSE as a function of inclination has been plotted and can be seen in Figure 3.37. One can conclude using

this figure that, aside from a non-unique coefficient b_3, a significant error in the approximation of the orbit is also present at high inclinations.

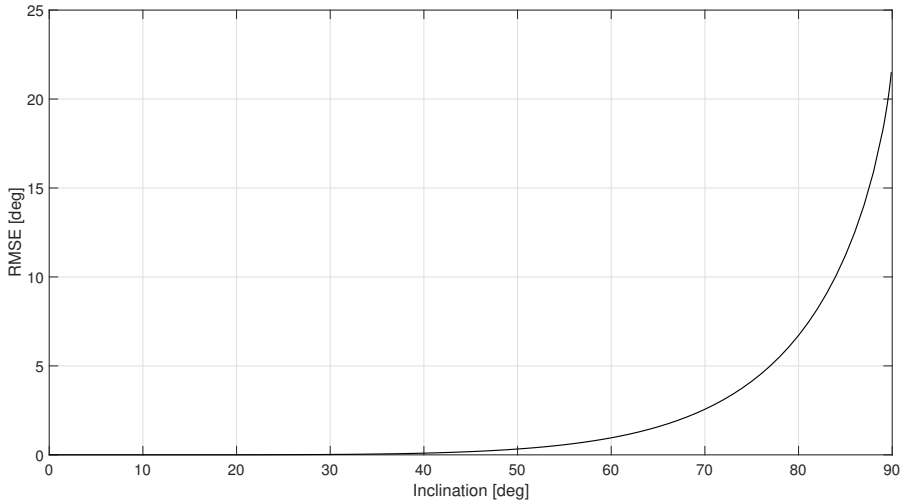


Figure 3.37: The RMSE of the third-order Fourier series as a function of inclination.

3.3.3. TRADE-OFF

In the previous two sections, the potential for the alternative shaping function and the third-order Fourier Series to simulate perturbed orbits was discussed. This section shall perform a trade-off and discuss the choice of the shaping function which shall be developed further.

Firstly, one can see from Figures 3.31 and 3.37 that the RMSE of the alternative shaping function is significantly smaller than that of the third-order Fourier series, especially at larger inclinations. Naturally, this is therefore a large advantage of the alternative shaping function.

Secondly, by comparing the number of coefficients, it can also be concluded that the alternative shaping function has less coefficients that should be estimated compared to the third-order Fourier series. While the coefficients will need to be rewritten such that they can vary as a function of the azimuthal angle, this is the case for both the Fourier series and the alternative shaping function. This in turn means that the usage of the Fourier series could result in a significant increase in the number of coefficients, each of which should be determined in order to obtain a trajectory. In particular, it is expected that a relatively large number of coefficients are needed to follow curves such as the ones of cases 5 and 6 in Figure 3.35. As increasing the number of coefficients also increases the computation time, the usage of less coefficients is favourable. Noteworthy is however that at least four coefficients are required in order to satisfy the boundary conditions on the out-of-plane shaping function [1].

From these observations, it can be concluded that the alternative shaping function has significantly more potential than both the spherical triangles and the third-order Fourier series. It is therefore decided that the alternative shaping function is further developed into a usable shaping function for the out-of-plane component of the spherical shaping method.

3.3.4. COEFFICIENT FUNCTIONS

With the general form of the shaping function decided upon, the coefficients currently present in it can be replaced by expressions such that the shaping function can directly model the cases presented earlier. It should be mentioned that, in an attempt to simplify the resulting shaping function, the expressions used will replace a^2 instead of the coefficient a. However, as a^2 cannot become smaller than zero, boundaries are placed upon the expressions that describe the variation of a^2 such that the result does not become smaller than zero either.

To find expressions for the coefficients, combinations of various functions will be used to model the test cases shown in Section 3.1. By substituting expressions for the different variables into Equation 3.15, the MATLAB

curve fitting toolbox can be used to fit the resulting shaping function to the variations of the elevation angle shown in Figures 3.1 to 3.13. Through a comparison of the RMSE of the various combinations, conclusions can be drawn with respect to their ability to simulate the test cases.

As can be seen in Table 3.3, a total of 12 different combinations are tested. In this table, the RMSE in degrees can be seen for the first test case. As shown in this table, a combination of $a_0 + a_1\theta + a_2\theta^2$ for a^2 and $b_0 + b_1\theta$ for the coefficient b was found to result in a slightly lower RMSE. The combination that resulted in the lowest RMSE and those that resulted in an RMSE within 25% of the best RMSE have been given a green background.

Table 3.3: The RMSE in degrees for the 12 different combinations when case 1 is evaluated.

	$b_0 + b_1\theta$	$b_0 + b_1\theta + b_2\theta^2$	$b_0 + b_1(\theta + b_2)^{b_3}$
$a_0 e^{a_1\theta + a_2} + a_3 e^{a_4\theta + a_5}$	$6.02 \cdot 10^{-6}$	$8.24 \cdot 10^{-8}$	$8.07 \cdot 10^{-7}$
$a_0 + a_1\theta$	$2.37 \cdot 10^{-10}$	$2.99 \cdot 10^{-11}$	$1.71 \cdot 10^{-5}$
$a_0 + a_1\theta + a_2\theta^2$	$1.71 \cdot 10^{-13}$	$1.42 \cdot 10^{-8}$	$2.79 \cdot 10^{-6}$
$a_0 + a_1(\theta + a_2)^{a_3}$	$3.72 \cdot 10^{-11}$	$1.10 \cdot 10^{-7}$	$9.43 \cdot 10^{-8}$

Interesting to note is that, while the third and fourth expressions for a^2 can be written as the second expression, the RMSE of combinations that use the third or fourth expression can be higher than those that use the second. A similar observation can be made for the various expressions of b. As this test case features a constant inclination, the values of a^2 and b should be constant and all combinations should result in the same RMSE. However, due to the addition of extra coefficients, it becomes more difficult for the coefficient estimation algorithm to reproduce this constant value. As such, a higher RMSE is found for certain combinations.

A similar observation can be made by looking at Table 3.4, where the RMSE in degrees for various combinations of expressions for a^2 and b are shown when test case 2 is evaluated. As done for Table 3.3, the combination that resulted in the lowest RMSE and those that resulted in an RMSE within 25% of the lowest RMSE have been highlighted in green.

Table 3.4: The RMSE in degrees for the 12 different combinations when case 2 is evaluated.

	$b_0 + b_1\theta$	$b_0 + b_1\theta + b_2\theta^2$	$b_0 + b_1(\theta + b_2)^{b_3}$
$a_0 e^{a_1\theta + a_2} + a_3 e^{a_4\theta + a_5}$	1.64	1.64	1.64
$a_0 + a_1\theta$	14.9	8.73	12.6
$a_0 + a_1\theta + a_2\theta^2$	7.71	10.5	4.67
$a_0 + a_1(\theta + a_2)^{a_3}$	1.64	1.64	1.66

Noteworthy here is that, even though the lowest RMSE is 1.64 degrees, the initial and final elevation angle differ by less than $1{\cdot}10^{-4}$ and $3{\cdot}10^{-3}$ degrees for each of the combinations with this RMSE. Using the radius at these positions, it can be found that this is equal to an error in the initial and final z-coordinate of approximately 12 and 351 meters respectively.

For test case 3, it can be noted from Table 3.5 that the RMSE is approximately equal for all combinations, except when $a_0 + a_1\theta$ is used for a^2 and $b_0 + b_1\theta$ is used for b.

Table 3.5: The RMSE in degrees for the 12 different combinations when test case 3 is evaluated.

	$b_0 + b_1\theta$	$b_0 + b_1\theta + b_2\theta^2$	$b_0 + b_1(\theta + b_2)^{b_3}$
$a_0 e^{a_1\theta + a_2} + a_3 e^{a_4\theta + a_5}$	$2.75 \cdot 10^{-2}$	$2.76 \cdot 10^{-2}$	$2.76 \cdot 10^{-2}$
$a_0 + a_1\theta$	1.12	$2.75 \cdot 10^{-2}$	$2.76 \cdot 10^{-2}$
$a_0 + a_1\theta + a_2\theta^2$	$2.76 \cdot 10^{-2}$	$2.75 \cdot 10^{-2}$	$2.76 \cdot 10^{-2}$
$a_0 + a_1(\theta + a_2)^{a_3}$	$2.76 \cdot 10^{-2}$	$2.80 \cdot 10^{-2}$	$2.80 \cdot 10^{-2}$

For test case 4, it was found that each of the chosen combinations had significant trouble with approximating the spacecraft continuing in its final orbit for a large number of revolutions once it reached this final orbit.

As such, it was tested instead whether the combinations could approximate the transfer itself. This means that only the part of the data that resembled the inclination change from 0 to 45 degrees was used to test the combinations.

When this test case is evaluated using the aforementioned change, it is found that all combinations result in an RMSE in the order of 10^{-1} degrees. However, as shown in Table 3.6, there are four combinations that result in a slightly lower RMSE.

Table 3.6: The RMSE in degrees for the 12 different combinations when test case 4 is evaluated.

	$b_0 + b_1\theta$	$b_0 + b_1\theta + b_2\theta^2$	$b_0 + b_1(\theta + b_2)^{b_3}$
$a_0 e^{a_1\theta+a_2} + a_3 e^{a_4\theta+a_5}$	$1.11 \cdot 10^{-1}$	$1.11 \cdot 10^{-1}$	$1.34 \cdot 10^{-1}$
$a_0 + a_1\theta$	$2.25 \cdot 10^{-1}$	$1.91 \cdot 10^{-1}$	$3.19 \cdot 10^{-1}$
$a_0 + a_1\theta + a_2\theta^2$	$1.20 \cdot 10^{-1}$	$3.59 \cdot 10^{-1}$	$2.07 \cdot 10^{-1}$
$a_0 + a_1(\theta + a_2)^{a_3}$	$3.35 \cdot 10^{-1}$	$3.19 \cdot 10^{-1}$	$3.19 \cdot 10^{-1}$

For test case 5, a similar problem to the one found for test case 4 was encountered. For this reason, the test case's trajectory was cut off at the point where it reached an inclination of 89.6 degrees. The reason for choosing this inclination as opposed to the final inclination of 89.99 degrees, is that the last 0.39 degrees change happens over a large number of revolutions in this transfer, meaning that the value of b changes very slowly in this part of the orbit. As this value changes very rapidly before this part of the orbit, the chosen combinations are unable to accurately model the orbit. To solve this problem, the final inclination was set to be 89.6 degrees instead.

When the inclination is increased from 0 to 89.6 degrees, the result is that the RMSE increases significantly for all combinations. This is shown in Table 3.7, where the RMSE in degrees for the various combinations is shown when test case 5 is evaluated.

Table 3.7: The RMSE in degrees for the 12 different combinations when test case 5 is evaluated.

	$b_0 + b_1\theta$	$b_0 + b_1\theta + b_2\theta^2$	$b_0 + b_1(\theta + b_2)^{b_3}$
$a_0 e^{a_1\theta+a_2} + a_3 e^{a_4\theta+a_5}$	16.2	9.37	7.87
$a_0 + a_1\theta$	12.8	10.4	10.9
$a_0 + a_1\theta + a_2\theta^2$	10.3	10.0	10.3
$a_0 + a_1(\theta + a_2)^{a_3}$	11.0	10.9	10.8

If a trajectory of which the inclination decreases from 89.98 to 0.1 degrees is approximated, the RMSEs in degrees shown in Table 3.8 are found.

Table 3.8: The RMSE in degrees for the 12 different combinations when test case 6 is evaluated.

	$b_0 + b_1\theta$	$b_0 + b_1\theta + b_2\theta^2$	$b_0 + b_1(\theta + b_2)^{b_3}$
$a_0 e^{a_1\theta+a_2} + a_3 e^{a_4\theta+a_5}$	5.94	1.72	1.71
$a_0 + a_1\theta$	9.23	8.50	9.08
$a_0 + a_1\theta + a_2\theta^2$	9.23	8.50	9.02
$a_0 + a_1(\theta + a_2)^{a_3}$	9.21	1.61	7.96

In this table, it can be seen that the use of either a linear or quadratic equation for a^2 results in a considerably higher RMSE. The cause of this is the aforementioned boundary set on the coefficients. In order to ensure that the equation resembling a^2 does not become smaller than zero, a minimum value of zero was set for a_0 and a_1, and a_0 to a_2 for the linear and quadratic expressions respectively. In the exponential and power functions, this was done for the a_0 and a_3, and the a_0 and a_1 coefficients. However, while these boundaries prevent the expression's result from becoming smaller than zero, they prevent the value from decreasing in the first place in the case of the linear and quadratic expression. As the value of a^2 decreases in this test case, a high RMSE is the result.

The RMSEs for test case 7 can be found in Table 3.9. It can be seen here that using an exponential function for a^2 and a power function for b results in an RMSE that is approximately 3 to 8 degrees less.

Table 3.9: The RMSE in degrees for the 12 different combinations when test case 7 is evaluated.

	$b_0+b_1\theta$	$b_0+b_1\theta+b_2\theta^2$	$b_0+b_1(\theta+b_2)^{b_3}$
$a_0e^{a_1\theta+a_2}+a_3e^{a_4\theta+a_5}$	13.8	5.17	4.29
$a_0+a_1\theta$	9.46	9.35	10.5
$a_0+a_1\theta+a_2\theta^2$	9.87	7.88	8.88
$a_0+a_1(\theta+a_2)^{a_3}$	10.93	9.75	8.77

As can be seen in Table 3.10, all combinations are able to accurately follow the trajectory found by Roegiers for a transfer to Neptune using a single revolution. The combination of using an exponential function for a^2 and a power function for b was found to result in the lowest RMSE.

Table 3.10: The RMSE in degrees for the 12 different combinations when test case 8 is evaluated.

	$b_0+b_1\theta$	$b_0+b_1\theta+b_2\theta^2$	$b_0+b_1(\theta+b_2)^{b_3}$
$a_0e^{a_1\theta+a_2}+a_3e^{a_4\theta+a_5}$	$7.53\cdot10^{-2}$	$2.72\cdot10^{-3}$	$2.02\cdot10^{-3}$
$a_0+a_1\theta$	$8.97\cdot10^{-2}$	$9.36\cdot10^{-3}$	$9.13\cdot10^{-2}$
$a_0+a_1\theta+a_2\theta^2$	$8.69\cdot10^{-2}$	$4.80\cdot10^{-3}$	$9.95\cdot10^{-2}$
$a_0+a_1(\theta+a_2)^{a_3}$	$9.45\cdot10^{-2}$	$1.37\cdot10^{-2}$	$1.54\cdot10^{-2}$

When a multi-revolution transfer to Neptune is approximated, the RMSEs shown in Table 3.11 are found. It can be seen in this table that all errors are approximately equal to each other, but that there are three combinations that result in a slightly larger RMSE.

Table 3.11: The RMSE in degrees for the 12 different combinations when test case 9 is evaluated.

	$b_0+b_1\theta$	$b_0+b_1\theta+b_2\theta^2$	$b_0+b_1(\theta+b_2)^{b_3}$
$a_0e^{a_1\theta+a_2}+a_3e^{a_4\theta+a_5}$	$7.13\cdot10^{-4}$	$7.12\cdot10^{-4}$	$7.12\cdot10^{-4}$
$a_0+a_1\theta$	$7.11\cdot10^{-4}$	$7.11\cdot10^{-4}$	$2.76\cdot10^{-3}$
$a_0+a_1\theta+a_2\theta^2$	$7.11\cdot10^{-4}$	$7.13\cdot10^{-1}$	$7.86\cdot10^{-3}$
$a_0+a_1(\theta+a_2)^{a_3}$	$7.26\cdot10^{-4}$	$7.13\cdot10^{-4}$	$7.13\cdot10^{-4}$

By evaluating Tables 3.3 to 3.11, it can be seen that the combinations using an exponential function for a^2 and either a quadratic or power function for b are often either the best result, or within 25% of the best result. The only case in which neither of them is the best result or within 25% of the best result, is case 1. However, it can be seen in Table 3.3 that both combinations do perform excellently. Furthermore, while the combination using an exponential function for a^2 and a quadratic form for b does not result in an RMSE within 25% of the best result for case 7, it does perform significantly better than the other combinations. As such, these two combinations are found to have the most potential.

3.3.5. RIGHT ASCENSION OF THE ASCENDING NODE CHANGES

As mentioned previously, the elevation shaping function currently used in the spherical shaping method is unable to accurately describe RAAN changes. Using the last two test cases shown in Section 3.1, the combinations that were found to have the most potential will be compared and the shaping function will be further altered such that it can describe RAAN changes.

Firstly, it can seen in Figure 3.13 that the maximum elevation angle is 45 degrees and the minimum elevation angle is approximately minus 40 degrees. Thus, it can concluded that the change of the elevation angle has an offset. As the shaping function cannot reproduce this with any of the combinations shown so far, an additional offset term c is added to the shaping function.

As changing the RAAN changes the point at which the orbit intersects the equatorial plane, the azimuthal angle at which this happens also changes. By comparing the change of the elevation angle as a function of

the azimuthal angle with a waveform, one could say that changing the RAAN causes the frequency of the waveform to change. This change in frequency is a function of the azimuthal angle and can be simulated by changing $cos(\theta + p)$ in the shaping function to an alternative form. Two expressions are tested, namely $cos\left(p_0\theta^{p_1} + p_2\theta + p_3\right)$ and $cos\left(p_0\theta^2 + p_2\theta + p_3\right)$. As both a quadratic function and a power function were found to have the most potential for describing the coefficient b, a total of four combinations will be tested for the two test cases described earlier.

For the test case 10, the RMSE in degrees for the four combinations can be seen in Table 3.12. As done previously, the combination with the lowest RMSE and those that resulted in an RMSE within 25% of the best RMSE have been highlighted in green.

Table 3.12: The RMSE in degrees for the 4 different combinations when test case 10 is evaluated.

	$b_0 + b_1\theta + b_2\theta^2$ [deg]	$b_0 + b_1(\theta + b_2)^{b_3}$ [deg]
$p_0\theta^{p_1} + p_2\theta + p_3$	$2.02 \cdot 10^{-2}$	$2.92 \cdot 10^{-1}$
$p_0\theta^2 + p_2\theta + p_3$	30.1	30.9

The RMSE for the four combinations when test case 11 is evaluated can be found in Table 3.13.

Table 3.13: The RMSE in degrees for the 4 different combinations when test case 11 is evaluated.

	$b_0 + b_1\theta + b_2\theta^2$ [deg]	$b_0 + b_1(\theta + b_2)^{b_3}$ [deg]
$p_0\theta^{p_1} + p_2\theta + p_3$	$7.92 \cdot 10^{-1}$	1.57
$p_0\theta^2 + p_2\theta + p_3$	24.3	15.2

From these two tables, it can be seen that the combination of $p_0\theta^{p_1} + p_2\theta + p_3$ and $b_0 + b_1\theta + b_2\theta^2$ results in the lowest RMSE in both cases. Interesting to note is also that, due to the fact that the inclination decreases and then increases again for the second test case, the equation describing a^2 should decrease and increase in value again as well. However, by plotting the change of a^2 using the coefficients found, it was observed that the value of a^2 does not change in this manner. Despite the fact that the change in a^2 was not as it theoretically should be, a low RMSE was still found. The cause of this is that the coefficient a does not change much around these inclinations. This can also be seen in Figure 3.27. As it is highly unusual for a satellite to first decrease and then increase its inclination again, this behaviour is not deemed to be disruptive.

3.4. CONCLUSIONS

In this chapter, the options that were among others found during the literature review were tested for their potential of forming the shaping function of the elevation angle. It was found that both the use of spherical triangles and the Fourier series did not result in a shaping function which could accurately describe the set of predefined test cases. Instead, an alternative shaping function was found to be capable of describing these test cases.

In order to obtain this alternative shaping function, different combinations of functions were tested for their potential to form the elevation shaping function. By evaluating the RMSE of each combination for each of the test cases, the most accurate combination was found. The resulting elevation shaping function can be seen in Equation 3.16.

$$\Phi(\theta) = \sqrt{\frac{1 + (a_0 e^{a_1\theta + a_2} + a_3 e^{a_4\theta + a_5})}{1 + (a_0 e^{a_1\theta + a_2} + a_3 e^{a_4\theta + a_5}) cos^2\left(p_0\theta^{p_1} + p_2\theta + p_3\right)}} cos\left(p_0\theta^{p_1} + p_2\theta + p_3\right) \cdot \left(b_0 + b_1\theta + b_2\theta^2\right) + c \tag{3.16}$$

This elevation shaping function will be the basis of all research done in the subsequent chapters.

4

IMPLEMENTATION

With the promising elevation shaping function found, the developed function can be implemented, validated and applied. This chapter will discuss the manner in which the function is implemented and trajectories are optimised.

4.1. SPHERICAL SHAPING METHOD

As mentioned previously, Novak's spherical shaping method has been adapted using the elevation shaping function found in Chapter 3. This section shall elaborate on the implementation and adaptation of this shape-based method.

4.1.1. INITIAL AND FINAL CONDITIONS

The initial and final conditions consist of the Keplerian elements at the start and end points. These can be converted into Cartesian and spherical coordinates with the equations shown in Appendix A. As such, the state vector at the initial and final position is [2]:

$$\begin{pmatrix} r_{i/f} & \theta_{i/f} & \phi_{i/f} & \dot{r}_{i/f} & \dot{\theta}_{i/f} & \dot{\phi}_{i/f} \end{pmatrix}^T \tag{4.1}$$

where the subscripts i and f indicate the initial and final values.

4.1.2. CHANGING PARAMETRISATION

Firstly, the initial and final conditions that are parametrised by time are changed such that they are parametrised by the azimuthal angle θ. The initial state vector defined in the spherical coordinate frame is shown in Equation 4.1. Furthermore, the velocity of the trajectory parametrised by time is [2]:

$$\mathbf{v} = \dot{\mathbf{r}} = \begin{pmatrix} v_r \\ v_\theta \\ v_\phi \end{pmatrix} = \begin{pmatrix} \dot{r} \\ r cos(\phi)\dot{\theta} \\ r\dot{\phi} \end{pmatrix} \tag{4.2}$$

To denote the parametrisation by azimuthal angle θ, the following notation is introduced [2]:

$$\tilde{\mathbf{v}} = \mathbf{r}' = \frac{d\mathbf{r}}{d\theta} \tag{4.3}$$

$$\tilde{\mathbf{a}} = \mathbf{r}'' = \frac{d^2\mathbf{r}}{d\theta^2} \tag{4.4}$$

where the tilde and prime indicate the parametrisation and derivative with respect to the azimuthal angle θ respectively.

To find the velocity of the trajectory parametrised by the azimuthal angle, $\dot{\mathbf{r}}$ can be divided by $\dot{\theta}$, as shown in Equation 4.5.

$$\tilde{\mathbf{v}} = \mathbf{r}' = \frac{dt}{d\theta}\frac{d\mathbf{r}}{dt} = \frac{1}{\dot{\theta}}\dot{\mathbf{r}} \tag{4.5}$$

Thus, by dividing Equation 4.2 by $\dot{\theta}$, the velocity vector parametrised by the azimuthal angle can be found. The result of this is shown in Equation 4.6 [2].

$$\tilde{\mathbf{v}} = \frac{dt}{d\theta}\begin{pmatrix} v_r \\ v_\theta \\ v_\phi \end{pmatrix} = \begin{pmatrix} \frac{dr}{d\theta} \\ rcos(\phi) \\ r\frac{d\phi}{d\theta} \end{pmatrix} = \begin{pmatrix} \tilde{v}_r \\ \tilde{v}_\theta \\ \tilde{v}_\phi \end{pmatrix} \tag{4.6}$$

Using this equation, the initial and final state vector parametrised by the azimuthal angle can be found [2]:

$$\begin{pmatrix} r_{i/f} & t_{i/f} & \phi_{i/f} & r'_{i/f} & t'_{i/f} & \phi'_{i/f} \end{pmatrix}^T \tag{4.7}$$

where $r'_{i/f}$ is obtained from Equation 4.8, which is in turn derived from Equation 4.6.

$$r'_{i/f} = \frac{r_{i/f}}{\dot{\theta}_{i/f}} \tag{4.8}$$

The equation used to find the derivative of the initial and final time with respect to the azimuthal angle is also found using Equation 4.6 and is shown in Equation 4.9.

$$t'_{i/f} = \frac{r_{i/f}cos(\phi_{i/f})}{v_{\phi i/f}} = \frac{1}{\dot{\theta}_{i/f}} \tag{4.9}$$

where the derivative of the azimuthal angle with respect to time can also be obtained from Equation 4.6, as shown in Equation 4.10 [2].

$$\dot{\theta}_{i/f} = \frac{v_{\theta i/f}}{r_{i/f} \cdot cos(\phi_{i/f})} \tag{4.10}$$

4.1.3. BOUNDARY CONDITIONS

The boundary conditions on the trajectory are listed below [2]:

$$\begin{aligned}
&R_i = R(\theta_i) = r_i \\
&R_f = R(\theta_f + 2\pi N_{rev}) = r_f \\
&\Phi_i = \Phi(\theta_i) = \phi_i \\
&\Phi_f = \Phi(\theta_f + 2\pi N_{rev}) = \phi_f \\
&T'_i = T'(\theta_i) = \frac{r_i cos(\phi_i)}{v_{\phi i}} \\
&T'_f = T'(\theta_f + 2\pi N_{rev}) = \frac{r_f cos(\phi_f)}{v_{\phi f}} \\
&R'_i = R'(\theta_i) = \tilde{v}_{ri} \\
&R'_f = R'(\theta_f + 2\pi N_{rev}) = \tilde{v}_{rf} \\
&\Phi'_i = \Phi'(\theta_i) = \frac{\tilde{v}_{\phi i}}{r_i} \\
&\Phi'_f = \Phi'(\theta_f + 2\pi N_{rev}) = \frac{\tilde{v}_{\phi f}}{r_f}
\end{aligned} \tag{4.11}$$

To match the boundary conditions on the derivative of time with respect to the azimuthal angle T', Novak expressed these boundary conditions as conditions on the initial and final R'' and Φ'' [1]:

$$\begin{aligned}
&R''(\theta_i) + \alpha_i \Phi''(\theta_i) = C_i \\
&R''(\theta_f) + \alpha_f \Phi''(\theta_f) = C_f
\end{aligned} \tag{4.12}$$

where α can be found using:

$$\alpha_{i/f} = \frac{R'_{i/f}\Phi'_{i/f}}{\Phi'^2_{i/f} + cos^2(\Phi_{i/f})} \tag{4.13}$$

and C is found with:

$$C_{i/f} = -\frac{\mu T'^2_{i/f}}{R^2_{i/f}} + 2\frac{R'^2_{i/f}}{R_{i/f}} + R_{i/f}\left(\Phi'^2_{i/f} + cos^2(\Phi_{i/f})\right) - R'_{i/f}\Phi'_{i/f}\frac{sin(\Phi_{i/f})cos(\Phi_{i/f})}{\Phi'^2_{i/f} + cos^2(\Phi_{i/f})} \tag{4.14}$$

4.1.4. SHAPING FUNCTIONS

For the radial shaping function, use is made of the function proposed by Novak [1]:

$$R(\theta) = \frac{1}{r_0 + r_1\theta + r_2\theta^2 + (r_3 + r_4\theta)cos(\theta) + (r_5 + r_6\theta)sin(\theta)} \tag{4.15}$$

It should be noted here that, while Novak and Roegiers both denote the coefficients in the radial shaping function as a_0 to a_6, they are defined here as r_0 to r_6 in order to prevent confusion with the coefficients of the elevation shaping function.

For the elevation shaping function, the function derived in Chapter 3 is used (Equation 3.16):

$$\Phi(\theta) = \sqrt{\frac{1 + (a_0 e^{a_1\theta + a_2} + a_3 e^{a_4\theta + a_5})}{1 + (a_0 e^{a_1\theta + a_2} + a_3 e^{a_4\theta + a_5})cos^2\left(p_0\theta^{p_1} + p_2\theta + p_3\right)}}cos\left(p_0\theta^{p_1} + p_2\theta + p_3\right)\cdot\left(b_0 + b_1\theta + b_2\theta^2\right) + c \tag{4.16}$$

In order to use Novak's method of solving the coefficients of these functions, both shaping functions should be functions of a dot product between a coefficient vector and a vector containing a set of sub-functions that depend on the azimuthal angle [2].

For the radial shaping function, the coefficient vector is:

$$\mathbf{r}_k = \begin{pmatrix} r_0 & r_1 & r_2 & r_3 & r_4 & r_5 & r_6 \end{pmatrix} \tag{4.17}$$

and the vector containing the set of sub-functions is:

$$\mathbf{R}_k(\theta) = \begin{pmatrix} 1 & \theta & \theta^2 & cos(\theta) & \theta cos(\theta) & sin(\theta) & \theta sin(\theta) \end{pmatrix} \tag{4.18}$$

In this vector, the first term is referenced as R_0, the second as R_1, the third as R_2 and so on. The radial shaping function can be obtained by dividing 1 by the dot product between these two vectors, and is thus a function of this dot product as required.

For the elevation shaping function, it should be noted from Equation 4.11 that four boundary conditions are available, as r_2 will be tuned with a Newton-Raphson loop such that the time-of-flight constraint is met. However, the elevation shaping function has more than four coefficients. As such, four coefficients will be determined using the boundary conditions and the remaining coefficients will be found by optimising the ΔV with the Monte-Carlo method and the Nelder-Mead method. The four coefficients that will be determined using the boundary conditions are b_0, b_1, b_2 and c, as the elevation shaping function can then easily be written in the required form discussed above.

Using this choice of coefficients, the coefficient vector becomes:

$$\mathbf{b}_j = \begin{pmatrix} b_0 & b_1 & b_2 & c \end{pmatrix} \tag{4.19}$$

and the vector containing the set of sub-functions becomes:

$$\mathbf{\Phi}_j(\theta) = \begin{pmatrix} \Gamma & \Gamma\theta & \Gamma\theta^2 & 1 \end{pmatrix} \tag{4.20}$$

where

$$\Gamma = \sqrt{\frac{1 + (a_0 e^{a_1\theta + a_2} + a_3 e^{a_4\theta + a_5})}{1 + (a_0 e^{a_1\theta + a_2} + a_3 e^{a_4\theta + a_5})cos^2\left(p_0\theta^{p_1} + p_2\theta + p_3\right)}}cos\left(p_0\theta^{p_1} + p_2\theta + p_3\right) \tag{4.21}$$

4.1.5. DERIVATIVES OF $\mathbf{R}_k$ AND $\mathbf{\Phi}_j$

In order to obtain the trajectory, the first, second and third derivatives of each element of the $\mathbf{R}_k$ and $\mathbf{\Phi}_j$ vectors is required. This section elaborates on these derivatives.

$\mathbf{R}_k$ DERIVATIVES

The first, second and third derivatives of the $\mathbf{R}_k$ vector's elements can be seen in Equation 4.22 [2].

$$\begin{aligned}
\mathbf{R}'_k(\theta) &= \begin{bmatrix} 0 & 1 & 2\theta & -sin(\theta) & cos(\theta)-\theta sin(\theta) & cos(\theta) & sin(\theta)+\theta cos(\theta) \end{bmatrix} \\
\mathbf{R}''_k(\theta) &= \begin{bmatrix} 0 & 0 & 2 & -cos(\theta) & -2sin(\theta)-\theta cos(\theta) & -sin(\theta) & 2cos(\theta)-\theta sin(\theta) \end{bmatrix} \\
\mathbf{R}'''_k(\theta) &= \begin{bmatrix} 0 & 0 & 0 & sin(\theta) & -3cos(\theta)+\theta sin(\theta) & -cos(\theta) & -3sin(\theta)-\theta cos(\theta) \end{bmatrix}
\end{aligned} \tag{4.22}$$

$\mathbf{\Phi}_j$ DERIVATIVES

Using Equations 4.20 and 4.21, it can be found that the first, second and third derivatives of the $\mathbf{\Phi}_j$ vector's elements are:

$$\begin{aligned}
\mathbf{\Phi}'_j(\theta) &= \begin{bmatrix} \Gamma' & \Gamma'\theta+\Gamma & \Gamma'\theta^2+2\Gamma\theta & 0 \end{bmatrix} \\
\mathbf{\Phi}''_j(\theta) &= \begin{bmatrix} \Gamma'' & \Gamma''\theta+2\Gamma' & \Gamma''\theta^2+4\Gamma'\theta+2\Gamma & 0 \end{bmatrix} \\
\mathbf{\Phi}'''_j(\theta) &= \begin{bmatrix} \Gamma''' & \Gamma'''\theta+3\Gamma'' & \Gamma'''\theta^2+6\Gamma''\theta+6\Gamma' & 0 \end{bmatrix}
\end{aligned} \tag{4.23}$$

where Γ is as shown in Equation 4.21.

In order to more easily find the derivatives of Γ, the additional coefficients shown in Equations 4.24 and 4.25 are defined.

$$\sigma = \left(a_0 e^{a_2+a_1\theta} + a_3 e^{a_5+a_4\theta}\right) \tag{4.24}$$

$$\lambda = \left(p_0\theta^{p_1} + p_2\theta + p_3\right) \tag{4.25}$$

By substituting these two coefficients into Equation 4.21, Equation 4.26 is found.

$$\Gamma = \tau cos(\lambda) \tag{4.26}$$

in which

$$\tau = \sqrt{\frac{1+\sigma}{1+\sigma cos^2(\lambda)}} \tag{4.27}$$

Using this equation, it can now be found that the derivatives of Γ are as shown in Equation 4.28. In this equation, sinuses and cosines have been abbreviated as s and c respectively.

$$\begin{aligned}
\Gamma' &= \tau' c(\lambda) - \tau s(\lambda)\lambda' \\
\Gamma'' &= \tau'' c(\lambda) - 2\tau'\lambda' s(\lambda) - \tau\lambda'' s(\lambda) - \tau\lambda'^2 c(\lambda) \\
\Gamma''' &= \tau''' c(\lambda) - 3\tau''\lambda' s(\lambda) - 3\tau'\lambda'' s(\lambda) - 3\tau'\lambda'^2 c(\lambda) - \tau\lambda''' s(\lambda) - 3\tau\lambda''\lambda' c(\lambda) + \tau\lambda'^3 s(\lambda)
\end{aligned} \tag{4.28}$$

The first derivative of τ with respect to θ is found using the symbolic computing environment Maple 2016 and is shown in Equation 4.29.

$$\tau' = \frac{\left(\frac{\sigma'}{\sigma c^2(\lambda)+1} - \frac{(c^2(\lambda)\sigma' - 2c(\lambda)s(\lambda)\sigma\lambda')(\sigma+1)}{(\sigma c^2(\lambda)+1)^2}\right)}{2\tau} = \frac{A}{B} \tag{4.29}$$

As shown in Equation 4.30, τ'' can now be obtained with the quotient rule by defining the numerator of Equation 4.29 as A and the denominator as B.

$$\tau'' = \frac{BA' - AB'}{B^2} = \frac{(2\tau)A' - A(2\tau')}{4\tau^2} \tag{4.30}$$

Furthermore, τ''' can now also be found using Equation 4.31.

$$\tau''' = \frac{B^2A'' - 2BA'B' - ABB'' + 2AB'^2}{B^3} = \frac{4\tau^2A'' - 8\tau A'\tau' - 4A\tau\tau'' + 8A\tau'^2}{8\tau^3} \tag{4.31}$$

To find the first and second derivative of the numerator of Equation 4.29, Maple 2016 is again used. The results are shown in Equations 4.32 and 4.33 respectively.

$$\begin{aligned}
A' = {} & \frac{\sigma''}{(\sigma c^2(\lambda)+1)} - \frac{(2\sigma'(\sigma' c^2(\lambda) - 2\sigma c(\lambda)\lambda' s(\lambda))}{(\sigma c^2(\lambda)+1)^2} + \frac{2(\sigma+1)(\sigma' c^2(\lambda) - 2\sigma c(\lambda)\lambda' s(\lambda))^2}{(\sigma c^2(\lambda)+1)^3} - \ldots \\
& \frac{(\sigma+1)(\sigma'' c^2(\lambda) - 4\sigma' c(\lambda)\lambda' + 2\sigma\lambda'^2 s^2(\lambda) - 2\sigma c(\lambda)\lambda'' s(\lambda) - 2\sigma c^2(\lambda)\lambda'^2)}{(\sigma c^2(\lambda)+1)^2}
\end{aligned} \tag{4.32}$$

$$
\begin{aligned}
A'' = &\frac{\sigma'''}{\sigma c^2(\lambda)+1} - \frac{3\sigma''(\sigma' c^2(\lambda) - 2\sigma c(\lambda)\lambda' s(\lambda))}{(\sigma c^2(\lambda)+1)^2} + \frac{6\sigma'(\sigma' c^2(\lambda) - 2\sigma c(\lambda)\lambda' s(\lambda))^2}{(\sigma c^2(\lambda)+1)^3} - \dots \\
&\frac{3\sigma'(\sigma'' c^2(\lambda) - 4\sigma' c(\lambda)\lambda' s(\lambda) + 2\sigma\lambda'^2 s^2(\lambda) - 2\sigma c(\lambda)\lambda'' s(\lambda) - 2\sigma c^2(\lambda)\lambda'^2)}{(\sigma c^2(\lambda)+1)^2} - \dots \\
&\frac{6(\sigma+1)(\sigma' c^2(\lambda) - 2\sigma c(\lambda)\lambda' s(\lambda))^3}{(\sigma c^2(\lambda)+1)^4} + \frac{6}{(\sigma c^2(\lambda)+1)^3}((\sigma+1)(\sigma' c^2(\lambda) - \dots \\
&2\sigma c(\lambda)\lambda' s(\lambda))(\sigma'' c^2(\lambda) - 4\sigma' c(\lambda)\lambda' s(\lambda) + 2\sigma\lambda'^2 s^2(\lambda) - \dots \\
&2\sigma c(\lambda)\lambda'' s(\lambda) - 2\sigma c^2(\lambda)\lambda'^2)) - \frac{1}{(\sigma c^2(\lambda)+1)^2}((\sigma+1)(\sigma''' c^2(\lambda) - \dots \\
&6\sigma'' c(\lambda)\lambda' s(\lambda) + 6\sigma'\lambda'^2 s^2(\lambda) - 6\sigma' c(\lambda)\lambda'' s(\lambda) - 6\sigma' c^2(\lambda)\lambda'^2 + 6\sigma\lambda' s^2(\lambda) \dots \\
&\lambda'' + 8\sigma\lambda'^3 s(\lambda)c(\lambda) - 2\sigma c(\lambda)\lambda''' s(\lambda) - 6\sigma c^2(\lambda)\lambda''\lambda'))
\end{aligned}
\tag{4.33}
$$

In order to evaluate the equations shown in this section, the first, second and third derivative of σ and λ are also required. These can be seen in Equation 4.34.

$$
\begin{aligned}
\sigma' &= \left(a_0 a_1 e^{a_2 + a_1\theta} + a_3 a_4 e^{a_5 + a_4\theta}\right) \\
\sigma'' &= \left(a_0 {a_1}^2 e^{a_2 + a_1\theta} + a_3 {a_4}^2 e^{a_5 + a_4\theta}\right) \\
\sigma''' &= \left(a_0 {a_1}^3 e^{a_2 + a_1\theta} + a_3 {a_4}^3 e^{a_5 + a_4\theta}\right) \\
\lambda' &= \left(p_0 p_1 \theta^{p_1 - 1} + p_2\right) \\
\lambda'' &= \left(p_0 p_1 \theta^{p_1 - 2}\left(p_1 - 1\right)\right) \\
\lambda''' &= \left(p_0 p_1 \theta^{p_1 - 3}(p_1 - 1)(p_1 - 2)\right)
\end{aligned}
\tag{4.34}
$$

4.1.6. COMPUTING THE COEFFICIENTS

With the boundary conditions defined and the derivatives computed, one can now find the coefficients such that the trajectory matches these conditions. Using the boundary conditions shown in Section 4.1.3, it can be shown that in order to match these conditions, Equation 4.35 should hold. For the full derivation of this equation, the reader is referred to Roegiers' master's thesis [2].

$$
\left[\mathbf{A}\right]\begin{pmatrix} r_0 & r_1 & r_3 & r_4 & r_5 & r_6 & b_0 & b_1 & b_2 & c \end{pmatrix}^T + \mathbf{A}_{r_2} = \mathbf{B} \tag{4.35}
$$

in which the **A** matrix is as shown in Equation 4.36 [2]. The various derivatives with respect to the azimuthal angle shown in this equation can be found in Section 4.1.5.

$$
\mathbf{A} = \begin{bmatrix}
R_0(\theta_i) & R_1(\theta_i) & R_3(\theta_i) & \dots & \dots & R_6(\theta_i) & 0 & 0 & 0 & 0 \\
R_0(\theta_f) & R_1(\theta_f) & R_3(\theta_f) & \dots & \dots & R_6(\theta_f) & 0 & 0 & 0 & 0 \\
R_0'(\theta_i) & R_1'(\theta_i) & R_3'(\theta_i) & \dots & \dots & R_6'(\theta_i) & 0 & 0 & 0 & 0 \\
R_0'(\theta_f) & R_1'(\theta_f) & R_3'(\theta_f) & \dots & \dots & R_6'(\theta_f) & 0 & 0 & 0 & 0 \\
-R_i^2 R_0''(\theta_i) & -R_i^2 R_1''(\theta_i) & -R_i^2 R_3''(\theta_i) & \dots & \dots & -R_i^2 R_6''(\theta_i) & \alpha_i \Phi_0''(\theta_i) & \dots & \dots & \alpha_i \Phi_3''(\theta_i) \\
-R_f^2 R_0''(\theta_f) & -R_f^2 R_1''(\theta_f) & -R_f^2 R_3''(\theta_f) & \dots & \dots & -R_f^2 R_6''(\theta_f) & \alpha_f \Phi_0''(\theta_f) & \dots & \dots & \alpha_f \Phi_3''(\theta_f) \\
0 & 0 & 0 & 0 & 0 & 0 & \Phi_0(\theta_i) & \dots & \dots & \Phi_3(\theta_i) \\
0 & 0 & 0 & 0 & 0 & 0 & \Phi_0(\theta_f) & \dots & \dots & \Phi_3(\theta_f) \\
0 & 0 & 0 & 0 & 0 & 0 & \Phi_0'(\theta_i) & \dots & \dots & \Phi_3'(\theta_i) \\
0 & 0 & 0 & 0 & 0 & 0 & \Phi_0'(\theta_f) & \dots & \dots & \Phi_3'(\theta_f)
\end{bmatrix}
\tag{4.36}
$$

Furthermore, the **B** and $\mathbf{A}_{r_2}$ vectors are as shown in Equations 4.37 and 4.38 respectively.

$$
\mathbf{B} = \begin{pmatrix} \frac{1}{R_i} & \frac{1}{R_f} & -\frac{\tilde{v}_{r_i}}{R_i^2} & -\frac{\tilde{v}_{r_f}}{R_f^2} & C_i - 2\frac{\tilde{v}_{r_i}^2}{R_i} & C_f - 2\frac{\tilde{v}_{r_f}^2}{R_f} & \Phi_i & \Phi_f & \frac{\tilde{v}_{\phi_i}}{R_i} & \frac{\tilde{v}_{\phi_f}}{R_f} \end{pmatrix}^T \tag{4.37}
$$

$$
\mathbf{A}_{r_2} = \begin{pmatrix} r_2 R_2(\theta_i) & r_2 R_2(\theta_f) & r_2 R_2'(\theta_i) & r_2 R_2'(\theta_f) & r_2 R_i^2 R_2''(\theta_i) & r_2 R_f^2 R_2''(\theta_f) & 0 & 0 & 0 & 0 \end{pmatrix}^T \tag{4.38}
$$

where r_2 has an initial value of 0 and will, as mentioned previously, be used later in a Newton-Raphson loop to match the time-of-flight constraint.

By rewriting Equation 4.35 as Equation 4.39 and substituting the various known values, the coefficients can now be found.

$$\begin{pmatrix} r_0 & r_1 & r_3 & r_4 & r_5 & r_6 & b_0 & b_1 & b_2 & c \end{pmatrix}^T = \left[\mathbf{A}\right]^{-1}\left(\mathbf{B}-\mathbf{A}_{r_2}\right) \tag{4.39}$$

4.1.7. EVALUATING THE SHAPING FUNCTIONS AND THEIR DERIVATIVES

Once the coefficients have been determined, the shaping functions $R(\theta)$ and $\Phi(\theta)$ can be evaluated using the equations shown below [2]:

$$\begin{aligned} R(\theta) &= \frac{1}{\mathbf{R}_k \cdot \mathbf{r}_k} \\ \Phi(\theta) &= \mathbf{\Phi}_j \cdot \mathbf{b}_j \end{aligned} \tag{4.40}$$

In order to make the computation of the derivatives easier, Roegiers proposed the auxiliary notation shown in Equation 4.41 [2].

$$Z(\theta) = \frac{1}{R(\theta)} = \mathbf{R}_k \cdot \mathbf{r}_k \tag{4.41}$$

It can now be found that the first, second and third derivative of the radial shaping function are as shown in Equation 4.42. For the derivation of these relations, the reader is referred to Roegiers' master's thesis [2].

$$\begin{aligned} R' &= -Z'R^2 \\ R'' &= -Z''R^2 + 2ZR'^2 \\ R''' &= -Z'''R^2 - 2RR'Z'' + 2Z'R'^2 + 4ZR'R'' \end{aligned} \tag{4.42}$$

where, for convenience, the fact that R and Z are functions of θ has been omitted and the derivatives of the auxiliary Z function are given by [2]:

$$\begin{aligned} Z' &= \mathbf{R}'_k \cdot \mathbf{r}_k \\ Z'' &= \mathbf{R}''_k \cdot \mathbf{r}_k \\ Z''' &= \mathbf{R}'''_k \cdot \mathbf{r}_k \end{aligned} \tag{4.43}$$

Furthermore, the first, second and third derivative of the elevation shaping function are given by [2]:

$$\begin{aligned} \Phi' &= \mathbf{\Phi}'_j \cdot \mathbf{b}_j \\ \Phi'' &= \mathbf{\Phi}''_j \cdot \mathbf{b}_j \\ \Phi''' &= \mathbf{\Phi}'''_j \cdot \mathbf{b}_j \end{aligned} \tag{4.44}$$

in which the equations for the first, second and third derivative of $\mathbf{R}_k$ and $\mathbf{\Phi}_j$ are as shown in Section 4.1.5.

4.1.8. EVALUATING THE CURVATURE

Next, the curvature along the trajectory should be checked. To do so, the time equation scalar function D is used. This function has the same sign as the curvature and is therefore positive when the trajectory is curved towards the central body and negative when it curves away from it. As such, the value of D is checked along the trajectory and the trajectory is discarded if the value of D ever becomes negative. The time equation scalar function at any point along the trajectory can be calculated using the derivatives shown in Section 4.1.7 and Equation 4.45 [2].

$$D(\theta) = -R'' + 2\frac{R'^2}{R} + R'\Phi'\frac{\Phi'' - sin(\Phi)cos(\Phi)}{\Phi'^2 + cos^2(\Phi)} + R\left(\Phi'^2 + cos^2(\Phi)\right) \tag{4.45}$$

4.1.9. SATISFYING THE TIME-OF-FLIGHT CONSTRAINT

In order to calculate the time-of-flight using the current value of r_2, Equation 4.46 is used [2]:

$$TOF = \int_{\theta_i}^{\theta_f + 2\pi N_r} \left(T'\right) d\theta \tag{4.46}$$

in which the first derivative of the time evolution function is as shown in Equation 4.47. To perform the integration, a Runge-Kutta 4 integrator is used. As the global truncation error of this method is $O(h^4)$, with h being the step size, this method is highly accurate [15]. Note also that Equation 4.47 shows that a positive value of D is required in order to obtain a physically correct time-of-flight.

$$T'(\theta) = \frac{dT}{d\theta} = \sqrt{\frac{DR^2}{\mu}} \tag{4.47}$$

Next, the time violation function can be written as follows [2]:

$$\Delta TOF = \begin{cases} TOF_{req} - TOF & \text{if } TOF_{req} \neq 0 \\ 0 & \text{if } TOF_{req} = 0 \end{cases} \tag{4.48}$$

where it should be noted that the implementation is written such that when a time-of-flight of zero is set, the time-of-flight is left free and the time violation function therefore returns 0.

Using a Newton-Raphson loop, the value of the free coefficient r_2 can now be adapted such that the current time-of-flight meets the required time-of-flight. The equation used in the iterative process of determining r_2 can be seen in Equation 4.49 [2].

$$r_{2new} = r_{2old} - \frac{\Delta TOF(r_{2old})}{\frac{d\Delta TOF}{dr_2}(r_{2old})} \tag{4.49}$$

As the required time-of-flight is constant, the derivative of Equation 4.48 with respect to r_2 becomes [2]:

$$\frac{d\Delta TOF}{dr_2}(r_{2old}) = -\frac{dT}{dr_2}(r_{2old}) \tag{4.50}$$

Substitution of this relation into Equation 4.49 results in Equation 4.51.

$$r_{2new} = r_{2old} + \frac{\Delta TOF(r_{2old})}{\frac{dT}{dr_2}(r_{2old})} \tag{4.51}$$

where the derivative of the time evolution function T with respect to the free coefficient is calculated using a two-point difference method:

$$\frac{dT}{dr_2}(r_{2old}) = \frac{T(r_{2old} + h) - T(r_{2old} - h)}{2h} \tag{4.52}$$

For the step size h, a value of $\sqrt{\epsilon}$ is chosen, with ϵ being the machine precision. As the machine precision is approximately $1 \cdot 10^{-16}$, a step size in the order of $1 \cdot 10^{-8}$ is used.

It can be seen in these equations that the process involves iteratively calculating the time-of-flight for various values of r_2. As the value of r_2 changes the values of the **B** vector discussed in Section 4.1.6, all calculations done in Sections 4.1.6 to 4.1.8 need to be repeated for each different value of r_2. This iterative process continues until a certain number of iterations is reached or the time violation function becomes smaller than a certain absolute tolerance. In this case, the maximum number of iterations and absolute tolerance were set equal to the values used to Roegiers, being 25 iterations and a tolerance of 10^{-4} days respectively [2].

4.1.10. COMPUTING THE CONTROL ACCELERATIONS

After the Newton-Raphson loop has been completed, the control vector can be found. This is done by first calculating the first and second derivative of the azimuthal angle with respect to time.

The first derivative of the azimuthal angle with respect to time can be calculated using Equation 4.53.

$$\dot{\theta} = \frac{1}{T'} = \sqrt{\frac{\mu}{DR^2}} \tag{4.53}$$

The second derivative of the azimuthal angle with respect to time can be found using Equation 4.54 [2].

$$\ddot{\theta} = -\frac{T''}{T'^3} \tag{4.54}$$

In this equation, T' is found using Equation 4.9 and T'' is calculated using Equation 4.55 [2].

$$T'' = \frac{1}{2\sqrt{\mu D}}\left(RD' + 2DR'\right) \tag{4.55}$$

in which D' is obtained from Equation 4.56 [2]:

$$\begin{aligned} D'(\theta) = \frac{dD}{d\theta} =& F_1 R' - R''' - \frac{2R'^3}{R^2} + \frac{4R'R''}{R} + F_4\Phi' R + \frac{2\Phi' R'(\Phi''' - \Phi' cos(2\Phi))}{F_3} + \dots \\ & \frac{F_2\Phi' R''}{F_1} + \frac{F_2 R'\Phi''}{F_1} - \frac{4F_4F_2\Phi'^2R'}{F_3^2} \end{aligned} \tag{4.56}$$

where

$$\begin{aligned} F_1 &= \Phi'^2 + cos^2(\Phi) \\ F_2 &= \Phi'' - \frac{sin(2\Phi)}{2} \\ F_3 &= cos(2\Phi) + 2\Phi'^2 + 1 \\ F_4 &= 2\Phi'' - sin(2\Phi) \end{aligned} \tag{4.57}$$

Next, the velocity vector $\tilde{\mathbf{v}}$ and acceleration vector $\tilde{\mathbf{a}}$ along the trajectory parametrised by the azimuthal angle can be calculated using Equations 4.6 and 4.58 [2].

$$\tilde{\mathbf{a}} = \begin{pmatrix} \tilde{a}_r \\ \tilde{a}_\theta \\ \tilde{a}_\phi \end{pmatrix} = \begin{pmatrix} R'' - R\left(\Phi'^2 + cos^2(\Phi)\right) \\ 2R'cos(\Phi) - 2R\Phi' sin(\Phi) \\ 2R'\phi' + R(\Phi'' + sin(\Phi)cos(\Phi) \end{pmatrix} \tag{4.58}$$

The control vector can then be found with Equation 4.59:

$$\dot{\theta}^2 \frac{d^2\mathbf{r}}{d\theta^2} + \ddot{\theta}\frac{d\mathbf{r}}{d\theta} = -\mu\frac{\mathbf{r}}{r^3} + \mathbf{u} \tag{4.59}$$

By substituting the values calculated at a certain position of the trajectory into this equation, the control vector $\mathbf{u}$ in the spherical reference frame at that position can be found.

To obtain the control acceleration in a tangential-normal-out-of-plane reference frame, Equation 4.60 can be used [2]:

$$\mathbf{u} = \begin{pmatrix} u_t \\ u_n \\ u_h \end{pmatrix} = \begin{pmatrix} \frac{\mu}{r^2}\mathbf{e}_r \cdot \mathbf{e}_t + \ddot{\theta}\tilde{\mathbf{v}} \cdot \mathbf{e}_t + \dot{\theta}^2\tilde{a} \cdot \mathbf{e}_t \\ \frac{\mu}{r^2}\mathbf{e}_r \cdot \mathbf{e}_n + \dot{\theta}^2\tilde{\mathbf{a}} \cdot \mathbf{e}_n \\ \dot{\theta}^2\tilde{\mathbf{a}} \cdot \mathbf{e}_h \end{pmatrix} \tag{4.60}$$

4.1.11. DETERMINING THE REQUIRED ΔV

Lastly, in order to determine the required ΔV, $|\mathbf{u}|\frac{dT}{d\theta}$ can simply be integrated over the interval $[\theta_i, \theta_f + 2\pi N_r]$. As done by Roegiers, a Runge-Kutta 4 integrator is used here as well [2]. This required ΔV can then be optimised through tuning of the initial and final positions as well as the required time-of-flight.

4.2. MONTE-CARLO METHOD

To determine the values of the coefficients that cannot be solved using the boundary conditions, the Monte-Carlo method is used as a first step.

The Monte-Carlo method involves generating a large number of random values for each of the coefficients and using the resulting ΔVs to obtain the combination that results in the lowest cost. These random values are generated within certain bounds. As the maximum inclination is known, these bounds can be determined by squaring the respective value for a found in Figure 3.27. This gives an initial guess for the maximum value that the exponential function needs to be able to reproduce. However, it should be noted here that, due to the fact that the exponential function replaces a^2, it should never become negative. If it does become negative, non-logical ΔVs are found. Therefore, the minimum bounds for the a_0 and a_3 coefficients are always set to zero.

It should furthermore also be mentioned that, that due to the random generation of the coefficients, the method is unable to find the absolute minimum. For this reason, the Nelder-Mead algorithm is used in order to further optimise the solution.

4.3. NELDER-MEAD METHOD

The Nelder-Mead optimisation method is an algorithm that uses a simplex to search for a function's minimum. As the method does not require the calculation of any derivatives, it is highly suitable for problems that feature non-smooth functions. In this case, it is used to optimise the additional coefficients that cannot be solved using the boundary conditions. By sorting the solutions found by the Monte-Carlo method with respect to their ΔV and by using the Nelder-Mead method on a number of the best points found, a more optimal solution can be obtained.

It is important to note that the original implementation of the Nelder-Mead method includes the unconstrained optimisation of coefficients, whereas the exponential part of the elevation shaping function should never become negative in the elevation shaping function. To prevent this from occurring, a check is performed throughout the Nelder-Mead algorithm to determine whether the new combination of coefficient values results in the exponential part becoming negative. If this is the case, the cost of using this combination of values is set to an extremely high value in order to move the Nelder-Mead method away from such combinations.

This section shall elaborate upon the Nelder-Mead algorithm and its use within the developed shaping method.

4.3.1. THE INITIAL SIMPLEX

The aforementioned simplex used to find the function's optimum can be described as an n-dimensional version of a triangle. For example, in one dimension it is a line, while in two and three dimensions it is a triangle and a tetrahedron respectively. It is worth noting that the simplex does not need to be symmetrical in any way and that an n-dimensional simplex is made up of n+1 points.

To construct the simplex around a given starting point, n additional points need to be defined. These points are placed at a small distance from the initial point along the direction of each dimension. As the Nelder-Mead method is used individually on a number of the best points found by the Monte-Carlo method, this initial point is different for each of the earlier obtained solutions that are to be optimised with the Nelder-Mead method. For each of the solutions to be optimised, the initial point is defined by its values for the 10 unknown coefficients of the elevation shaping function, where the initial guesses for these coefficients were found through the Monte-Carlo method.

The i^{th} additional point as defined with the use of the unit vector of the i^{th} dimension is given by:

$$\mathbf{x}_i = \mathbf{x}_1 + h(x_1, i) \cdot \mathbf{u}_i \tag{4.61}$$

where $\mathbf{u}_i$ is the unit vector in the i^{th} coordinate, $\mathbf{x}_1$ is the initial point and $\mathbf{x}_i$ is the i^{th} additional point of the simplex [16]. The step size $h(x_1, i)$ is defined as 0.05 if the value of $\mathbf{x}_1$ in the i^{th} coordinate is non-zero and 0.0025 if it is. Thus, if the initial point would be set equal to $(0, 1)$, the second point in the simplex would

be $(0.0025, 1)$ and the third point would be $(0, 1.05)$. The values 0.05 and 0.0025 are equal to those used by MATLAB's *fminsearch* function [17].

Once the initial simplex has been defined, the algorithm can start its first iteration.

4.3.2. ORDERING

At the start of every iteration, the points are ordered based on their cost. Using this ordering, the worst, second worst and best point are found and the indices h, s and l are given to these points respectively.

4.3.3. DETERMINING THE CENTROID

As a second step, the centroid of all points except the worst ($\mathbf{x}_h$) is determined using Equation 4.62 [16].

$$\mathbf{c} = \frac{1}{n} \sum_{i \neq h} \mathbf{x}_i \tag{4.62}$$

4.3.4. TRANSFORMATION

Once the centroid has been found, the current simplex can be transformed into the next iteration's simplex. This new simplex is obtained through a number of steps, each of which will be elaborated upon.

REFLECTION

Firstly, the reflection point $\mathbf{x}_r$ is found using Equation 4.63. If $f(\mathbf{x}_l) \leq f(\mathbf{x}_r) < f(\mathbf{x}_s)$, meaning that the cost of $\mathbf{x}_r$ is lower than the second-worst point but not lower than the current best, $\mathbf{x}_h$ is replaced by $\mathbf{x}_r$ and the algorithm continues to the next iteration [18].

$$\mathbf{x}_r = \mathbf{c} + \alpha(\mathbf{c} - \mathbf{x}_h) \tag{4.63}$$

In this equation α is the reflection parameter and is usually set equal to 1. This reflection step is done in an attempt to move the simplex in the opposite direction of $\mathbf{x}_h$. For a two-dimensional problem, this step can be illustrated as shown in Figure 4.1, where the previous simplex is shown in blue and the new simplex is shown in red [19].

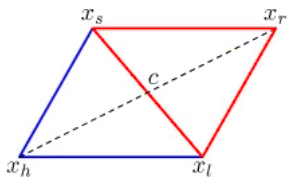


Figure 4.1: An illustration of the reflection step [19].

EXPANSION

If the reflection point $\mathbf{x}_r$ is better than the current best point $\mathbf{x}_l$, the expansion point $\mathbf{x}_e$ is determined using Equation 4.64 in an attempt to further optimise the reflection point. In this equation, γ is the expansion parameter and is usually set equal to 2 [18].

$$\mathbf{x}_e = \mathbf{c} + \gamma(\mathbf{x}_r - \mathbf{c}) \tag{4.64}$$

If $f(\mathbf{x}_e)$ is better than $f(\mathbf{x}_r)$, the point $\mathbf{x}_h$ is replaced by $\mathbf{x}_e$ and the algorithm continues to the next iteration. If the reflection point results in a lower cost than the expansion point, $\mathbf{x}_r$ replaces $\mathbf{x}_h$ instead. This step of the algorithm can be illustrated using Figure 4.2 [19].

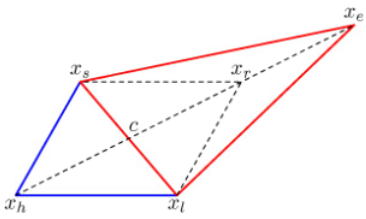


Figure 4.2: An illustration of the expansion step [19].

CONTRACTION

If the reflection point $\mathbf{x}_r$ is instead worse than the second worst point $\mathbf{x}_s$, a contraction is attempted instead using the best of the two points $\mathbf{x}_h$ and $\mathbf{x}_r$. The method used to determine the contraction point depends on which one of these two points is better.

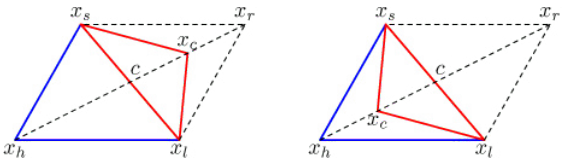
If $f(\mathbf{x}_s) \leq f(\mathbf{x}_r) < f(\mathbf{x}_h)$, the contraction point is determined using Equation 4.65 [18]. If this contraction point results in a cost lower than or equal to the one of the reflection point, the contraction point $\mathbf{x}_c$ replaces $\mathbf{x}_h$ and the algorithm continues to the next iteration. This version of the contraction can be seen in Figure 4.3a [19].

$$\mathbf{x}_c = \mathbf{c} + \beta(\mathbf{x}_r - \mathbf{c}) \tag{4.65}$$

If instead the reflection point $\mathbf{x}_r$ is worse than the current worst point $\mathbf{x}_h$, the contraction point is determined using Equation 4.66 [18]. If this contraction point results in a cost lower than that of the worst point, this point replaces $\mathbf{x}_h$ in the simplex and the algorithm continues to the next iteration. This alternative version of the contraction is illustrated in Figure 4.3b [19].

$$\mathbf{x}_c = \mathbf{c} + \beta(\mathbf{x}_h - \mathbf{c}) \tag{4.66}$$

where β is the contraction parameter and is usually set equal to 0.5.



(a) External contraction. (b) Internal contraction.

Figure 4.3: An illustration of the contraction step [19].

SHRINKAGE

If contraction of the simplex does not result in a better simplex, the entire simplex is redefined through shrinkage. This new simplex consists of the best point and points that are redefined with respect to this best point and the previous points. Once this new simplex has been found, the algorithm continues to the next iteration. The i^{th} point of the new simplex is found using Equation 4.67 [18].

$$\mathbf{x}_i = \mathbf{x}_l + \delta(\mathbf{x}_i - \mathbf{x}_l) \tag{4.67}$$

where δ is the shrinkage parameter and is equal to 0.5 in most implementations.

4.3.5. TERMINATION CONDITIONS

The aforementioned process is continued until one of the following termination conditions is met:

- A set maximum number of iterations is reached
- All points are less than a predefined distance away from each other
- The difference between the costs of each of the points in a simplex is less than a certain predefined value

Once one of these conditions is met, the Nelder-Mead optimisation algorithm stops its search and returns the best point found so far and the corresponding final simplex.

5

VALIDATION

Before the spherical shaping method with the new elevation shaping function can be used, it should be validated. This is done by validating each component of the developed C++ software, as well as by using a number of internal and external validation cases to confirm that the obtained results are correct.

This chapter shall first discuss the validation that ensures that the software used to generate reference cases is installed correctly in Section 5.1. Afterwards, the validation of the ΔV calculation and the implementation of the Monte-Carlo and Nelder-Mead methods are discussed in Sections 5.2 and 5.3. Lastly, the validation using internal and external test cases is elaborated upon in Sections 5.4 and 5.5.

5.1. INSTALLATION OF EXISTING SOFTWARE

The software written by Roegiers for her research on the spherical shaping method will be used to partially validate the software written for this thesis. Therefore, as a first step in the validation process, it can be confirmed that Roegiers' software has been installed correctly by evaluating whether usage of this software generates the same results as those discussed in her MSc thesis. To do so, results will be generated for a number of test cases defined in Roegiers' MSc thesis and these results will be compared to the values found by Roegiers. If these results are equal to one another, Roegiers' software is installed and used correctly, and can be used to validate the software written for this thesis.

This section will start off with a comparison between the results found for a two-dimensional transfer between elliptical orbits in Section 5.1.1. After this, three-dimensional transfers between elliptical orbits and the effect of large inclinations will be discussed in Sections 5.1.2 and 5.1.3 respectively. Information discussed in this section has been taken from *Application of the Spherical Shaping Method to a Low-Thrust Multiple Asteroid Rendezvous Mission* by Roegiers, unless specifically mentioned otherwise [2].

5.1.1. TWO-DIMENSIONAL TRANSFER BETWEEN ELLIPTICAL ORBITS

Firstly, the relatively simple transfer between two co-planar elliptical orbits can be considered. The case names as defined by Roegiers and the initial and final Keplerian states $(a, e, i, \Omega, \omega, \theta)$ of the two cases that are considered can be seen in Table 5.1.

Table 5.1: The initial and final Keplerian states of cases G19 and G25.

Case	Initial Keplerian State	Final Keplerian State
G19	(1, 0.1, 0, 0, 0, 0)	(3, 0.1, 0, 0, 0, 0)
G25	(1, 0.1, 10, 0, 0, 0)	(3, 0.1, 10, 0, 0, 0)

The following two sections shall compare the ΔV and control accelerations for both transfers.

REQUIRED ΔV

By using the Keplerian states shown in Table 5.1 as inputs for the software developed by Roegiers, a comparison between the results obtained with this software and those shown in Roegiers' MSc thesis can be made. To further evaluate the software's installation, the two cases shown above are run twice: once using a free time-of-flight and once using a required time-of-flight of 1000 days. The required ΔV found for each of these cases, as well as the the ΔV found by Roegiers, can be seen in Table 5.2. In this table, the *t* suffix indicates that a required time-of-flight has been set for the test case.

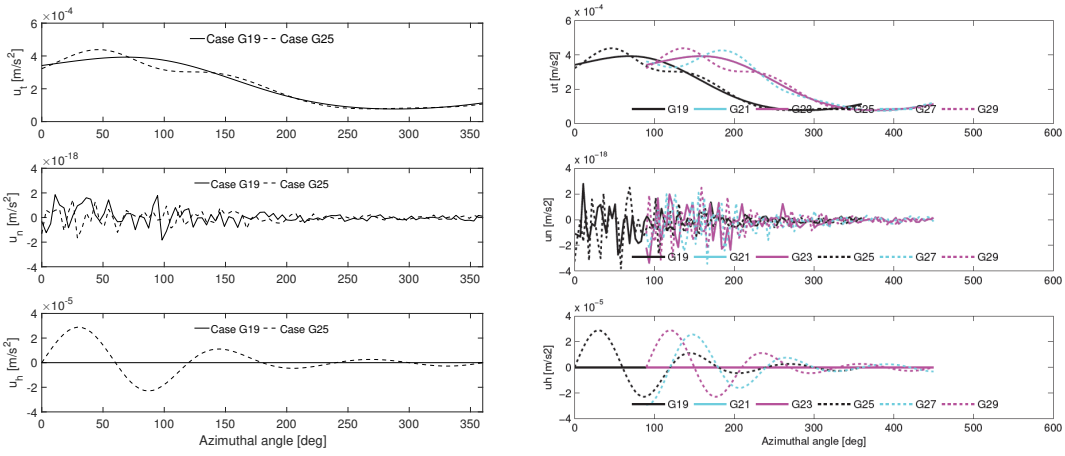
Table 5.2: A comparison between the ΔVs shown in Roegiers' MSc thesis and the values found using her software for cases G19 and G25.

Case	TOF [days]	$\Delta V_{Roegiers}$ [km/s]	ΔV [km/s]	Absolute Error [km/s]
G19	Free	12.33	12.33	$< 1{\cdot}10^{-2}$
G25	Free	12.34	12.34	$< 1{\cdot}10^{-2}$
G19t	1000	12.29	12.29	$< 1{\cdot}10^{-2}$
G25t	1000	12.30	12.30	$< 1{\cdot}10^{-2}$

From this table, it can be noted that the required ΔV found by the software is equal to the one elaborated upon in Roegiers' MSc thesis. It can furthermore be seen that, while the two cases should require an equal amount of ΔV for the same time-of-flight setting, a different ΔV is found to be needed; as was found by Roegiers, the elevation angle influences the ΔV required.

CONTROL ACCELERATIONS

To confirm this difference in ΔV required, control accelerations were generated for the four cases shown in Table 5.2 using the provided code. The control accelerations in the tangential, normal and out-of-plane directions for the cases G19 and G25 can be seen Figure 5.1.

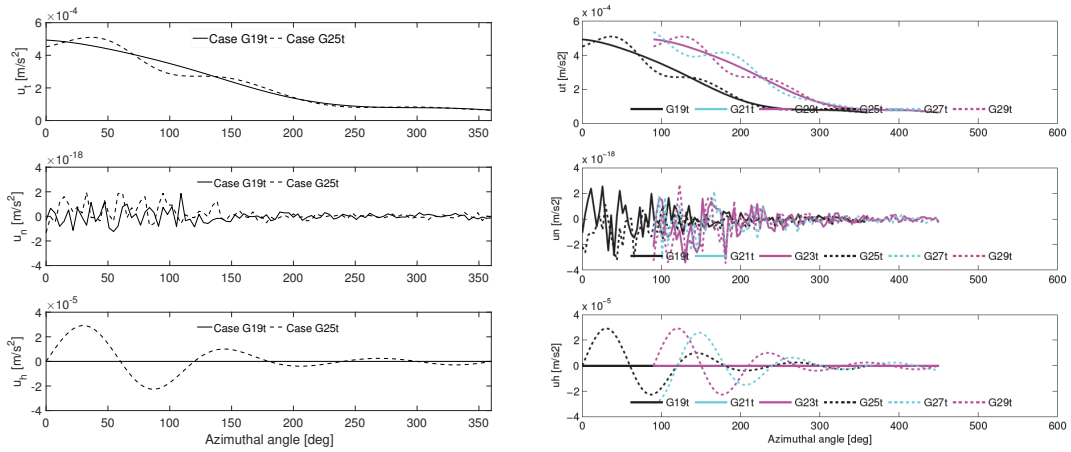


(a) The newly generated control accelerations.

(b) The control accelerations found by Roegiers.

Figure 5.1: The control accelerations corresponding to cases G19 and G25.

Furthermore, a comparison between the control accelerations obtained from Roegiers' software and those shown in her MSc thesis for cases G19t and G25t can be seen in Figure 5.2.



(a) The newly generated control accelerations.

(b) The control accelerations found by Roegiers.

Figure 5.2: The control accelerations corresponding to cases G19t and G25t.

It can clearly be seen in Figures 5.1 and 5.2 that the control accelerations change when a different initial inclination is used. Noteworthy is also that the control accelerations shown in these figures correspond to those found by Roegiers, thus confirming that the software has been installed and used as intended.

5.1.2. THREE-DIMENSIONAL TRANSFER BETWEEN ELLIPTICAL ORBITS

Now that the two-dimension transfers have been shown to be reproducible, one can determine whether the three-dimensional transfer between elliptical orbits can also be reproduced. To do so, results are generated for four cases using the aforementioned software and their required ΔVs and control profiles are again compared to those shown in Roegiers' MSc thesis. As can be seen in Table 5.3, an inclination change of 10 degrees is performed using two different initial inclinations and right ascensions of the ascending node.

Table 5.3: The initial and final Keplerian states of cases G31, G37, G35 and G41.

Case	Initial Keplerian State	Final Keplerian State
G31	$(1, 0.1, 0, 0, 0, 0)$	$(3, 0.1, 10, 0, 0, 0)$
G37	$(1, 0.1, 5, 0, 0, 0)$	$(3, 0.1, 15, 0, 0, 0)$
G35	$(1, 0.1, 0, 0, 90, 0)$	$(3, 0.1, 10, 90, 0, 0)$
G41	$(1, 0.1, 5, 0, 90, 0)$	$(3, 0.1, 15, 90, 0, 0)$

REQUIRED ΔV

The four transfers shown in Table 5.3 are again evaluated for the case in which the time-of-flight is free, and for the case in which the required time-of-flight has been set to 1000 days. A comparison between the ΔVs found using the software and the values elaborated upon in Roegiers' MSc thesis can be seen in Table 5.4.

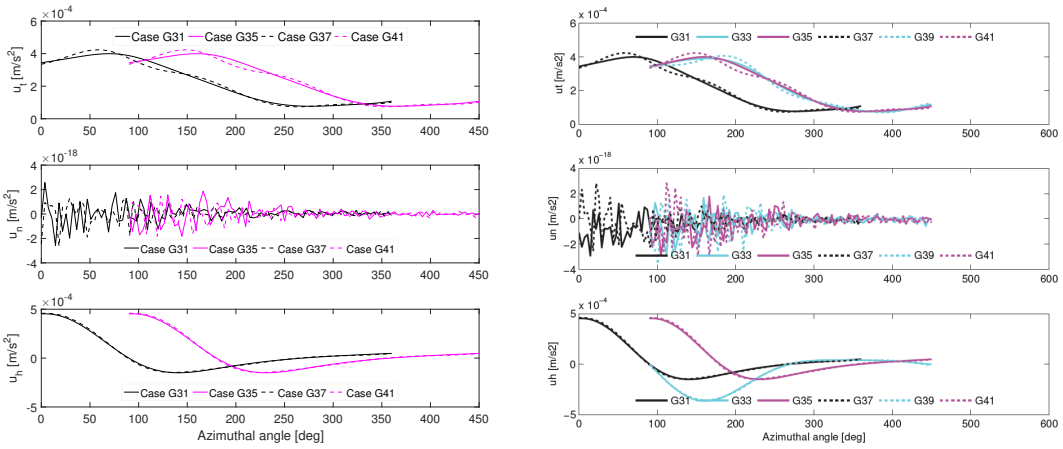
Table 5.4: A comparison between the ΔV found by Roegiers and those found with the provided software for cases G31, G37, G35 and G41.

Case	TOF [days]	$\Delta V_{Roegiers}$ [km/s]	ΔV [km/s]	Absolute Error [km/s]
G31	Free	13.75	13.75	$< 1{\cdot}10^{-4}$
G37	Free	13.77	13.77	$< 1{\cdot}10^{-4}$
G35	Free	13.75	13.75	$< 1{\cdot}10^{-4}$
G41	Free	13.77	13.77	$< 1{\cdot}10^{-4}$
G31t	1000	13.63	13.63	$< 1{\cdot}10^{-4}$
G37t	1000	13.65	13.65	$< 1{\cdot}10^{-4}$
G35t	1000	13.63	13.63	$< 1{\cdot}10^{-4}$
G41t	1000	13.65	13.65	$< 1{\cdot}10^{-4}$

Aside from the fact that the ΔVs obtained from the provided software are equal to those found by Roegiers, thus showing that the provided C++ code functions as intended, it can also be noted that the inclination at which the transfer is started influences the required amount of ΔV and that the right ascension of the ascending node (RAAN), as expected, does not.

CONTROL ACCELERATIONS

With the use of the four transfers shown in Table 5.3, a set of control accelerations is generated for the case in which the time-of-flight is free, and for the case in which the required time-of-flight is 1000 days. The control accelerations for the former case can be seen in Figure 5.3.

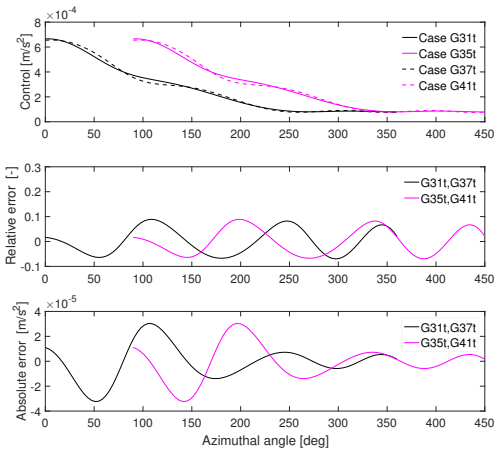


(a) The newly generated control accelerations.

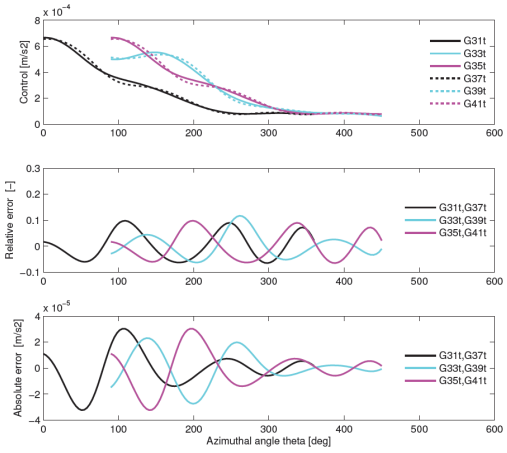
(b) The control accelerations found by Roegiers.

Figure 5.3: The control accelerations corresponding to cases G31, G37, G35 and G41.

While no tangential, normal and out-of-plane control accelerations are presented for cases G31t, G37t, G35t and G41t by Roegiers, the ΔV difference for these cases can still be investigated by comparing the control magnitudes. This comparison can be seen in Figure 5.4.



(a) The newly generated control profiles.



(b) The control profile found by Roegiers.

Figure 5.4: The control profile corresponding to cases G31t, G37t, G35t and G41t.

As was found for the two-dimensional transfer, the initial inclination has an effect upon the control accelerations that are generated. Furthermore, by comparing the accelerations of cases G31 and G35 shown in Figure 5.3, it can be seen that changing the RAAN solely shifts the graph over the x-axis. As expected, the RAAN therefore does not influence the magnitude of control accelerations.

5.1.3. LARGE INCLINATIONS

To further confirm the results of the provided code and this code's properties, one can observe how the required ΔV changes when the initial inclination is increased even further. The initial and final Keplerian states of a number of co-planar transfers at increasing inclinations can be seen in Table 5.5. For these cases, a free time-of-flight is used. To indicate that these cases are used to evaluate the effect of large inclinations, the suffix *i* has been added to these cases' names.

Table 5.5: The initial and final Keplerian states of cases P1i, P7i, P9i and P10i.

Case	Initial Keplerian State	Final Keplerian State
P1i	(1, 0, 0, 0, 0, 0)	(3, 0, 0, 0, 0, 0)
P7i	(1, 0, 20, 0, 0, 0)	(3, 0, 20, 0, 0, 0)
P9i	(1, 0, 40, 0, 0, 0)	(3, 0, 40, 0, 0, 0)
P10i	(1, 0, 50, 0, 0, 0)	(3, 0, 50, 0, 0, 0)

A comparison between the required ΔV for each of these four transfers can be seen in Table 5.6. As the required ΔV should theoretically be equal for all four transfers, the last column of this table shows the relative difference with respect to case P1i.

Table 5.6: A comparison between the ΔV found by Roegiers and those found using the provided software for cases P1i, P7i, P9i and P10i.

Case	$\Delta V_{Roegiers}$ [km/s]	ΔV [km/s]	Absolute Error [km/s]	(Pii-P1i)/P1i
P1i	12.31	12.31	$< 1{\cdot}10^{-4}$	0
P7i	12.83	12.83	$< 1{\cdot}10^{-4}$	$4.22{\cdot}10^{-2}$
P9i	47.57	47.57	$< 1{\cdot}10^{-4}$	2.86
P10i	-, D<0	-, D<0	-	-

From this table, one can conclude that the error compared to the P1i case increases as the inclination of the coplanar transfer increases. Furthermore, it can be seen that the spherical shaping method is unable to generate a transfer orbit when the inclination is set to 50 degrees due to the trajectory's curvature becoming negative.

A last property of the provided software that can be investigated, is the consequence of splitting up a large inclination change into a number of smaller inclination changes. In this case, the sum of the control profiles of the latter should be equal to the control profile of the former. To investigate this, the three transfers shown in Table 5.7 can be considered. The suffix ic has been added to these cases' names in order to indicate that they are used to test the splitting up a transfer's inclination change.

Table 5.7: The initial and final Keplerian states of cases G7ic, G8ic and G10ic.

Case	Initial Keplerian State	Final Keplerian State
G7ic	(1, 0, 0, 0, 0, 0)	(1, 0, 10, 0, 0, 0)
G8ic	(1, 0, 10, 0, 0, 0)	(1, 0, 20, 0, 0, 0)
G10ic	(1, 0, 0, 0, 0, 0)	(1, 0, 20, 0, 0, 0)

As can be seen in Table 5.7, the aforementioned property can be evaluated by summing the control profiles of cases G7ic and G8ic, and comparing the result with case G10ic. The control profiles for cases G7ic, G8ic and G10ic, as well as the summation of the control profiles for cases G7ic and G8ic can be seen in Figure 5.5.

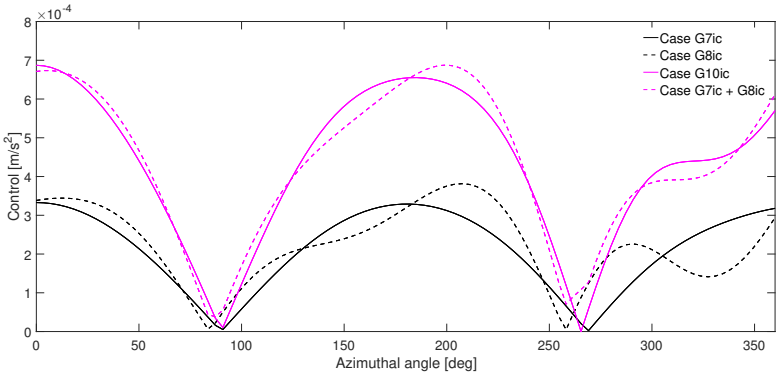


Figure 5.5: The control profiles corresponding to cases G7ic, G8ic and G10ic.

It is clear from this figure that the summation of the two control profiles does not equal the control profile of case G10ic. As also discussed by Roegiers, the cause of this is the aforementioned influence of the initial inclination upon the control profile.

5.2. VALIDATION OF ΔV CALCULATION

As mentioned in Section 5.1, the software written and validated by Roegiers can be used to partially validate the software written for this thesis. To do so, the original elevation shaping function used by Roegiers and Novak was also implemented in the newly developed software. As the sole difference between the original and improved spherical shaping method is the elevation shaping function, using the original function in the newly developed software should result in the exact same ΔVs as those found by Roegiers.

By changing the elevation shaping function used by the software written for this thesis to the original function, the ΔVs shown in Table 5.8 were found for five different cases. For reference, the ΔV found by using Roegiers' software and the absolute difference are shown as well.

Table 5.8: A comparison between the ΔV found using Roegiers' software and the software developed for this thesis.

Case	TOF [days]	$\Delta V_{Roegiers}$ [km/s]	ΔV [km/s]	Absolute Difference [km/s]
P1i	Free	12.31	12.31	$< 1{\cdot}10^{-13}$
P9i	Free	47.57	47.57	$< 1{\cdot}10^{-13}$
G10ic	Free	17.05	17.05	$< 1{\cdot}10^{-13}$
G35t	1000	13.63	13.63	$< 1{\cdot}10^{-8}$
G41t	1000	13.65	13.65	$< 1{\cdot}10^{-9}$

In this table, it can be seen that the same ΔVs are found when the elevation shaping function is changed to the original function. As such, it can be concluded that the code, with the exception of the calculations regarding the alternative elevation shaping function, has been validated.

The part of the ΔV calculation that is still to be validated, is the calculation of $\mathbf{\Phi}_j$, $\mathbf{\Phi}'_j$, $\mathbf{\Phi}''_j$ and $\mathbf{\Phi}'''_j$. To do this, the symbolic computing environment Maple 2016 was used. While Maple 2016 was also used to find these derivatives, only part of the derivation was done using this software. By instead fully deriving the derivatives using Maple and by comparing the output when a known input is used, the derivatives can be validated.

As a known input, $1 \cdot 10^5$ combinations of coefficients were used. These combinations were randomly generated using a uniform distribution and the following bounds:

Table 5.9: The boundaries used for the random generation of coefficient values.

Coefficient	a_0	a_1	a_2	a_3	a_4	a_5	p_0	p_1	p_2	p_3	θ
Minimum	0.00	-5.00	-5.00	0.00	-5.00	-5.00	-2π	-5.00	-5.00	-2π	0
Maximum	5.00	5.00	5.00	5.00	5.00	5.00	2π	5.00	5.00	2π	2π

An overview of the maximum relative differences between the two methods of calculating $\mathbf{\Phi}_j$, $\mathbf{\Phi}'_j$, $\mathbf{\Phi}''_j$ and $\mathbf{\Phi}'''_j$ can be seen in Table 5.10. In the calculation of these relative differences, the values obtained from the equations that were fully derived using Maple were used as reference values.

Table 5.10: The maximum relative difference between the values found when Maple 2016 is used for parts of the derivation of $\mathbf{\Phi}_j$, $\mathbf{\Phi}'_j$, $\mathbf{\Phi}''_j$ and $\mathbf{\Phi}'''_j$, and the values found when the equations are used of which the derivation was done completely by Maple 2016.

	$j=0$	$j=1$	$j=2$	$j=3$
$\mathbf{\Phi}_j$ [-]	$4.69 \cdot 10^{-16}$	$4.58 \cdot 10^{-16}$	$6.09 \cdot 10^{-16}$	0
$\mathbf{\Phi}'_j$ [-]	17.00	$7.30 \cdot 10^{-9}$	$9.45 \cdot 10^{-9}$	0
$\mathbf{\Phi}''_j$ [-]	8.00	8.00	$6.14 \cdot 10^{-3}$	0
$\mathbf{\Phi}'''_j$ [-]	12.00	$1.03 \cdot 10^{2}$	6.64	0

It can be seen here that the majority of the maximum relative differences are sufficiently small, but also that there are a number of large differences. To obtain more insight into the cause of this, the corresponding absolute differences have been calculated as well and have been summarised in Table 5.11.

Table 5.11: The maximum absolute difference between the values found when the derivation of $\mathbf{\Phi}_j$, $\mathbf{\Phi}'_j$, $\mathbf{\Phi}''_j$ and $\mathbf{\Phi}'''_j$ is done completely by Maple 2016 and the values found using the equations of which the derivation was only partially done by Maple 2016.

	$j=0$	$j=1$	$j=2$	$j=3$
$\mathbf{\Phi}_j$ [rad]	$4.44 \cdot 10^{-16}$	$2.66 \cdot 10^{-15}$	$6.66 \cdot 10^{-16}$	0
$\mathbf{\Phi}'_j$ [rad/rad]	$4.72 \cdot 10^{-16}$	$6.97 \cdot 10^{-11}$	$7.32 \cdot 10^{-10}$	0
$\mathbf{\Phi}''_j$ [1/rad]	$3.73 \cdot 10^{-9}$	$2.33 \cdot 10^{-8}$	$7.30 \cdot 10^{-4}$	0
$\mathbf{\Phi}'''_j$ [1/rad^2]	$2.13 \cdot 10^{-14}$	$5.74 \cdot 10^{-13}$	$2.07 \cdot 10^{-10}$	0

By comparing Tables 5.10 and 5.11, it can be observed that all significantly large relative differences correspond to small absolute differences. For example, the relative difference of $1.03 \cdot 10^2$ of $\mathbf{\Phi}'''_1$ corresponds to an absolute difference of $5.74 \cdot 10^{-13}$ 1/rad^2. This absolute difference is nearing the machine precision of approximately $1 \cdot 10^{-16}$ when one considers that such machine errors propagate throughout the numerous equations used to calculate the derivatives. As such, this indicates that the derivatives were correctly derived and that they can therefore be considered as validated.

5.3. VALIDATION OF MONTE-CARLO AND NELDER-MEAD METHODS

In order to validate the implementation of the Monte-Carlo and Nelder-Mead methods, they were applied together to the Rastrigin function by taking the 15 best points found by the Monte-Carlo method as input for the Nelder-Mead algorithm. The Rastrigin function is one of the commonly used function to test optimisation algorithms and has several local minima. The Rastrigin function is shown in Equation 5.1, in which d is the dimension [20].

$$f(\mathbf{x}) = 10d + \sum_{i=1}^{d} \left[x_i^2 - 10cos(2\pi x_i) \right] \tag{5.1}$$

To test the algorithm, two dimensions were used. The input domain for the function is $x_i \in [-5.12, 5.12]$ for all $i = 1, ..., d$ and the global minimum is $f(\mathbf{x}) = 0$ at $\mathbf{x} = \mathbf{0}$. Using these conditions and Equation 5.1, Figure 5.6 can be generated.

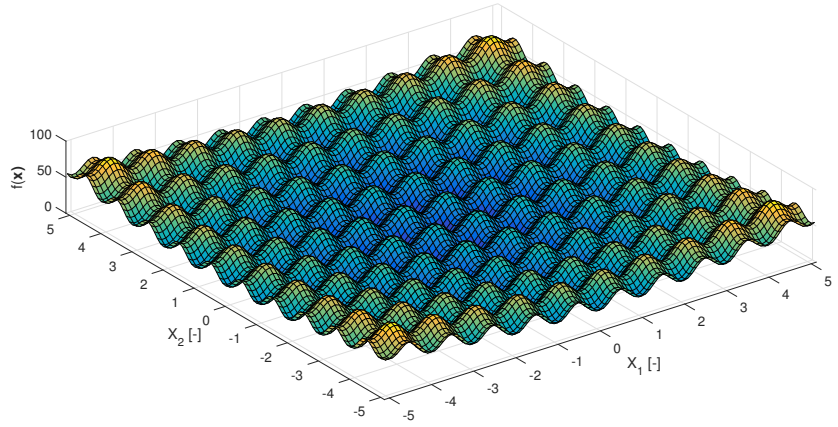


Figure 5.6: The Rastrigin function over the domain $x_1, x_2 \in [-5.12, 5.12]$.

With the use of the settings shown in Table 5.12, the combination of algorithms found the minimum $f(\mathbf{x}) = 4.29 \cdot 10^{-10}$ at $\mathbf{x} = [1.45 \cdot 10^{-6}, -2.41 \cdot 10^{-7}]$. The combination was therefore capable of finding the global minimum with a high accuracy, thus validating the implementation of the Monte-Carlo and Nelder-Mead methods. The evolution of the simplex that found the global minimum is shown in Figure 5.7.

Table 5.12: The settings used to validate the implementation of the Monte-Carlo and Nelder-Mead methods.

MC Samples	200
NM Sample Size	15
NM Distance Threshold	$1 \cdot 10^{-15}$
NM ΔV Threshold	$1 \cdot 10^{-10}$
NM Max. Iterations	2000

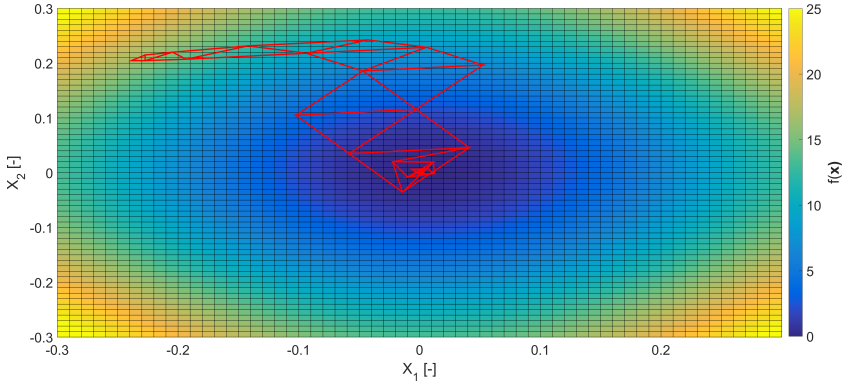


Figure 5.7: The evolution of the simplex that converged to the global minimum.

5.4. INTERNAL VALIDATION

To internally validate the developed shaping function, the test cases shown in Tables 5.1, 5.3 and 5.5 are used. They are elaborated upon in Sections 5.4.1, 5.4.2 and 5.4.3 respectively. By using the same test cases as shown in Section 5.1, a comparison can be made.

5.4.1. PLANAR TEST CASES

The first part of the validation consists of a number of planar test cases. In order to be able to compare the new shaping function with the original one, the test cases shown in Table 5.1 will be evaluated. To do so, the following settings are used for the Monte-Carlo and Nelder-Mead methods:

Table 5.13: The settings used to internally validate the developed shaping function.

MC Samples	$1{\cdot}10^{5}$
NM Sample Size	50
NM Distance Threshold	$1{\cdot}10^{-15}$
NM ΔV Threshold	$1{\cdot}10^{-10}$
NM Max. Iterations	2000
Integration Step Size	$2\pi/100$

and the boundaries shown in Table 5.14 are used for the generation of random values for each variable in the Monte-Carlo method. These values were found by using Figure 3.27 as an initial approximation of the coefficient a. The boundaries on the phasing coefficients p_0 to p_3 were set to be large enough such that the whole function could be shifted extensively.

Table 5.14: The boundaries used for the Monte-Carlo method.

Coefficient	a_0	a_1	a_2	a_3	a_4	a_5	p_0	p_1	p_2	p_3
Minimum	0.00	-0.05	-0.05	0.00	-0.05	-0.05	-2π	-5.00	-5.00	-2π
Maximum	0.05	0.05	0.05	0.05	0.05	0.05	2π	5.00	5.00	2π

Furthermore, it should be noted that for each of the test cases in this and the following sections, an initial true anomaly of $1\cdot10^{-12}$ degrees was used instead of 0 degrees. This was done to prevent a singularity caused by the new elevation shaping function. Using these settings, the ΔVs shown in Table 5.15 were found for the planar test cases.

Table 5.15: The ΔVs found for the planar test cases.

Case	TOF [days]	ΔV [km/s]
G19	Free	12.329
G25	Free	12.330
G19t	1000	12.286
G25t	1000	12.291

As an equal change in the initial and final inclination does not have an impact on the ΔV, G19 and G25, as well as G19t and G25t should be equal to one another. However, it can be seen from these results that a small difference in ΔV is nonetheless present. By comparing these results with those of the original spherical shaping method shown in Table 5.2, it can be seen that the new elevation shaping function does perform better than the previous one, as the difference in ΔV due to the inclination decreases.

Furthermore, while at a first glance it may appear counter-intuitive that the time-constrained cases have a lower ΔV than those with a free time-of-flight, this can be explained by looking more closely at the usage of the spherical shaping method and the matching of the time-of-flight as described in Section 4.1. When the time-of-flight is left unconstrained, the r_2 coefficient is set to zero and the other coefficients are solved such that the other boundary conditions are met. When the time-of-flight is constrained, a Newton-Raphson loop

is used to find the value of r_2 that matches the time-of-flight. As the coefficients are thus solved purely such that the boundary conditions are met, the coefficients found in both cases are not necessarily the most optimal ones in terms of ΔV. Neither case therefore includes the varying of the coefficients to optimise the ΔV when the boundary conditions have been set. Instead, this optimisation of the ΔV can be done by optimising the boundary conditions such as the departure and arrival conditions and the time-of-flight.

By plotting the control accelerations of case G19 and case G25, Figure 5.8 can be obtained.

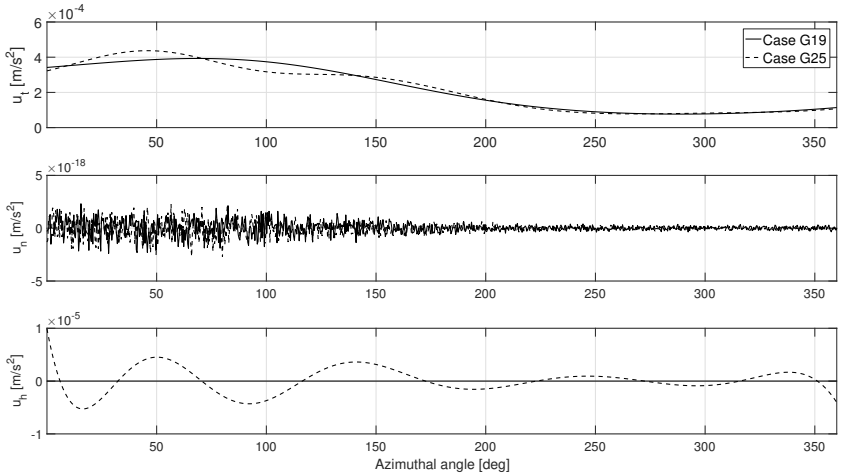


Figure 5.8: The control accelerations found for cases G19 and G25.

As can be noted from this figure, the normal component of the thrust is practically zero. This is as expected, since it was set to zero in the derivation of the first derivative of time with respect to the azimuthal angle [2]. Furthermore, it can be seen that both the out-of-plane and radial thrust acceleration vary when the transfer is placed at an inclination, even though these should be the same for the two cases due to the fact that these accelerations are shown in the tangential-normal-out-of-plane reference frame. This is also the case for test cases G19t and G25t, as can be seen in Figure 5.9.

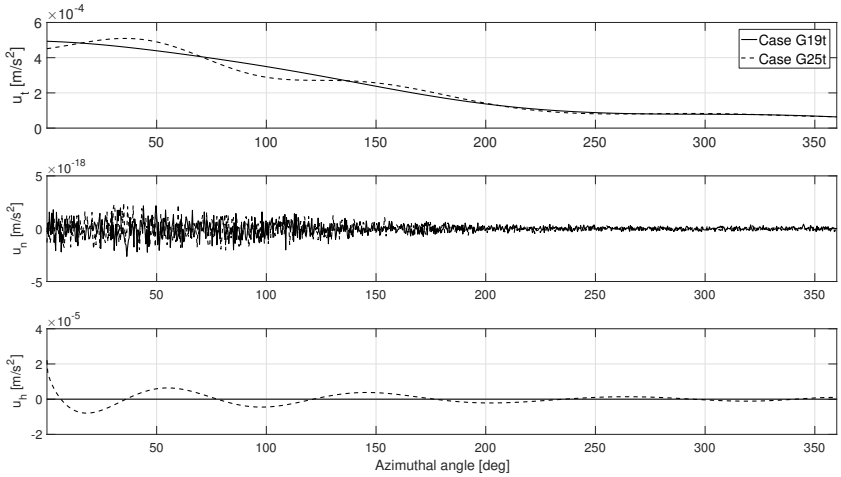


Figure 5.9: The control accelerations found for cases G19t and G25t.

5.4.2. NON-PLANAR TEST CASES

In order to further test the developed elevation shaping function, the four non-planar test cases shown in Table 5.3 were evaluated as well for both a free time-of-flight and a required time-of-flight of 1000 days. The ΔVs found for each of these test cases can be seen in Table 5.16.

Table 5.16: The ΔVs found for the non-planar test cases.

Case	TOF [days]	ΔV [km/s]
G31	Free	13.394
G37	Free	13.386
G35	Free	13.392
G41	Free	13.388
G31t	1000	13.381
G37t	1000	13.364
G35t	1000	13.370
G41t	1000	13.367

It can be seen from these results that a shift in the RAAN slightly impacts the ΔV found. However, this is likely to be caused by the method used to determine the coefficients, as p_3 can change the phase of the elevation shaping function as required. Furthermore, these results confirm the conclusion drawn from Table 5.15 that an equal shift in the initial and final inclination results in a different value for the ΔV. Nonetheless, by comparing these results with those shown in Table 5.4, it can again be seen that this shift in inclination has a smaller impact than before. The control profiles corresponding to the values shown in Table 5.16 can be seen in Figures 5.10 and 5.11.

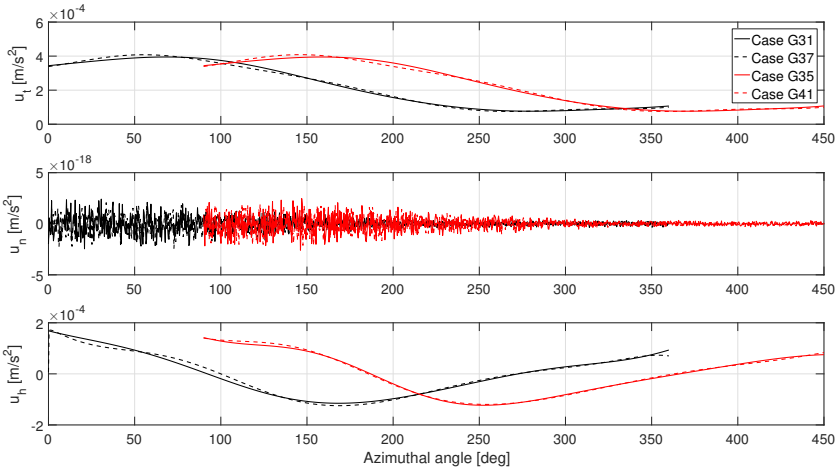


Figure 5.10: The control accelerations found for cases G31, G37, G35 and G41.

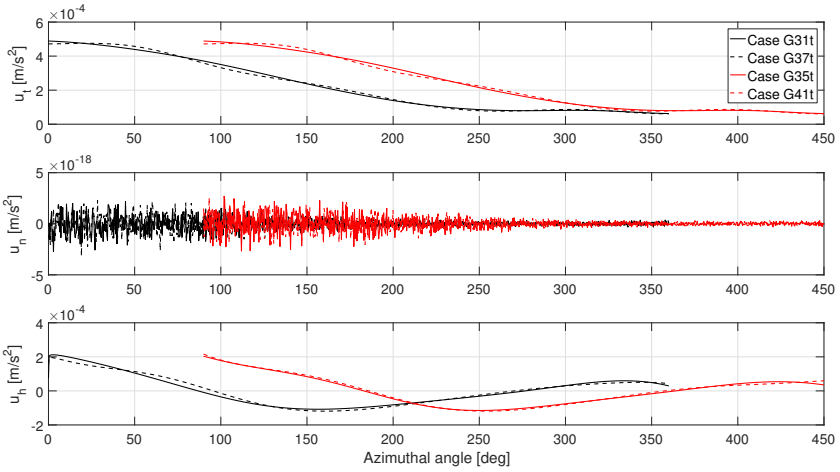


Figure 5.11: The control accelerations found for cases G31t, G37t, G35t and G41t.

5.4.3. HIGH INCLINATIONS

Thirdly, one can look at how the shaping function performs when transfers are done at high inclinations. The test cases shown in Table 5.5 that were used by Roegiers to evaluate the original elevation shaping function were used to evaluate the new elevation shaping function as well. The resulting ΔVs can be seen in Table 5.17. The last column of this table shows the relative error in ΔV with respect to case P1i, as all four transfers should theoretically have the same required ΔV.

Table 5.17: The ΔVs found for the test cases shown in Table 5.5 and their relative difference to case P1i.

Case	TOF [days]	ΔV [km/s]	(Pii-P1i)/P1i
P1i	Free	12.31	0
P7i	Free	12.43	$9.77 \cdot 10^{-3}$
P9i	Free	18.19	$4.77 \cdot 10^{-1}$
P10i	Free	31.02	1.52

By comparing these results with those shown in Table 5.6, it can be concluded that the relative error in ΔV shown in the last column has decreased by at least a factor 4 for these test cases. While the curvature furthermore does not become negative at 50 degrees as it does when the original elevation shaping function is used, the error in ΔV nonetheless becomes significantly large. This error in ΔV can also be observed through the control accelerations shown in Figure 5.12.

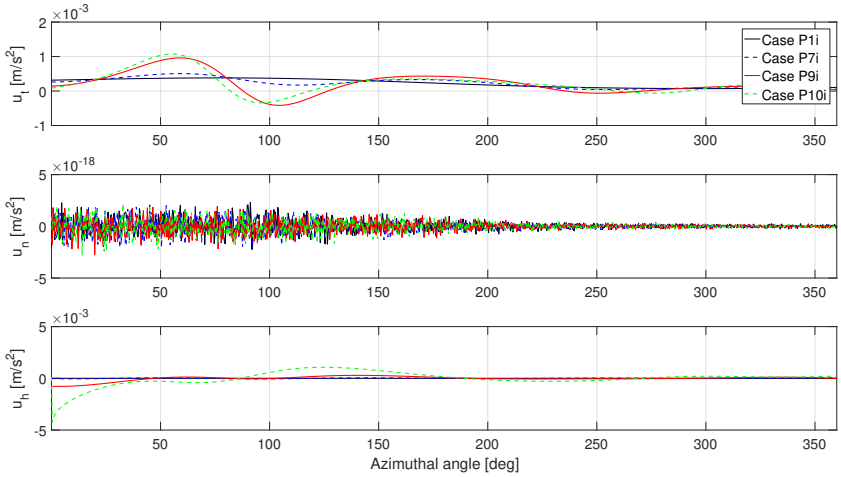


Figure 5.12: The control accelerations found for cases P1i, P7i, P9i and P10i.

5.4.4. CHANGE IN RIGHT ASCENSION OF THE ASCENDING NODE

As mentioned during the development of the new elevation shaping function, another limitation of the currently available software is the ability to accurately calculate the ΔV when a change of the RAAN is performed under an inclination. To test the new elevation shaping function's ability to produce results for such cases, a number of reference cases are generated first. These reference cases consist of changing the RAAN of a circular orbit at a small inclination of $1 \cdot 10^{-5}$ degrees, as the RAAN is undefined at an inclination of 0 degrees. The initial and final Keplerian states of the test cases can be found in Table 5.18.

Table 5.18: The initial and final Keplerian states of the test cases of which the results will be used as reference.

Case	Initial Keplerian State	Final Keplerian State
P1Omega0	$(1, 0, 10^{-5}, 0, 0, 0)$	$(3, 0, 10^{-5}, 10, 0, 0)$
P2Omega0	$(1, 0, 10^{-5}, 10, 0, 0)$	$(3, 0, 10^{-5}, 20, 0, 0)$
P5Omega0	$(1, 0, 10^{-5}, 0, 0, 0)$	$(3, 0, 10^{-5}, 60, 0, 0)$
P6Omega0	$(1, 0, 10^{-5}, 60, 0, 0)$	$(3, 0, 10^{-5}, 120, 0, 0)$
P9Omega0	$(1, 0, 10^{-5}, 0, 0, 0)$	$(3, 0, 10^{-5}, 120, 0, 0)$
P10Omega0	$(1, 0, 10^{-5}, 120, 0, 0)$	$(3, 0, 10^{-5}, 240, 0, 0)$

It can be seen in this table that the test cases consist of a number of pairs and that the initial and final Keplerian states of the two cases in each pair were chosen such that they should have the same required ΔV.

The ΔVs found for the cases shown above when the new elevation shaping function is used can be seen in Table 5.19.

Table 5.19: The ΔVs corresponding to the test cases shown in Table 5.18.

Case	ΔV [km/s]
P1Omega0	12.33
P2Omega0	12.33
P5Omega0	12.40
P6Omega0	12.40
P9Omega0	12.46
P10Omega0	12.46

With the reference values generated, the elevation shaping function's capability to produce accurate results when both the initial and final orbit have a non-zero inclination can be evaluated. This is done by giving each of the orbits shown in Table 5.18 an inclination of 40 degrees. The resulting test cases are shown in Table 5.20.

Table 5.20: The test cases used to test the new elevation shaping function's capability to produce trajectories for cases that have an inclination and a change in RAAN.

Case	Initial Keplerian State	Final Keplerian State
P1Omega	$(1, 0, 40, 0, 0, 0)$	$(3, 0, 40, 10, 0, 0)$
P2Omega	$(1, 0, 40, 10, 0, 0)$	$(3, 0, 40, 20, 0, 0)$
P5Omega	$(1, 0, 40, 0, 0, 0)$	$(3, 0, 40, 60, 0, 0)$
P6Omega	$(1, 0, 40, 60, 0, 0)$	$(3, 0, 40, 120, 0, 0)$
P9Omega	$(1, 0, 40, 0, 0, 0)$	$(3, 0, 40, 120, 0, 0)$
P10Omega	$(1, 0, 40, 120, 0, 0)$	$(3, 0, 40, 240, 0, 0)$

When the new elevation shaping function is used to generate trajectories for these cases, the ΔVs shown in Table 5.21 are found. The ΔVs found when the previous elevation shaping function is used are shown here for reference as well.

Table 5.21: The ΔVs found for the cases shown in Table 5.20 when the previous and new elevation shaping functions are used.

Case	$\Delta V_{previous}$ [km/s]	ΔV_{new} [km/s]
P1Omega	49.49	20.95
P2Omega	49.49	19.63
P5Omega	44.15	26.78
P6Omega	44.15	26.31
P9Omega	41.47	39.47
P10Omega	41.47	37.79

Firstly, it can be noticed that the ΔVs found are significantly larger than those shown in Table 5.19 and that they are not the same for each of the two cases within a pair. This is likely caused by the optimisation technique, as the p_3 coefficient should be able to change the phase of the elevation shaping function as required and a similar observation was made in Section 5.4.2. Nonetheless, the ΔV for each case is lower than the one found using the previous elevation shaping function.

It can furthermore be concluded from these results that the new elevation shaping function, despite the predictions made during the development, does not seem to be capable of accurately producing results for the cases shown above. Part of the reason for this is the error introduced by the high inclination. As case P9i shown in Table 5.17 consists of a comparable transfer but without the RAAN change, it is estimated that the aforementioned error for these cases is similarly in the order of 6 km/s.

By subtracting this estimated error from the results shown in Table 5.21 and by comparing them with those shown in Table 5.19, the ΔV for cases P1Omega and P2Omega comes relatively close to the ones of cases P1Omega0 and P2Omega0. However, the ΔV of cases P5Omega and onwards still have a significant difference with their counterparts shown in Table 5.18.

For reference, the total control accelerations have been plotted as well in Figure 5.13. In this figure, the top graph depicts the control accelerations corresponding to the ΔVs of cases P1Omega0 to P10Omega0 shown in Table 5.19 and the bottom graph depicts the control accelerations corresponding to the ΔVs of cases P1Omega to P10Omega shown in the last column of Table 5.21.

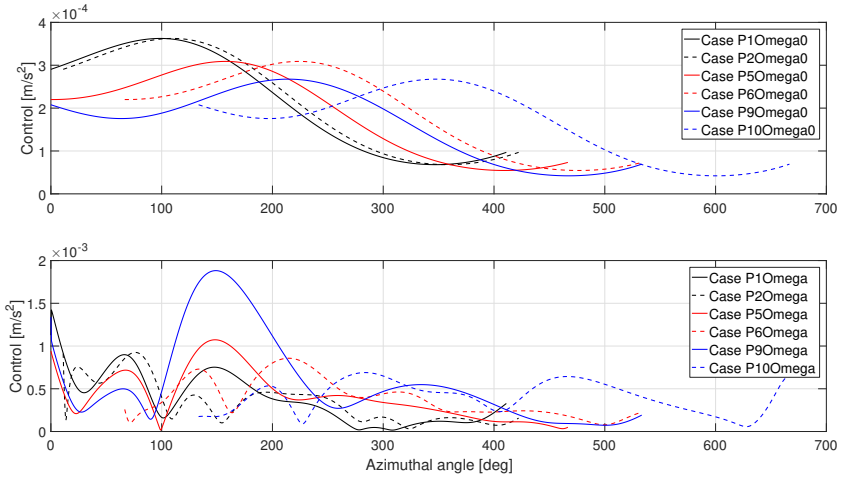


Figure 5.13: Top: the control profiles of cases P1Omega0 to P10Omega0. Bottom: the control profiles of cases P1Omega to P10Omega.

5.4.5. LOOSENING OF BOUNDARY CONDITIONS

In an attempt to further improve the developed shaping function and the results produced by it, case P10i has been investigated further. By calculating the variation of the elevation angle of the departure and target orbits using Equation 3.12 and plotting them together with the change in elevation angle of the trajectory, Figure 5.14 can be obtained.

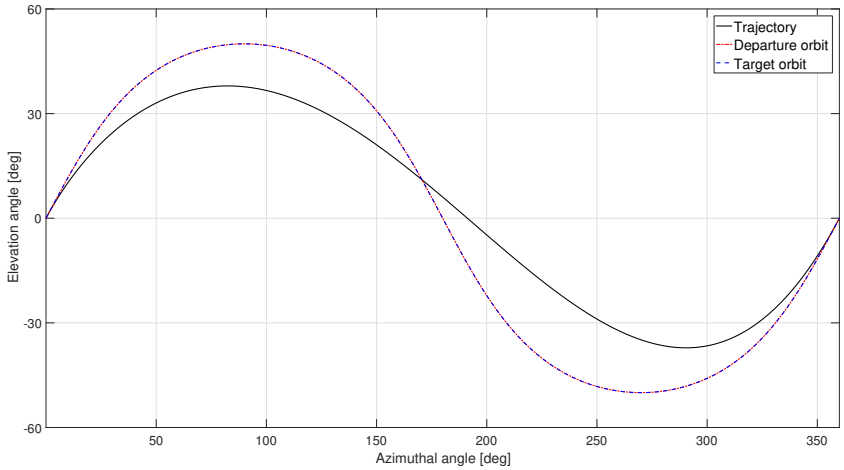


Figure 5.14: The elevation angle as a function of the azimuthal angle for the trajectory, departure and target orbits of case P10i.

As the inclination of this case does not change, one can compare the elevation angle of the trajectory with that of the departure and target orbits. It was found that the root-mean-square error (RMSE) between the orbits was 12.40 degrees. This error is significantly larger than the errors that were predicted in Section 3.3.4.

As the MATLAB implementation that was used in Section 3.3.4 to calculate the RMSE did not strictly enforce the boundary conditions, whereas the C++ implementation discussed in Chapter 4 does, loosening the boundary conditions may reduce the RMSE.

To test this concept, the settings shown in Table 5.13 were applied to the modelling of a Keplerian orbit of which the initial and final Keplerian elements are shown in Table 5.22.

Table 5.22: The initial and final Keplerian states for the test case used to evaluate the effect of loosening the boundary conditions.

Initial Keplerian State	Final Keplerian State
$(1, 0, 50, 0, 0, \theta_i)$	$(1, 0, 50, 0, 0, \theta_f)$

To model the loosening of the boundary conditions, the initial and final Cartesian coordinates and velocities were varied in each direction by a random value within predetermined boundaries δ_{pos} and δ_{vel} respectively. This random variation was included in the Monte-Carlo method and the resulting boundary conditions were saved for each generation. For the Nelder-Mead method, the new boundary conditions corresponding to each best point evaluated was then treated as a constant value and the elevation shaping function's coefficients were further optimised using their respective boundary conditions.

When θ_i and θ_f are set to $1 \cdot 10^{-12}$ and 360 degrees respectively, the results shown in Table 5.23 are found. In this table, the RMSE and maximum difference in elevation angle with respect to the theoretical elevation angle found through Equation 3.12 are shown as well.

Table 5.23: The ΔVs, RMSEs and maximum differences with respect to the theoretical elevation angle when θ_i and θ_f are set to $1 \cdot 10^{-12}$ and 360 degrees.

$[\delta_{pos}$ [km], δ_{vel} [km/s]]	ΔV [km/s]	RMSE [deg]	Max $\Delta\phi$ [deg]
[0, 0]	31.63	7.53	12.95
[10, 0.01]	24.69	4.05	7.32
[100, 0.1]	24.26	4.35	10.39
[1000, 1.0]	21.77	2.91	7.33

In this table, it can first of all be seen that even a relatively small misfit of approximately 2.9 degrees already results in a large ΔV of 21.77 km/s. It is predicted that this effect will increase as the inclination increases.

Furthermore, it can be seen that loosening the boundary conditions greatly reduces the ΔV required, but that the RMSE and maximum difference in elevation angle do not always decrease with the ΔV. The former can also be seen in Figure 5.15, as this graph shows that the trajectory approaches the theoretical case when the boundary conditions are loosened. The theoretical value in this figure is the elevation angle of the Keplerian orbit shown in Table 5.22.

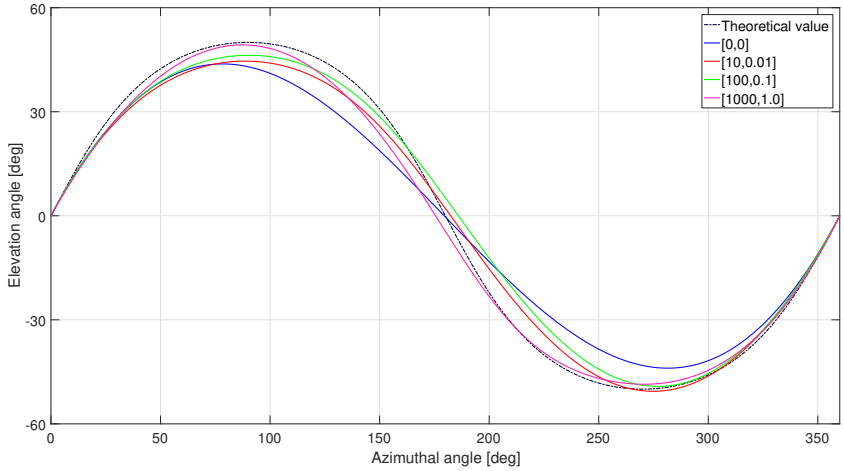


Figure 5.15: The variation of the elevation angle as a function of the azimuthal angle using various limits for the boundary conditions.

When θ_f is changed to 180 degrees, the ΔVs, RMSEs and maximum elevation angle differences shown in Table 5.24 can be obtained.

Table 5.24: The RMSEs and maximum differences with respect to the theoretical elevation angle, as well as the ΔVs found when θ_i and θ_f are set to $1 \cdot 10^{-12}$ and 180 degrees.

$[\delta_{pos}$ [km], δ_{vel} [km/s]]	ΔV [km/s]	RMSE [deg]	Max $\Delta\phi$ [deg]
[0, 0]	2.23	0.41	0.51
[10, 0.01]	4.00	1.36	1.89
[100, 0.1]	1.57	0.06	0.12
[1000, 1.0]	5.60	2.54	3.38

Noteworthy is that all ΔVs, RMSEs and maximum elevation angle differences are significantly smaller than those shown in Table 5.23 and that, using the aforementioned settings, the third row results in the lowest ΔV. Furthermore, it can be seen that the ΔV does not always decrease when the boundary conditions are loosened. As the ΔV for cases with more loose boundary conditions should theoretically be at least as low as the less loose boundary conditions, this can indicate that $1 \cdot 10^5$ Monte-Carlo samples and a sample size of the 50 best points for the Nelder-Mead algorithm may not be enough. Interesting to note is also that for this test case, the RMSE and maximum elevation angle difference appear to have a correlation with the resulting ΔV, unlike the test case shown in Table 5.23. For reference, the variation of these trajectories' elevation angles can be found in Appendix G.

If θ_i and θ_f are set to 90 and 270 degrees respectively, the values shown in Table 5.25 can be found.

Table 5.25: The ΔVs, RMSEs and maximum differences with respect to the theoretical elevation angle when θ_i and θ_f are set to 90 and 270 degrees.

$[\delta_{pos}$ [km], δ_{vel} [km/s]]	ΔV [km/s]	RMSE [deg]	Max $\Delta\phi$ [deg]
[0, 0]	0.77	0.07	0.19
[10, 0.01]	0.74	0.10	0.24
[100, 0.1]	0.79	0.06	0.12
[1000, 1.0]	1.26	1.02	1.83

From this, it can be seen that the elevation shaping function is more capable of modelling this case than the cases shown in Tables 5.23 and 5.24. However, in this case there also seems to be no correlation between the RMSE and ΔV found. It can furthermore be noted that the second set of bounds results in the lowest ΔV in this case and the ΔV found increases when the boundary conditions are set more loose. This reinforces the hypothesis that a larger Monte-Carlo and Nelder-Mead sample size may be needed. For the results shown in Table 5.25, the variation of the elevation angle as a function of the azimuthal angle can be found in Appendix G as well.

In order to obtain more insight in the influence of using more Monte-Carlo and Nelder-Mead samples on the ΔVs found, the test case shown in Table 5.23 was evaluated with an increased number of Monte-Carlo and Nelder-Mead samples. The new settings can be seen in Table 5.26.

Table 5.26: The settings used to evaluate the influence of using more Monte-Carlo and Nelder-Mead samples.

MC Samples	$1{\cdot}10^{6}$
NM Sample Size	200
NM Distance Threshold	$1{\cdot}10^{-15}$
NM ΔV Threshold	$1{\cdot}10^{-10}$
NM Max. Iterations	2000

Using these settings, the following results were found:

Table 5.27: The ΔVs, RMSEs and maximum differences with respect to the theoretical elevation angle when θ_i and θ_f are set to $1 \cdot 10^{-12}$ and 360 degrees and the settings shown in Table 5.26 are used.

[δ_{pos} [km], δ_{vel} [km/s]]	ΔV [km/s]	RMSE [deg]	Max $\Delta\phi$ [deg]
[0, 0]	14.27	5.82	14.27
[10, 0.01]	24.40	3.85	7.06
[100, 0.1]	24.17	3.81	8.79
[1000, 1.0]	20.58	3.62	6.74

While it can be seen in this table that each of the ΔVs is lower than the corresponding ΔV shown in Table 5.23 when the same values for δ_{pos} and δ_{vel} are used, the values are still very large with respect to the theoretical ΔV of zero. Furthermore, by comparing Tables 5.23 and 5.27, it can be seen that increasing the Monte-Carlo and Nelder-Mead sample size has resulted in looser boundary conditions no longer requiring at most the same ΔV as more strict ones. As discussed above, this was also observed for the test cases shown in Tables 5.24 and 5.25. This potentially indicates that an even larger Monte-Carlo and Nelder-Mead sample size is required. As the computation time with the current settings is already extremely long, the usage of a different method to determine the additional coefficients for the elevation shaping function would be recommended. Nonetheless, Table 5.23 shows that the concept of loosening the boundary conditions has the potential to significantly lower the ΔV found.

5.5. EXTERNAL VALIDATION

To externally validate the developed elevation shaping function, two test cases are evaluated. These test cases were used by both Novak and Roegiers in the development of the spherical shaping method, and the results were compared to those of the pseudo-equinoctial shaping method. As such, the results obtained using the new elevation shaping function can be compared with those obtained with other shape-based methods.

It should be noted that Roegiers used a different model for the targets' orbits and that it is not known which model Novak used. Therefore, instead of evaluating a range of departure dates and time-of-flights and propagating the target's orbit, the trajectory corresponding to the optimal departure date and time-of-flight found by Roegiers is used as test case. As the code written by Roegiers is available and its installation has been validated, the initial and final Keplerian elements used by Roegiers can be extracted and used as input for the software that contains the new elevation shaping function.

5.5.1. EARTH - MARS TRANSFER

As a first test case for the external validation, an Earth - Mars transfer is evaluated. For this case, the settings shown in Table 5.28 were used together with the bounds for the Monte-Carlo method shown in Table 5.14.

Table 5.28: The settings used to externally validate the developed shaping function using an Earth - Mars transfer.

MC Samples	$1{\cdot}10^{5}$
NM Sample Size	50
NM Distance Threshold	$1{\cdot}10^{-15}$
NM ΔV Threshold	$1{\cdot}10^{-10}$
NM Max. Iterations	2000
Integration Step Size	$2\pi/100$

A comparison between the various shaping methods and their results is shown in Table 5.29. In this table, "+LQ" indicates that the trajectory was improved using a linear quadratic controller and "+DITAN" indicates that the Direct Interplanetary Trajectory Analysis (DITAN) software was used to optimise the trajectory [1].

Table 5.29: A comparison between the ΔV found using existing software and the spherical shaping method using the new elevation shaping function.

	Sph. by Novak [2]	Sph. by Roegiers [2]	Pseudo-equin. [2]	Sph. by Vroom
Launch Date	Oct. 22, 2021	Oct. 22, 2021	Oct. 22, 2021	Oct. 22, 2021
Arrival Date	June 8, 2024	June 8, 2024	June 8, 2024	June 8, 2024
TOF [days]	960	960	960	960
Revolutions	1	1	1	1
TOF shaping	?	yes	?	yes
ΔV [km/s]	12.57	5.75	7.07	5.75
ΔV (+LQ) [km/s]	6.01	-	5.89	-
ΔV (+DITAN) [km/s]	5.89	-	5.89	-

It can be seen in this table that the ΔV found is similar the one found by Roegiers and that these two implementations find a lower ΔV than the ones found by Novak with the spherical and pseudo-equinoctial shaping methods. One reason for this might be that Novak used a different method to shape the time in order to find the proper value of r_2 and to match the required time-of-flight [2]. In his PhD thesis, a second way to do so is elaborated upon, but it is unclear whether this method was implemented in the original shaping method [1]. For this thesis, the method as described in Section 4.1.9 was used in order to be able to compare the obtained results with those of the spherical shaping method as implemented by Roegiers. Nonetheless, due to the similarity between the ΔVs found, it is concluded that the new elevation shaping function has been verified with respect to this test case.

5.5.2. EARTH - NEPTUNE TRANSFER

The second test case that is evaluated, is an Earth - Neptune transfer. The same settings are used as done for the Earth - Mars transfer, here repeated in Table 5.30 for convenience.

Table 5.30: The settings used to find the Earth - Neptune transfer.

MC Samples	$1{\cdot}10^{5}$
NM Sample Size	50
NM Distance Threshold	$1{\cdot}10^{-15}$
NM ΔV Threshold	$1{\cdot}10^{-10}$
NM Max Iterations	2000
Integration Step Size	$2\pi/100$

As described earlier, the initial and final Keplerian elements are set to those found by Roegiers due to the difference in orbit models used. Furthermore, an initial tangential velocity of 3 km/s relative to Earth was

used in this test case in order to limit the peak thrust of the transfer and because it would also be likely that a mission like this would be injected directly into an Earth escape trajectory by a launcher [1]. By using these input parameters, the results shown in Table 5.31 are found.

Table 5.31: The comparison between the ΔV found for the Earth - Neptune transfer using existing software and the spherical shaping method using the new elevation shaping function.

	Sph. by Novak	Sph. by Roegiers	Sph. by Vroom
Launch Date	?	May 10, 2021	May 10, 2021
Arrival Date	?	April 26, 2058	April 26, 2058
TOF [days]	?	13500	13500
Revolutions	0	0	0
Initial Relative Velocity [km/s]	3	3	3
ΔV [km/s]	14.99	15.48	15.43
ΔV (+DITAN) [km/s]	13.34	-	-

From these results, it can be concluded that a similar ΔV is obtained as the one found by Roegiers, but that the ΔV found by Novak is lower. One potential explanation for this, might be the difference in departure date and time-of-flight. Unfortunately, the optimal time-of-flight and departure date are not mentioned in Novak's PhD thesis [1]. However, as the ΔV found is still similar to the non-optimised value found by Novak and Roegiers, the implementation of the spherical shaping method that uses the new elevation shaping function is considered to have been verified with respect to this second external validation case as well.

6

APPLICATION

Now that the new elevation shaping function has been validated, it can be used to generate transfers for additional problems. In this chapter, two problems will be considered: a transfer from Earth to the dwarf planet Makemake and a transfer from Earth to the comet 2003 EH1. These targets were chosen due to their high inclinations and because neither of these celestial bodies has been visited yet.

To obtain these transfers, the following boundaries were used for the coefficients of the Monte-Carlo method:

Table 6.1: The boundaries used for the coefficients in the Monte-Carlo method.

Coefficient	a_0	a_1	a_2	a_3	a_4	a_5	p_0	p_1	p_2	p_3
Minimum	0.00	-4.00	-4.00	0.00	-4.00	-4.00	-2π	-5.00	-5.00	-2π
Maximum	4.00	4.00	4.00	4.00	4.00	4.00	2π	5.00	5.00	2π

It can be seen in this table that the bounds of the coefficients are significantly larger than those used during the validation. This was done to allow the shaping function to take on a wider variety of shapes. However, an additional method was used in order to guide the Monte-Carlo method. As the exponential function that contains the coefficients a_0 to a_5 replaces a^2 in Equation 3.15, the theoretical maximum value of a^2 is known through the graph shown in Figure 3.27 and the range of azimuthal angles is known as well. Therefore, it can be checked whether the exponential function ever becomes larger than the theoretical maximum value of a^2. If this is the case, too large coefficients were chosen by the Monte-Carlo method for the exponential function and another set of random values for a_0 to a_5 is then generated for this Monte-Carlo sample instead. These attempts at finding appropriate values for a_0 to a_5 do not count towards the maximum number of Monte-Carlo samples.

To find the theoretical value of a^2 as a function of inclination, the values used to create Figure 3.27 were first squared in order to obtain a graph showing how the value of a^2 changes with inclination. A least-squares approximation was then made of the resulting graph, such that values for a^2 could be found for arbitrary inclinations. To increase the accuracy of the approximation, the graph was split into a number of parts and a separate least-squares approximation was made for each section. The function that was used to approximate each section can be seen in Equation 6.1.

$$a^2 = \left| c_0 e^{c_1 \cdot i} + c_2 e^{c_3 \cdot i} \right| \tag{6.1}$$

in which i is the inclination in radians.

The bounds for each section of the graph, their respective values for coefficients c_0 to c_3 and root-mean-square-errors (RMSE) can be seen in Table 6.2.

Table 6.2: The inclination bounds, values of coefficients and the RMSEs of the least-squares approximations.

Inclination Bounds [deg]	c_0	c_1	c_2	c_3	RMSE [-]
[0 70)	$2.23 \cdot 10^{-2}$	4.12	$-2.75 \cdot 10^{-2}$	-4.31	$2.69 \cdot 10^{-2}$
[70 85]	$1.48 \cdot 10^{-3}$	6.37	$6.24 \cdot 10^{-18}$	28.95	$1.22 \cdot 10^{-1}$
(85 87]	$1.63 \cdot 10^{-9}$	16.00	$3.24 \cdot 10^{-23}$	36.84	1.06
(87 88]	$2.81 \cdot 10^{-16}$	25.28	$5.46 \cdot 10^{-29}$	45.82	2.33
(88,90]	$1.00 \cdot 10^{4}$	0	0	0	-

It can be seen in this table that when the inclination is larger than 88 degrees, the theoretical value of a^2 is set to be a constant value. The reason for this, is that at these inclinations, a different value for a^2 corresponds to a very small difference in the inclination (see Figure 3.27) and that it becomes difficult to obtain a least-squares approximation without splitting the graph into a large number of additional sections. By setting the theoretical value to a sufficiently large constant value, the aforementioned check for the chosen values of the coefficients a_0 to a_5 can still be made. As the approximation is not fully accurate, the resulting theoretical value of a^2 should not be treated as the maximum value directly. Instead, the theoretical value of a^2 was multiplied with an additional factor f_a in order to obtain an upper limit for a^2 and to circumvent the inaccuracy of the least-squares approximation.

The search space created by the variation of the departure time, number of revolutions and time-of-flight was first evaluated using a grid search. Due to the computation time needed through the usage of the Monte-Carlo and Nelder-Mead methods, a small Monte-Carlo and Nelder-Mead sample size was used to generate trajectories for each point in the grid. The Nelder-Mead algorithm was furthermore only used if a ΔV lower than a certain threshold was found by the Monte-Carlo method. In this way, the computation time of certain combinations of departure times, number of orbits and time-of-flights that were not promising was limited. The ΔV required for the software to use the Nelder-Mead algorithm was set to 50 km/s for both the transfer to Makemake and the transfer to 2003 EH1. Additional settings used for the initial grid search can be found in Table 6.3.

Table 6.3: The settings used for the initial grid search.

MC Samples	2000
NM Sample Size	5
NM Distance Threshold	$1 \cdot 10^{-15}$
NM ΔV Threshold	$1 \cdot 10^{-10}$
NM Max. Iterations	100
Integration Step Size	$2\pi/20$
Factor f_a	1.5

Once the optimal point in the grid was found, the trajectory using the departure date, number of revolutions and time-of-flight of this point was further optimised using the settings shown in Table 6.4.

Table 6.4: The settings used to further optimise the best solution found by the grid search.

MC Samples	$1 \cdot 10^{5}$
NM Sample Size	50
NM Distance Threshold	$1 \cdot 10^{-15}$
NM ΔV Threshold	$1 \cdot 10^{-10}$
NM Max. Iterations	2000
Integration Step Size	$2\pi/100$
Factor f_a	1.5

6.1. EARTH - MAKEMAKE TRANSFER

As mentioned previously, one of the two transfers that the spherical shaping method with the new elevation shaping function was applied to, was a transfer from Earth to the dwarf planet Makemake. The Keplerian elements of Earth and Makemake on January 1, 2020 (7304.5 MJD2000) were obtained from the HORIZONS web-interface developed by the Jet Propulsion Laboratory (JPL) and can be seen in Table 6.5 [21].

Table 6.5: The Keplerian elements of Earth and Makemake at 7304.5 MJD2000 [21].

	a [AU]	e [-]	i [deg]	ω [deg]	Ω [deg]	θ [deg]
Departure Orbit	1.00	$1.67 \cdot 10^{-2}$	$-3.22 \cdot 10^{-3}$	-5.16	$1.08 \cdot 10^{2}$	-3.25
Target Orbit	45.48	$1.60 \cdot 10^{-1}$	28.90	79.63	$2.95 \cdot 10^{2}$	$1.69 \cdot 10^{2}$

To propagate these Keplerian elements for the different launch dates used in the grid search, use was made of the equations presented in Appendix C and *Fundamentals of Astrodynamics* by Wakker [4].

Firstly, the eccentric anomaly at a reference time t_0 is found from the true anomaly using Equation 6.2. In this equation, E is the eccentric anomaly, e is the eccentricity and θ is the true anomaly [4].

$$tan\left(\frac{E}{2}\right) = \sqrt{\frac{1-e}{1+e}} tan\left(\frac{\theta}{2}\right) \tag{6.2}$$

This eccentric anomaly can then be converted to the mean anomaly with Equation 6.3, where M is the mean anomaly [4].

$$M = E - e \cdot sin(E) \tag{6.3}$$

Using the mean anomaly at reference time t_0, the propagated mean anomaly M_{prop} can be obtained from Equation 6.4.

$$M_{prop} = M_0 + n \cdot (t - t_0) \tag{6.4}$$

where n is the mean motion and can be found with Equation 6.5. In this equation μ is the standard gravitational parameter.

$$n = \sqrt{\frac{\mu}{a^3}} \tag{6.5}$$

This new mean anomaly can then be converted back to find the propagated eccentric anomaly. However, as Equation 6.3 does not have a closed-form solution, a Newton-Raphson method is used to find the value of the eccentric anomaly that corresponds to a certain mean anomaly. By iteratively solving Equation 6.6 until the difference between E_{k+1} and E_k is below a certain value, the eccentric anomaly can be found [4]. For this iterative process, a threshold of $1 \cdot 10^{-20}$ radians was used for the difference between E_{k+1} and E_k, and the maximum number of iterations was set to $1 \cdot 10^5$ in order to ensure the accuracy of the eccentric anomaly found.

$$E_{k+1} = E_k - \frac{E_k - e \cdot sin(E_k) - M_{prop}}{1 - e \cdot cos(E_k)} \tag{6.6}$$

Depending on the eccentricity, the following initial value for E_k was used:

$$E = \begin{cases} M & \text{e<0.8} \\ \pi & \text{e>=0.8} \end{cases} \tag{6.7}$$

Once the propagated eccentric anomaly has been found, the propagated true anomaly can be determined using Equation 6.8. This can then be used together with the other Keplerian elements to find the propagated initial Cartesian elements. While in practise the other Keplerian elements also change over time, they are considered to be constant in this case.

$$tan\left(\frac{\theta}{2}\right) = \sqrt{\frac{1+e}{1-e}} tan\left(\frac{E}{2}\right) \tag{6.8}$$

For the grid search, the minima, maxima and step sizes shown in Table 6.6 were used for the launch date, time-of-flight and the number of revolutions. The launch window in this case was set such that it includes slightly more than two synodic periods. This synodic period can, with the use of the equation shown in Section 2.2.6, be found to be 366.67 days [22].

Table 6.6: The minima, maxima and step sizes for the launch date, time-of-flight and the number of revolutions.

	Minimum	Step size	Maximum
Launch Date [MJD2000]	7304.5	40	8184.5
Time-of-Flight [days]	11000	2000	40000
Number of Revolutions [-]	0	1	2

It assumed in this test case that the mission will be injected into an Earth escape trajectory using a launcher, as was also done for the Earth - Neptune transfer. The initial tangential velocity relative to Earth is therefore set to 3 km/s (C_3 = 9 km^2/s^2). Using this initial relative velocity together with the inputs shown in Table 6.3 and the grid shown in Table 6.6, the following mission is found:

Table 6.7: The best trajectory found by the grid search for the Earth - Makemake transfer.

Launch Date	Oct. 2, 2021
Arrival Date	Feb 8, 2112
TOF [days]	33000
Revolutions	0
Initial Relative Velocity [km/s]	3
ΔV [km/s]	16.58
Peak thrust acceleration [m/s^2]	$2.17 \cdot 10^{-3}$

If this mission is further optimised with the parameters shown in Table 6.4, the following results are obtained:

Table 6.8: The orbit characteristics of the Earth - Makemake transfer after further optimisation.

Launch Date	Oct. 2, 2021
Arrival Date	Feb 8, 2112
TOF [days]	33000
Revolutions	0
Initial Relative Velocity [km/s]	3
ΔV [km/s]	16.53
Peak thrust acceleration [m/s^2]	$2.19 \cdot 10^{-3}$

By comparing the two tables shown above, it can be seen that the ΔV is slightly lower after optimisation. Furthermore, the ΔV of 16.53 km/s is also lower than the one obtained by evaluating the grid using the currently existing spherical shaping method, this being 23.96 km/s. While the ΔV could also be compared to that of a high-thrust mission, this is deemed to not provide much additional insight, as it is known that the ΔV of the high-thrust case will be smaller. The reason for this, is that a low-thrust spacecraft continuously thrusts against the gravitational pull, whereas a high-thrust spacecraft does not due to its short thrusting time.

The control accelerations corresponding to the mission shown in Table 6.8 can be seen in Figure 6.1.

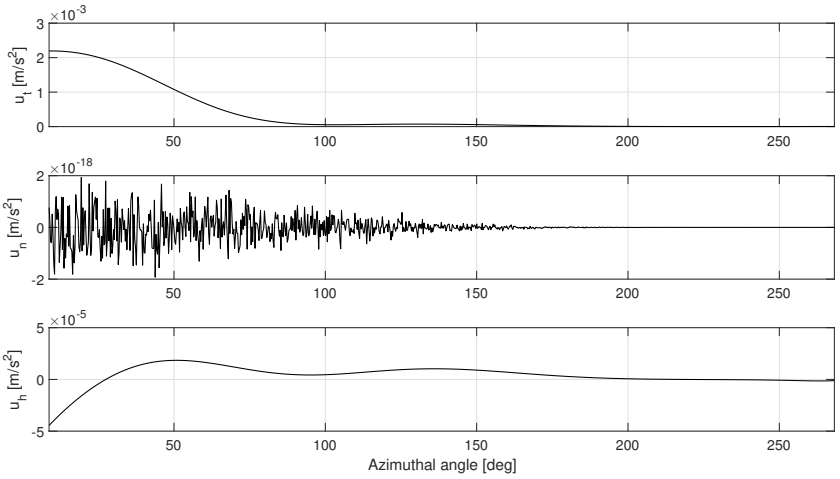


Figure 6.1: The control accelerations of the Earth - Makemake transfer.

Furthermore, the trajectory itself has been plotted as well and can be seen in Figure 6.2.

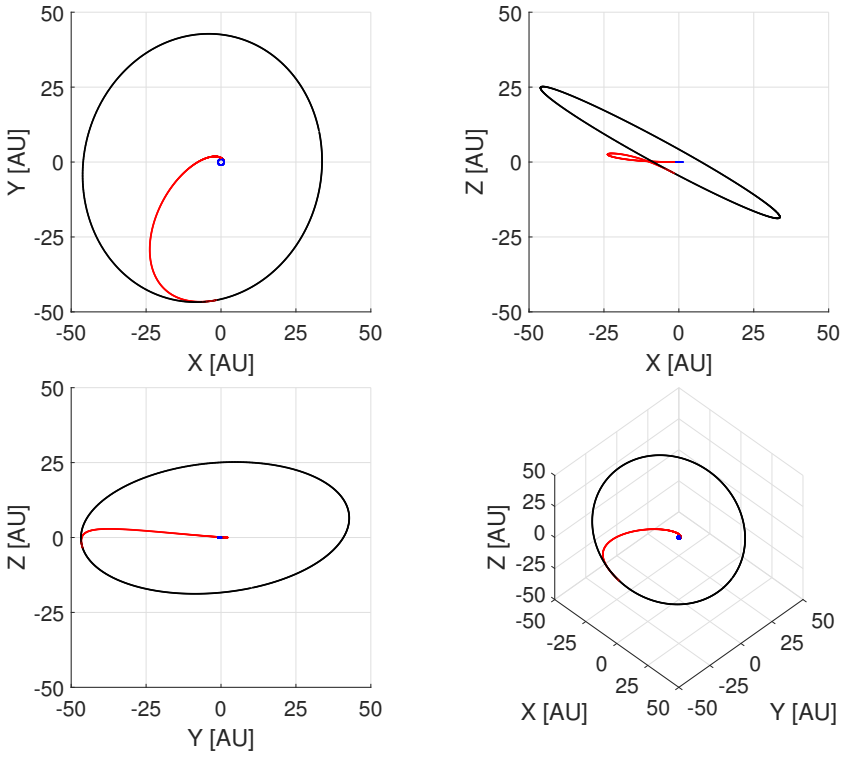


Figure 6.2: The Earth - Makemake transfer in the Cartesian reference frame.

From these figures, it can first of all be seen that the resulting trajectory is smooth and the control accelerations are very modest. Furthermore, it appears that the high inclination does not cause any odd behaviour. While the trajectory is likely not the most optimal one due to the chosen grid, this result does show that the new elevation shaping function is capable of producing trajectories to high inclination targets and could thus be used in the future to for instance generate pork-chop plots for similar problems.

6.2. EARTH - 2003 EH1 TRANSFER

Secondly, the spherical shaping method was applied together with the new elevation shaping function to a transfer from Earth to the comet 2003 EH1, the latter of which has a very large inclination of approximately 70 degrees. Using JPL's HORIZONS web-interface, the Keplerian elements shown in Table 6.9 were found for January 1, 2020 (7304.5 MJD2000) [21]. These Keplerian elements were propagated in the same manner as was done for the Earth - Makemake transfer.

Table 6.9: The Keplerian elements of Earth and 2003 EH1 at 7304.5 MJD2000 [21].

	a [AU]	e [-]	i [deg]	Ω [deg]	ω [deg]	θ [deg]
Departure Orbit	1.00	$1.67 \cdot 10^{-2}$	$-3.22 \cdot 10^{-3}$	-5.16	$1.08 \cdot 10^{2}$	-3.25
Target Orbit	3.12	$6.19 \cdot 10^{-1}$	70.84	$2.83 \cdot 10^{2}$	$1.72 \cdot 10^{2}$	77.82

The minima, maxima and step sizes shown in Table 6.10 were used for the launch date, time-of-flight and the number of revolutions. The launch window in this case is also chosen such that it encompasses slightly more than two synodic periods. As the period of 2003 EH1 is 5.52 years, Equation 2.5 can be used to find that the comet's synodic period is approximately 446.33 days [22]. By using a launch date variation of 900 days, the aforementioned two synodic periods can be included.

Table 6.10: The minima, maxima and step sizes for the launch date, time-of-flight and the number of revolutions.

	Minimum	Step size	Maximum
Launch Date [MJD2000]	7304.5	30	8204.5
Time-of-Flight [days]	500	250	4500
Number of Revolutions [-]	0	1	2

Using these inputs and the same initial relative velocity as used for the Earth - Makemake transfer, the following mission is found:

Table 6.11: The characteristics of the best trajectory found by the grid search.

Launch Date	Sep. 22, 2021
Arrival Date	Dec. 9, 2029
TOF [days]	3000
Revolutions	0
Initial Relative Velocity [km/s]	3
ΔV [km/s]	24.10
Peak thrust acceleration [m/s^2]	$1.49 \cdot 10^{-3}$

By optimising this mission using the inputs shown in Table 6.4, the mission shown in Table 6.12 is the result.

Table 6.12: The trajectory characteristics after further optimisation.

Launch Date	Sep. 22, 2021
Arrival Date	Dec. 9, 2029
TOF [days]	3000
Revolutions	0
Initial Relative Velocity [km/s]	3
ΔV [km/s]	22.97
Peak thrust acceleration [m/s^2]	$8.36 \cdot 10^{-4}$

It can be seen from these two tables that in this case, the optimisation slightly lowers the ΔV and that the resulting peak thrust acceleration is again very modest. Highly interesting is the fact that this ΔV is also much lower than the one found by evaluating the grid with the currently existing spherical shaping method. The ΔV found by using the previous elevation shaping function is 50.77 km/s. This shows that the new elevation

shaping function is capable of producing transfers to targets with high inclinations that have a much more attainable ΔV.

If the transfer, departure and target orbits are plotted, Figure 6.3 can be obtained.

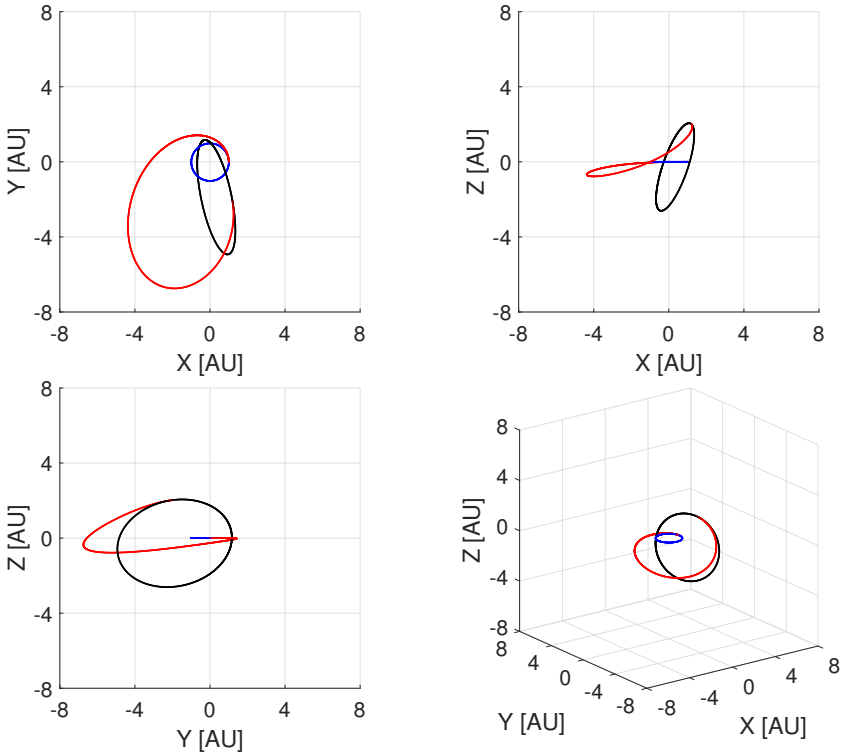


Figure 6.3: The Earth - 2003 EH1 transfer in the Cartesian reference frame.

By inspecting the trajectory, it can first of all be noticed that this trajectory is also smooth and as expected. It furthermore can be seen that the majority of the inclination change happens at a large distance from the Sun. As this is also theoretically the most efficient way of changing the inclination, it is excellent to notice that the algorithm found a transfer that performs its inclination change at this point to be the cheapest. This change of inclination can also be seen by observing the control accelerations shown in Figure 6.4. It is shown here that a large portion of the out-of-plane thrust acceleration occurs near the end of the trajectory.

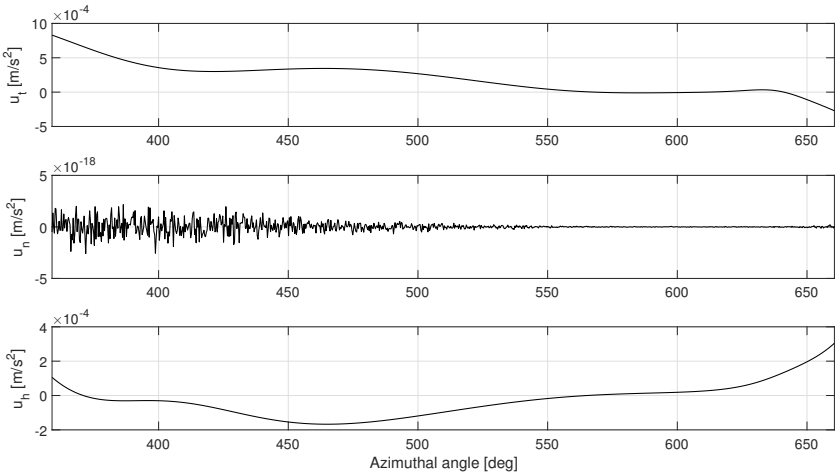


Figure 6.4: The control accelerations of the Earth - 2003 EH1 transfer.

6.3. FLEXIBLE BOUNDARY CONDITIONS

In an attempt to further improve the solutions found, the concept of using flexible boundary conditions that was introduced in Section 5.4.5 is applied to the trajectories found in this chapter.

6.3.1. EARTH - MAKEMAKE TRANSFER

When the methodology described in Section 5.4.5 is applied to the Earth - Makemake transfer shown in Table 6.8, the results shown in Table 6.13 can be obtained.

Table 6.13: The ΔVs and peak thrust accelerations found when the Earth - Makemake transfer is evaluated using various limits for the variation of the position and velocity boundary conditions.

$[\delta_{pos}$ [km], δ_{vel} [km/s]]	ΔV [km/s]	Peak thrust acceleration [m/s^2]
[0, 0]	16.53	$2.19 \cdot 10^{-3}$
[10, 0.01]	16.53	$2.20 \cdot 10^{-3}$
[100, 0.1]	16.46	$2.11 \cdot 10^{-3}$
[1000, 1.0]	14.54	$1.49 \cdot 10^{-3}$

From this table, it can be concluded that the ΔV for the Earth - Makemake transfer decreases as the boundary conditions are loosened. It was found that the cause of this is the change in velocity boundary conditions. For example, when the initial velocity of the spacecraft is higher due to the flexible boundary conditions, less ΔV is required to match the final conditions. As such, it can be concluded that using flexible boundary conditions does not improve the ΔV beyond what is caused by the change in velocity boundary conditions. Thus, the use of flexible boundary conditions does not improve the result found by the new elevation shaping function in this case. When the trajectories corresponding to the results shown in Table 6.13 are plotted, Figure 6.5 can be found.

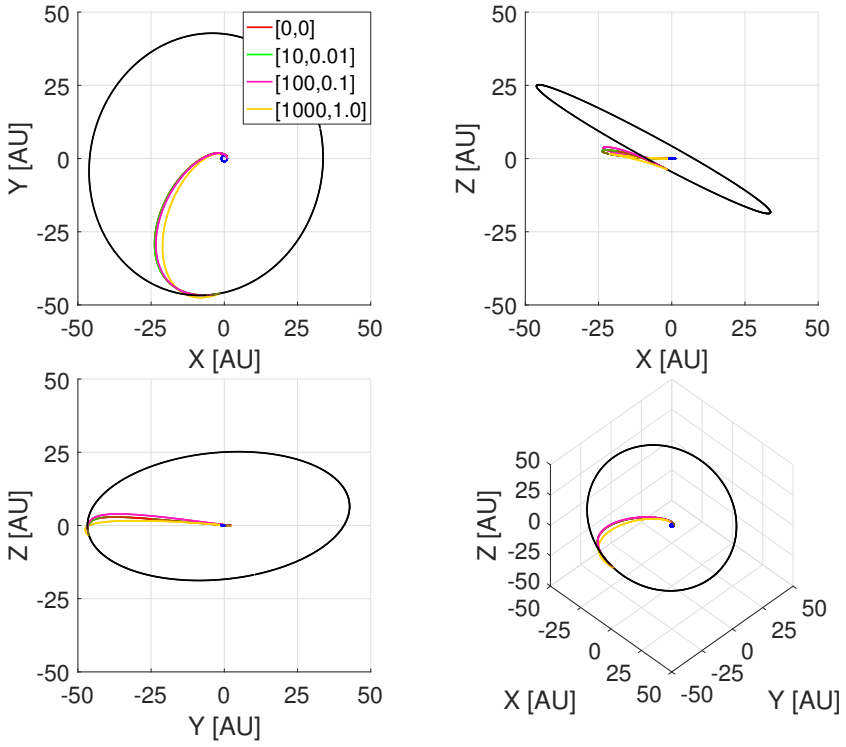


Figure 6.5: Comparison between the Earth - Makemake transfers using various restrictions on the boundary conditions.

It can be seen from this figure that these trajectories are all smooth as expected and that they closely coincide. This indicates that the new elevation shaping function is also capable of producing viable trajectories for this case when the boundary conditions are loosened.

6.3.2. EARTH - 2003 EH1 TRANSFER

The same methodology described previously can also be applied to the Earth - 2003 EH1 transfer shown in Table 6.12. By varying the boundary conditions, the results shown in Table 6.14 can be obtained.

Table 6.14: The ΔVs and peak thrust accelerations found when the Earth - 2003 EH1 transfer is evaluated using various limits for the variation of the position and velocity boundary conditions.

[δ_{pos} [km], δ_{vel} [km/s]]	ΔV [km/s]	Peak thrust acceleration [m/s^2]
[0, 0]	22.97	$8.36 \cdot 10^{-4}$
[10, 0.01]	22.95	$7.85 \cdot 10^{-4}$
[100, 0.1]	22.94	$7.19 \cdot 10^{-4}$
[1000, 1.0]	21.05	$5.35 \cdot 10^{-4}$

From these results, it can be concluded that the ΔV of the Earth - 2003 EH1 transfer also decreases as the boundary conditions are set to be more flexible. It was found that in this case, it also caused by the change in velocity boundaries, as was the case for the Earth - Makemake transfer. As such, loosening the boundary conditions does not seem to improve the result found for this case either. Nonetheless, when the four trajectories are plotted, it can be seen that in this case they are also all smooth and highly similar. As mentioned previously, this shows that loosening the boundary conditions still allows the new elevation shaping function to generate trajectories to high inclination targets.

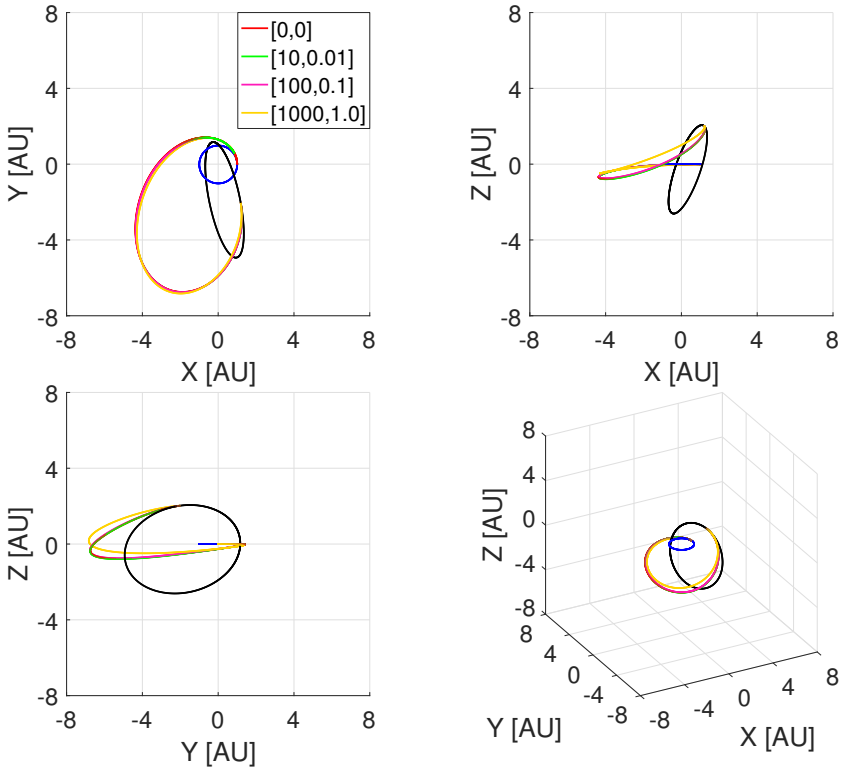


Figure 6.6: Comparison between the Earth - 2003 EH1 transfers using various restrictions on the boundary conditions.

6.4. SOLUTION RELIABILITY

To evaluate the reliability of the solutions found, the optimisation of the mission shown in Table 6.7 is performed an additional five times with different seeds for the pseudorandom number generator used by the Monte-Carlo method. By comparing the ΔVs found, the reliability of the solutions can be evaluated. The ΔVs obtained for the five additional runs can be see in Table 6.15.

Table 6.15: The ΔVs of the five additional optimisation runs.

Run	#1	#2	#3	#4	#5
ΔV [km/s]	16.528	16.528	16.524	16.528	16.527

By comparing the ΔVs shown in Table 6.15 with each other and the ΔV of 16.528 km/s found previously, it can be seen that the maximum variation in ΔV is approximately 4 m/s. This small variation in ΔV can also be observed by comparing the control accelerations shown in Figures 6.7 and 6.1.

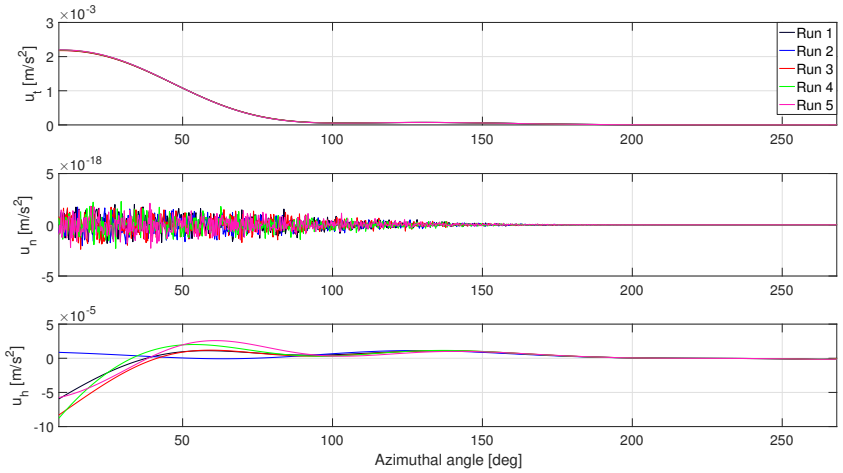


Figure 6.7: The control accelerations of the five additional optimisation runs.

It can be seen here that tangential control accelerations are equal and that the maximum difference between the out-of-plane control accelerations is less than $8{\cdot}10^{-5}$ m/s^2. This similarity in control accelerations also causes the five corresponding trajectories to be highly similar, as can be seen in Figure 6.8.

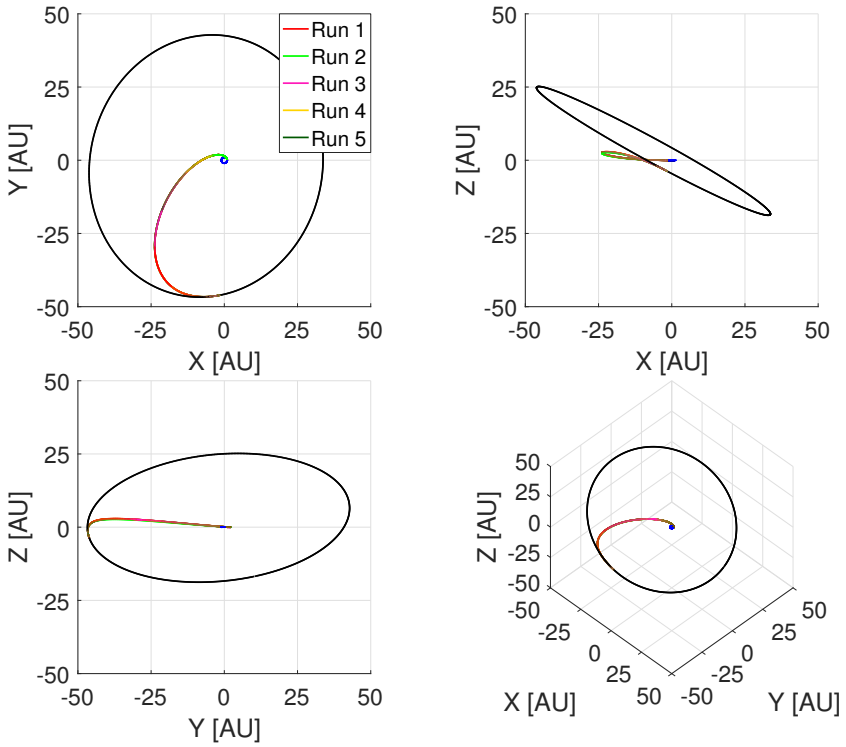


Figure 6.8: The trajectories of the five additional optimisation runs.

The small differences in the ΔVs, control accelerations and the trajectories show that the solution is reliable when the best point found by the grid search is used. However, it should be noted here that Table 6.15 shows that a lower ΔV than the one shown in Table 6.8 was found in these additional runs and that the values of the shaping function's coefficients were found to be significantly different for each run. These two observations support the hypothesis mentioned in Chapter 5 that the optimisation algorithm has some difficulty finding the global minimum. The grid search itself should therefore also be run multiple times and the results should be compared in order to obtain a full analysis of the solution reliability. As this analysis could not be done during this thesis, it is highly recommended that future research evaluates this optimisation procedure further. For reference, the values of the shaping function's coefficients are shown in Appendix H.

7

CONCLUSIONS AND RECOMMENDATIONS

In this chapter, the numerous conclusions drawn throughout the development process of a more accurate elevation shaping function for the spherical shaping method are summarised. Furthermore, a number of recommendations for future research regarding this function are giving in this chapter as well.

7.1. CONCLUSIONS

As a large portion of this thesis can effectively be broken down into four phases, the conclusions drawn have been divided into these phases as well. These phases consist of the development, implementation, validation and application of the new elevation shaping function.

7.1.1. DEVELOPMENT

During the development, the ability of three potential methods to obtain the new elevation shaping function was evaluated with the use of a number of test cases. These methods included the usage of spherical triangles, Fourier series and an alternative function that was found during the development, the first of which was found to have least potential of resulting in an elevation shaping function.

It was concluded that in order to be able to accurately follow perturbed orbits, the coefficients of the Fourier series and the alternative function should have the capacity to change with the azimuthal angle. A number of candidate functions capable of describing these changes were found by splitting each perturbed orbit into numerous unperturbed orbits and by using the MATLAB curve fitting application to generate graphs showing the variation of the function's coefficients as a function of the azimuthal angle for each case. Using this methodology, the Fourier series could be discarded due to the larger number of coefficients needed and the significantly higher root-mean-square error (RMSE) at high inclinations.

The various combinations of the aforementioned candidate functions were then evaluated for their ability to model the predefined test cases. After substitution of the best combination into the alternative function, the new elevation shaping function was found.

7.1.2. IMPLEMENTATION

To implement the new elevation shaping function, use was made of the spherical shaping method's currently existing methodology. By replacing the previous elevation shaping function with the new function and keeping the same process of finding trajectories, the new elevation shaping function was incorporated in the spherical shaping method.

As four boundary conditions were available for the elevation shaping function but 14 coefficients needed to be solved, it was concluded that the coefficients b_0, b_1, b_2 and c could be found using these boundary conditions and the remaining coefficients could be found through optimisation. This optimisation process consisted of using the Monte-Carlo method to perform an initial global search and a further optimisation of a predefined number of the best solutions found with the Nelder-Mead method.

7.1.3. VALIDATION

By using a number of internal and external validation cases, it was found that the new elevation shaping function is much more capable of modelling transfers at high inclinations than the previous one and that the curvature does not become negative at inclinations where it would if this previous elevation shaping function was used. Despite the new elevation shaping function being more capable of modelling these transfers, a significant error was still found when transfers with high inclinations were simulated. A similar conclusion was drawn for cases with a high inclination where a Keplerian orbit or a change in RAAN was simulated. With regards to the former, it was noticed that when a Keplerian orbit with an inclination of 50 degrees was simulated, even a small RMSE with respect to the theoretical case resulted in a large ΔV. It was predicted that this effect will increase as the inclination increases.

By comparing the RMSE predicted during the development with the RMSE observed during the validation, it was noticed that the RMSE was significantly higher than initially predicted using the MATLAB curve fitting toolbox. As the aforementioned toolbox does not strictly enforce the boundary conditions, whereas the spherical shaping method does, it was investigated whether using more flexible position and velocity boundary conditions could improve the solution found.

It was determined that using these flexible boundary conditions could indeed significantly reduce the ΔV found when the aforementioned Keplerian orbit at an inclination of 50 degrees was approximated. Furthermore, it was found that the new elevation shaping function is much more capable of modelling half a Keplerian orbit than a full revolution. When half a Keplerian orbit was approximated, it was however seen that loosening the boundary conditions did not always improve the ΔV found. It was concluded that this was caused by the optimisation procedure having difficulty finding the global minimum, as the ΔV found when the boundary conditions are set to be more flexible should not be higher than when the boundary conditions are more strict. Lastly, it was also noticed that increasing the number of Monte-Carlo samples and the number of points optimised with the Nelder-Mead algorithm greatly reduced the ΔV as well. However, the resulting computation time became impractical when this was done and this increase was therefore not considered for further usage.

7.1.4. APPLICATION

It was found that the spherical shaping method with the new elevation shaping function was capable of producing smooth trajectories to the dwarf planet Makemake and the comet 2003 EH1, despite the latter having an inclination of approximately 70 degrees. It was noticed that the ΔVs found were also significantly smaller than those of trajectories obtained using the previous elevation shaping function.

In the Earth - 2003 EH1 transfer, it was furthermore seen that a large portion of the inclination change was performed at a large distance from the Sun. As this is the most efficient way of changing the inclination, it can be concluded that the new elevation shaping function is therefore highly promising.

The concept of using flexible boundary conditions was furthermore applied to the Earth - Makemake and Earth - 2003 EH1 transfers. However, no improvement of the ΔV beyond what was caused by the change in velocity boundary conditions was seen here. Nonetheless, as the trajectories obtained were all also smooth and highly similar, the new elevation shaping function was found to still be capable of producing viable trajectories for these cases when flexible boundary conditions were used.

Lastly, the reliability of the solutions was evaluated by performing the secondary optimisation of the Earth - Makemake transfer found by the grid search five additional times with different seeds for the pseudorandom number generator used by the Monte-Carlo method. By comparing the resulting ΔVs, it was found that the maximum variation in ΔV was approximately 4 m/s and that the control accelerations were highly similar. While these small differences show the solution's reliability, a solution with a lower ΔV than the one of the initial solution was also found in these five additional runs. Furthermore, it was also noticed that the determined values of the shaping function's coefficients were significantly different for each run. The latter two observations support the conclusion that the optimisation method has difficulty finding the global minimum.

7.2. RECOMMENDATIONS

Based on the development process and the results found, a number of recommendations can be made for the further development of the spherical shaping method and the new elevation shaping function.

7.2.1. OPTIMISATION TECHNIQUE

As mentioned in this thesis, the optimisation technique used was found to be suboptimal. This could among others be seen in the fact that a higher ΔV was sometimes found when the boundary conditions were loosened, whereas a lower or equal ΔV should have been the result. As such, it is highly recommended that future research looks more into the optimisation process.

For example, one could look at replacing the Monte-Carlo method. Due to the dimension of the problem, a large number of function evaluations is required to obtain a good distribution over the search space. A good alternative could for instance be the use of Sobol sampling. Where the Monte-Carlo method uses random generation of coefficient values, Sobol sampling generates these values quasi-randomly. This is done in a manner that ensures coverage throughout the search space. Another option would be to further increase the number of Monte-Carlo samples and the number of points optimised using the Nelder-Mead algorithm, as this was found to significantly reduce the ΔV during the thesis. However, as this would increase the computation time drastically, it may be necessary to use parallel computing to prevent the computation time from increasing too much.

Another option would be to reduce the number of Monte-Carlo samples needed through assuming certain values for coefficients in particular cases. One such case could be when the inclination stays constant. As a^2 in the original function is dependent on the inclination, it should stay constant when the inclination does not change. Therefore, the exponential part of the elevation shaping function which replaces a^2 should stay constant in such a case as well, meaning that the coefficients a_1 to a_5 could theoretically be set to zero. Similarly, this could also be done for the coefficients p_0 and p_1 that simulate the change in right ascension of the ascending node. However, as restricting the coefficient values could also exclude more optimal trajectories in certain cases, care should be taken when assuming certain values for coefficients.

Furthermore, the points found by the Monte-Carlo method are currently sorted based on their ΔV and the N best points are optimised using the Nelder-Mead method. A potential problem with this is that multiple points can lay close to each other and that the Nelder-Mead optimisation of these points therefore finds the same local minimum. To prevent this, an additional check could be implemented that requires the points to be a certain distance away from each other. If the Euclidean distance to the next point to be optimised using the Nelder-Mead method is smaller than this amount, a different point that was found by the Monte-Carlo method or Sobol sampling could be chosen instead.

7.2.2. LOOSENING OF THE BOUNDARY CONDITIONS

Lastly, future research could look more into the loosening of the boundary conditions. It was found that by doing so, the ΔV could be improved beyond the decrease in ΔV needed to meet the lower velocity boundary conditions. As the RMSE of some of the validation cases is furthermore still larger than was predicted during the development of the shaping function and loosening the boundary conditions could resolve this, further research into this is highly recommended.

A

COORDINATE SYSTEM TRANSFORMATIONS

In the application of the spherical shaping method, various coordinate system transformations are used. This appendix elaborates on these transformations.

A.1. ORBITAL ELEMENTS TO CARTESIAN COORDINATES

To transform a spacecraft's position and velocity known in terms of orbital elements into Cartesian coordinates, the set of equations shown below can be used. For the derivation of these equations, the reader is referred to the book *Fundamentals of Astrodynamics* by Wakker [4].

The relation between the spacecraft's position and velocity in rectangular coordinates and orbital elements is shown below as Equation A.1 [4].

$$\begin{aligned} x &= l_1\xi + l_2\eta \qquad \dot{x} = \frac{\mu}{H}\left[-l_1 s\theta + l_2(e + c(\theta))\right] \\ y &= m_1\xi + m_2\eta \qquad \dot{y} = \frac{\mu}{H}\left[-m_1 s(\theta) + m_2(e + c(\theta))\right] \\ z &= n_1\xi + n_2\eta \qquad \dot{z} = \frac{\mu}{H}\left[-n_1 s(\theta) + n_2(e + c(\theta))\right] \end{aligned} \tag{A.1}$$

where:

$$\begin{aligned} l_1 &= c(\omega)c(\Omega) - s(\omega)s(\Omega)cos(i) \qquad l_2 = -s(\omega)c(\Omega) - c(\omega)s(\Omega)c(i) \\ m_1 &= c(\omega)s(\Omega) + s(\omega)c(\Omega)c(i) \qquad m_2 = -s(\omega)s(\Omega) + c(\omega)c(\Omega)c(i) \\ n_1 &= s(\omega)s(i) \qquad n_2 = c(\Omega)s(i) \end{aligned} \tag{A.2}$$

and:

$$\xi = rcos(\theta) \qquad \eta = rsin(\theta) \qquad H = \sqrt{\mu a(1-e^2)} \tag{A.3}$$

A.2. CARTESIAN COORDINATES TO SPHERICAL COORDINATES

To obtain the spherical coordinates from the Cartesian coordinates, Equation A.4 can be used [2].

$$\begin{aligned} r &= \sqrt{x^2 + y^2 + z^2} \\ \theta &= atan\left(\frac{y}{x}\right) \\ \phi &= asin\left(\frac{z}{\sqrt{x^2 + y^2 + z^2}}\right) \end{aligned} \tag{A.4}$$

Furthermore, the velocities in the spherical coordinate system can be found with Equation A.5 [2].

$$\begin{pmatrix} v_r \\ v_\theta \\ v_\phi \end{pmatrix} = \begin{bmatrix} cos(\phi)cos(\theta) & cos(\phi)sin(\theta) & sin(\phi) \\ -sin(\theta) & cos(\theta) & 0 \\ -sin(\phi)cos(\theta) & -sin(\phi)sin(\theta) & cos(\phi) \end{bmatrix} \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} \tag{A.5}$$

B

REFERENCE FRAMES

In this thesis, a number of reference frames are used. This appendix will therefore discuss a multitude of inertial and rotation reference frames and the corresponding transformation matrices.

B.1. INERTIAL REFERENCE FRAMES

From the laws discussed in Section 2.2.1, it can be noted that the first law defines an inertial reference frame as a frame in which a particle is in a state of rest or uniform motion if no force is acting upon it. Furthermore, it can be seen that if an inertial reference frame is defined, the second law describes a relation between the time rate of change of linear momentum and the resultant force acting upon the particle. An inertial reference frame should therefore be defined to be able to use Newton's second law of motion.

B.1.1. HELIOCENTRIC ARIES ECLIPTIC (XYZ)

For interplanetary trajectory planning, mainly the heliocentric Aries ecliptic (HAE) system is used. This inertial reference frame has its XY-plane lying in the ecliptic plane, Z-axis towards the ecliptic north pole, X-axis pointing towards the vernal equinox Υ, Y-axis complementing the system and the Sun as its origin, as can be seen in Figure B.1 [23]. In this definition, the ecliptic is the plane of the Earth's orbit around the Sun and the equinox is the intersection of this ecliptic with the Earth's equatorial plane [24].

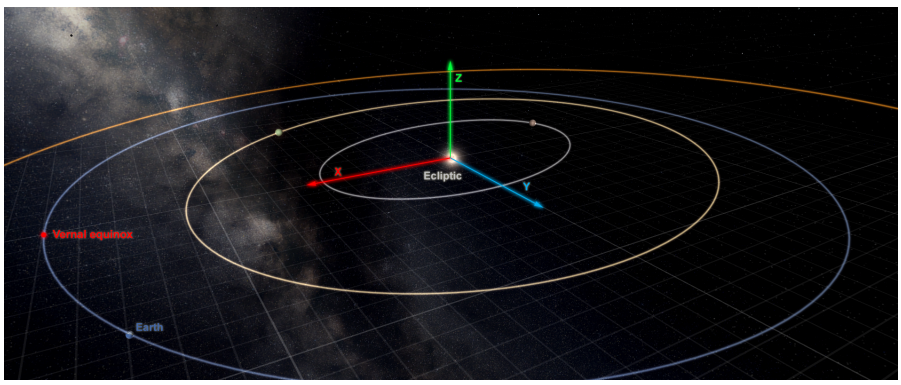


Figure B.1: The axes of the heliocentric Aries ecliptic (HAE) reference frame [23].

B.1.2. PERIFOCAL (PQW)

A second inertial system that can be used, is the perifocal reference frame. This frame consists of a P-axis pointing towards the perihelion of the orbit of the object, an in-orbit Q-axis which is perpendicular to the P-axis and a W-axis that completes the system. As such, the PQ plane coincides with the orbital plane and the system's origin is the Sun [25]. This can also be seen in Figure B.2 [23].

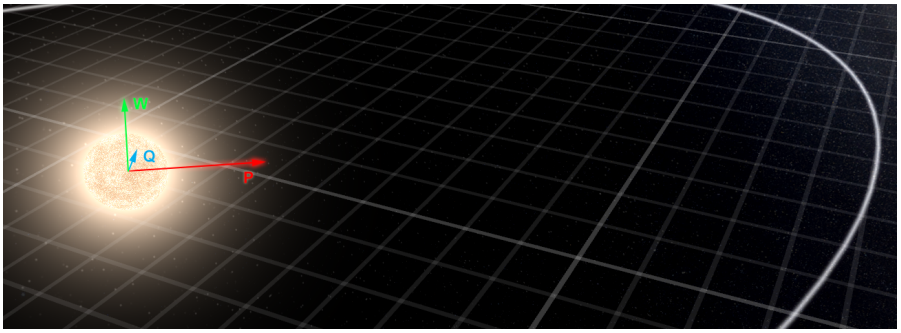


Figure B.2: The perifocal (PQW) reference frame and its axes [23].

B.2. ROTATING REFERENCE FRAMES

Rotating reference frames, as the name suggests, rotate with respect to an inertial reference frame and are thus non-inertial coordinate systems. This subsection will discuss the three most often used rotating reference frames.

B.2.1. RADIAL-ORTHORADIAL-OUT-OF-PLANE (IJK)

This frame uses the Sun as its origin and an I-axis that points towards the orbiting body. Furthermore, the in-plane J-axis is defined perpendicular to this I-axis and the K-axis then complements the system. This reference frame can be considered to be a "local coordinate system", due to it rotating as the body orbits the Sun [25]. Again, a graphical representation of this reference frame can be seen in Figure B.3 [23].

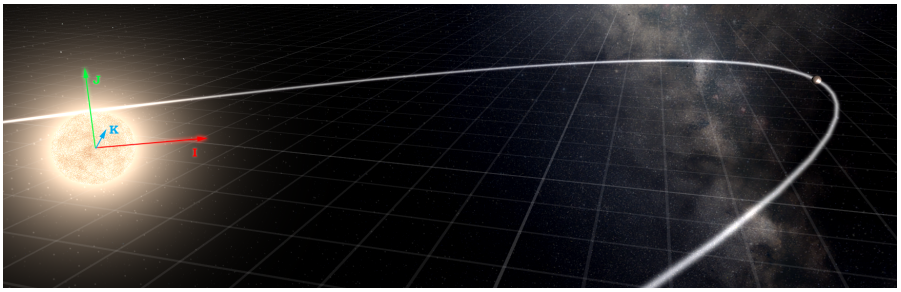


Figure B.3: A graphical representation of the radial-orthoradial-out-of-plane (IJK) reference frame [23].

B.2.2. RADIAL-TRANSVERSE-NORMAL (RSW)

Furthermore, one can consider the radial-transverse-normal frame. This frame, also known as the RSW-frame, uses the orbiting object as its origin and an R-axis that is radially coming from the Sun. As a secondary and tertiary axis, it uses an in-plane S-axis perpendicular to the R-axis and a W-axis that completes the right-handed system [26]. The RS-plane therefore again coincides with the orbital plane, as can be seen in Figure B.4 [23]. It can be noted here that the RSW-frame is essentially a translated version of the IJK-frame, as can also be found by comparing the figure shown below and Figure B.3.

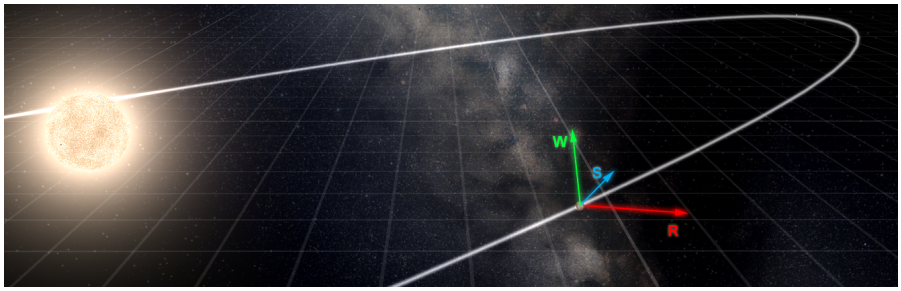


Figure B.4: The radial-transverse-normal (RSW) reference frame and the direction of its axes [23].

B.2.3. VELOCITY-FIXED (CDE)

Lastly, the velocity-fixed reference frame can be introduced. This frame has its C-axis pointing along the velocity vector and the D-axis perpendicular to the C-axis, in the orbital plane and thus pointing towards the central body in case of a circular orbit [27]. The E-axis complements the system as shown in Figure B.5 [23].

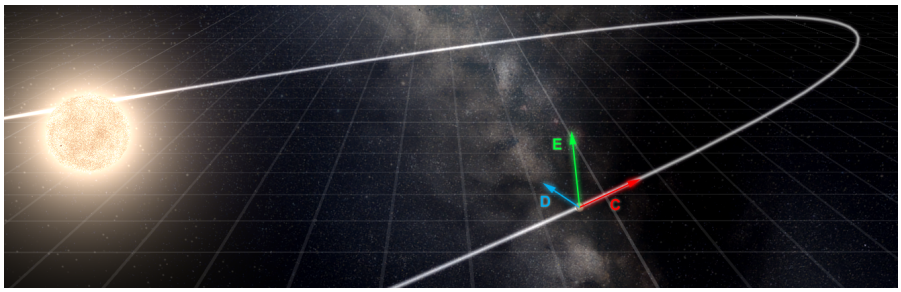


Figure B.5: The definition of the velocity-fixed (CDE) reference frame [23].

B.3. TRANSFORMATIONS

In order to change between the different reference frames mentioned previously, a number of transformation matrices are required. This section will elaborate on these matrices.

B.3.1. TRANSFORMATION MATRICES

For every rotation about one of the axes, the following three transformation matrices can be defined [25]:

$$\eta_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\eta) & sin(\eta) \\ 0 & -sin(\eta) & cos(\eta) \end{bmatrix} \quad \eta_y = \begin{bmatrix} cos(\eta) & 0 & -sin(\eta) \\ 0 & 1 & 0 \\ sin(\eta) & 0 & cos(\eta) \end{bmatrix} \quad \eta_z = \begin{bmatrix} cos(\eta) & sin(\eta) & 0 \\ -sin(\eta) & cos(\eta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

These three matrices form the fundamental rotation matrices and can be used to transform vectors from one reference frame into another, and will be used to do so in the upcoming sections.

B.3.2. RADIAL-ORTHORADIAL-OUT-OF-PLANE TO HELIOCENTRIC ARIES ECLIPTIC

To transform the radial-orthoradial-out-of-plane (IJK) frame into the heliocentric Aries ecliptic (XYZ) frame, it can first be noted that the position vector in the initial reference frame is $\mathbf{r} = [r\ 0\ 0]^T$ and that the velocity vector is given by $\mathbf{V} = [\dot{r}\ r\dot{\nu}\ 0]^T$. The transformation can now be carried out with the following three steps:

1. Rotation of $-(\omega+\theta)$ about the K-axis
2. Rotation of $(-i)$ about the new X'-axis
3. Rotation of $(-\Omega)$ about the resulting Z"-axis

Substitution of these angles into the fundamental transformation matrices results in Equation B.1 [25].

$$\mathbf{r}^{(XYZ)} = [-\Omega_z]\,[-i_x]\,[-(\Omega+\theta)_z]\,\mathbf{r}^{(ijk)} \tag{B.1}$$

By writing out the equation shown above and defining $cos()$ as $c()$, $sin()$ as $s()$ and $(\omega+\theta)$ as χ, Equation B.2 can now be found.

$$\begin{aligned}\begin{bmatrix}X\\Y\\Z\end{bmatrix} &= \begin{bmatrix}c(\Omega) & -s(\Omega) & 0\\ s(\Omega) & c(\Omega) & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & c(i) & -s(i)\\ 0 & s(i) & c(i)\end{bmatrix}\begin{bmatrix}c(\chi) & -s(\chi) & 0\\ s(\chi) & c(\chi) & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}r\\0\\0\end{bmatrix}\\ &= \mathbf{T}_{(IJK2XYZ)}\mathbf{r}^{(IJK)}\end{aligned} \tag{B.2}$$

where the complete transformation matrix $\mathbf{T}_{(IJK2XYZ)}$ can also be rewritten to:

$$\mathbf{T}_{(IJK2XYZ)} = \begin{bmatrix}c(\Omega)c(\chi)-s(\Omega)s(\chi)c(i) & -c(\Omega)s(\chi)-s(\Omega)c(\chi)c(i) & s(\Omega)s(i)\\ s(\Omega)c(\chi)+c(\Omega)s(\chi)c(i) & -s(\Omega)s(\chi)+c(\Omega)c(\chi)c(i) & -c(\Omega)s(i)\\ s(\chi)s(i) & c(\chi)s(i) & c(i)\end{bmatrix} \tag{B.3}$$

For these equations, the properties $cos(-\eta) = cos(\eta)$ and $sin(-\eta) = -sin(\eta)$ were used. The transformed velocity vector can now also be found by simply multiplying the initial vector $[\dot{r}\ r\dot{\upsilon}\ 0]^T$ with this transformation matrix.

B.3.3. RADIAL-ORTHORADIAL-OUT-OF-PLANE TO PERIFOCAL

From the figures shown in Section B.1, one can note that both reference frames have the Sun as their origin and have a plane that coincides with the orbital plane. As such, the transformation only consists of a rotation of $-\theta$, the true anomaly, around the Z-axis. The transformation matrix $\mathbf{T}_{(IJK2PQW)}$ becomes [25]:

$$\mathbf{T}_{(IJK2PQW)} = \begin{bmatrix}cos(\theta) & -sin(\theta) & 0\\ sin(\theta) & cos(\theta) & 0\\ 0 & 0 & 1\end{bmatrix} \tag{B.4}$$

B.3.4. RADIAL-TRANSVERSE-NORMAL TO PERIFOCAL

To convert between the radial-transverse-normal frame and the perifocal frame, one should rotate $-\theta$ around the Z-axis and translate the frame such that it uses the origin of the perifocal frame. The transformation matrix $\mathbf{T}_{(RSW2PQW)}$ is thus [26]:

$$\mathbf{T}_{(RSW2PQW)} = \begin{bmatrix}cos(\theta) & -sin(\theta) & 0\\ sin(\theta) & cos(\theta) & 0\\ 0 & 0 & 1\end{bmatrix} + \mathbf{r}_{\theta}^{PQW} \tag{B.5}$$

where $\mathbf{r}_{\theta}^{PQW}$ denotes the vector that translates the frame such that it uses the central body as its origin instead of the orbiting object.

B.3.5. RADIAL-TRANSVERSE-NORMAL TO HELIOCENTRIC ARIES ECLIPTIC

As mentioned earlier, the difference between the IJK-Frame and the RSW-Frame is a simple translation. As such, the rotation matrices are the same. In order to convert between the RSW- and XYZ-Frame, Equation B.3 can thus be used as rotation matrix and a vector $\mathbf{r}_{\theta}^{XYZ}$ containing the required translation can be added. The transformation matrix is therefore [26]:

$$\mathbf{T}_{(RSW2XYZ)} = \begin{bmatrix}c(\Omega)c(\chi)-s(\Omega)s(\chi)c(i) & -c(\Omega)s(\chi)-s(\Omega)c(\chi)c(i) & s(\Omega)s(i)\\ s(\Omega)c(\chi)+c(\Omega)s(\chi)c(i) & -s(\Omega)s(\chi)+c(\Omega)c(\chi)c(i) & -c(\Omega)s(i)\\ s(\chi)s(i) & c(\chi)s(i) & c(i)\end{bmatrix} + \mathbf{r}_{\theta}^{XYZ} \tag{B.6}$$

where for ease of reading, χ is defined as being equal to $(\omega+\theta)$.

B.3.6. PERIFOCAL TO HELIOCENTRIC ARIES ECLIPTIC

It can be noted that, if the true anomaly is set to zero, the radial-orthoradial-out-of-plane (IJK) and perifocal (PQW) systems would align. The matrix found for the IJK-to-XYZ transformation can then be used to convert the perifocal PQW-Frame to the heliocentric Aries ecliptic XYZ-Frame [25]. In short, the transformation matrix becomes:

$$\mathbf{T}_{(PQW2XYZ)} = \begin{bmatrix}c(\Omega)c(\omega)-s(\Omega)s(\omega)c(i) & -c(\Omega)s(\omega)-s(\Omega)c(\omega)c(i) & s(\Omega)s(i)\\ s(\Omega)c(\omega)+c(\Omega)s(\omega)c(i) & -s(\Omega)s(\omega)+c(\Omega)c(\omega)c(i) & -c(\Omega)s(i)\\ s(\omega)s(i) & c(\omega)s(i) & c(i)\end{bmatrix} \tag{B.7}$$

B.3.7. VELOCITY-FIXED TO RADIAL-TRANSVERSE-NORMAL

In order to complete the set of transformation matrices, a transformation between the velocity-fixed reference frame and the previously mentioned reference frames is required. From the figures shown in Section B.2.3, it can be noted that a simple rotation around the out-of-plane axis is sufficient to transform the CDE-frame to the RSW-frame. The transformation matrix therefore becomes:

$$\mathbf{T}_{(CDE2RSW)} = \begin{bmatrix} cos(\alpha) & sin(\alpha) & 0 \\ -sin(\alpha) & cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{B.8}$$

where α is the angle between the radius vector and the spacecraft's velocity. If one defines the flight path angle γ as the angle between the local horizontal and the velocity vector, it can be found that $\alpha = 90^\circ - \gamma$, as shown in Figure B.6. By using the transformation matrices previously discussed in this chapter, the velocity-fixed reference frame can now be transformed to the other reference frames.

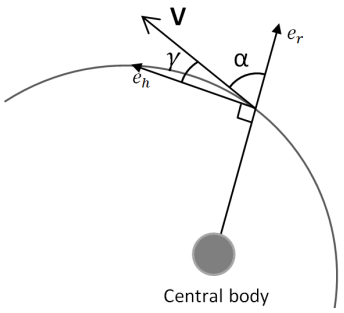


Figure B.6: The definition of the flight path angle and α with respect to the local horizontal vector $\mathbf{e}_h$, the velocity vector $\mathbf{V}$ and the radial unit vector $\mathbf{e}_r$.

C

PROPERTIES OF KEPLER ORBITS

As mentioned in Section 2.2.4, Kepler orbits are the solutions found to the two-body problem when the mass of the central body is assumed to be much larger than that of the satellite. Properties of the elliptical, parabolic and hyperbolic orbits can be seen in Table C.1 [28]. Due to a circular orbit being a special case of the elliptical one, it is not shown separately in this table.

Table C.1: Mathematical properties of elliptical, parabolic and hyperbolic orbits [28].

	Ellipse	Parabola	Hyperbola
Parametric equation	$\frac{x^2}{a^2}+\frac{y^2}{b^2}=1$	$x^2=4qy$	$\frac{x^2}{a^2}-\frac{y^2}{b^2}=1$
Focal distance r	$a(1-ecos(E))$ or $\frac{a(1-e^2)}{1+ecos(\theta)}$	$q+\frac{D^2}{2}$ or $\frac{2q}{1+cos(\theta)}$	$a(1-ecosh(F))$ or $\frac{a(1-e^2)}{1+ecos(\theta)}$
Eccentricity e	$0 \leq e < 1$	1	> 1
Semi-major axis a	$0 \leq a < \infty$	∞	< 0
Mean motion n	$\sqrt{\mu/a^3}$	$\sqrt{\mu/p^3}$	$\sqrt{\mu/(-a^3)}$
Mean anomaly M	$n(t-t_0)$	$n(t-t_0)$	$n(t-t_0)$
Eccentric anomaly E,D,F	$tan\left(\frac{E}{2}\right)=\left(\frac{1-e}{1+e}\right)^{1/2}tan\left(\frac{\theta}{2}\right)$	$D=\sqrt{2q}\cdot tan\left(\frac{\theta}{2}\right)$	$tanh\left(\frac{F}{2}\right)=\left(\frac{1-e}{1+e}\right)^{1/2}tan\left(\frac{\theta}{2}\right)$
True anomaly θ	$tan\left(\frac{\theta}{2}\right)=\left(\frac{1+e}{1-e}\right)^{1/2}tan\left(\frac{E}{2}\right)$	$\theta=atan2\left(\left(\frac{p-r}{r},\frac{D_p}{r\sqrt{2q}}\right)\right)$	$tan\left(\frac{\theta}{2}\right)=\left(\frac{1+e}{1-e}\right)^{1/2}tan\left(\frac{F}{2}\right)$

D

VERIFICATION OF EQUATIONS FOUND IN LITERATURE

Several equations obtained from literature are used in this thesis. In order to verify these equations, this appendix shall elaborate on their derivations.

D.1. RELATIONS FOUND USING SPHERICAL TRIANGLES

As mentioned in Section 3.1.1, Vinti et al. found several relations between the orbital elements using spherical triangles. This section shall elaborate upon the derivation process of the four relations shown in Equation 3.1. For convenience, these four relations can also be found in Equation D.1 [13].

$$\begin{aligned} sin\left(\phi\right) &= sin\left(i\right) sin\left(\omega+\theta\right) \\ cos\left(\phi\right) &= cos\left(\alpha-\Omega\right) cos\left(\omega+\theta\right) + sin\left(\alpha-\Omega\right) sin\left(\omega+\theta\right) cos\left(i\right) \\ cos\left(\omega+\theta\right) &= cos\left(\phi\right) cos\left(\alpha-\Omega\right) \\ cos\left(\phi\right) sin\left(\alpha-\Omega\right) &= sin\left(\omega+\theta\right) cos\left(i\right) \end{aligned} \tag{D.1}$$

where the definition of the variables can be seen in Figure 2.1 [4].

It can first be noted that to derive the relations shown in Equation D.1, the law of sines and Figure 2.1 can be used to find the following relation:

$$\frac{sin\left(\phi\right)}{sin\left(i\right)} = \frac{sin\left(\omega+\theta\right)}{sin\left(\pi/2\right)} \tag{D.2}$$

As $sin\left(\pi/2\right)$ is equal to one, this equation can be rewritten to:

$$sin\left(\phi\right) = sin\left(i\right) sin\left(\omega+\theta\right) \tag{D.3}$$

This is the first relation shown in Equation D.1. Furthermore, as shown in Equation D.4, the second relation can be obtained by using the law of cosines.

$$cos\left(\phi\right) = cos\left(\alpha-\Omega\right) cos\left(\omega+\theta\right) + sin\left(\alpha-\Omega\right) sin\left(\omega+\theta\right) cos\left(i\right) \tag{D.4}$$

In a similar manner, the following equation can also be found by using the law of cosines and Figure 2.1:

$$cos\left(\omega+\theta\right) = cos\left(\phi\right) cos\left(\alpha-\Omega\right) + sin\left(\phi\right) sin\left(\alpha-\Omega\right) cos\left(\pi/2\right) \tag{D.5}$$

As $cos\left(\pi/2\right)$ is equal to zero, this can be rewritten to Equation D.6. This is the third relation shown in Equation D.1.

$$cos\left(\omega+\theta\right) = cos\left(\phi\right) cos\left(\alpha-\Omega\right) \tag{D.6}$$

To find the last relation, one should first multiply Equation D.4 by $sin\left(\alpha-\Omega\right)$. This results in Equation D.7.

$$cos\left(\phi\right) sin\left(\alpha-\Omega\right) = cos\left(\alpha-\Omega\right) sin\left(\alpha-\Omega\right) + sin^2\left(\alpha-\Omega\right) sin\left(\omega+\theta\right) sin\left(i\right) \tag{D.7}$$

By substituting Equation D.6 into this equation, the following relation can be obtained:

$$cos(\phi)\, sin(\alpha-\Omega) = cos^2(\alpha-\Omega)\, cos(\phi)\, sin(\alpha-\Omega) + sin^2(\alpha-\Omega)\, sin(\omega+\theta)\, cos(i) \tag{D.8}$$

This equation can then be rewritten to Equation D.9.

$$cos(\phi)\, sin(\alpha-\Omega)\left(1-cos^2(\alpha-\Omega)\right) = sin^2(\alpha-\Omega)\, sin(\omega+\theta)\, cos(i) \tag{D.9}$$

As the last part of the left-hand side of this equation is equal to $sin^2(\alpha-\Omega)$, this equation can be rewritten to:

$$cos(\phi)\, sin(\alpha-\Omega)\, sin^2(\alpha-\Omega) = sin^2(\alpha-\Omega)\, sin(\omega+\theta)\, cos(i) \tag{D.10}$$

By dividing the equation shown above by $sin^2(\alpha-\Omega)$, the fourth relation shown in Equation D.1 can now be obtained.

$$cos(\phi)\, sin(\alpha-\Omega) = sin(\omega+\theta)\, cos(i) \tag{D.11}$$

D.2. VARIATION OF THE ELEVATION ANGLE

In this thesis, a relation between the inclination i, right ascension of the ascending node (RAAN) Ω, azimuthal angle θ and the elevation angle ϕ found by Novak is used. The relation can be seen in Equation D.12 [1]. To verify this equation, it is re-derived in this section.

$$sin(\phi) = \frac{sin(\theta-\Omega)\, sin(i)}{\sqrt{sin^2(\theta-\Omega) + cos^2(\theta-\Omega)\, cos^2(i)}} \tag{D.12}$$

One can start the derivation by assuming that the following two equations hold:

$$\begin{aligned} (\boldsymbol{e}_r \times \boldsymbol{\pi}) \cdot \boldsymbol{e}_z &= 0 \\ (\boldsymbol{\Omega} \times \boldsymbol{\pi}) \times \boldsymbol{e}_z &= \boldsymbol{0} \end{aligned} \tag{D.13}$$

The definition of these vectors can be seen in Figure D.1 [1].

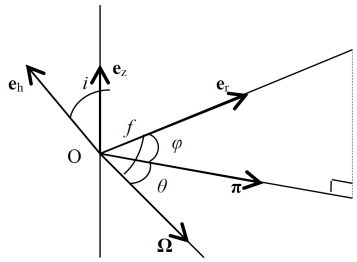


Figure D.1: The definition of the vectors used in the derivation [1].

In this figure, $\boldsymbol{e}_r$, $\boldsymbol{e}_h$ and $\boldsymbol{\Omega}$ are the unit vectors in the direction of the radius vector, angular momentum and RAAN, $\boldsymbol{\pi}$ is the unit vector in the direction of the orthogonal projection of $\boldsymbol{e}_r$ onto the equatorial plane and $\boldsymbol{e}_z$ is the unit vector perpendicular to this plane. Furthermore, i is the inclination, θ is the azimuthal angle, ϕ is the elevation angle and f is the summation of the argument of pericenter and the true anomaly [1].

From Figure D.1, it can be concluded that the first assumption shown in Equation D.12 is valid due to $(\boldsymbol{e}_r \times \boldsymbol{\pi})$ being perpendicular to $\boldsymbol{e}_z$ and that the second assumption is valid due to the fact that $(\boldsymbol{\Omega} \times \boldsymbol{\pi})$ is parallel to $\boldsymbol{e}_z$.

Next, one can write $(\boldsymbol{\Omega} \times \boldsymbol{\pi}) \cdot (\boldsymbol{\Omega} \times \boldsymbol{e}_r)$ as shown below:

$$(\boldsymbol{\Omega} \times \boldsymbol{\pi}) \cdot (\boldsymbol{\Omega} \times \boldsymbol{e}_r) = sin(\theta) \cdot \boldsymbol{e}_z \cdot (\boldsymbol{\Omega} \times \boldsymbol{e}_r) \tag{D.14}$$

which, using the geometric properties of the dot product and cross product, can be rewritten to:

$$(\boldsymbol{\Omega} \times \boldsymbol{\pi}) \cdot (\boldsymbol{\Omega} \times \boldsymbol{e}_r) = sin(\theta) \cdot \boldsymbol{e}_z \left(sin(f) \cdot \boldsymbol{e}_h\right) \tag{D.15}$$

Using the same set of equations to rewrite the dot product and cross product, one can find that this in turn can be written as shown in Equation D.16.

$$(\boldsymbol{\Omega} \times \boldsymbol{\pi}) \cdot (\boldsymbol{\Omega} \times \boldsymbol{e}_r) = sin(\theta)\, sin\big(f\big)\, \boldsymbol{e}_z \boldsymbol{e}_h = sin(\theta)\, sin\big(f\big)\, cos(i) \tag{D.16}$$

Furthermore, one can note that instead of writing $(\boldsymbol{\Omega} \times \boldsymbol{\pi}) \cdot (\boldsymbol{\Omega} \times \boldsymbol{e}_r)$ as shown in Equation D.14, it can also be written as shown below:

$$(\boldsymbol{\Omega} \times \boldsymbol{\pi}) \cdot (\boldsymbol{\Omega} \times \boldsymbol{e}_r) = ((\boldsymbol{\Omega} \times \boldsymbol{\pi}) \times \boldsymbol{\Omega}) \cdot \boldsymbol{e}_r \tag{D.17}$$

which, as the length of the unit vector $\boldsymbol{\Omega}$ is 1, can be rewritten to:

$$(\boldsymbol{\Omega} \times \boldsymbol{\pi}) \cdot (\boldsymbol{\Omega} \times \boldsymbol{e}_r) = \boldsymbol{\pi} \cdot \boldsymbol{e}_r - cos(\theta)\, (\boldsymbol{\Omega} \cdot \boldsymbol{e}_r) \tag{D.18}$$

Using the geometric properties of the dot product and cross product mentioned earlier, Equation D.19 can now be found.

$$(\boldsymbol{\Omega} \times \boldsymbol{\pi}) \cdot (\boldsymbol{\Omega} \times \boldsymbol{e}_r) = cos\big(\phi\big) - cos(\theta)\, cos\big(f\big) \tag{D.19}$$

As Equation D.16 and D.19 should be equal, Equation D.20 can be obtained.

$$cos\big(\phi\big) = cos(\theta)\, cos\big(f\big) + sin(\theta)\, sin\big(f\big)\, cos(i) \tag{D.20}$$

Next, it can be noted that as $(\boldsymbol{\Omega} \times \boldsymbol{\pi})$ is in the same direction as $\boldsymbol{e}_z$ and since $(\boldsymbol{e}_r \times \boldsymbol{\pi}) \cdot \boldsymbol{e}_z = 0$, the following equation holds as well:

$$(\boldsymbol{e}_r \times \boldsymbol{\pi}) \cdot (\boldsymbol{\Omega} \times \boldsymbol{\pi}) = 0 \tag{D.21}$$

By again using the properties of the dot product and cross product, this can be rewritten to:

$$(\boldsymbol{e}_r \times \boldsymbol{\pi}) \cdot (\boldsymbol{\Omega} \times \boldsymbol{\pi}) = cos\big(f\big) - cos(\theta)\, cos\big(\phi\big) = 0 \tag{D.22}$$

from which one can now find:

$$cos\big(f\big) = cos(\theta)\, cos\big(\phi\big) \tag{D.23}$$

By substituting Equation D.20 into the equation shown above, Equation D.24 is the result.

$$cos\big(f\big) = cos(\theta)\, \big(sin(\theta)\, sin\big(f\big)\, cos(i) + cos(\theta)\, cos\big(f\big)\big) \tag{D.24}$$

By dividing this equation by $cos\big(f\big)$, one now obtains the equation shown below:

$$1 = cos(\theta)\, \big(sin(\theta)\, tan\big(f\big)\, cos(i) + cos(\theta)\big) \tag{D.25}$$

which is in turn equal to Equation D.26.

$$1 = cos(\theta)\, sin(\theta)\, tan\big(f\big)\, cos(i) + cos^2(\theta) \tag{D.26}$$

If $cos^2(\theta)$ is now subtracted from both sides, Equation D.27 is the result.

$$1 - cos^2(\theta) = cos(\theta)\, sin(\theta)\, tan\big(f\big)\, cos(i) \tag{D.27}$$

By using the fact that $sin^2(\theta) + cos^2(\theta) = 1$, this can be rewritten to:

$$sin^2(\theta) = cos(\theta)\, sin(\theta)\, tan\big(f\big)\, cos(i) \tag{D.28}$$

If this equation is subsequently divided by $cos(\theta)\, sin(\theta)$, Equation D.29 is found.

$$tan(\theta) = cos(i)\, \frac{sin\big(f\big)}{cos\big(f\big)} = cos(i)\, tan\big(f\big) \tag{D.29}$$

If one now uses the properties $z = r\, sin(i)\, sin\big(f\big)$ and $sin\big(\phi\big) = z/r$, Equation D.30 can be derived.

$$sin\big(\phi\big) = sin(i)\, sin\big(f\big) \rightarrow sin\big(f\big) = \frac{sin\big(\phi\big)}{sin(i)} \tag{D.30}$$

When Equation D.30 is now substituted into Equation D.29 and if the property $sin^2(f) + cos^2(f) = 1$ is used, Equation D.31 can now be obtained.

$$tan(\theta) = cos(i) \frac{sin(\phi)}{sin(i)\sqrt{1 - \frac{sin^2(\phi)}{cos^2(\phi)}}} \tag{D.31}$$

which, if multiplied by $\sqrt{1 - sin^2(\phi)/cos^2(\phi)}$, results in:

$$tan(\theta)\sqrt{1 - \frac{sin^2(\phi)}{cos^2(\phi)}} = cos(i) \frac{sin(\phi)}{sin(i)} \tag{D.32}$$

By squaring the equation shown above, Equation D.33 is now found.

$$tan^2(\theta)\left(1 - \frac{sin^2(\phi)}{cos^2(\phi)}\right) = cos^2(i) \frac{sin^2(\phi)}{sin^2(i)} \tag{D.33}$$

Once the brackets have been removed from this equation, Equation D.34 is the result:

$$tan^2(\theta) - tan^2(\theta) \frac{sin^2(\phi)}{cos^2(\phi)} = cos^2(i) \frac{sin^2(\phi)}{sin^2(i)} \tag{D.34}$$

By adding $tan^2(\theta)\, sin^2(\phi)/sin^2(i)$ to this equation, the following equation is found:

$$tan^2(\theta) = cos^2(i) \frac{sin^2(\phi)}{cos^2(\phi)} + \frac{tan^2(\theta)\, sin^2(\phi)}{sin^2(i)} \tag{D.35}$$

which, by introducing brackets can be rewritten to:

$$tan^2(\theta) = sin^2(\phi)\left(\frac{cos^2(i) + tan^2(\theta)}{sin^2(i)}\right) \tag{D.36}$$

By attempting to keep only $sin^2(\phi)$ on the right-hand side, Equation D.37 is now obtained.

$$tan^2(\theta) \frac{sin^2(i)}{cos^2(i) + tan^2(\theta)} = sin^2(\phi) \tag{D.37}$$

If the tangent is now written in its cosine and sine component, the following equation is found:

$$\frac{sin^2(\theta)}{cos^2(\theta)} \frac{sin^2(i)}{cos^2(i) + \frac{sin^2(\theta)}{cos^2(\theta)}} = sin^2(\phi) \tag{D.38}$$

which, when combined into a single fraction results in:

$$\frac{sin^2(\theta)\, sin^2(i)}{cos^2(\theta)\, cos^2(i) + sin^2(\theta)} = sin^2(\phi) \tag{D.39}$$

Once the square root is taken of this equation and θ is replaced by $(\theta - \Omega)$ for generalisation, the equation shown in Novak's PhD thesis is found [1].

E

VARIATION OF THE FOURIER SERIES COEFFICIENTS

As mentioned in Section 3.3.2, the variation of each of the coefficients in the third-order Fourier series was determined for a number of test cases. This appendix contains the plots for the coefficients that were not discussed in Section 3.3.2.

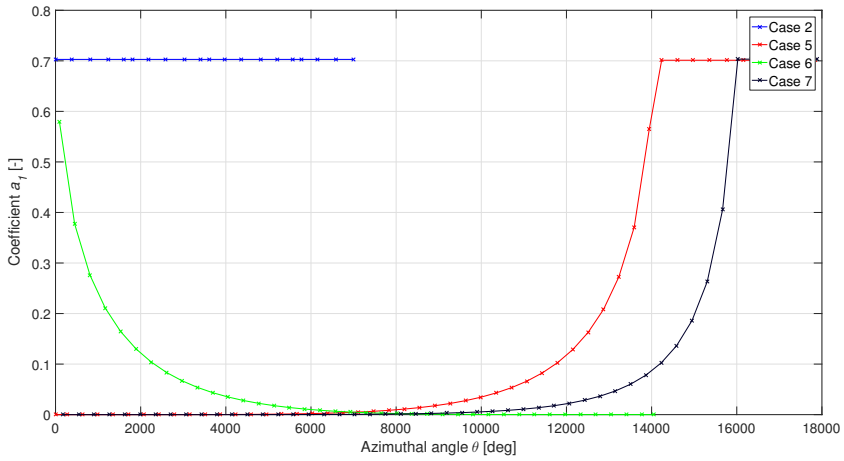


Figure E.1: The variation of the coefficient a_1 for cases 2, 5, 6 and 7.

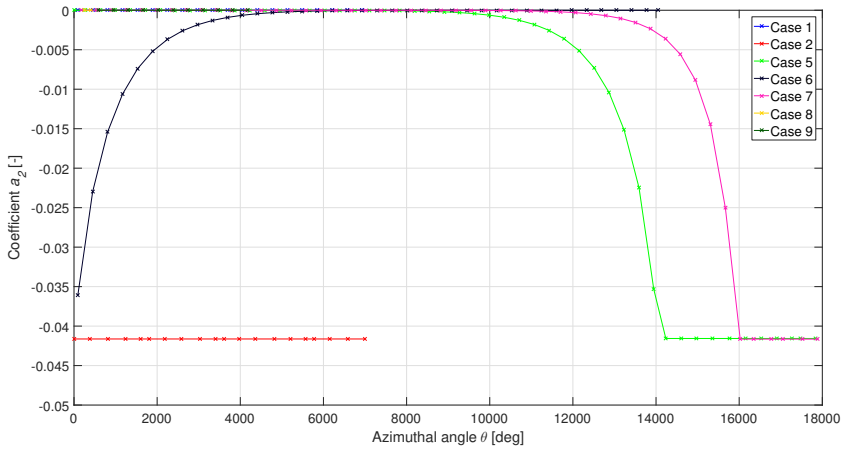


Figure E.2: The value of the coefficient a_2 as a function of the azimuthal angle for cases 1, 2, 5, 6 ,7, 8 and 9.

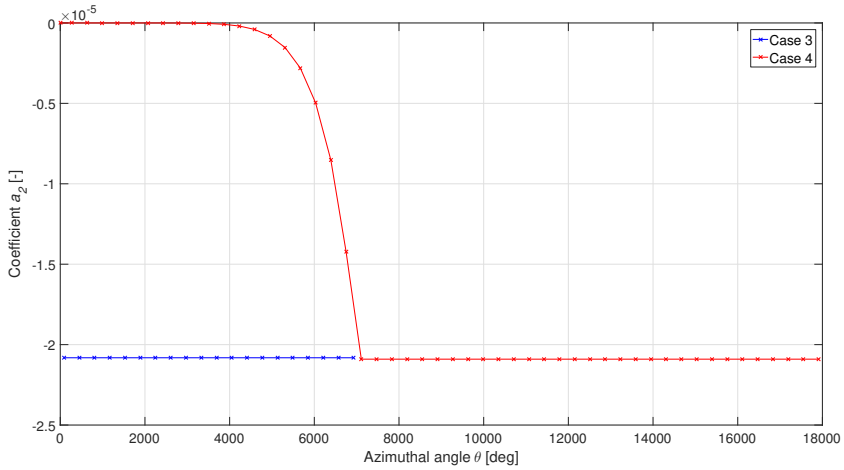


Figure E.3: The variation of the coefficient a_2 for cases 3 and 4.

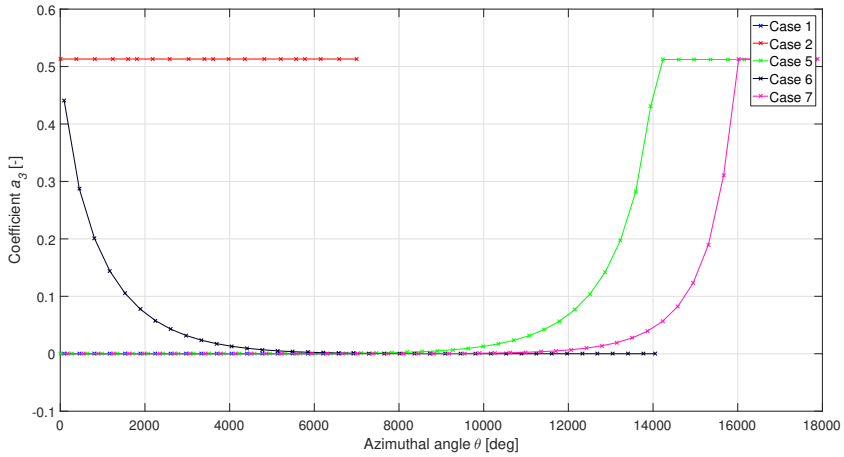


Figure E.4: The variation of the coefficient a_3 for cases 1, 2, 5, 6 and 7.

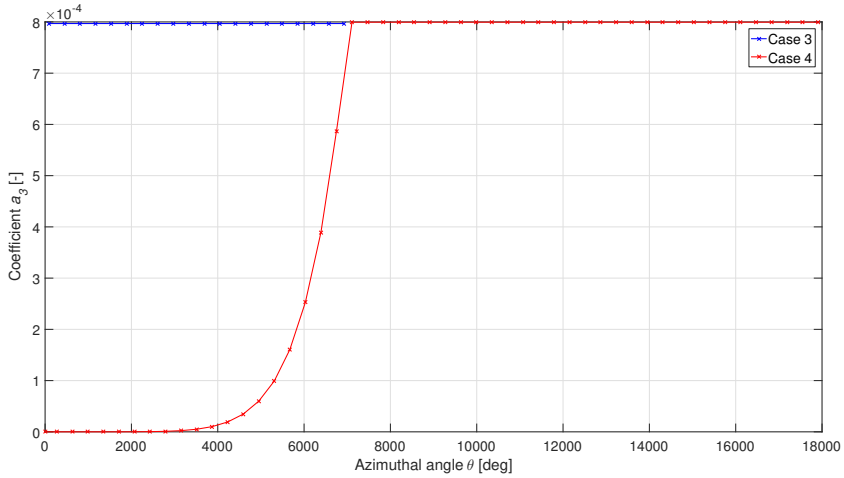


Figure E.5: The value of the coefficient a_3 as a function of the azimuthal angle for cases 3 and 4.

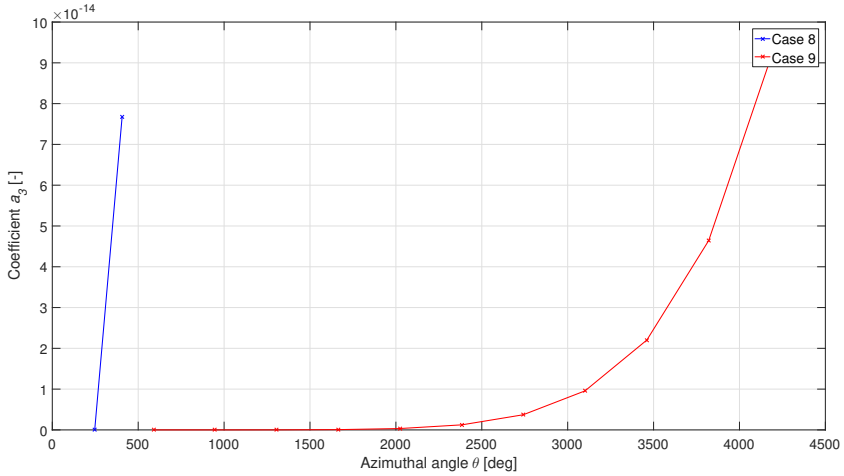


Figure E.6: The variation of the coefficient a_3 for cases 8 and 9.

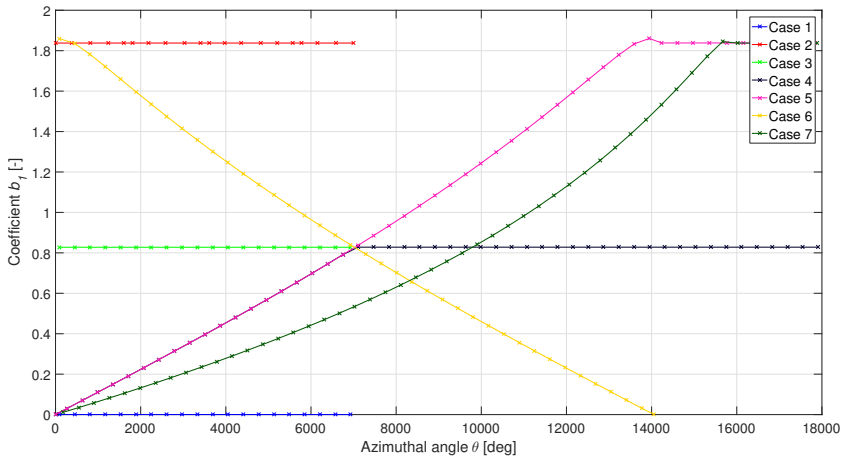


Figure E.7: The value of the coefficient b_1 versus the azimuthal angle for cases 1 to 7.

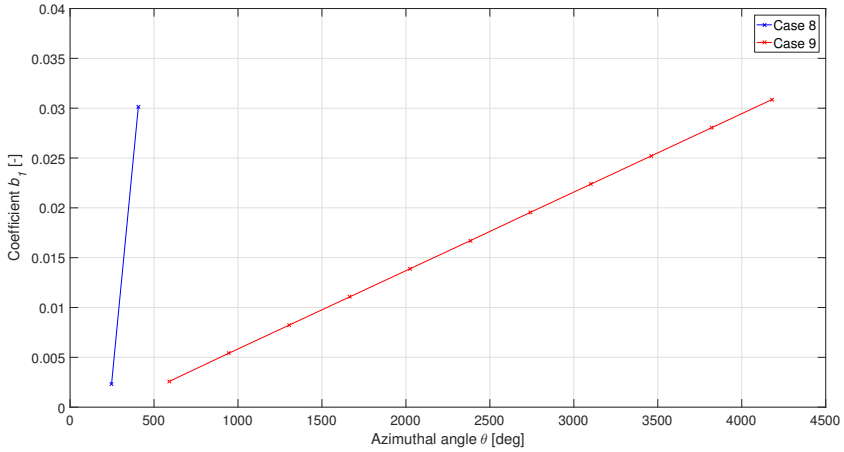


Figure E.8: The variation of the coefficient b_1 for cases 8 and 9.

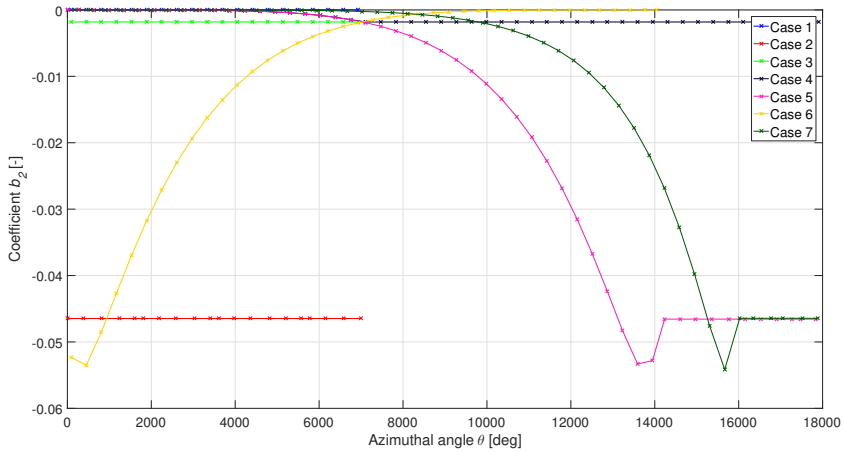


Figure E.9: The variation of the coefficient b_2 for cases 1 to 7.

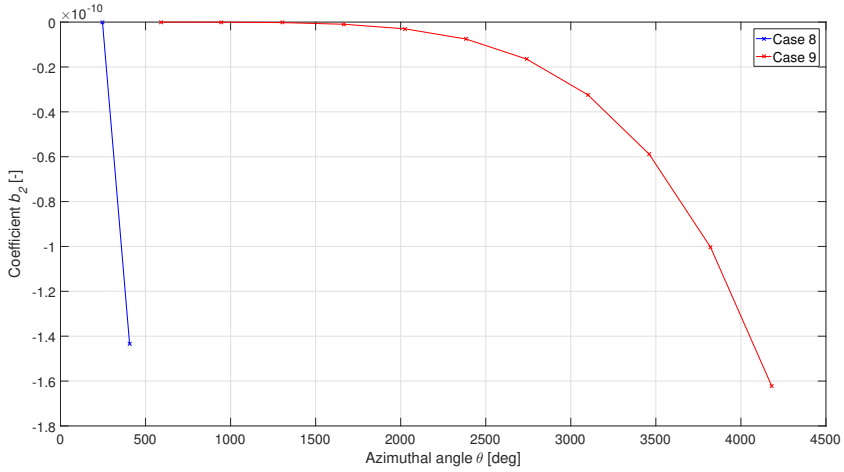


Figure E.10: The value of the coefficient b_2 versus the azimuthal angle for cases 8 and 9.

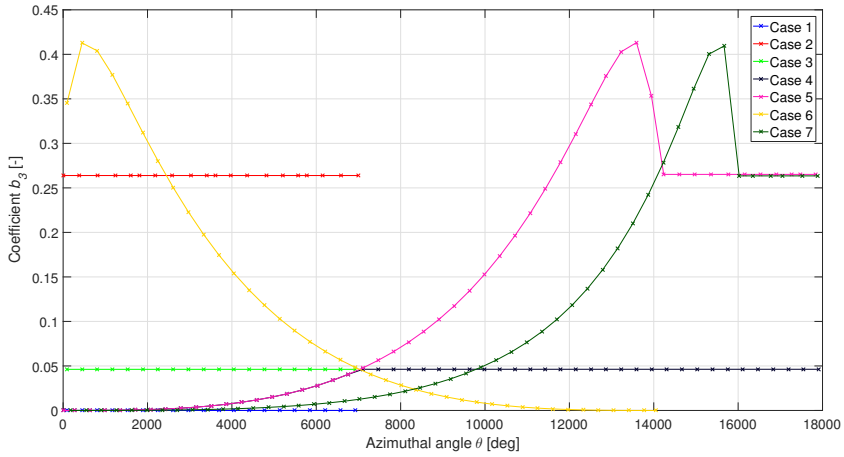


Figure E.11: The value of the coefficient b_3 versus the azimuthal angle for cases 1 to 7.

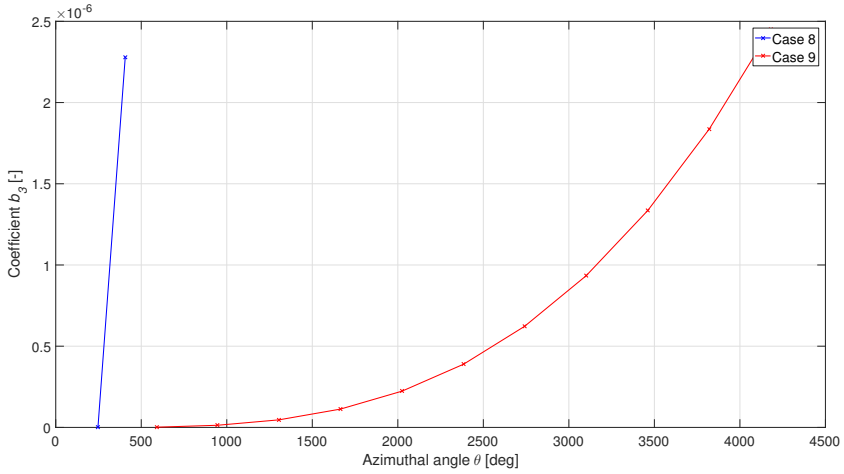


Figure E.12: The value of the coefficient b_3 versus the azimuthal angle for cases 8 and 9.

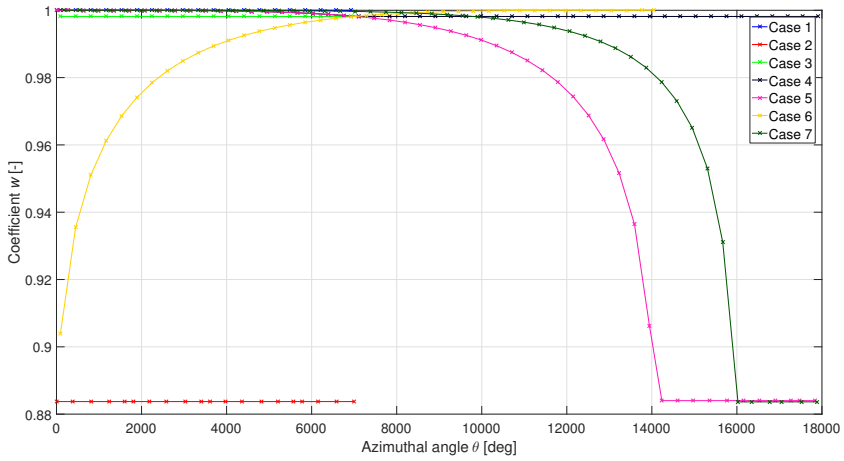


Figure E.13: The value of the coefficient w versus the azimuthal angle for cases 1 to 7.

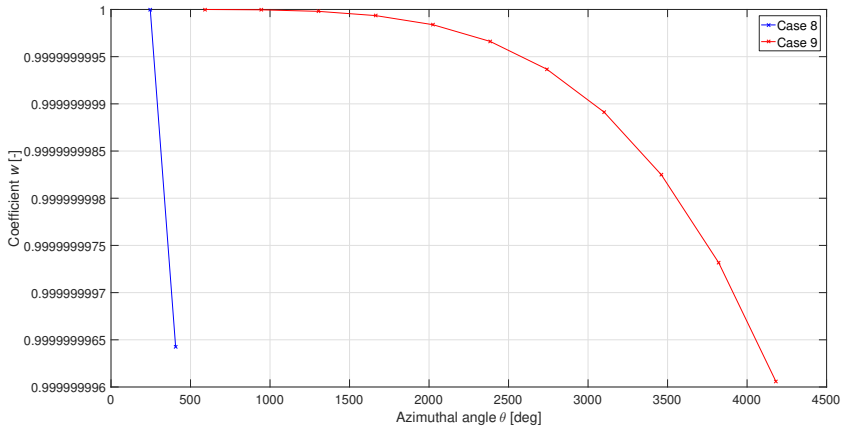


Figure E.14: The variation of the coefficient w for cases 8 and 9.

F

FOURIER COEFFICIENTS AS FUNCTION OF INCLINATION

To illustrate the behaviour of the coefficients used in the third-order Fourier series, their variation has been plotted as a function of inclination. This appendix contains the plots for the coefficients that are not discussed in Section 3.3.2.

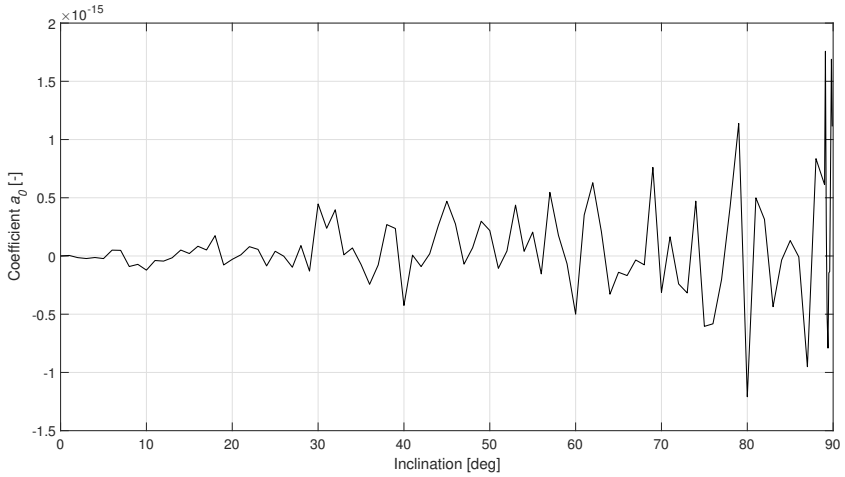


Figure F.1: The variation of the coefficient a_0 as a function of inclination.

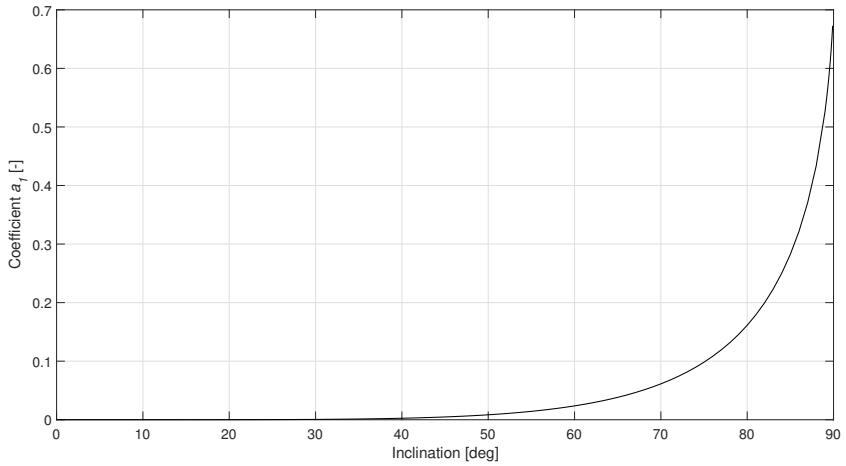


Figure F.2: The change of the coefficient a_1 as a function of inclination.

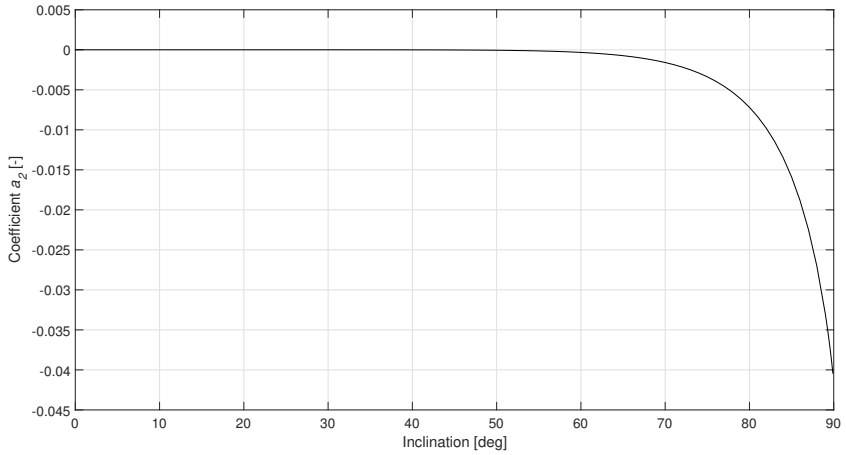


Figure F.3: The value of the coefficient a_2 versus the inclination.

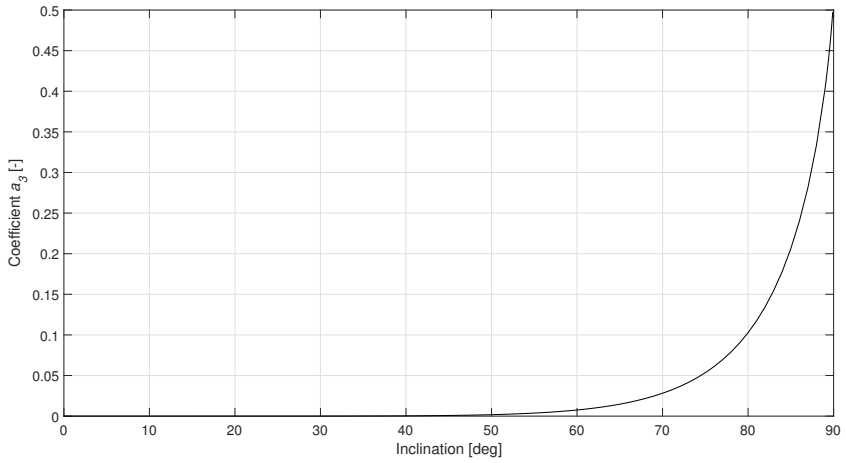


Figure F.4: The variation of the coefficient a_3 as a function of inclination.

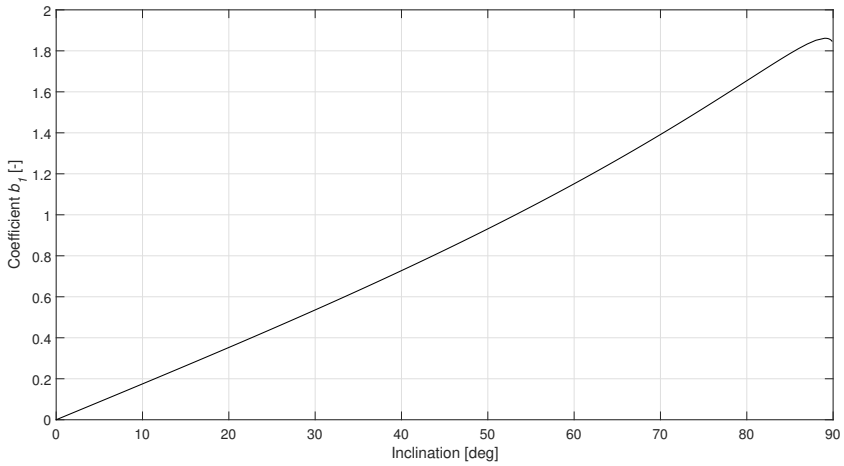


Figure F.5: The value of the coefficient b_1 versus the inclination.

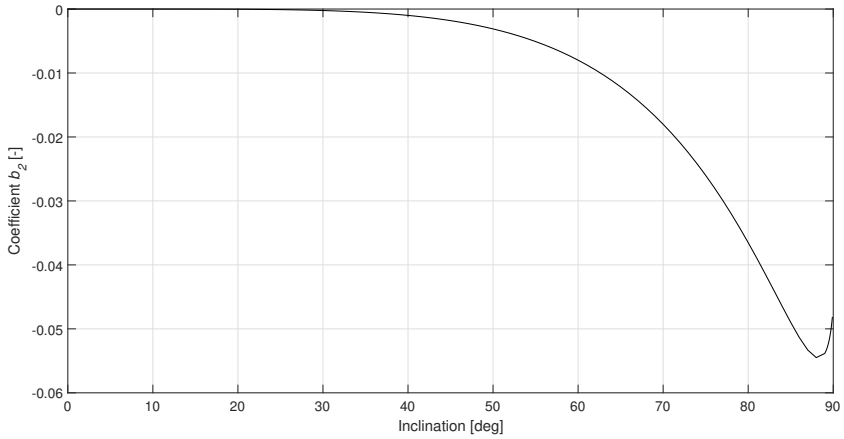


Figure F.6: The change of the coefficient b_2 as a function of inclination.

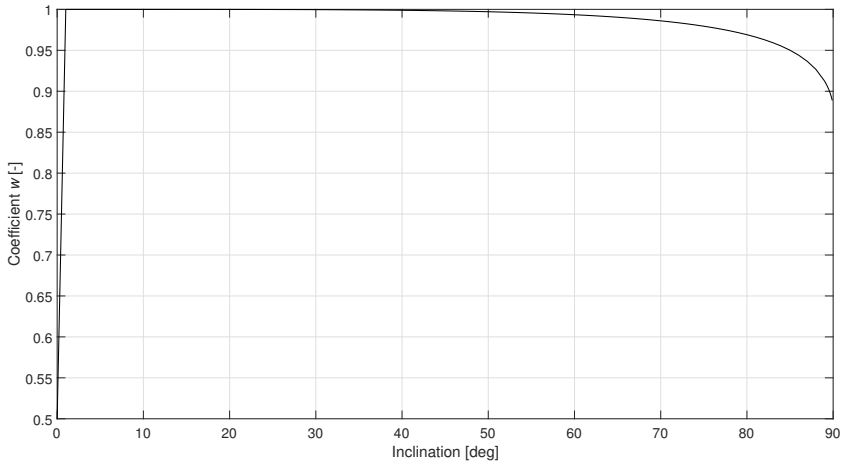


Figure F.7: The value of the coefficient w as a function of inclination.

G

Out-Of-Plane Motion for Cases with Flexible Boundary Conditions

During the validation of the developed elevation shaping function, it was tested whether relaxing the boundary conditions improves the resulting trajectory. This was done by attempting to follow a circular orbit with a constant inclination of 50 degrees and semi-major axis of 1 AU. This appendix shows the variation of the elevation angle as a function of the azimuthal angle when various limits for the boundary conditions are used. In the plots shown in this appendix, the elevation angle of the aforementioned circular orbit is plotted as the theoretical value.

When the circular orbit is followed from a true anomaly of 0 to 180 degrees, Figure G.1 can be obtained. With regards to the legend, it should be mentioned that the values between the brackets indicate the maximum difference in Cartesian coordinates in kilometres and velocities in kilometres per second respectively. The same maximum variation was used for each direction in the Cartesian coordinate system and the same settings were used for both the initial and final states.

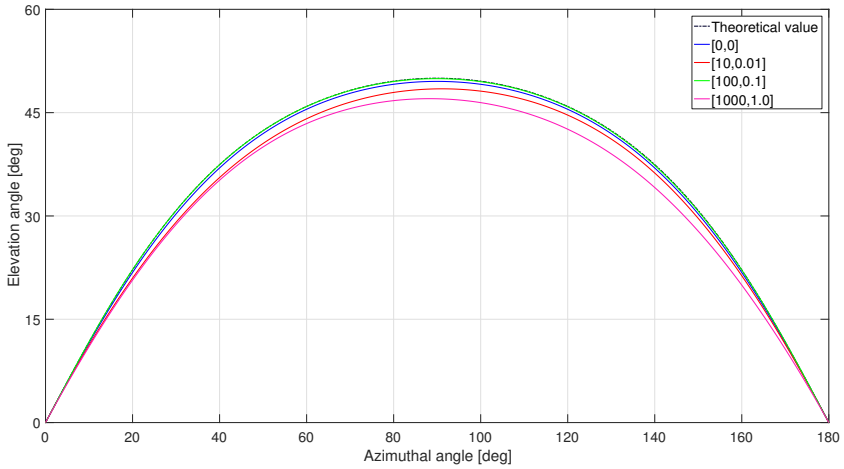


Figure G.1: The variation of the elevation angle as a function of the azimuthal angle when various limits for the boundary conditions are used and a circular orbit at 50 degrees inclination with a semi-major axis of 1 AU is followed from a true anomaly of 0 to 180 degrees.

If the orbit is instead followed from a true anomaly of 90 to 270 degrees, Figure G.2 is the result.

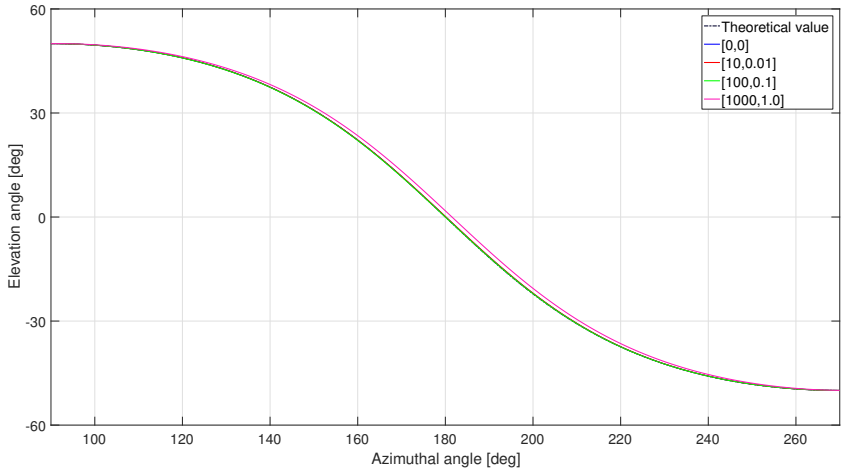


Figure G.2: The variation of the elevation angle as a function of the azimuthal angle when various limits for the boundary conditions are used and a circular orbit at 50 degrees inclination with a semi-major axis of 1 AU is followed from a true anomaly of 90 to 270 degrees.

H

EARTH - MAKEMAKE TRANSFER COEFFICIENT VALUES

To evaluate the robustness of the solution methodology, the optimisation of the Earth - Makemake transfer found by the grid search was performed five additional times with different seeds for the pseudorandom number generator used by the Monte-Carlo method. The determined values of the shaping function's coefficients can be seen Tables H.1 to H.3.

Table H.1: The values of the coefficients r_0 to r_6 for the five additional optimisation runs.

Run	r_0	r_1	r_2	r_3	r_4	r_5	r_6
1	1.08	$-7.94\cdot10^{-1}$	$1.18\cdot10^{-1}$	$-8.36\cdot10^{-2}$	$-9.22\cdot10^{-2}$	$8.93\cdot10^{-1}$	$2.01\cdot10^{-1}$
2	1.08	$-7.94\cdot10^{-1}$	$1.18\cdot10^{-1}$	$-8.36\cdot10^{-2}$	$-9.22\cdot10^{-2}$	$8.93\cdot10^{-1}$	$2.01\cdot10^{-1}$
3	1.08	$-7.94\cdot10^{-1}$	$1.18\cdot10^{-1}$	$-8.36\cdot10^{-2}$	$-9.22\cdot10^{-2}$	$8.93\cdot10^{-1}$	$2.01\cdot10^{-1}$
4	1.08	$-7.94\cdot10^{-1}$	$1.18\cdot10^{-1}$	$-8.36\cdot10^{-2}$	$-9.22\cdot10^{-2}$	$8.93\cdot10^{-1}$	$2.01\cdot10^{-1}$
5	1.08	$-7.94\cdot10^{-1}$	$1.18\cdot10^{-1}$	$-8.36\cdot10^{-2}$	$-9.22\cdot10^{-2}$	$8.93\cdot10^{-1}$	$2.01\cdot10^{-1}$

Table H.2: The values of the coefficients a_0 to p_0 for the five additional optimisation runs.

Run	a_0	a_1	a_2	a_3	a_4	a_5	p_0
1	-2.29	-2.29	-5.80	1.84	-0.40	-2.11	$2.08\cdot10^{-3}$
2	$2.29\cdot10^{-9}$	3.86	$2.44\cdot10^{-1}$	$5.39\cdot10^{-1}$	3.29	-23.1	$-2.22\cdot10^{-4}$
3	$2.55\cdot10^{-4}$	5.49	-18.2	2.03	-4.60	-12.3	$-6.80\cdot10^{-3}$
4	8.00	$-5.55\cdot10^{-1}$	-2.00	1.30	$-2.11\cdot10^{-1}$	-4.27	$3.87\cdot10^{-3}$
5	18.4	-1.04	-3.28	$2.27\cdot10^{-1}$	1.54	-1.92	$-3.64\cdot10^{-3}$

Table H.3: The values of the coefficients p_1 to c for the five additional optimisation runs.

Run	p_1	p_2	p_3	b_0	b_1	b_2	c
1	4.89	$-5.94\cdot10^{-1}$	-5.68	$5.54\cdot10^{-2}$	$-1.77\cdot10^{-2}$	$8.13\cdot10^{-3}$	$-4.69\cdot10^{-2}$
2	6.22	$2.04\cdot10^{-1}$	3.61	$-4.91\cdot10^{-2}$	$-4.76\cdot10^{-3}$	$-2.35\cdot10^{-3}$	$-4.38\cdot10^{-2}$
3	4.25	$7.45\cdot10^{-1}$	$-6.53\cdot10^{-1}$	$3.31\cdot10^{-2}$	$-1.62\cdot10^{-2}$	$7.93\cdot10^{-3}$	$-2.65\cdot10^{-2}$
4	4.52	$-6.75\cdot10^{-1}$	-2.38	$-6.19\cdot10^{-2}$	$1.99\cdot10^{-2}$	$-8.58\cdot10^{-3}$	$-5.18\cdot10^{-2}$
5	3.67	$5.96\cdot10^{-1}$	$-2.61\cdot10^{-1}$	$-1.13\cdot10^{-1}$	$1.12\cdot10^{-2}$	$-1.47\cdot10^{-2}$	$1.10\cdot10^{-1}$

BIBLIOGRAPHY

[1] Novak, D. M., "Methods and tools for preliminary low thrust mission analysis", PhD thesis, University of Glasgow, 2012.

[2] Roegiers, T., "Application of the Spherical Shaping Method to a Low-Thrust Multiple Asteroid Rendezvous Mission: Implementation, Limitations and Solutions", Master's thesis, Delft University of Technology, 2015.

[3] Vroom, A., "On the out-of-plane component of shape-based methods", Literature Review, Delft University of Technology, 2015.

[4] Wakker, K., *Fundamentals of Astrodynamics*, TU Delft Library, January 2015.

[5] Gondelach, D. J., "A Hodographic-Shaping Method for Low-Thrust Trajectory Design", Master's thesis, Delft University of Technology, July 2012.

[6] Noomen, R., "Flight and Orbital Mechanics: Perturbations" [Lecture Notes AE2230-I], Delft University of Technology, March 2015.

[7] Doyle, J., "Engine built to catch a comet begins endurance test", Jet Propulsion Laboratory, July 1996, [online], `http://www.jpl.nasa.gov/news/releases/96/ioneng.html` [retrieved 8 July 2015].

[8] Petropoulos, A. E. and Longuski, J. M., "Automated Design of Low-Thrust Gravity-Assist Trajectories", PhD thesis, School of Aeronautics and Astronautics, Purdue University 2000.

[9] Wall, B. J., "Shape-Based Approximation Method for Low-Thrust Trajectory Optimization", *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 2008, pp. 18–21.

[10] Wall, B. J. and Novak, D. M., "A 3D Shape-Based Approximation Method for Low-Thrust Trajectory Design", *Advances in the Astronautical Sciences*, Vol. 142, 2011.

[11] Vasile, M., De Pascale, P., and Casotto, S., "On the Optimality of a Shape-based Approach on Pseudo-Equinoctial Elements", *Acta Astronautica*, Vol. 61, No. 1, 2007, pp. 286–297.

[12] Gondelach, D. J. and Noomen, R., "Hodographic-Shaping Method for Low-Thrust Interplanetary Trajectory Design", *Journal of Spacecraft and Rockets*, Vol. 52, No. 3, May-June 2015.

[13] Vinti, J., Der, G., and Bonavito, A., *Orbital and Celestial Mechanics*, Vol. 177 of *Progress in Astronautics and Aeronautics*, American Institute of Aeronautics and Astronautics, Inc., 1998.

[14] Stack Exchange, "'Cosine'-esque function with flat peaks and valleys", 2012, [online], `https://math.stackexchange.com/q/100655` [retrieved 24 October 2016].

[15] Ryan, J. K., "Global truncation error and ODE systems" [Lecture Notes WI3097TU], Delft University of Technology, October 2012.

[16] Sachin Joglekar's Blog, "Nelder-Mead Optimization", 2016, [online], `https://codesachin.wordpress.com/2016/01/16/nelder-mead-optimization/` [retrieved 6 March 2017].

[17] Gao, F. and Han, L., "Implementing the Nelder-Mead simplex algorithm with adaptive parameters", *Computational Optimization and Applications*, Vol. 51, No. 1, 2012, pp. 259–277.

[18] Scholarpedia, "Nelder-Mead algorithm", 2009, [online], `http://www.scholarpedia.org/article/Nelder-Mead_algorithm` [retrieved 6 March 2017].

[19] Satola, B. J., "The Amoeba Method Algorithm (Simplex Nelder-Mead)", December 2016, [online], `http://docs.chejunkie.com/amoeba-method-algorithm-simplex-nelder-mead/` [retrieved 6 March 2017].

[20] Surjanovic, S. and Bingham, D., "Virtual Library of Simulation Experiments: Test Functions and Datasets", January 2015, [online], https://www.sfu.ca/~ssurjano/optimization.html [retrieved 23 May 2017].

[21] NASA Jet Propulsion Laboratory, "HORIZONS Web-Interface", 1996, [online], https://ssd.jpl.nasa.gov/horizons.cgi [retrieved 26 May 2017].

[22] Chamberlin, A. and Yeomans, D., "JPL Small-Body Database Browser", 2010, [online], https://ssd.jpl.nasa.gov/sbdb.cgi [retrieved 12 June 2016].

[23] Giant Army, "Universe Sandbox v2.2" [Computer Software], 2015, [online], http://universesandbox.com/ [retrieved 27 August 2015].

[24] Braeunig, R. A., "Interplanetary Flight", 2012, [online], http://www.braeunig.us/space/interpl.htm [retrieved 6 July 2015].

[25] Lutze, F. H., "Orbit in Space and Coordinate Systems", August 2003, [online], http://www.dept.aoe.vt.edu/~lutze/AOE4134/9OrbitInSpace.pdf [retrieved 7 July 2015].

[26] Parker, J. S., "Spaceflight Dynamics" [Lecture Notes ASEN 5050], University of Colorado, October 2014, [online], http://ccar.colorado.edu/asen5050/ASEN5050/Lectures_files/lecture19.pdf [retrieved 7 July 2015].

[27] Roegiers, T., "Optimisation of a multiple asteroid rendezvous mission", Literature Review, Delft University of Technology, 2012.

[28] Wertz, J. R., *Orbit & Constellation Design & Management*, Microcosm Press, Springer, 2001.